

IFIP AICT 683

Deepak Puthal
Saraju Mohanty
Baek-Young Choi (Eds.)



Internet of Things

Advances in Information and Communication Technology

6th IFIP International Cross-Domain Conference, IFIPIoT 2023
Denton, TX, USA, November 2–3, 2023
Proceedings, Part I

1 Part I

 Springer



Editor-in-Chief

Kai Rannenber, *Goethe University Frankfurt, Germany*

Editorial Board Members

TC 1 – Foundations of Computer Science

Luís Soares Barbosa , *University of Minho, Braga, Portugal*

TC 2 – Software: Theory and Practice

Michael Goedicke, *University of Duisburg-Essen, Germany*

TC 3 – Education

Arthur Tatnall , *Victoria University, Melbourne, Australia*

TC 5 – Information Technology Applications

Erich J. Neuhold, *University of Vienna, Austria*

TC 6 – Communication Systems

Burkhard Stiller, *University of Zurich, Zürich, Switzerland*


TC 7 – System Modeling and Optimization

Lukasz Stettner, *Institute of Mathematics, Polish Academy of Sciences, Warsaw, Poland*

TC 8 – Information Systems

Jan Pries-Heje, *Roskilde University, Denmark*


TC 9 – ICT and Society

David Kreps , *National University of Ireland, Galway, Ireland*

TC 10 – Computer Systems Technology

Achim Rettberg, *Hamm-Lippstadt University of Applied Sciences, Hamm, Germany*


TC 11 – Security and Privacy Protection in Information Processing Systems

Steven Furnell , *Plymouth University, UK*

TC 12 – Artificial Intelligence

Eunika Mercier-Laurent , *University of Reims Champagne-Ardenne, Reims, France*

TC 13 – Human-Computer Interaction

Marco Winckler , *University of Nice Sophia Antipolis, France*

TC 14 – Entertainment Computing

Rainer Malaka, *University of Bremen, Germany*

IFIP Advances in Information and Communication Technology

The IFIP AICT series publishes state-of-the-art results in the sciences and technologies of information and communication. The scope of the series includes: foundations of computer science; software theory and practice; education; computer applications in technology; communication systems; systems modeling and optimization; information systems; ICT and society; computer systems technology; security and protection in information processing systems; artificial intelligence; and human-computer interaction.

Edited volumes and proceedings of refereed international conferences in computer science and interdisciplinary fields are featured. These results often precede journal publication and represent the most current research.

The principal aim of the IFIP AICT series is to encourage education and the dissemination and exchange of information about all aspects of computing.

More information about this series at <https://link.springer.com/bookseries/6102>

Deepak Puthal · Saraju Mohanty ·
Baek-Young Choi
Editors


Internet of Things

Advances in Information and Communication Technology

6th IFIP International Cross-Domain Conference, IFIPIoT 2023
Denton, TX, USA, November 2–3, 2023
Proceedings, Part I

Editors

Deepak Puthal 
Khalifa University
Abu Dhabi, United Arab Emirates

Saraju Mohanty 
University of North Texas
Denton, TX, USA

Baek-Young Choi 
University of Missouri at Kansas City
Kansas City, MO, USA

ISSN 1868-4238 ISSN 1868-422X (electronic)
IFIP Advances in Information and Communication Technology
ISBN 978-3-031-45877-4 ISBN 978-3-031-45878-1 (eBook)
<https://doi.org/10.1007/978-3-031-45878-1>

© IFIP International Federation for Information Processing 2024

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

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

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

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

Paper in this product is recyclable.

Preface

6th IFIP International Conference on Internet of Things (IFIP IoT 2023)

The rapid evolution of technology has led to the development of the Internet of Things (IoT), a network of physical objects that are embedded with sensors, software, and network connectivity, enabling them to collect and exchange data. The IoT is transforming our digital landscape, and the IFIP Internet of Things (IFIP-IoT) 2023 conference is a crucial platform for scholars, researchers, and practitioners to come together, share ideas, and advance this transformative field.

This edited book is a compilation of cutting-edge research and developments presented at the IFIP-IoT conference. The conference serves as a dynamic hub where experts from diverse backgrounds come together to explore the multifaceted aspects of IoT, from its technological foundations to its far-reaching implications for society, industry, and beyond.

The chapters in this book are a testament to the collaborative spirit of the IFIP-IoT community. They offer insights into the latest innovations, challenges, and opportunities in IoT, covering a wide array of topics, including IoT architectures, security and privacy, data analytics, edge computing, and applications in various domains. These contributions not only reflect the state of the art in IoT research but also provide valuable perspectives that pave the way for future breakthroughs.

The IFIP-IoT Conference is an annual IFIP event dedicated to IoT research, innovation, and applications, emphasizing the multidisciplinary nature of IoT. IoT encompasses topics from network protocols and embedded systems to analytics, machine learning, and social, legal, ethical, and economic considerations, enabling services in e-health, mobility, energy, manufacturing, smart cities, agriculture, and more. Security, privacy, and societal aspects are essential in IoT deployment. IFIP-IoT covers these diverse areas, seeking papers showcasing technical advancements, research, innovation, pilot results, and policy discussions. Contributors include researchers, users, organizations, ICT industry experts, authorities, and regulators.

IFIP-IoT welcomed full and short paper submissions, with full papers being original and unpublished elsewhere. Poster presentations were limited to student papers. The conference program featured keynotes, plenary talks, tutorials, technical sessions, special sessions, expert panels, a research demo session (RDS), and a student research forum (SRF). New tracks like “SRF” and “RDS” aimed to enhance event participation.

The paper submission guidelines include an 18-page limit for full papers, which applied to both regular and special sessions, as well as an 8-page limit for short papers, applicable to any session, including SRF and RDS. To ensure a thorough review process, we implemented a four-tier review mechanism within EDAS, consisting of TPC-Chairs, Track Chairs, TPC members, and dedicated reviewers. We took measures to address conflicts of interest by appointing multiple TPC chairs and multiple track

chairs for each track. Additionally, we imposed a limit of 2 papers maximum for PC members. SRF encouraged student first-author papers with an 8-page limit, while RDS papers also had an 8-page limit and may or may not feature student first authors. It's important to note that TPC members were permitted to co-author papers with their students in both SRF and RDS. Furthermore, our conference included Regular tracks/sessions that accept submissions from any authors, as well as Special Sessions/Tracks proposed by established researchers, with submissions received by invitation.

The IFIP-IoT conference had six regular tracks, each focusing on a different aspect of IoT:

- **Hardware/Software Solutions for IoT and CPS (HSS):** This track covered the design, development, and implementation of hardware and software solutions for IoT and cyber-physical systems (CPS).
- **Electronics and Signal Processing for IoT (ESP):** This track focused on the use of electronics and signal processing techniques for IoT applications.
- **Artificial Intelligence and Machine Learning Technologies for IoT (AMT):** This track explored the use of artificial intelligence (AI) and machine learning (ML) technologies for IoT applications.
- **Cyber Security/Privacy/Trust for IoT and CPS (SPT):** This track addressed the security, privacy, and trust challenges of IoT and CPS systems.
- **IoT or CPS Applications and Use Cases (APP):** This track presented case studies and applications of IoT and CPS technologies.
- **Networking and Communications Technology for IoT (NCT):** This track focused on the networking and communication technologies used in IoT systems.

Leading IoT experts from around the globe proposed special sessions on cutting-edge IoT topics. These session organizers then invited other established researchers to submit papers to their sessions. We are pleased to announce that the following special sessions took place and contributed excellent research papers to the IFIP-IoT 2023 conference:

- **AI and Big Data for Next-G Internet of Medical Things (IoMT):** This special session explored the use of AI and big data for the next generation of IoMT systems.
- **Blockchain for IoT-Driven Systems (BIOt):** This special session examined the use of blockchain for IoT-driven systems.
- **Edge AI for Smart Wearables (EAW):** This special session focused on the use of edge AI for smart wearables.
- **Energy-Aware Security for IoT (EAS):** This special session addressed the security challenges of IoT systems, with a focus on energy efficiency.
- **IoT for Smart Healthcare (SHC):** This special session explored the use of IoT for smart healthcare applications.
- **IoT for Wearables and Smart Devices (IWS):** This special session focused on the use of IoT for wearables and smart devices.
- **Metaverse for IoT (MIOt):** This special session examined the use of the metaverse for IoT applications.

- **Security by Design for IoT (SbD):** This special session discussed the importance of security by design for IoT systems.
- **Technologies for Smart Agriculture (TSA):** This special session explored the use of IoT technologies for smart agriculture.

In addition to the regular tracks and special sessions mentioned earlier, we introduced two sessions to support graduate students, early career researchers, and ongoing projects through short papers:

- **Student Research Forum (SRF):** This session was designed to provide valuable opportunities for research scholars and graduate students. Presentations in this session were in a concise oral or poster format.
- **Research Demo Session (RDS):** Authors in this session had the chance to showcase live demonstrations and hardware prototypes of their research.

We are grateful to the authors who contributed their expertise to this volume, and we commend their dedication to advancing the field of IoT. We would also like to acknowledge the reviewers whose insightful feedback ensured the quality and rigor of the included chapters.

We hope that this edited book will serve as a valuable resource for researchers, educators, policymakers, and industry professionals alike, fostering a deeper understanding of IoT and inspiring further innovation in this transformative domain. As the IFIP-IoT conference continues to evolve and grow, we look forward to witnessing the continued impact of this vibrant community on the ever-expanding Internet of Things.

Deepak Puthal
Saraju Mohanty
Baek-Young Choi

Research Demo Session Chairs

Amit Joshi Malaviya National Institute of Technology Jaipur, India
Sibi Sethuraman Vellore Institute of Technology AP, India

Finance Chair

Bibhudutta Rout University of North Texas, USA

Registration Chair

Alakananda Mitra University of Nebraska Lincoln, USA

Publicity Chairs

Umamaheswara Tida North Dakota State University, USA
Uttam Ghosh Meharry Medical College, USA
Hemanta Mondal National Institute of Technology Durgapur, India
Sudeendra Kumar PES University, India
Dhruva Ghai Oriental University, India
Sujit Biswas University of East London, UK
Theocharis Theocharides University of Cyprus, Cyprus

Industry Liaison Chair

Robert Karam University of South Florida, USA

Panel Chairs

Hao Zheng University of South Florida, USA
Alex Chiumento University of Twente, The Netherlands

Women in Engineering Chairs

Jaya Dofe California State University, Fullerton, USA
Banee Bandana Das SRM University AP, India

Steering Committee Chair

Srinivas Katkoori University of South Florida, USA

Track Chairs

Regular Track - Artificial Intelligence and Machine Learning Technologies for IoT (AMT)

Sejun Song University of Missouri Kansas City, USA
Yu Chen Binghamton University, USA

Regular Track - Cyber Security/Privacy/Trust for IoT and CPS (SPT)

Filippo Malandra University at Buffalo, USA
Kaustubh Dhondge Glaukes Labs, USA

Regular Track - Electronics and Signal Processing for IoT (ESP)

Narayan Panigrahi Center for AI and Robotics, India
Venkataramana Badarla Indian Institute of Technology Tirupati, India

Regular Track - Hardware/Software Solutions for IoT and CPS (HSS)

Cihan Tunc University of North Texas, USA
Tauhidur Rahman Florida International University, USA

Regular Track - IoT or CPS Applications and Use Cases (APP)

Peeta Basa Pati Amrita Vishwa Vidyapeetham, India
Pradip Sharma University of Aberdeen, UK

Regular Track - Networking and Communications Technology for IoT (NCT)

Sergio Trilles Universitat Jaume I, Spain
Xuyun Zhang Macquarie University, Australia

Regular Track - Research Demo Session (RDS)

Amit Joshi Malviya National Institute of Technology, India

Regular Track - Research Demo Session (RDS); Special Track - Metaverse for IoT (MIoT)

Sibi Chakkaravarthy VIT-AP University, India
Sethuraman

Regular Track - Student Research Forum (SRF)

Chenyun Pan University of Texas at Arlington, USA
Mike Borowczak University of Central Florida, USA

Regular Track - Women in Engineering (WIE)

Banee Das SRM University Andhra Pradesh, India
Jaya Dofe California State University Fullerton, USA

Special Track - AI and Big Data for Next-G Internet of Medical Things (IoMT)

Uttam Ghosh Meharry Medical College, USA

Special Track - Blockchain for IoT-Driven Systems (BIOt)

Ashok Kumar Pradhan SRM University Andhra Pradesh, India
Sujit Biswas University of East London, UK

Special Track - Edge AI for Smart Wearables (EAW)

Bashir Morshed Texas Tech University, USA

Special Track - Energy-Aware Security for IoT (EAS)

Sriram Sankaran Amrita University, India
Swapnoneel Roy University of North Florida, USA

Special Track - IoT for Smart Healthcare (SHC)

Abhishek Sharma LNM Institute of Information Technology, India
Prateek Jain Nirma University, India

Special Track - IoT for Wearables and Smart Devices (IWS)

Ramanujam E. National Institute of Technology Silchar, India
Thinagarani Perumal University Putra Malaysia, Malaysia

Special Track - Metaverse for IoT (MIoT)

Lei Chen Georgia Southern University, USA
Meenalosini Cruz Georgia Southern University, USA

Special Track - Security by Design for IoT (SbD)

Saswat Ram SRM University Andhra Pradesh, India
Venkata Yanambaka Texas Woman's University, USA

Special Track - Technologies for Smart Agriculture (TSA)

Alakananda Mitra University of Nebraska Lincoln, USA
Laavanya Rachakonda University of North Carolina Wilmington, USA

Technical Program Committee

Artificial Intelligence and Machine Learning Technologies for IoT (AMT)

Showmik Bhowmik Jadavpur University, India
Jayson Boubin Ohio State University, USA
Saptarshi Chatterjee Jadavpur University, India
Te-Chuan Chiu National Tsing Hua University, Taiwan
Uma Choppali Dallas College - Eastfield Campus, USA

Soham Das	Microsoft, USA
Hongsheng Hu	Data61 CSIRO, Australia
Agbotiname Imoize	University of Lagos, Nigeria/Ruhr University Bochum, Germany
Hokeun Kim	Hanyang University, South Korea
Uma Maheswari B.	Amrita School of Engineering, India
Adnan Mahmood	Macquarie University, Australia
Pradip Pramanick	Tata Consultancy Services, India
Rajan Shankaran	Macquarie University, Australia
Sicong Shao	University of Arizona, USA
Yuan-Yao Shih	National Chung Cheng University, Taiwan
Ronghua Xu	Michigan Technological University, USA
Xiaonan Zhang	Florida State University, USA

IoT or CPS Applications and Use Cases (APP)

Showmik Bhowmik	Jadavpur University, India
Jayson Boubin	Ohio State University, USA
Saptarshi Chatterjee	Jadavpur University, India
Hongsheng Hu	Data61 CSIRO, Australia
Uma Maheswari B.	Amrita School of Engineering, India
Adnan Mahmood	Macquarie University, Australia
Pradip Pramanick	Tata Consultancy Services, India
Rajan Shankaran	Macquarie University, Australia
Praveen Shukla	Babu Banarasi Das University, India

Electronics and Signal Processing for IoT (ESP)

Showmik Bhowmik	Jadavpur University, India, India
Jayson Boubin	Ohio State University, USA
Saptarshi Chatterjee	Jadavpur University, India
Hongsheng Hu	Data61 CSIRO, Australia
Tanmay Kasbe	Shri Vaishnav Vidyapeeth Vishwavidyalaya Indore, India
Uma Maheswari B.	Amrita School of Engineering, India
Adnan Mahmood	Macquarie University, Australia
Tapas Patra	Odisha University of Technology and Research, India
Pradip Pramanick	Tata Consultancy Services, India
Md Abu Sayeed	Eastern New Mexico University, USA
Rajan Shankaran	Macquarie University, Australia
Vikas Tiwari	C.R. Rao AIMSCS, India

Hardware/Software Solutions for IoT and CPS (HSS)

Showmik Bhowmik	Jadavpur University, India
Jayson Boubin	Ohio State University, USA
Saptarshi Chatterjee	Jadavpur University, India
Uma Choppali	Dallas College - Eastfield Campus, USA

Garima Ghai	Oriental University Indore, India
Hongsheng Hu	Data61 CSIRO, Australia
Uma Maheswari B.	Amrita School of Engineering, India
Adnan Mahmood	Macquarie University, Australia
Ram Mohanty	UNSW Canberra, Australia
Pradip Pramanick	Tata Consultancy Services, India
Xiao Sha	Stony Brook University, USA
Rajan Shankaran	Macquarie University, Australia

Networking and Communications Technology for IoT (NCT)

Showmik Bhowmik	Jadavpur University, India
Jayson Boubin	Ohio State University, USA
Saptarshi Chatterjee	Jadavpur University, India
Joy Dutta	Khalifa University, United Arab Emirates
Umashankar Ghugar	GITAM University, India
Hongsheng Hu	Data61 CSIRO, Australia
Uma Maheswari B.	Amrita School of Engineering, India
Adnan Mahmood	Macquarie University, Australia
Pradip Pramanick	Tata Consultancy Services, India
Rajan Shankaran	Macquarie University, Australia

Contents – Part I

Artificial Intelligence and Machine Learning Technologies for IoT (AMT)

An Optimized Graph Neural Network-Based Approach for Intrusion Detection in Smart Vehicles 3
Pallavi Zambare and Ying Liu

Evaluation of the Energy Viability of Smart IoT Sensors Using TinyML for Computer Vision Applications: A Case Study 18
Adriel Monti De Nardi and Maxwell Eduardo Monteiro

Simulated Annealing Based Area Optimization of Multilayer Perceptron Hardware for IoT Edge Devices 34
Rajeev Joshi, Lakshmi Kavya Kalyanam, and Srinivas Katkooori

Machine Learning-Based Multi-stratum Channel Coordinator for Resilient Internet of Space Things 48
Md Tajul Islam, Sejun Song, and Baek-Young Choi

A Schedule of Duties in the Cloud Space Using a Modified Salp Swarm Algorithm 62
Hossein Jamali, Ponkoj Chandra Shill, David Feil-Seifer, Frederick C. Harris Jr., and Sergiu M. Dascalu

Layer-Wise Filter Thresholding Based CNN Pruning for Efficient IoT Edge Implementations 76
Lakshmi Kavya Kalyanam, Rajeev Joshi, and Srinivas Katkooori

Energy-Aware Security for IoT (EAS)

Shrew Distributed Denial-of-Service (DDoS) Attack in IoT Applications: A Survey 97
Harshdeep Singh, Vishnu Vardhan Baligodugula, and Fathi Amsaad

Honeypot Detection and Classification Using Xgboost Algorithm for Hyper Tuning System Performance 104
Vinayak Musale, Pranav Mandke, Debajyoti Mukhopadhyay, Swapnoneel Roy, and Aniket Singh

Electromagnetic Fault Injection Attack on ASCON Using ChipShouter 114
Varun Narayanan and Sriram Sankaran

Edge AI for Smart Wearables (EAW)

A Smart Health Application for Real-Time Cardiac Disease Detection
and Diagnosis Using Machine Learning on ECG Data 135
Ucchwas Talukder Utsha, I Hua Tsai, and Bashir I. Morshed

Reinforcement Learning Based Angle-of-Arrival Detection for
Millimeter-Wave Software-Defined Radio Systems 151
Marc Jean and Murat Yuksel

Empowering Resource-Constrained IoT Edge Devices: A Hybrid Approach
for Edge Data Analysis 168
Rajeev Joshi, Raaga Sai Somesula, and Srinivas Katkoori

Energy-Efficient Access Point Deployment for Industrial IoT Systems 182
*Xiaowen Qi, Jing Geng, Mohamed Kashef, Shuvra S. Bhattacharyya,
and Richard Candell*

Hardware/Software Solutions for IoT and CPS (HSS)

FAMID: False Alarms Mitigation in IoMT Devices 199
Shakil Mahmud, Myles Keller, Samir Ahmed, and Robert Karam

Dynamic Task Allocation and Scheduling for Energy Saving in Edge
Nodes for IoT Applications 218
Shubhangi K. Gawali, Lucy J. Gudino, and Neena Goveas

Deep Learning Based Framework for Forecasting Solar Panel
Output Power 229
*Prajnyajit Mohanty, Umesh Chandra Pati,
and Kamalakanta Mahapatra*

AI and Big Data for Next-G Internet of Medical Things (IoMT)

EHR Security and Privacy Aspects: A Systematic Review 243
Sourav Banerjee, Sudip Barik, Debashis Das, and Uttam Ghosh

SNN Based Neuromorphic Computing Towards Healthcare Applications 261
*Prasenjit Maji, Ramapati Patra, Kunal Dhibar,
and Hemanta Kumar Mondal*

Crossfire Attack Detection in 6G Networks with the Internet of Things (IoT)	272
<i>Nicholas Perry and Suman Bhunia</i>	

IoT for Wearables and Smart Devices (IWS)

Prediction of Tomato Leaf Disease Plying Transfer Learning Models	293
<i>B. S. Vidhyasagar, Koganti Harshagnan, M. Diviya, and Sivakumar Kalimuthu</i>	
Video Captioning Based on Sign Language Using YOLOV8 Model	306
<i>B. S. Vidhyasagar, An Sakthi Lakshmanan, M. K. Abishek, and Sivakumar Kalimuthu</i>	
Improvement in Multi-resident Activity Recognition System in a Smart Home Using Activity Clustering	316
<i>E. Ramanujam, Sivakumar Kalimuthu, B. V. Harshavardhan, and Thinagaran Perumal</i>	

Metaverse for IoT (MIoT)

Forensics Analysis of Virtual Reality Social Community Applications on Oculus Quest 2	337
<i>Samuel Ho and Umit Karabiyik</i>	
Objective Emotion Quantification in the Metaverse Using Brain Computer Interfaces	353
<i>Anca O. Muresan, Meenalosini V. Cruz, and Felix G. Hamza-Lup</i>	
MetaHap: A Low Cost Haptic Glove for Metaverse.	362
<i>S. Sibi Chakkaravarthy, Marvel M. John, Meenalosini Vimal Cruz, R. Arun Kumar, S. Anitha, and S. Karthikeyan</i>	

Technologies for Smart Agriculture (TSA)

CroPAiD: Protection of Information in Agriculture Cyber-Physical Systems Using Distributed Storage and Ledger	375
<i>Sukrutha L. T. Vangipuram, Saraju P. Mohanty, and Elias Kougianos</i>	
Sana Solo: An Intelligent Approach to Measure Soil Fertility	395
<i>Laavanya Rachakonda and Samuel Stasiewicz</i>	

Smart Agriculture – Demystified 405
Alakananda Mitra, Saraju P. Mohanty, and Elias Kougianos

Student Research Forum (SRF)

**WeedOut: An Autonomous Weed Sprayer in Smart Agriculture Framework
Using Semi-Supervised Non-CNN Annotation** 415
*Kiran Kumar Kethineni, Alakananda Mitra, Saraju P. Mohanty,
and Elias Kougianos*

**ALBA: Novel Anomaly Location-Based Authentication in IoMT
Environment Using Unsupervised ML** 424
Fawaz J. Alruwaili, Saraju P. Mohanty, and Elias Kougianos

**A Configurable Activation Function for Variable Bit-Precision DNN
Hardware Accelerators** 433
*Sudheer Vishwakarma, Gopal Raut, Narendra Singh Dhakad,
Santosh Kumar Vishvakarma, and Dhruva Ghai*

**Authentication and Authorization of IoT Edge Devices Using Artificial
Intelligence** 442
Muhammad Sharjeel Zareen, Shahzaib Tahir, and Baber Aslam

Secure Dynamic PUF for IoT Security 454
Shailesh Rajput and Jaya Dofe

Author Index 463

Contents – Part II

IoT or CPS Applications and Use Cases (APP)

IoT Based Real Time Monitoring of Delhi-NCR Air Pollution Using Low Power Wide Area Network.	3
<i>Prem Chand Jain, Vandavasi Satya Charan, Chitumalla Sahith, and Rohit Singh</i>	
A Survey of Pedestrian to Infrastructure Communication System for Pedestrian Safety: System Components and Design Challenges	14
<i>Pallavi Zambare and Ying Liu</i>	
Computer Vision Based 3D Model Floor Construction for Smart Parking System.	36
<i>Jayaprakash Patra, Satyajit Panda, Vipul Singh Negi, and Suchismita Chinara</i>	
3D Visualization of Terrain Surface for Enhanced Spatial Mapping and Analysis.	49
<i>Pant Shivam and Panigrahi Narayan</i>	
Generic Medicine Recommender System with Incorporated User Feedback . . .	64
<i>Sneh Shah, Varsha Naik, Debajyoti Mukhopadhyay, and Swapnoneel Roy</i>	
Recall-Driven Precision Refinement: Unveiling Accurate Fall Detection Using LSTM	74
<i>Rishabh Mondal and Prasun Ghosal</i>	

Blockchain for IoT-Driven Systems (BIoT)

Application of Blockchain Based e-Procurement Solution for Mitigating Corruption in Smart Cities Using Digital Identities	87
<i>Arish Siddiqui, Kazi Tansen, and Hassan Abdalla</i>	
Blockchain Based Framework for Enhancing Cybersecurity and Privacy in Procurement	101
<i>Arish Siddiqui, Kazi Tansen, and Hassan Abdalla</i>	

Block-Privacy: Privacy Preserving Smart Healthcare Framework: Leveraging Blockchain and Functional Encryption	114
<i>Bhaskara Santhosh Egala, Ashok Kumar Pradhan, and Shubham Gupta</i>	
zkHealthChain - Blockchain Enabled Supply Chain in Healthcare Using Zero Knowledge	133
<i>G. Naga Nithin, Ashok Kumar Pradhan, and Gandharba Swain</i>	
Blockchain-Based Secure Noninvasive Glucometer and Automatic Insulin Delivery System for Diabetes Management	149
<i>Divi Gnanesh, Gouravajjula Lakshmi Sai Bhargavi, and G. Naga Nithin</i>	
An Efficient and Secure Mechanism for Ubiquitous Sustainable Computing System	160
<i>G. Naga Nithin</i>	
Networking and Communications Technology for IoT (NCT)	
Understanding Security Challenges and Defending Access Control Models for Cloud-Based Internet of Things Network	179
<i>Pallavi Zambare and Ying Liu</i>	
Fog Computing in the Internet of Things: Challenges and Opportunities	198
<i>Iqra Amin Shah, Mohammad Ahsan Chishti, and Asif I. Baba</i>	
A 2-Colorable DODAG Structured Hybrid Mode of Operations Architecture for RPL Protocol to Reduce Communication Overhead	212
<i>Alekha Kumar Mishra, Sadhvi Khatik, and Deepak Puthal</i>	
Security by Design for IoT (SbD)	
Role-Based Access Control in Private Blockchain for IoT Integrated Smart Contract	227
<i>Darwish Al Neyadi, Deepak Puthal, Joy Dutta, and Ernesto Damiani</i>	
VXorPUF: A Vedic Principles - Based Hybrid XOR Arbiter PUF for Robust Security in IoMT	246
<i>Md Ishtyaq Mahmud, Pintu Kumar Sadhu, Venkata P. Yanambaka, and Ahmed Abdelgawad</i>	
Easy-Sec: PUF-Based Rapid and Robust Authentication Framework for the Internet of Vehicles.	262
<i>Pintu Kumar Sadhu and Venkata P. Yanambaka</i>	

IoT for Smart Healthcare (SHC)

- FortiRx: Distributed Ledger Based Verifiable and Trustworthy Electronic Prescription Sharing 283
Anand Kumar Bapatla, Saraju P. Mohanty, and Elias Kougianos
- Survival: A Smart Way to Locate Help 302
Laavanya Rachakonda and Kylie Owen
- Federated Edge-Cloud Framework for Heart Disease Risk Prediction Using Blockchain 309
Uttam Ghosh, Debashis Das, Pushpita Chatterjee, and Nadine Shillingford

Cyber Security/Privacy/Trust for IoT and CPS (SPT)

- Understanding Cybersecurity Challenges and Detection Algorithms for False Data Injection Attacks in Smart Grids 333
Pallavi Zambare and Ying Liu
- Comprehensive Survey of Machine Learning Techniques for Detecting and Preventing Network Layer DoS Attacks 347
Niraj Prasad Bhatta, Ashutosh Ghimire, Al Amin Hossain, and Fathi Amsaad
- Power Analysis Side-Channel Attacks on Same and Cross-Device Settings: A Survey of Machine Learning Techniques 357
Ashutosh Ghimire, Vishnu Vardhan Baligodugula, and Fathi Amsaad

Research Demo Session (RDS)

- Lite-Agro: Exploring Light-Duty Computing Platforms for IoAT-Edge AI in Plant Disease Identification 371
Catherine Dockendorf, Alakananda Mitra, Saraju P. Mohanty, and Elias Kougianos
- FarmIns: Blockchain Leveraged Secure and Reliable Crop Insurance Management System 381
Musharraf Alruwaill, Anand Kumar Bapatla, Saraju P. Mohanty, and Elias Kougianos
- PTSD Detection Using Physiological Markers 390
Laavanya Rachakonda and K. C. Bipin

A Signal Conditioning Circuit with Integrated Bandgap Reference for
Glucose Concentration Measurement 396
Riyaz Ahmad, Amit Mahesh Joshi, and Dharmendar Boolchandani

Detection of Aircraft in Satellite Images using Multilayer Convolution
Neural Network 402
Swaraj Agarwal, Narayan Panigarhi, and M. A. Rajesh

PEP: Hardware Emulation Platform for Physiological Closed-Loop Control
Systems 410
Shakil Mahmud, Samir Ahmed, and Robert Karam

Author Index 419

Artificial Intelligence and Machine Learning Technologies for IoT (AMT)



An Optimized Graph Neural Network-Based Approach for Intrusion Detection in Smart Vehicles

Pallavi Zambare^(✉) and Ying Liu^(✉)

Texas Tech University, Lubbock, USA
{pzambare, Y.Liu}@ttu.edu

Abstract. Due to recent developments in vehicle networks (VNs), more and more people want their electric cars to have access to sophisticated networking features and perform a variety of advanced activities. Several technologies are installed in these autonomous vehicles to aid the drivers. Cars have unquestionably become smarter as we've been able to install more and more gadgets and applications on them. As a result, the security of assistant and self-drive systems in automobiles becomes a life-threatening concern, since hostile attacks that cause traffic accidents may infiltrate smart cars. This research provides a technique based on the Graph Neural Network (GNN) deep learning (DL) model for detecting intrusions in VNs. This model can recognize attack patterns and classify threats accordingly. Experimentation utilizes car-hacking datasets. These files include DoS attacks, fuzzy attacks, driving gear spoofing, and RPM gauge spoofing. The car-hacking dataset is converted into image files, and the GNN model provided works with the newly produced image dataset. The findings of comparing our model to DL models indicate that our GNN model is superior. In specific, the test case scenarios may identify abnormalities with respect to detection F1-score, recall, precision, and accuracy to ensure accurate detection and identification of potential false alarm concerns.

Keywords: Intrusion Detection System · Internet of Vehicles · Internet of Things · Deep Learning Techniques · Graph Neural Network

1 Introduction

With the emergence of self-driving vehicles and Vehicle to Everything, automobiles are wirelessly linked to a variety of gadgets. Automobiles are susceptible to malicious attacks that may result in major traffic accidents. Several intrusion analysis and detection approaches have been offered to alleviate these difficulties. The Controller Area Network (CAN) is an essential component of a connected car and is susceptible to malicious attacks. These attacks can be carried out by hackers who use various techniques to exploit vulnerabilities in the vehicle's systems. The attack includes DOS attack, Fuzzy attack, Spoofing attack, etc. In

all cases, in-vehicle intrusion attacks [1] can have serious consequences, including loss of control over the vehicle, compromise of sensitive data, and potentially life-threatening situations. It is important for vehicle manufacturers to take steps to secure their systems and protect against these types of attacks. Figure 1 shows the general scenario that is carried out by hackers to inject cyber-attack in vehicles. Because of this, a lot of researchers have tried to come up with reliable ways to find and analyze attacks. In the identification of the potential attacks, a number of IDS based on Many classification approaches are proposed by different authors which include the use of machine learning (ML) methods such as support vector machine (SVM), decision tree (DT), naive Bayes classifier (NB), logistic regression (LR), and deep convolutional neural network (CNN) [2–5] to detect network attacks. Current machine learning approaches for detecting attacks on moving cars often utilize cyber threat studies that pair routing data with potential attack profiles derived from behavioral analysis techniques including packet collecting, feature comparisons, and packet filtering. In order to classify potential attacks, determine the true nature of the attack, identify the attacked ECU, and initiate countermeasures, routing data of CAN devices is used. Hence, security administrators have turned to DNN approaches such as LSTM, RNN, and CNN to identify complex threats from a variety of vectors. In most cases, deep CNN provides a novel approach to enhancing the accuracy of threat identification in network intrusion detection (NID) and decreasing the occurrence of false positives (FPR). Nevertheless, a single base classifier doesn't adequately fit the data distribution, either because of an excessive amount of bias or an excessive amount of variance [5].

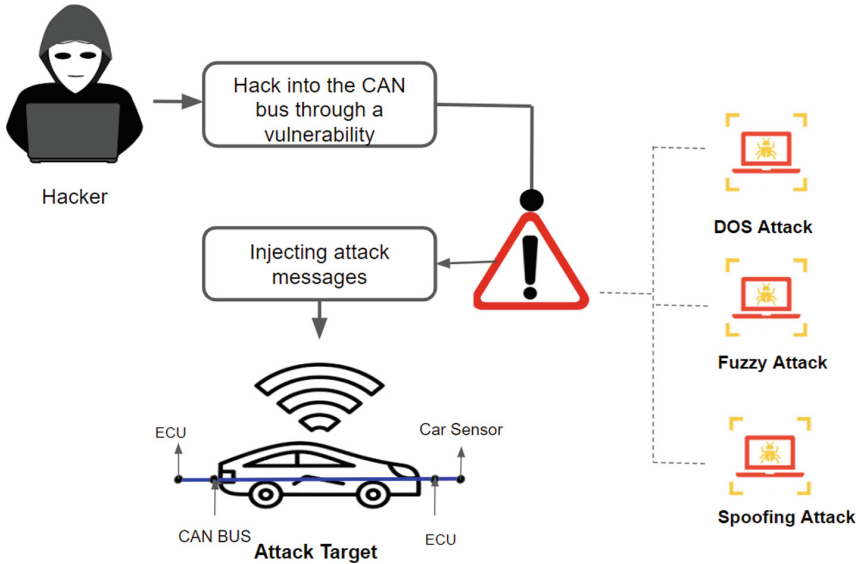


Fig. 1. In vehicle intrusion attack by hacker

Although existing approaches are very effective at detecting attacks, they do not account for the inherent uncertainty in such endeavors. It requires a significant investment of time and effort from humans to label the data set, and there has to be a very large number of training examples used by these conventional approaches for ML and DL. DL-based approaches have been developed to manage a variety of challenging graph issues related to networks. GNN is a kind of DL-based model that operates on graph data. Zhou Cui et al. have made extensive use of them. [7] Successful development of GNNs was motivated by CNNs, the state-of-the-art models for many machine vision applications. CNNs are able to extract multi-scale spatial characteristics via their filters, which may subsequently be merged to create high-quality representations. CNN, however, has a hard time learning from and mining graph data. Authors Jing et al. According to Zhou Cui et al., the ML community has attempted to apply the efficacy of CNNs to graphs as Euclidean data structure is a unique form of a graph. [7]

The typical representation of a node in a GNN is its weighted sum of the representations of its neighbors. This procedure will provide the nodes with more data and a more accurate representation of themselves if the signal is sufficiently consistent in reference to the underlying network. The states held by GNNs, which are connectionist models, are able to get information from their neighbors at any distance. According to Zhang et al., [8], they have been used to simulate the interdependence of graphs in a variety of graph-based tasks, which include node classification, edge prediction, edge clustering, node clustering, and graph classification.

In this study, we used GNN to deal with a wide range of classification problems related to threats. The most important things this work has done are:

- We give a detailed review of existing DL models and suggest a new framework for effective cyber-attack detection in vehicle networks using CNN and GNN techniques.
- We suggest a method for transforming data that can turn vehicle network traffic data into images so that different cyber-attack patterns are easier to spot.
- We test the proposed method on a benchmark cyber-security dataset for car hacking that includes both data from inside the car and from outside it. We then compare the model's performance to that of other state-of-the-art methods.

The rest of the paper is organized as follows; In the Sect. 2, we present a literature review of ML, DL, and graph network modes. In Sect. 3 the detailed proposed system is discussed which includes text-to-image conversion of dataset and DL models namely CNN and GNN design pipeline. In Sect. 4, we review research on the theoretical and empirical outcomes and analyses of CNNs and GNNs, as well as related research efforts. We finish with the suggested system in Sect. 5

2 Literature Review

2.1 Traditional IDS

Most of the traditional ways to find intrusions are based on statistical analysis [9], threshold analysis [10], and signature analysis [11]. While these techniques are effective at revealing malicious traffic behavior, they need the input of expert knowledge from security researchers. This renders their predetermined rules and limits mostly useless. In addition, this “experience” is really simply a description of the hitherto unquantifiable behavior of hostile traffic. Because of this, these methods can’t be easily adapted to the Internet of today, which has a lot more network data and more unpredictable network attacks.

2.2 IDS Based on Machine Learning

By using machine learning, it has now become possible to categorize and group network traffic for safety purposes. Simple machine learning methods were tried out by early researchers, including k-nearest neighbor (KNN) [12], self-organizing maps (SOM) [14] and SVM [13] to solve classification and clustering problems in other fields. These algorithms worked well on DARPA, NSL-KDD, KDD99, and other intrusion detection datasets. These datasets aren’t up to date, which is a shame. Moreover, they have normal data and data about attacks that are too simple. It is hard to simulate the highly complex network environment of today with these datasets. As our work in this study shows, it is difficult to achieve optimal results by applying traditional algorithms when the new malicious traffic dataset is passed.

2.3 IDS Based on DNN

Most of the time, the ML algorithm’s performance is usually affected by the way data is displayed. [15]. The author uses a DNN approach called representation learning (or “feature learning”) to discover the underlying causes of data variation. Spectral clustering and DNN methods are combined by Ma et al. [16] to identify suspicious actions. Niyaz et al. [17] employed deep belief networks to create a flexible and effective IDS. Yet, these studies construct their models to acquire representations from user-created traffic characteristics. The entire potential of DNN is not being used. The detection rate, accuracy, and false alarm rate may all be increased by using an enhanced traffic feature set, as shown by the work of Eesa et al. [18]. As in Natural language processing (NLP) and computer vision [19], it should be able to obtain characteristics directly from raw traffic data.

CNN and RNN are the two DNN models that are used the most often. The CNN feeds on the raw data without any pre-processing. It has few parameters and requires little input data, thus there’s no need for image reconstruction or feature extraction. In the field of image recognition, CNNs have been shown to work very well [20]. CNNs can do well with certain types of network traffic and

protocols if they are trained quickly. Fan and Ling-zhi [21] used a multilayer CNN to get very accurate features. The convolution layer was connected to the sampling layer below, and this model did better on the KDD99 dataset than traditional detection algorithms like SVM. But CNN can only look at one input package at a time; it can't look at information about timing in a given traffic scenario. In a real attack traffic scenario, a single packet is just normal data. When a lot of packets are sent at once or in a short amount of time, this packet is considered to be malicious traffic. In this situation, the CNN doesn't work, which could mean that a lot of alerts are missed.

2.4 IDS Based on Graph-Based Deep Learning Algorithms

Text categorization is a significant NLP issue that has existed for a while. In the traditional way of classifying texts, bag-of-words features are used. When a text is shown as a graph of words, you can better understand the meaning of words that aren't next to each other or that are far apart. Peng et al. [22] Using a graph-CNN-based DL model to initially turn texts into graphs of words. Niepert et al. [23] then utilize graph convolution operations to combine the word graphs. Zhang et al. [24] suggest that text be encoded using the Sentence LSTM. They think of the whole sentence as a single state that is made up of sub-states for each word and a state for the sentence as a whole. They use global sentence-level representations for operations like grouping objects. Using these techniques, a phrase or piece of writing may be seen as a network of word nodes. When Yao et al. [25] construct the corpus graph, they see words as nodes and documents. To understand how words and documents are embedded in the corpus graph, they employ the Text GCN. Text categorization may also have difficulty with sentiment analysis, for which Tai et al. [26] propose a Tree-LSTM method.

3 Proposed System

In this study, we provide an Analytical Model for In-Vehicle NID. In order to anticipate the stability of NID once suspicious network flows have been gathered, the proposed NID model employs the GNN model with fine-tuned parameters. The three steps of the process for classifying behavior are displayed in Figure 2 together with the general structure of the model: (1) Pre-processing of data, (2) Text to Image Conversion, (3) applying DL-based model (CNN and GNN) (4) model validation and performance analysis.

3.1 Data Pre-processing

First, data sources like the HCRL Car-Hacking dataset were used to get the training sample data [27]. The HCRL dataset has about 4 million messages from 30 to 40 min of CAN traffic. For our plan, we used the driving gear from this set to test DoS attacks, spoofing attacks, etc. In each of these attacks, messages were attacked for one out of approximately 25 IDs. So, only messages that had these IDs were looked at.

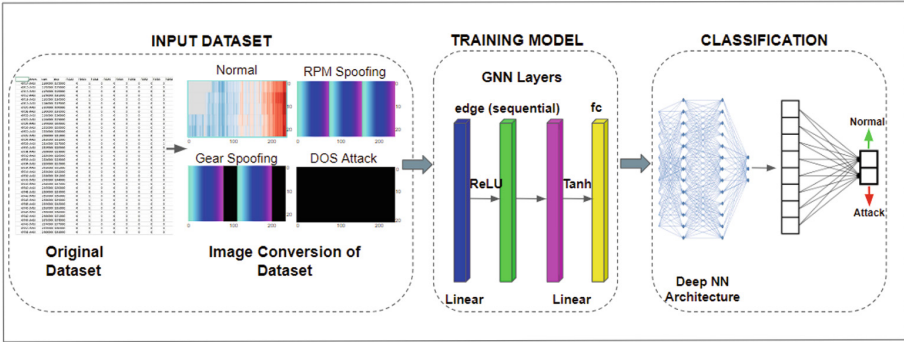


Fig. 2. Proposed System Architecture

- Step 1. Dataset Cleaning.** In the IDS experiment, the whole Car-hacking dataset was chosen because it was a large enough sample to test how well the proposed GNN classifier worked. To get a good look at how well the model works, we rebuilt the experimental data set to create five major types of attacks, including Controller Area Network (CAN). The dataset has both normal traffic and traffic that was injected as an attack. There are four kinds of attacks: flooding, spoofing, replaying, and fuzzing. To reduce the training data samples, duplicate messages are removed.
- Step 2. Normalization.** In the car-hacking dataset, all the fields were thought of as symbolic features. We first changed the symbols for the network packets. Then, attack categories of flags were turned into numbers (1–5) based on the different types of attacks. Normal traffic was given the number ‘0’ through a process called “one-hot encoding” (Fig. 3).

	Timestamp	CAN ID	DLC	DATA(0)	DATA(1)	DATA(2)	DATA(3)	DATA(4)	DATA(5)	DATA(6)	DATA(7)	Flag	Type	label
0	1.478198e+09	0130	8.0	18	80	00	ff	1b	80	0a	e4	R	Normal	0
1	1.478198e+09	0131	8.0	00	80	00	00	4c	7f	0a	9a	R	Normal	0
2	1.478198e+09	0140	8.0	00	00	00	00	08	25	2a	47	R	Normal	0
3	1.478198e+09	0370	8.0	00	20	00	00	00	00	00	00	R	Normal	0
4	1.478198e+09	043f	8.0	00	40	60	ff	7e	cb	08	00	R	Normal	0
...
4659648	1.478193e+09	316	8.0	45	29	24	ff	29	24	0	ff	T	RPM	5
4659649	1.478193e+09	316	8.0	45	29	24	ff	29	24	0	ff	T	RPM	5
4659650	1.478193e+09	316	8.0	45	29	24	ff	29	24	0	ff	T	RPM	5
4659651	1.478193e+09	316	8.0	45	29	24	ff	29	24	0	ff	T	RPM	5
4659652	1.478193e+09	316	8.0	45	29	24	ff	29	24	0	ff	T	RPM	5

Fig. 3. Original dataset sample [27]

3.2 Dataset Generation (Text to Image Conversion)

Normalizing the data is the first step in the process of transforming it. Considering the values of pixels in an image range from 0 to 255, so the scale for network data should also range from 0–255. Quantile normalization is one of the ways that data values can be changed so that they all fall in the same range. We used quantile normalization in the proposed framework because it is very good at handling outliers. Using the quantile normalization approach, feature distributions are transformed into normal distributions, and the values of the features are recalculated based on the normal distribution. Hence, most of the values for the variables are quite near to the median values, which makes it simple to handle any outliers [15]. Upon the completion of the normalization process, the data is next divided into chunks according to the timestamps and sizes of the features included within the network traffic datasets. Figure 4 shows the steps that are taken to get the feature map from the dataset. These steps are feature extraction and normalization. Each segment of 27 consecutive samples from the Car-Hacking dataset, which includes 9 significant features are converted into an image with the form 28283 and a total of 243 feature values. As a result, each altered image is now a three-channel, square, color image (RGB). Figure 5 demonstrates the stark differences between the feature maps created by malware and by legitimate applications utilizing the HRCL dataset.

```
transform = transforms.Compose([transforms.Grayscale(num_output_channels=1),
                               transforms.ToTensor(),
                               transforms.Normalize((0.1307,), (0.3081,))])
```

Fig. 4. steps to generate feature map of car-hacking dataset

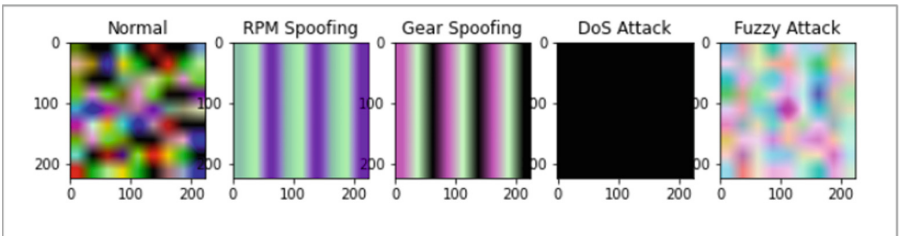


Fig. 5. Feature map (image) generated by car-hacking dataset.

3.3 Training Phase

In this step, the data from the experiment were split into train and test sets. The training set had 80% of the data, but the testing set only had 20 % of the data. Also, the training and testing set had both attack files and regular files. During the model training phase, the CNN and GNN models were used to correctly group cyber threats into different categories. In the experiment, the fine-tuned CNN and GNN were taught to find network intrusions by looking at how collected samples behaved. The model was assessed using the softmax function, and the learning rate was changed using the cross-entropy function to lower the classification error and hasten training.

3.4 Validation Phase

During the model validation phase, the CNN classifier and the proposed GNN classifier were used to compare how well the trained model worked. After the training of the basic classifiers, the results were merged using the training data taken from different subsets at random. At long last, a meta-classifier was educated to categorize the dangers posed by the samples. In practice, the choice of hyperparameters has a big effect on how DL models are trained and how well they can classify data. During the validation phase, the batch size, weight matrix, and epochs were set based on the results of the learning phase.

- **Step 1. CNN Classifier**

CNN classifier for images involves a number of steps, including data collection, preparation, model architecture design, training, validation, testing, optimization, and deployment. The success of each of these processes is essential to the overall success of the final model, thus each step in the procedure needs to be carefully monitored. The below section describes the CNN (2D) used in the classification of in vehicle network intrusion.

CNN is a modified version of a deep neural net that works by looking at how pixels near each other relate to each other. At first, it uses randomly defined patches as input, and then it changes those patches as it learns. During training, the network uses these changed patches to forecast and confirm the outcome during testing and validation. The convolutional layer, the fully connected layer, and the pooling layer are the three primary layers of a CNN, as shown in Fig. 6. The first layer figures out what the neurons that are linked to local areas are sending out. Each one is worked out by multiplying the weights by the area. The network determines the width and height of a filter, and the depth of the filter is the same as the depth of the input. Sub-sampling is another important transformation that can be used in different ways (max pooling, min pooling, and average pooling) depending on what is needed. The user can choose the size of the pooling filter, which is usually an odd number. The pooling layer is in charge of reducing the number of dimensions of the data, which is a good way to avoid overfitting. For effective classification, the output is sent to a fully connected layer after employing a convolution and pooling layer combination.

• Step 2. GNN Classifier

Graph Convolutional Networks have been introduced by Kipf et al. [15] at the University of Amsterdam. GNNs are a type of DL algorithm used for processing data that has a graph structure. The adjacency matrix is a key component of GNNs, as it encodes the structure of the graph. Here are the steps involved in using a GNN for car-hacking dataset (images) classification:

1. Construct a graph representation of the image: In order to use GNNs, we first represented the image as a graph. We have achieved this by representing each pixel in the image as a node in the graph and then connecting it with adjacent pixels with edges.
2. Graph Construction: The next step is to construct the graph using an adjacency matrix. The adjacency matrix represents the connections between the nodes in the graph. In the case of image data, we use a 2D grid graph with edges connecting adjacent pixels.
3. Data Normalization as per Kipf & Welling et al. [15].

$$m_{N(u)} = \sum_{v \in N(u)} \frac{h_v}{\sqrt{|N(u)||N(v)|}} \quad (1)$$

4. Neural Network Architecture: For layer architecture generation we have used Graph Convolutional Neural Networks (GCNs).
5. Feature Extraction: Use the GNN to obtain features from the graph. Here the features extracted are the pixel values, edges, and pixel gradients.
6. Classification: Finally, use the extracted features to classify the images. This is done by using a fully connected layer.
7. Tuning: To improve the performance of the model we have adjusted the hyperparameters of the model, such as learning rate, number of hidden layers, and regularization parameters.

The detailed proposed GNN architecture used is shown in Figure 7.

4 Result and Discussion

A Implementation Details

We do a number of tests to find out how well the proposed model works. We made the size of the node representation 28 and used images made from the text as input data. In other experiments, we also change the size of the features. The size of the training batches is set to 64, and the Adam optimizer is used to train the model. The Adam optimizer is a replacement for stochastic gradient descent as an optimization algorithm for training DL models.

B Dataset Description

The dataset of car-hacking is used for experimental evaluation [6] which includes Spoofing the drive gear, fuzzy attack, DoS attack, and spoofing the RPM gauge. Currently, every dataset has 300 message injection invasions. The average incursion lasted for 3 to 5s, and the total CAN traffic for each dataset was 30 to 40 min. We converted the car-hacking text dataset into an image (28 *28) and gives a label to each image.

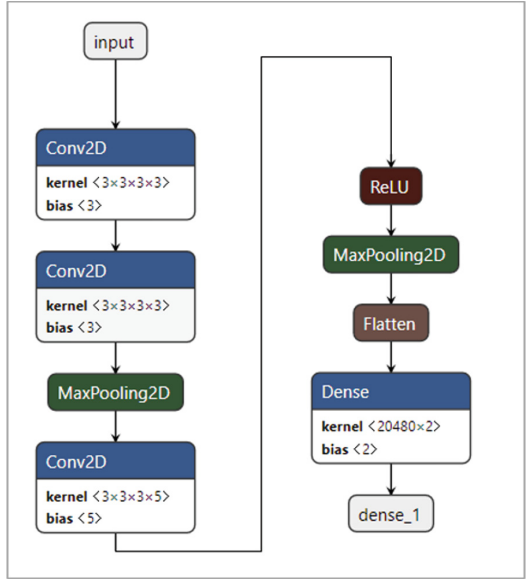


Fig. 6. Proposed fine-tune CNN model for classification of car-hacking dataset

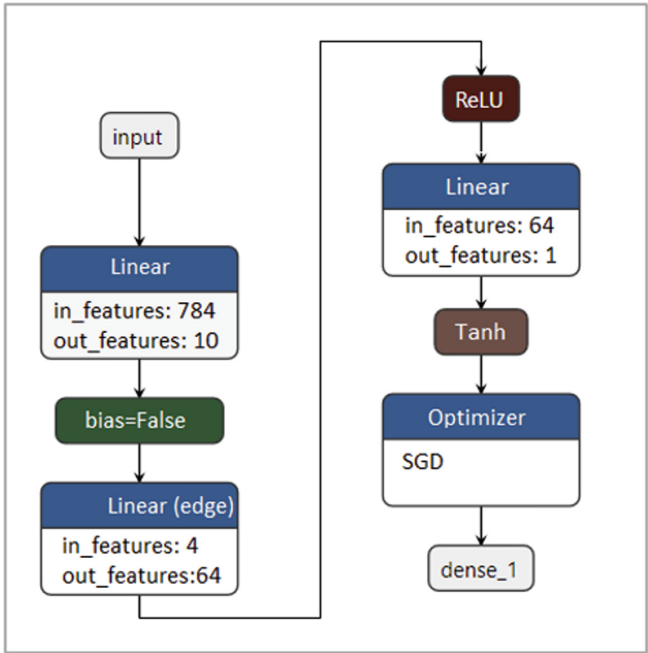


Fig. 7. Proposed fine-tune GNN model for classification of car-hacking dataset

C Performance Parameters

In this paper, detecting intrusions in a vehicle network is a multiclass problem, and the result of the detection is a DOS attack on the vehicle network. We use the F1-score, precision, recall, and accuracy as evaluation metrics to figure out how well our proposed model works. In the confusion matrix, the variables TP, FP, TN, and FN stand for the following: true negative (TN), true positive (TP), false negative (FN), and false positive (FP). we used the below formula for the calculation;

Accuracy (ACC):

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

Precision (PRE):

$$PRE = \frac{TP}{TP + FP} \quad (3)$$

Recall (REC):

$$REC = \frac{TP}{TP + FN} \quad (4)$$

F1 -Measure (F1):

$$F1 = 2 * \frac{PRE * REC}{PRE + REC} \quad (5)$$

D Result Analysis

We used the proposed GNN classifier to ensure that the prediction accuracy of a DL learning classifier, like GNN, is better than that of other DL classifiers. This was done to prove that the developed model was good at making predictions. In this study, the model parameters for binary classification on GNN and CNN were set using a 10-fold cross-validation scheme.

Table 1 provides the fine-tuned parameters for two classifiers: the CNN Classifier and the GNN Classifier. The CNN Classifier uses ReLU as the activation function and employs the Adam optimizer with a batch size of 64. It undergoes 25 epochs during fine-tuning. On the other hand, the GNN Classifier uses ReLU and Tanh as the activation function, and it uses SGD as the optimizer with a batch size of 28. Similar to the CNN Classifier, it also undergoes 25 epochs during fine-tuning. These fine-tuned parameters help adapt the pre-trained models to specific tasks, leading to improved performance and accuracy.

Table 1. Tuning Parameters for CNN and GNN

Parameters	CNN Classifier	GNN Classifier
Activation Function	Relu	Relu/Tanh
Optimizer	adam	SGD
Batch Size	64	28
Number of Epoch	25	25

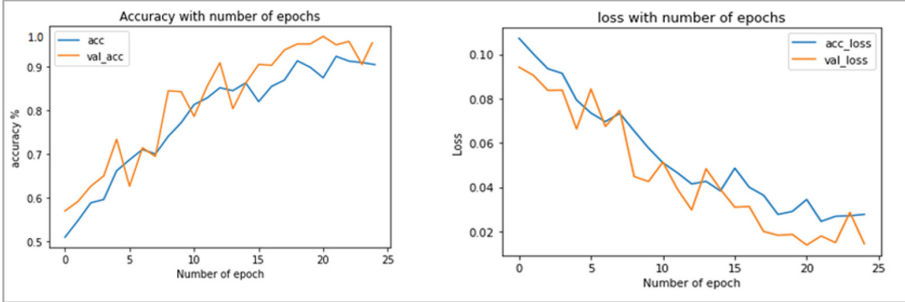


Fig. 8. Performance comparison graph of CNN (a) accuracy graph (b) loss graph.

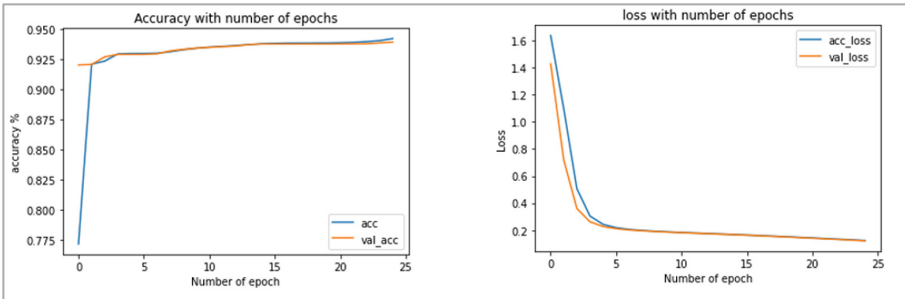


Fig. 9. Performance comparison graph of CNN (a) accuracy graph (b) loss graph.

Table 1 shows the parameters that can be used to fine-tune the DL classifier. Figure 8 shows a comparison of the accuracy and loss graphs of CNN algorithms with hyperparameters that have been fine-tuned.

Figure 9 shows the accuracy and loss graph comparison of CNN algorithms with fine-tuning of hyperparameters. Figure 9 shows the accuracy and loss graph comparison of GNN algorithms with fine-tuning of hyperparameters. GNN gives better accuracy compared to CNN as can be seen in Fig. 10

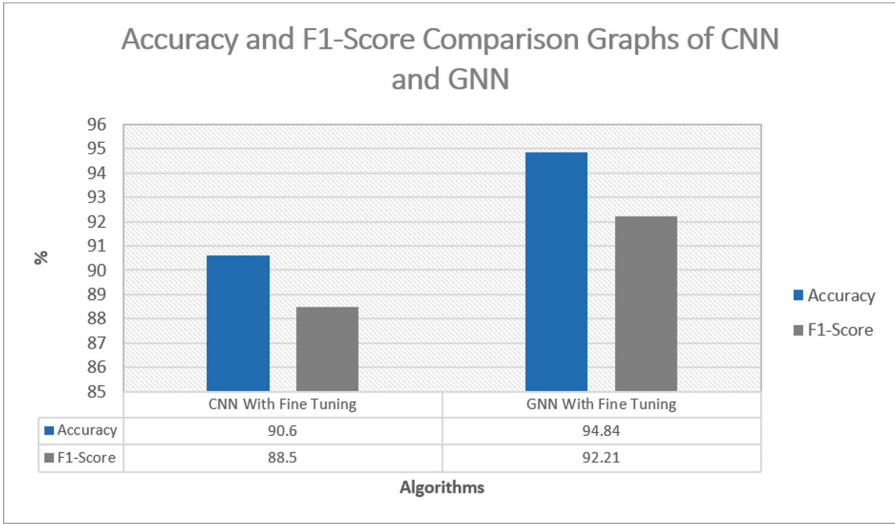


Fig. 10. Accuracy and F1-Score comparison graph of CNN and GNN

5 Conclusion

The use of DL techniques (CNN) and proposed graph-based techniques (GNN) for attack detection in in-Vehicle Network systems has shown promising results. With the increasing use of connected vehicles, the security of these systems is paramount to ensure the safety of passengers and prevent malicious attacks. We have used the car-hacking dataset in experiments where we have converted text data into image data and passed it to the proposed classifier. Models such as CNNs, GNN have been employed to identify different types of attacks in vehicle network systems, such as DoS attacks, fuzzy attacks, spoofing the drive gear, and spoofing the RPM gauge with the data stream. These models analyze large amounts of data in real time, detect anomalous behavior, and provide accurate predictions with high accuracy rates. We achieve an accuracy of 95 % using the GNN classifier with fine-tuning of hyperparameters. Future research should concentrate on enhancing these models' scalability and efficiency, as well as inventing new ways for detecting complex threats in real-time.

References

1. Khan, J., Lim, D.W., Kim, Y.S.: Intrusion detection system can-bus in-vehicle networks based on the statistical characteristics of attacks. *Sensors* **23**(7), 3554 (2023)
2. Hossain, M.D., Inoue, H., Ochiai, H., Fall, D., Kadobayashi, Y.: LSTM-based intrusion detection system for in-vehicle can bus communications. *IEEE Access* **8**, 185489–185502 (2020)
3. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **31**(7), 1235–1270 (2019)

4. Javed, A.R., Ur Rehman, S., Khan, M.U., Alazab, M., Reddy, T.: Canintelliids: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE Trans. Netw. Sci. Eng.* **8**(2), 1456–1466 (2021)
5. Zhu, H., et al.: Space-efficient optical computing with an integrated chip diffractive neural network. *Nat. Commun.* **13**(1), 1044 (2022)
6. Song, H.M., Woo, J., Kim, H.K.: In-vehicle network intrusion detection using deep convolutional neural network. *Veh. Commun.* **21**, 100198 (2020)
7. Zhou, J., et al.: Graph neural networks: a review of methods and applications. *AI open* **1**, 57–81 (2020)
8. Zhang, H., Xu, M.: Graph neural networks with multiple kernel ensemble attention. *Knowl.-Based Syst.* **229**, 107299 (2021)
9. Verma, A., Ranga, V.: Statistical analysis of cids-001 dataset for network intrusion detection systems using distance-based machine learning. *Procedia Comput. Sci.* **125**, 709–716 (2018)
10. Xu, H., Mueller, F.: Machine learning enhanced real-time intrusion detection using timing information. In: *International Workshop on Trustworthy and Real-time Edge Computing for Cyber-Physical Systems* (2018)
11. Wang, Y., Meng, W., Li, W., Li, J., Liu, W.X., Xiang, Y.: A fog-based privacy-preserving approach for distributed signature-based intrusion detection. *J. Parallel Distrib. Comput.* **122**, 26–35 (2018)
12. Xu, H., Fang, C., Cao, Q., Fu, C., Yan, L., Wei, S.: Application of a distance-weighted knn algorithm improved by moth-flame optimization in network intrusion detection. In: *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pp. 166–170. *IEEE* (2018)
13. Teng, S., Wu, N., Zhu, H., Teng, L., Zhang, W.: Svm-dt-based adaptive and collaborative intrusion detection. *IEEE/CAA J. Automatica Sinica* **5**(1), 108–118 (2017)
14. Liu, J., Xu, L.: Improvement of SOM classification algorithm and application effect analysis in intrusion detection. In: Patnaik, S., Jain, V. (eds.) *Recent Developments in Intelligent Computing, Communication and Devices*. *AISC*, vol. 752, pp. 559–565. Springer, Singapore (2019). https://doi.org/10.1007/978-981-10-8944-2_65
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016)
16. Ma, T., Wang, F., Cheng, J., Yu, Y., Chen, X.: A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors* **16**(10), 1701 (2016)
17. Sun, P., et al.: Dl-ids: Extracting features using CNN-LSTM hybrid network for intrusion detection system. *Secur. Commun. Netw.* **2020**, 1–11 (2020)
18. Eesa, A.S., Orman, Z., Brifcani, A.M.A.: A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Syst. Appl.* **42**(5), 2670–2679 (2015)
19. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT press, Cambridge (2016)
20. Wang, Z.: The applications of deep learning on traffic identification. *BlackHat USA* **24**(11), 1–10 (2015)
21. Jia, F., Kong, L.: Intrusion detection algorithm based on convolutional neural network. *Trans. Beijing Inst. Technol.* **37**(12), 1271–1275 (2017)
22. Peng, Y., Choi, B., Xu, J.: Graph learning for combinatorial optimization: a survey of state-of-the-art. *Data Sci. Eng.* **6**(2), 119–141 (2021)

23. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: International Conference on Machine Learning, pp. 2014–2023. PMLR (2016)
24. Zhang, Y., Liu, Q., Song, L.: Sentence-state LSTM for text representation. arXiv preprint [arXiv:1805.02474](https://arxiv.org/abs/1805.02474) (2018)
25. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7370–7377 (2019)
26. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint [arXiv:1503.00075](https://arxiv.org/abs/1503.00075) (2015)
27. HCRL Dataset. <https://ocslab.hksecurity.net/Datasets/car-hacking-dataset>



Evaluation of the Energy Viability of Smart IoT Sensors Using TinyML for Computer Vision Applications: A Case Study

Adriel Monti De Nardi  and Maxwell Eduardo Monteiro ^(✉) 

Instituto Federal do Espírito Santo Campus Serra, Serra, ES 29166-630, Brazil

maxmonte@ifes.edu.br

<https://ppcomp.serra.ifes.edu.br/>

Abstract. TinyML technology, situated at the intersection of Machine Learning, Embedded Systems, and the Internet of Things (IoT), presents a promising solution for a wide range of IoT domains. However, achieving successful deployment of this technology on embedded devices necessitates optimizing energy efficiency. To validate the feasibility of TinyML on embedded devices, extensive field research and real-world experiments were conducted. Specifically, a TinyML computer vision model for people detection was implemented on an embedded system installed in a turnstile at a Federal Institute. The device accurately counts people, monitors battery levels, and transmits real-time data to the cloud. Encouraging results were obtained from the prototype, and experiments were performed using a lithium battery configuration with three batteries in series. Hourly voltage consumption analysis was conducted, and the findings were illustrated through graphical representations. The camera sensor prototype exhibited a consumption rate of 1.25 V per hour, whereas the prototype without the camera sensor displayed a more sustainable consumption rate of 0.93 V per hour. This field research contributes to advancing TinyML applications and enriching studies regarding its integration with IoT and computer vision.

Keywords: tinyml · iot · embedded systems · computer vision

1 Introduction

The Internet is a global network that connects millions of computers [1]. Its technological advancement has brought significant transformations to society, with impacts on both the public and private sectors, as well as social, political, and economic contexts [2]. According to the “Digital: Global Overview Report” study, published in April 2023, the number of active users worldwide reached 5.18 billion, representing approximately 63% of the global population. The report indicates a significant increase in the number of internet users worldwide over

the past decade, with a doubled growth rate. Based on this trend, it is expected that this number will continue to grow in the coming years [3]. It is believed that the growth potential of the internet is moving towards a pervasive computing ratio [4].

With the advancement of the Internet and the potential for growth of connected objects, the term Internet of Things (IoT) has emerged over the years. IoT is a network in which “things” or devices embedded with sensors are interconnected through private or public networks. These devices can be remotely controlled to perform specific functionalities, and information can be shared through the networks using communication protocols. Smart things or devices can range from simple accessories to large machines, all equipped with some type of sensor [5].

This technology is increasingly present in everyday life. There are various application solutions in the development of IoT. For example, in smart homes, we have the Smart Home, which includes security systems and smart energy applications. In the context of intelligent transportation, we find fleet tracking as an example. In the field of smart healthcare, we have the surveillance of patients with chronic diseases. Finally, we can mention Smart City projects with real-time monitoring solutions, such as parking availability and smart street lighting” [6]. “A recent example occurred in the city of Christchurch, New Zealand, after an earthquake in 2011. The city was rebuilt and took the opportunity to implement a Smart City solution, installing sensors that collect real-time data on various aspects, from traffic flow to water quality. This provided greater urban efficiency and productivity, along with relatively low implementation costs for these IoT devices [7].

These intelligent solutions mentioned above, connecting so many ‘things,’ lead us to a new generation, an evolution of IoT called Internet of Things 2.0 (IoT 2.0), which involves data inference with artificial intelligence in devices, sensors, and actuators. IoT 2.0 will be a key to the digital transformation of an internet-connected society, aiming to generate intelligence from devices and enable real-time data sharing” [8]. “One of the possibilities for the use of Internet of Things 2.0 is Tiny Machine Learning¹ (TinyML). This technology involves the use of machine learning on low-cost and low-power microcontrollers that have limitations in memory and storage. In other words, extremely small devices execute machine learning models” [9]. “TinyML emerges with the purpose of addressing the challenges of memory and storage, enabling analysis of sensor data on the device and involving hardware, algorithm, and software considerations [10].

A recent application developed using TinyML technology is capable of detecting precise alcohol concentrations through temperature, humidity, and alcohol sensors on a low-power, small-sized microcontroller, resulting in high latency. This embedded device collects data and implements a machine learning model, processing the data directly on the device during testing. The study used Google Colab to build a TinyML model, with the assistance of a micro-library called

¹ <https://www.tinyml.org/home/>.

TensorFlow Lite. The objective of this study was to improve the accuracy in alcohol detection, considering the variation in environmental conditions [11].

1.1 The Problem and Proposal of the Study

Over the past few decades, there has been dedicated research effort towards improving embedded technologies for use in resource-constrained environments on the internet” [12]. “A successful implementation of IoT requires, among other desirable requirements, efficient power consumption by embedded devices, which is particularly important in the IoT application field. With the recent emergence of TinyML, this technology offers a potential application for IoT, standing out for its use of machine learning on small-scale embedded devices operating with limited battery resources and requiring optimization. This study can contribute to exploring and providing innovative solutions for the use of TinyML in the development of future applications [13].

The study of energy consumption, among other factors, is of great interest to the TinyML community, aiming for better development and improvement of this technology” [14]. “Embedded devices typically consist of a microcontroller with a processing unit connected to sensors. These sensors, responsible for data collection, can affect the battery life, especially when it comes to camera sensors. In this work, a camera sensor will be used, which poses a significant challenge for embedded systems. While the literature on TinyML demonstrates the possibility of using this technology in building IoT devices, in my research, I have not found studies that provide practical support in a real IoT scenario, utilizing TinyML and exploring its energy behavior. Therefore, this research aims to test the viability of TinyML as a sustainable device for IoT, analyzing battery consumption and durability, as well as proposing techniques and methods to enhance its performance [15].

However, in practice, in a case study and field test research, what would be the performance of this embedded device with TinyML technology? What would be the battery consumption of the device when operating in a specific application? What would be the durability of the device when operating in a real network? What would be the operational availability profile of the TinyML device? External environmental factors such as sunlight and rain, would they hinder its usefulness and application? Would the energy consumption results be significant?

Currently, researchers and experts are working on enhancing TinyML-based IoT for devices. One recommended device for this purpose is the microcontroller (MCU), which offers an increasingly popular option due to its affordability, convenience, and portability“ [16]. “This work aims to contribute to the advancement of future studies on TinyML in these intelligent IoT devices, providing beneficial indicators and identifying areas for improvement. An example is the evaluation of device durability in a real-world environment.

As mentioned earlier, the integration of machine learning (ML) into IoT devices aims to reduce the demand for memory, storage, and processing power,

with the goal of reaching a broader audience and offering new options for personal and professional services and applications. According to the reviewed literature, the purpose of this study is to conduct a case study of a ready-made computer vision solution, implementing it on an embedded device using TinyML technology and applying it in an IoT scenario. Real-time data will be captured through a camera sensor and stored in the network. Subsequently, an energy feasibility study will be conducted, analyzing the lithium battery consumption of the embedded device and investigating its behavior and utilized techniques. Furthermore, improvements will be proposed and presented through results and graphs. By the end of this work, it will be possible to evaluate whether the solution meets the project's expectations and demands, as well as provide insights for future research.

1.2 Limitations and Objectives of the Prototype

The prototype that will be developed has certain limitations aimed at delimiting the scope of the field test research in its development and evaluation. These limitations are as follows:

Limited Functionality: The prototype is designed to demonstrate a specific set of features and functionalities. It may not cover the full range of capabilities that can be achieved in a fully developed system. Therefore, the limitation of functionality will be that of the proposed people counting feature and its battery level.

Scale and Scope: In a real IoT network, there are millions of devices and various IoT gadgets. However, due to the project being conducted by a single person, the IoT network will be limited, with few available gateways, resulting in a reduced statistical significance compared to a more robust application. With that said, the prototype is typically built on a smaller scale and may not be representative of full-scale deployment. It focuses on addressing specific objectives and may not cover all possible scenarios or use cases.

Performance Constraints: Due to the nature of the prototype, there may be performance constraints in terms of speed, accuracy, or efficiency. These limitations are taken into account during the design and evaluation process. In this project, these constraints include the available budget and the use of a limited number of devices, restricted to those available for the field test. This will also affect the coverage of the IoT network space that will be utilized.

Validation and Verification: The prototype aims to validate and verify the feasibility of certain concepts or technologies. It may not be a fully validated and verified solution ready for commercial deployment.

Time and Labor: This work will address only a single TinyML application, with only two researchers involved in the research. Therefore, the established time limit for completing the master's project must be respected.

The overall objective of this work is to evaluate the feasibility of TinyML technology in embedded devices in an intelligent IoT scenario. To achieve this, specific objectives include selecting an embedded intelligence application in IoT devices, along with the corresponding technologies; integrating an embedded

intelligence application solution using TinyML; developing an information controller to send and store data in a computer network; choosing the proposed field test environment and planning the validation of the selected embedded devices; and publishing and comparing the results obtained from the smart devices.

1.3 Related Works

According to [17], billions of devices connected to the IoT are powered by batteries. The IoT is applied in various areas such as Smart Cities, Smart Energy, and Smart Environment. The performance of these areas relies on the optimized use of battery life, which becomes a challenge. As mentioned by [18], it is crucial to understand in detail the charge and energy consumption of battery-powered devices.

In Fig. 1, a comparison of active energy consumption is presented between traditional Machine Learning (ML) devices and those supported by TinyML. On the horizontal axis, we have the NDP100 (consumption of 150 μ W) and Cortex-M7 (consumption of 23.5 mW) devices, which belong to the TinyML system, while the RasPi 4 (consumption of 3.4 W) and 6049GP-TRT (consumption of 2000 W) belong to the MLPerf systems. By analyzing the vertical axis, we can conclude that TinyML systems can have up to four times less energy consumption compared to state-of-the-art ML systems, representing a significant energy saving.

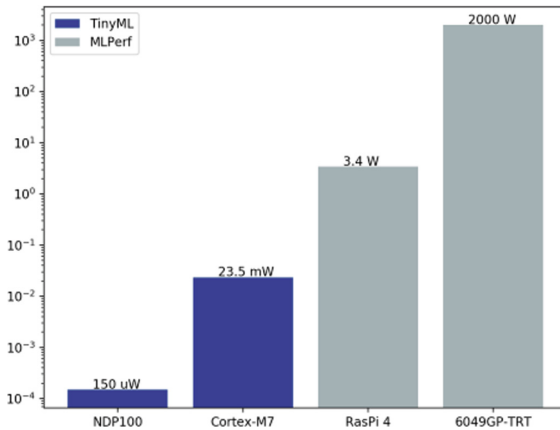


Fig. 1. A comparison of active energy consumption between ML systems and those supported by TinyML [19].

According to the study by Khan et al. [5], TinyML has the capability to run computer vision models. The study highlights research in Machine Learning (ML) and computer vision across various domains, divided into different fields. The research fields include Weather Forecasting, Agriculture, Medical Sciences,

Professional Sports, and Facial Recognition, each representing 6%. The fields of Industries, Professional Sports, and City Traffic Levels have a representation of 12% each, while Biological Sciences and Human Activity are the most representative fields, each with 19%.

Regarding ML applications in computer vision, it is observed that they have been successfully used in various areas such as weather forecasting, biological sciences, expression reading, food security, species classification, sports, traffic flow monitoring, and predictive maintenance in industries. Some areas, such as biological sciences, human activity interpretation, traffic management, and professional sports, are emerging. The study concluded that object detection, classification, and prediction are the most frequently performed tasks in computer vision with ML. All these areas mentioned in the research can be applied in the context of the Internet of Things (IoT), emphasizing the importance of studies on energy efficiency in this field.

The study by Jolly et al. [18] addresses the end-of-life behavior of batteries in IoT devices. The study utilizes measurements from the battery of the IoT device Keysight X8712A at different voltage levels. It was observed that the charge consumption varies as the voltage decreases, providing valuable insights for optimizing battery life.

Several strategies to extend battery life are presented, such as disabling non-essential features, reducing sensor measurement frequency, minimizing data transmission, and alerting users when the battery level is low. These measures aim to improve customer satisfaction and add value to IoT devices.

Overall, these studies highlight the importance of energy management and conservation in IoT devices, offering insights and strategies to optimize energy consumption and extend battery life. This is crucial for the performance and efficiency of these devices, providing benefits to end-users.

2 Method

To achieve the objectives of this work and assess the energy availability and behavior of TinyML technology in IoT devices using a computer vision model, this chapter discusses the materials and methods that will be employed. It begins by describing the methods, which include field research and the project development flow, followed by an explanation of how the TinyML application was developed. Lastly, the materials and technologies used in the development of this work will be presented in detail.

2.1 Study Design

The project will be applied in an empirical case study research. Conducting experimental research is widely used to study software and protocol design for IoT use cases. These experiments can be used to verify or even refute theories or simulations of the researched model, which is a crucial point for the test we will use through case studies. In addition to the initial project, suggestions

and improvements for its study will be proposed in this work. Regarding field research, guidance is provided on how to conduct field research with market-oriented research lessons, which can be divided into four activities: choosing what to study, whom to study with, conducting the research, and elaborating the results [20]. Based on these four activities, we have chosen what to study and whom to study with, focusing on the energy efficiency of a pre-trained computer vision TinyML model. Some changes were made to the code, using appropriate tools for inference on embedded devices and applying it in a real IoT scenario, which was tested at the Federal Institute of Espirito Santo. After conducting the research, the results were elaborated and discussed. Various different tools are used for experimental research in IoT systems, including the implementation of the approach itself, software platforms, frameworks, and hardware. Additionally, tools are used to schedule, execute, control, and evaluate the necessary experiment [21].

Defining the field research methodology for the prototype to run its experiment in the case study, there is a flow to be followed before developing ML to TinyML. The following flowchart in Fig. 2 addresses each stage of the development flow of the chosen ML model until its implementation on an embedded device. The first stage is “Choose a model,” which will be an existing application model for computer vision in the TensorFlow Lite version. The next stage is “Convert the model,” but this work does not aim to convert the TensorFlow model to the lite version, as the lite version of the model has already been selected, skipping this part. In the third stage, “Perform inference with the model on the microcontroller,” a custom framework will be used for this procedure to bring intelligence to the embedded system. Finally, we have the optional last stage, “Optimize your model.” In this stage, the model was customized to count people, aiming to have a computer vision application to verify the battery’s viability. In the optimization development stage, the C++ language was used, and then the model inference was executed on the microcontroller through a Linux terminal [22].

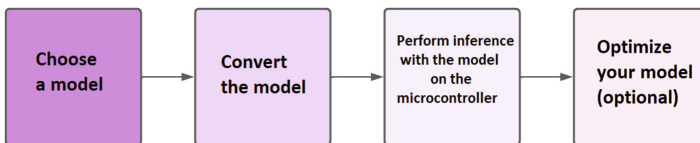


Fig. 2. Flowchart of the work development [22].

2.2 Prototype Development

In this section, we will describe the development of the prototype. Firstly, among the various existing TinyML models, it was recommended to conduct an experiment that involves the use of a camera sensor, which is a critical aspect in

IoT. Therefore, a TinyML computer vision model was chosen that detects people using the camera sensor. The code was optimized to count the number of people once the detection is performed, and this technology was inferred in the embedded system. The entire process, from model selection to inference on the embedded system, will be detailed. The study of the lithium battery's performance over time is achieved through a voltage divider, and the development of another controller that retrieves information about the people count and current battery level, and sends the data over the network to be saved in a non-relational database.

TinyML Model for Computer Vision: The Espressif platform provides resources for application developers to realize their ideas, particularly for IoT applications. Therefore, it was chosen for development in this work along with the TinyML model. Before selecting the TinyML computer vision model, it is necessary to install and configure the ESP-IDF framework. The programming guide can be found on the Espressif website [23], as referenced. According to the documentation, a few prerequisites need to be installed, such as CMake and Ninja build tools. To obtain the chosen model, Git needs to be installed for cloning. The framework will be operated through the command line of the Linux Ubuntu 20.04 LTS operating system, which was the latest stable version at the time of development. The ESP-IDF framework was also installed via the Linux terminal, and version 4.1 (release version) was used [24].

The first step involves installing the aforementioned prerequisites. The second step is to install ESP-IDF, followed by the third step, which is setting up the tools. In step 4, the environment variables are set up, and then the TinyML project with computer vision can be developed.

The directory structure includes a term called “Component,” where functionalities can be implemented. In this project, the chosen model comes pre-installed and configured with the esp32-camera component (for accessing the camera sensor) and the tflmicro library (TFL library for microcontrollers). The model is no longer located in the original directory developed on the TensorFlow Lite GitHub repository [25].

The model detects people through the camera sensor. The author added a counter during the detection process and activated a digital output pin to communicate with the ESP8266 board. After optimizing the code, the project is built, flashed, or monitored directly from the terminal [26]. Now, the model is ready to be used practically in a real-world environment.

Data Controller for Network Transmission: For the data controller for network transmission, the Arduino IDE was used to develop the code. When the computer vision application detects and counts a person, the output pin on the ESP32-CAM signals this by activating an LED, indicating that a person has been detected. This information is then sent to the ESP8266. In addition to obtaining a people count, this microcontroller also includes a voltage divider to calculate the level of the lithium battery, with the value obtained from an

analog pin. Furthermore, a second count was implemented in the prototype to track the duration of its operation while the battery powers the entire system. The ESP8266 microcontroller sends the data through its Wi-Fi connection to a gateway for further transmission on the network [27].

2.3 Materials and Technologies

In this section on Materials and Technologies, we discuss the hardware components used to assemble the prototype, the software tools utilized for application development, and the complementary accessories that, when combined with the hardware and software, form the complete prototype.

ESP32-CAM: The ESP32 CAM board is a low-cost microcontroller equipped with Wi-Fi and camera sensor, ideal for projects involving computer vision, IP cameras, and video streaming. Supporting both OV2640 and OV7640 cameras, along with built-in flash, it has a low-power 32-bit CPU and a clock speed of 160MHz. It has various built-in sleep modes and a CAM SRAM board, enabling easy implementation of low-power projects. Its Wi-Fi connectivity allows for live data visualization with high-quality, real-time images. Encoding can be done using the FTDI cable or the Module B (also called shield), which provides a micro USB port for easier microcontroller programming, as well as additional hardware protection. With all these features, the ESP32 CAM board is an excellent option for IoT projects and others that require a microcontroller with computer vision capabilities [28]

ESP8266: In the proposed project, the NodeMCU is used to capture the battery level at the time of person detection, total people count, and the runtime of the application. The NodeMCU is a low-cost microcontroller that runs firmware on the ESP8266 WiFi SOC, developed by Espressif Systems. This hardware is an opensource platform and can be worked with IoT due to its features. The NodeMCU is based on the ESP-12 module and has 16 digital I/O pins and one analog pin. Additionally, it has Wi-Fi connectivity features and operates at a voltage of 3.3V. Using the Wi-Fi connection, it sends this data through a gateway to the network, where the database receives them in real-time, allowing them to be viewed through a computer or smartphone [28]

Framework ESP-IDF: According to Espressif's official documentation, the Espressif IoT Development Framework (ESP-IDF) is an application development framework intended for the System-on-Chips (SoCs) of the ESP32, ESP32-S, and ESP32-C series. The use of ESP-IDF allows the configuration of the software development environment for hardware based on the ESP32 chip, as well as enabling the modification of the board on which the application will be used. It is possible to customize the menu of sensors and applications, as well as build and update the firmware on an ESP32 board according to the specific needs of the project [29].

Firestore Realtime Database: Developed by Google, Firestore Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in real time. For each connected client, the data is available and remains visible even when the app is offline. It allows for the creation of cross-platform applications for Arduino, Apple, Android, and JavaScript. All clients share a single instance of the Realtime Database and receive automatic updates with the latest data received. The cloud database is a NoSQL, non-relational database, where there is no direct mapping of classes to the database. In this work, we collect battery level and people count information through sensors, with the objective of gathering data from these sensors and sending the information over the internet to a cloud server. Hence, Firestore was a suitable choice [30].

3 Case Study

For the assembly of the prototype, in each case study, all the hardware components were mounted on a sturdy clipboard material using wires and screws to provide additional stability. In addition to some parts, hot glue was used for enhanced protection. A protoboard was used, along with jumpers, to interconnect the microcontrollers and the 18650 lithium batteries, which power the system. Diodes and resistors were also utilized.

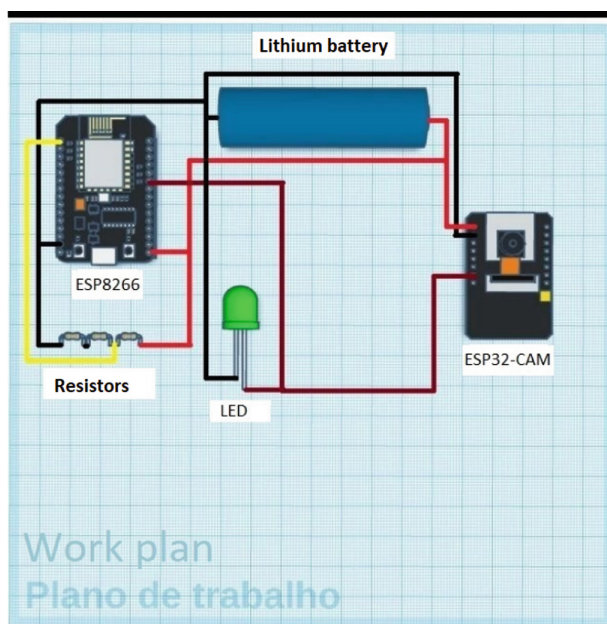


Fig. 3. Prototype Work Plan 1

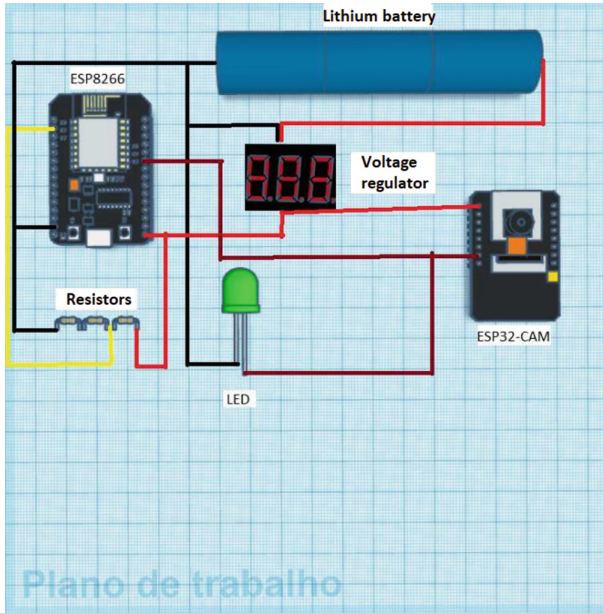


Fig. 4. Prototype Work Plan 2

For the computer vision application, a system capable of recognizing people through a camera sensor was chosen. With the model inferred in the embedded system, the code was modified to count the number of people detected and activate an output port indicating that the person was detected. Firstly, the application detects and counts people through a camera sensor using artificial intelligence. Then, the controller sends the data over the network and stores it in a non-relational database. With this system, it is possible to monitor the presence of people in a given space and collect valuable information for analysis and decision-making [14]. The TinyML system and the control system are interconnected through a protoboard, while the lithium battery provides all the necessary power for the system. As shown in Fig. 3 we can see how the first prototype was assembled, using a single battery module. From there, improvements were made for a second prototype, as shown in Fig. 4 where a case with three batteries in series and a voltage regulator were used to power the circuit at 5 V.

3.1 Results and Discussion

To generate the results, we selected the Federal Institute of Espírito Santo (IFES) as the field research location. IFES is an academic center that offers courses ranging from technical programs to doctoral degrees. Currently, it has 22 campuses, with a total of 1300 professors and 36,000 students. Among these campuses, we chose the Serra Unit to install the prototype. According to information from the

Academic Records Coordination (CRA) and the General Coordination of People Management (CGGP), the Serra Campus has 1679 students and 180 staff members, including permanent, temporary, and reinstated personnel. In terms of physical space, the Serra Campus consists of 09 blocks, with a total land area of 150,000 square meters. The prototype was installed at the entrance, near the turnstiles, where there is a higher flow of people accessing the Serra Campus [31].

With the data collected from the non-relational database, using the high-level language Python [32] and the Google Colaboratory (Colab) tool [33], graphs were plotted to visualize the proposed study. Each 18650 model battery used in the prototype has a capacity of 3000mAh and a current rating of 30A.

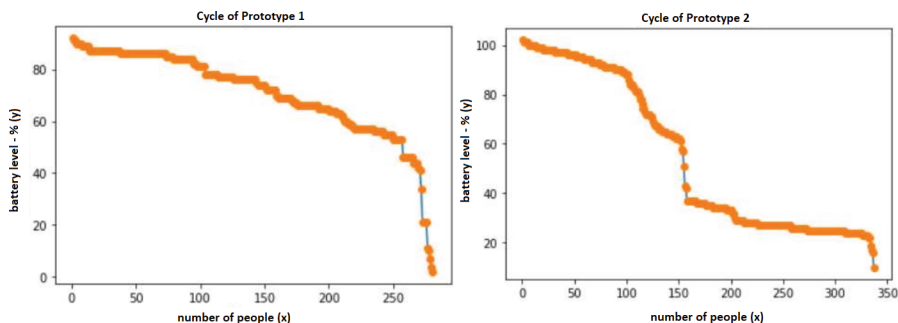


Fig. 5. Prototypes cycle - Axis x(People) / Axis y(Battery Level)

First Prototype: In this prototype, we used a single 18650 battery connected to the lithium charger module. Since the circuit requires 5V, the battery has 4.2V, but the module circuit has a 5V output, which can power the prototype circuit this way. As can be seen in the left of the image as shown in the Fig. 5 281 people were detected on the x-axis, and on the y-axis, the battery level consumed. The cycle detected a total of 281 people in a time of 2 h and 21 min. In the next prototype, we will use a multimeter to obtain more research data on battery power consumption during circuit use.

Second Prototype: In this prototype, three 18650 battery models were used, connected in series through a battery case, allowing an output of up to 12V. To power the circuit, a voltage regulator was used to convert the 12V voltage to 5V. Two tests were performed, one with the camera sensor module connected to detect people, and another without the camera sensor module to compare battery consumption. A graph was also generated to analyze the consumption of the three batteries over a six-hour interval, which was the maximum time the circuit with the camera sensor module was able to support, generating significant results.

Without Camera Sensor Module: In this test, no people were detected, so it was not necessary to generate a counting graph. However, in to the right of the image as shown in the Fig. 5, a graph of the voltage consumption of the batteries in series without the camera sensor module was presented. On the x-axis, we have a time interval of up to 6 h, and on the yaxis, the initial voltage of the 3 batteries, allowing for visualization of their behavior. Battery 3 had its consumption changed between 4 and 5 h of prototype use, a longer time compared to the graph in Fig. 6. All batteries were measured every hour using a multimeter. Initially, when the circuit began to be powered, the

Without Camera Sensor Module: In this test, no people were detected, so it was not necessary to generate a counting graph. However, in Fig. 6, a graph of the voltage consumption of the batteries in series without the camera sensor module was presented. On the x-axis, we have a time interval of up to 6 h, and on the yaxis, the initial voltage of the 3 batteries, allowing for visualization of their behavior. Battery 3 had its consumption changed between 4 and 5 h of prototype use, a longer time compared to the graph in Fig. 6. All batteries were measured every hour using a multimeter. Initially, when the circuit began to be powered, the three batteries in series totaled 11.86 V. In the end, the batteries presented a total voltage of 5.58 V.

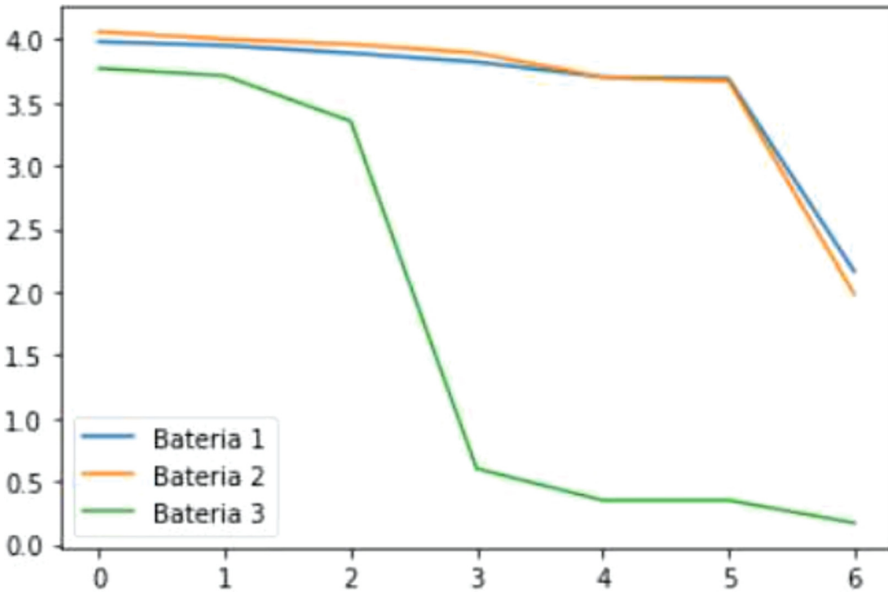


Fig. 6. Batteries with camera sensor - X-axis(Hours)/Yaxis(Battery voltage)

Final Considerations: Firstly, it is important to highlight that accurately measuring the consumption of a lithium battery is a challenge that has been addressed by various researchers, such as [Kravari et al. 2017] and [Jolly 2019]. To overcome this difficulty, we used a multimeter to obtain more precise measurements from prototype 2, allowing for a more accurate analysis of the consumption of the TinyML model in conjunction with the circuit and microcontrollers. In the first test with prototype 1, no significant changes were observed in its behavior, as shown in the graph of Fig. 4. The proposed TinyML model has a relevant accuracy for people detection. However, the objective of this work was not to precisely measure people detection, but rather to evaluate how TinyML, in conjunction with a computer vision application, behaves in the context of IoT in terms of battery consumption. Thus, our focus was to present data on the lifespan of the lithium battery that powers the entire embedded device. Considering that a 3000 mAh, 30 A model 18650 lithium battery has a maximum voltage of 4.2 V and a minimum of 3.7 V, we analyzed two different cases with prototype 2. In the first case, where the camera sensor was used, the batteries in series started with 11.81 V and ended with 4.31 V after 6 h of use, totaling a consumption of 7.5 V, or 1.25 V/h. In the second case, where the camera sensor was not used, the batteries started with a total of 11.86 V and ended with 6.28 V after 6 h, totaling a consumption of 5.58 V, or 0.93 V/h. These data indicate that the camera sensor has a significant energy expenditure in a computer vision application with machine learning. Based on the results obtained, we can conclude, according to [Banbury et al. 2020], that TinyML presents much higher energy efficiency in microcontrollers with low processing, memory, and storage. Compared to large-scale machine learning models, this experiment would be unfeasible. Although the application serves its functions for several hours, improvements can be proposed to further increase the battery life in an IoT scenario. An interesting solution to this problem was presented by [Srinivasan et al. 2019], who propose energy management and conservation strategies. Thus, we suggest some improvements in future works for this study.

4 Conclusion and Future Work

Although existing literature on TinyML points to its potential use in IoT devices, no studies were found that prove its energy viability in this field test context. Given this gap, the present study was developed with the aim of evaluating the energy viability of intelligent IoT sensors using TinyML for computer vision applications in a real-world environment. TinyML technology emerges as a promising solution to reduce memory, storage, and computational processing in IoT devices. Additionally, computer vision applications in IoT environments have significant potential in various areas, as indicated by previous studies in this work, both in the IoT and computer vision scenarios. Field research conducted at a Federal Institute revealed the voltage expenditures of the prototype and allowed for the identification of improvements to advance the use of TinyML technology. To increase battery life in application usage, for future studies, it is

suggested to add an Arduino Uno that receives 12V and supplies 5V to the microcontroller, along with the data to be sent every 10 min in deep sleep mode. During this period, the data controller enters a low-power state. Another suggestion is to power the circuit through a solar kit to prolong the life of the lithium battery. These ideas of disabling resources in the microcontroller during specific time intervals, such as Wi-Fi, which consumes high energy, are mentioned in related works. With these improvements and suggestions for future work, a longer battery life is expected, contributing to the advancement of TinyML technology in the field of IoT.

Acknowledgments. I would like to express my sincere gratitude for the incredible teaching I received during my academic journey. I know that everything I have learned would not have been possible without the commitment, dedication, and passion that you put into your classes every day. The knowledge I have gained thanks to you is something I will carry with me forever. Additionally, I would like to thank FAPES for granting me the scholarship that made it possible for me to pursue my master's degree at IFES. Without this opportunity, I would not have had access to all the learning and academic community I have encountered. Finally, I would like to thank IFIP-IOT for giving me the chance to publish my research work. It is an honor to be able to contribute to the academic community and I hope that my work can be useful to those who read it.

References

1. Vangie, B.: What is the Internet?—Webopedia (2021). <https://www.webopedia.com/definitions/internet/>. Accessed 11 Feb 2021
2. Zimmer, E.F., et al.: A influência da internet na saúde psicossocial do adolescente: revisão integrativa. *Revista Brasileira de Enfermagem, SciELO Brasil* **73**(2) (2020)
3. Datare Portal. Digital 2023: Global Overview Report (2023). <https://datareportal.com>. Accessed 01 Jan 2023
4. Umar, F.M., et al.: A review on internet of things (iot). *Int. J. Comput. Appl.* **113**(1), 1–7 (2015)
5. Ahmad, K.M., Khaled, S.: IoT security: review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **82**, 395–411 (2018)
6. Felix, W., Kristina, F.: Internet of Things. *Bus. Inf. Syst. Eng.* **57**(3), 221–224 (2015)
7. Sarah, M.: Lamppost shines a light on smart cities (2015). <https://www.ft.com/content/53b285c8-851d-11e4-ab4e-00144feabdc0>. Accessed 11 Feb 2021
8. Stefano, N., et al.: Iot 2.0 and the internet of transformation (2020)
9. Ji, L., et al.: Mccnet: tiny deep learning on IoT devices. *arXiv preprint arXiv:2007.10319* (2020)
10. de Prado, M., et al.: Robust navigation with tinyml for autonomous mini-vehicles. *arXiv preprint arXiv:2007.00302* (2020)
11. Vitthalrao, L.S., et al.: Alcohol sensor calibration on the edge using tiny machine learning (tiny-ml) hardware. In: *ECS Meeting Abstracts*, p. 1848. IOP PUBLISHING (2020)
12. Lachit, D., Swapna, B.: Tinyml meets IoT: a comprehensive survey. *Internet Things* **16**, 100461 (2021)

13. Somayya, M., et al.: Internet of things (IoT): a literature review. *J. Comput. Commun.* **3**(05), 164 (2015)
14. Stanislava, S.: TinyML for Ubiquitous Edge AI (2021)
15. Chandrasekar, V., et al.: Democratization of AI, albeit constrained iot devices & tiny ml, for creating a sustainable food future. In: *IEEE 2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 525–530 (2020)
16. Hiroshi, D., Roberto, M., Jan, H.: Bringing machine learning to the deepest IoT edge with tinyml as-a-service (2020)
17. Kalliopi, K., Theodoros, K., An, P.: Towards an iot-enabled intelligent energy management system. In: *IEEE 2017 18th International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering (ISEF) Book of Abstracts*, pp. 1–2 (2017)
18. Brad, J.: The last thing IoT device engineers think about: end of battery life behavior for IoT devices. In: *IEEE 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 837–840 (2019)
19. Colby, R.B., et al.: Benchmarking tinyml systems: challenges and direction. *arXiv preprint arXiv:2003.04821* (2020)
20. Jaworski, B.J., Kohli, A.K.: Conducting field-based, discovery-oriented research: lessons from our market orientation research experience. *AMS Rev.* **7**, 4–12 (2017)
21. Bin, Z.Y., et al.: Internet of things (IoT): operating system, applications and protocols design, and validation techniques (2018)
22. TensorFlow. TensorFlow Lite (2022). <https://www.tensorflow.org/lite/guide?hl=pt-br>. Accessed 21 Mar 2022
23. Espressif ESP-IDF (2022). <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>. Accessed 22 Mar 2022
24. Ubuntu, U.: <https://ubuntu.com/download>. Accessed 21 July 2021
25. Chugh, S.: <https://github.com/sunnychugh/tensorflow/>. Accessed 01 Aug 2021
26. TensorFlow Blog (2021). <https://blog.tensorflow.org/2020/08/announcing-tensorflow-lite-micro-esp32.html>. Accessed 01 Aug 2021
27. EzContents blog (2021). <https://ezcontents.org/esp8266-battery-level-meter>. Accessed 01 Aug 2021
28. Abhinab, S., Diwan, R.: IoT based load automation with remote access surveillance using esp 32 camand esp 8266 module. *Ann. Romanian Soc. Cell Biol.*, 6904–6914 (2021)
29. Espressif ESP-IDF. Disponível em (2022). <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>. Acesso em 22 Mar 2022
30. Firebase (2022). <https://firebase.google.com/docs/database?hl=pt>. Accessed 25 Mar 2022
31. Ministério da Educação (2022). <https://www.ifes.edu.br/>. Accessed 13 May 2022
32. Welcome to Python (2022). <https://www.python.org/> Accessed 13 May 2022
33. Google Colab. (2022). <https://colab.research.google.com/>. Accessed 13 May 2022
34. Xian-Da, Z.: Machine learning. In: *A Matrix Algebra Approach to Artificial Intelligence*, pp. 223–440. Springer, Heidelberg (2020)
35. Korablyov, D.: <https://medium.com/@dmytro.korablyov/first-steps-with-esp32-and-tensorflow-lite-for-microcontrollers-c2d8e238accf>. Accessed 02 May 2022



Simulated Annealing Based Area Optimization of Multilayer Perceptron Hardware for IoT Edge Devices

Rajeev Joshi^(✉), Lakshmi Kavya Kalyanam, and Srinivas Katkoori

Department of Computer Science and Engineering, University of South Florida,
Tampa, FL 33620, USA

{rajeevjoshi,lakshmikavya,katkoori}@usf.edu

Abstract. The deployment of highly parameterized Neural Network (NN) models on resource-constrained hardware platforms such as IoT edge devices is a challenging task due to their large size, expensive computational costs, and high memory requirements. To address this, we propose a Simulated Annealing (SA) algorithm-based NN optimization approach to generate area-optimized hardware for multilayer perceptrons on IoT edge devices. Our SA loop aims to change hidden layer weights to integer values and uses a two-step process to round new weights that are proximate to integers to reduce the hardware due to operation strength reduction, making it a perfect solution for IoT devices. Throughout the optimization process, we prioritize SA moves that do not compromise the model's efficiency, ensuring optimal performance in a resource-constrained environment. We validate our proposed methodology on five MLP benchmarks implemented on FPGA, and we observe that the best-case savings are obtained when the amount of perturbation (p) is 10% and the number of perturbations at each temperature (N) is 10,000, keeping constant temperature reduction function (α) at 0.95. For the best-case solution, the average savings in Lookup Tables (LUTs) and filpflops (FFs) are 24.83% and 25.76%, respectively, with an average model accuracy degradation of 1.64%. Our proposed SA-based NN optimization method can significantly improve the deployment of area-efficient NN models on resource-constrained IoT edge devices without compromising model accuracy, making it a promising approach for various IoT applications.

Keywords: Machine learning · Edge-AI · Metaheuristic · Smart Embedded systems · FPGA · Optimization

1 Introduction

Deep Neural Networks' success has been largely attributed to the construction of highly complex larger neural networks (NNs). The excessive parameterization of these NNs results in extremely accurate models. This enables these models

to perform more effectively and accurately across a range of applications, such as image classification [12], object detection, etc. However, this also significantly raises the cost of their use in terms of resource utilization. These highly parameterized NN models require massive computational costs and memory requirements, making distribution more difficult. Larger NN models also take longer to run and necessitate more hardware resources. This is really a serious concern when designing a NN hardware model for resource-constrained environments for real-world applications.

The surge in the use of IoT edge devices because of their portability, lightweight, and low power requirements [8, 11] demands an efficient NN hardware model. Hence, there is a great demand for lightweight, compressed versions of these NN models that can be easily deployed for IoT applications such as home automation, healthcare, automobiles, transportation, etc. As a result, researchers are focusing on designing lightweight, compressed NN models with high accuracy. Some of the research topics that have been studied include pruning, quantization, knowledge distillation, the co-design of NN architecture and hardware, and efficient NN model architecture design [6]. There are still a ton of untapped research areas that can be immensely useful for creating efficient compressed NN models for resource-constrained environments.

We propose a metaheuristic algorithm, SA based NN optimization methodology to build an energy-efficient, lightweight, and compressed NN hardware model for resource-constrained environments. In our proposed methodology, we fine tune the micro-architectural parameters (neuron weights) of the hidden layer of the MLP. We consider a single hidden-layer MLP for this work. The fine-tuning procedure is carried out in two parts. The SA algorithm is customized to generate an optimized MLP model. A subset of hidden layer neuron weights is used for perturbation in the custom-modified SA algorithm. During the perturbation, the generated weights whose values are close to integer are rounded to the next-nearest integer value. Once the optimized MLP model is generated, we apply hardware optimization approaches to further compress it. We use operator strength reduction techniques such as bit shifting for replacing multiplication operations for all the weights whose values are 1 and multiple of power of 2, (2^m , where m is the indices). We prune down all the weight with 0 values. We also simplify multiplication operations of the weights whose values are multiple of power of ($2^m + 1$) and ($2^m + 2$) using the multiplier strength reduction and addition operations. This results in an optimized, lightweight, and compressed MLP model. Based on an estimation of the hardware resources utilized by a single unit of the IEEE-754 single-precision FP32 multiplier and adder architecture, we evaluate the performance of our hardware MLP model inference architecture. The hardware resource utilization of a single unit of a multiplier and an adder is 60 LUTs and 51 LUTs, respectively.

In our initial exploratory research [9], we conducted three different experiments where we kept the constant value of α at 0.95, p at 10 %, and varied the value of N from 100 to 10,000. Our results showed that there was a notable increase in the savings of LUTs and FFs as N increased. On the most favorable

outcome, we attain an average savings of 24.45% in FFs and 25.51% in LUTs. Furthermore, in our subsequent research work [10], we conducted twelve different experiments where we kept the value of p constant at 10% and varied the values of α and N from 0.80 to 0.99 and 100 to 10,000, respectively. We also performed resizing of the registers to decrease the memory requirements for storing all integer weights, which resulted in significant savings in FFs, area, and power when implementing an ASICs based design. For area, FFs, and power, the optimal scenario yields savings of 27.53%, 27.71%, and 27.28%, respectively.

In this paper, we extend upon our preliminary research by investigating how the values of p and N affect the results when α is held constant at 0.95 and also simplifying the multiplication operations of the integer weights to further save the memory. We perform an extensive experiment to validate our NN model using five well-known classification datasets. Twelve different experimental observations have been performed using different perturbation amounts, p of hidden layer neuron weights. For each p , we conduct different experiments by varying the number of iterations, N , used in the custom-modified SA algorithm. The values of N used in this work are 100, 1,000, and 10,000, respectively, for all the datasets. We find the best optimal solution for each of the five classification datasets with $p = 10\%$ and $N = 10,000$. Overall, we observe an average savings of 24.83% in LUTs and 25.76% in FFs as compared to the regular NN model. These experiments are performed to validate our proof-of-concept. The experimental findings show promising results that can be further explored to design a much more complex NN model.

The rest of the paper is structured as follows: Sect. 2 presents background on NN and multilayer perceptron (MLP), SA, and an overview of the existing literature. Section 3 presents our proposed methodology. Section 4 reports experimental comparisons and outcomes along with a discussion of the findings. Finally, we present the conclusion and potential future perspectives in Sect. 5.

2 Background and Related Work

In this section, we review the contemporary works on NNs, simulated annealing, and related work to design efficient NN models.

2.1 Neural Networks and Multilayer Perceptron

Neural networks mimic the activity of the human brain, enabling AI, machine learning, and deep learning systems to spot patterns and solve common issues. These networks, also known as “artificial neural networks” are subfields of machine learning. These networks have nodes that are interconnected and arranged in layers. In general, they have three layers: an input layer, single or multiple hidden layers, and an output layer. The term “Multilayer Perceptron” refers to a fully connected multilayer neural network. Figure 1 shows an MLP with a single hidden layer. Each of these layers computes and passes on

the values to the next layer, which is calculated using Eq. 1.

$$H_i = \sum_{i=1}^n x_i * w_i + b_i \quad (1)$$

The nodes, or neurons, are assigned a weight W and a threshold, and after performing the necessary computation, they are forwarded further [4,13]. The hidden layers are computationally heavy, and all the neurons are connected to all the layers in the previous layer and the subsequent layers, to be called fully connected layers.

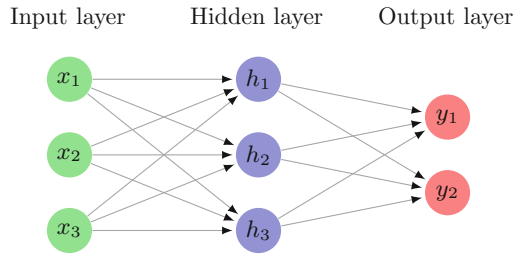


Fig. 1. A 3-input 2-output Multilayer Perceptron

2.2 Simulated Annealing

The simulated annealing algorithm is a global search optimization method inspired by the metallurgical annealing process. It is a technique used to solve unconstrained and constrained optimization issues. The fundamental approach for implementing this analogy to the annealing method consists of generating randomized spots in the proximity of the current optimal point and evaluating the problem functions there [14]. In ML, this algorithm can be used to determine the optimal features during the feature selection procedure by simulating this process. If the value of a cost function is less than its current best value, the point is accepted, and the best function value is adjusted. The point is accepted or refused based on whether or not the function value is greater than the best value found to date. In contrast to other local search algorithms, simulated annealing performs admirably even when applied to non-linear objective functions, whereas those other algorithms are unlikely to do so. It utilizes randomization as a component of its search process. The probability of accepting poorer answers is large at the beginning of the search algorithm and reduces as it progresses, allowing the algorithm to first identify the global optima region, escape the local optima, then proceed to the optima itself.

2.3 Related Work

Several works have been published in the literature to build a lightweight, compressed version of NN models. Hu et al. [7] proposed a NN model compression technique called the one-shot pruning-quantization method (OPQ) that

solves the compression allocation analytically using weight parameters that have already been trained. This method avoids the need for manual tuning or iterative optimization. Unstructured and structured pruning are the two main subcategories of pruning methods. Lee et al. [15] presented an unstructured pruning approach that prunes the unimportant weights of a NN once during initialization before training. Similarly, Park et al. [18] presented a simple unstructured magnitude-based pruning method that extends the single-layer optimization to a multi-layer optimization. A structured pruning-based general framework for compressing the NN based on neuron importance levels is presented in [21]. Lin et al. [16] proposed a structured pruning based global and dynamic pruning (GDP) approach to prune unnecessary filters for CNN acceleration. A shift operation-based spatial information collection strategy as an alternative to spatial convolutions is presented in [20]. In this method, end-to-end trainable shift-based modules are created by combining shifts and point-wise convolutions. Some of the works that have been studied regarding optimizing the micro-architecture and macro-architecture of NN for designing efficient NN model architectures are presented in [17].

3 Proposed Work

This section gives an overview of our proposed work. We present a metaheuristic algorithm-based optimization technique for designing efficient inference hardware models for NNs.

3.1 SA Algorithm-Based NN Optimization

We propose a NN optimization method for developing a fine-tuned and compressed NN model based on the SA algorithm. The optimization methodology that we develop focuses mainly on fine-tuning the micro-architectural elements, such as neuron weights, of the hidden layers of MLP networks. We apply the fine-tuning process in two steps. First, we created an optimized and compressed MLP model using the custom-modified SA algorithm. Then, we apply the different hardware optimization techniques. We apply operator strength reduction, implemented using bit shifting, to handle the multiplication operations for multiples of power of 2 (2^m , where m is the indices). We meticulously prune the hidden layer neurons, which consist of weights with 0 values to slim the overall parameters of MLP. We also reduce the multiplication operations of all the hidden layer neurons consisting of weights with value 1. We further simplify the multiplication of the weights with values as multiples of $(2^m + 1)$ and $(2^m + 2)$ using operator strength reduction and addition operations. Figure 2 illustrates the hardware optimization process.

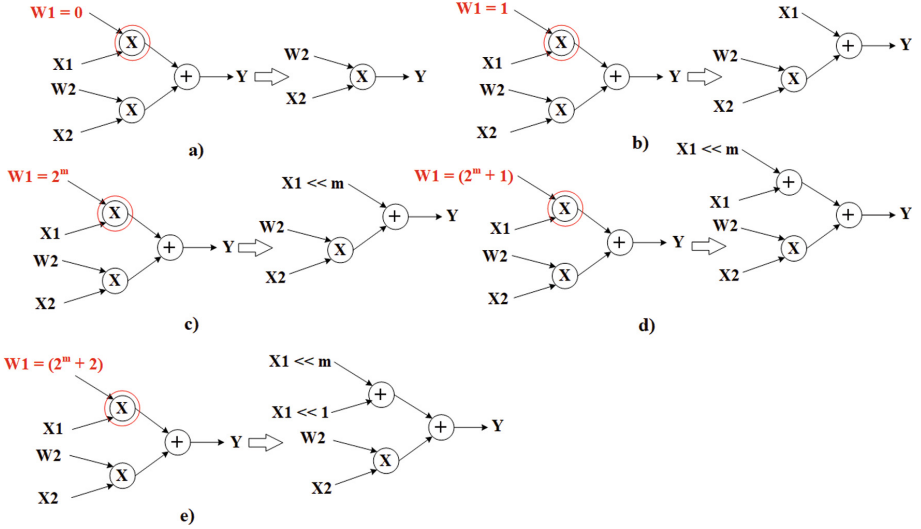


Fig. 2. Illustrations of the hardware optimization process. a) Pruning of the terms with $W = 0$ b) Removal of the multiplier operator with $W = 1$. c) Multiplier strength reduction for weight with value as of form 2^m using bit shifting operations. d) Simplification of the multiplication of the weights with value as a multiple of $(2^m + 1)$ using bit shifting operation. e) Simplification of the multiplication of the weight with value as a multiple of $(2^m + 2)$ using bit shifting operation and an adder operator.

The Algorithm 1 depicts our proposed SA algorithm-based MLP optimization method. The details of the proposed approach are described as follows:

- First, we prepare the training dataset, D , along with the pre-trained single-layered MLP model with weights, W and biases, B . The pre-trained MLP model contains its parameters in IEEE-754 single-precision FP32 format.
- The SA algorithm's various parameters are subsequently initialized. As a starting point, we choose the random solution along with the starting annealing temperature T_{init} , and the temperature reduction function, α . The starting annealing temperature is kept at 100. As we run the SA algorithm, the temperature T decays with α , which is given by $T = \alpha * T$.
- Then, we select the specific percentage, (perturbation amount (p)) of the hidden layer neuron weights, W_p for the perturbation of all the neuron weights of the hidden layer, W_h at random, where $W_p \subseteq W_h$.
- We also specify the number of iterations, N , before running the SA algorithm. Once the SA algorithm is run, we perturb each neuron's weight of the hidden layer at random in each iteration of the training. The W_p is proportional to the T in our proposed methodology.
- For each iteration, the analysis of the newly generated hidden layer neuron weights, W' is performed. If some of the W' are proximate to the integer value, we round them to the nearest neighbor integer.

Algorithm 1. Our proposed methodology optimization process

Input: A pre-trained IEEE-754 FP32 single hidden layered MLP model with W weights, B biases and training dataset, D .

Output: Optimized MLP model

```

1: Select the initial random solution, starting annealing temperature,  $T_{init}$  and
   temperature reduction function,  $\alpha$ .
2: Select a specific %  $p$  of the hidden layer neuron weights,  $W_p$  at random to perturb
   all the weights,  $W_h$  of the hidden layer neurons.  $\triangleright W_p \subseteq W_h$ 
3:  $N \leftarrow$  Number of iterations
4: While  $T > T_{final}$  do
5:   for each iteration  $i$  in  $N$  do
6:     repeat
7:       Perturb each neuron weight of the hidden layer random.  $\triangleright W_p \propto T$ 
8:       Train the MLP model and generate new hidden layer neuron weights,  $W'$ .
9:       if some of the  $W'$  values are  $\approx$  integer then
10:        Round them to the proximate integer value.
11:        Measure the predictive performance.
12:        if performance criteria are met then
13:          Accept  $W'$  and the solution.
14:        else
15:          Calculate acceptance probability,  $P(acceptance)$ .
16:          Generate random number,  $R$ .  $\triangleright R \in [0, 1]$ 
17:          if  $R > P(acceptance)$  then
18:            Reject  $W'$ .
19:          else
20:            Accept  $W'$  and the solution.
21:          end if
22:        end if
23:      end if
24:    until all the datasets from  $D$  are selected.
25:  end for
26:  Reduce the  $T$ .  $\triangleright T = \alpha * T$ 
27: end

```

- Then the predictive performance is calculated in terms of the accuracy of the model. If there is an increase in the predictive performance, we accept the newly generated hidden layer neuron weights, W' and the solution. If not, we compute the acceptance probability, $P(acceptance)$.
- After calculating the acceptance probability, generate a random number, $R \in [0, 1]$. If R is greater than $P(acceptance)$, we discard the W' else we accept the W' and the solution. The equation of the acceptance probability, $P(acceptance)$ is given by:

$$P(acceptance) = \begin{cases} \exp(-\Delta C/T), & \text{if } \Delta C \geq 0 \\ 1 & \text{if } \Delta C < 0 \end{cases} \quad (2)$$

(3)

where, $\Delta C = \text{new cost function} - \text{old cost function}$.

The chances of accepting a worse solution decrease as the number of iterations increases. The smaller the temperature, the lower the acceptance probability. Additionally, the larger the ΔC , the lower the acceptance probability.

- Once the number of iterations reaches the maximum number of iterations, N_{max} we reduce the T by a factor of α . Again, the model is trained for N until the model converges to an optimal solution or the final annealing temperature reaches T_{final} .

For this proposed work, the value of α is kept constant at 0.95. To verify our proof-of-concept, we extensively run the experiments for three different perturbation amounts: $p = 5\%$, 10% , 15% , and 20% , in order to generate the optimal MLP model. With each perturbation amount, we run the experiment using three different iteration values: $N = 100, 1,000, \text{ and } 10,000$.

Once the training is complete, we obtain the optimal neuron weights for the hidden layer of the optimized MLP model. The hidden layer neuron weights are further fine-tuned using pruning and hardware optimization techniques. All the weights with 0 values are pruned away. The weights with value 1 and 2^m are further reduced using operator strength reduction. This reduces the number of multiplication operators. Additionally, we further reduce the multiplication of the integer weight values which are multiples of the form $(2^m + 1)$ and $(2^m + 2)$ by the application of bit shifting and addition operations. This reduces the neural parameters of the hidden layer of the MLP model. This contributes to a substantial decrease in the memory footprint and computational cost of the MLP model. As a result, a lightweight, effective, and efficient condensed MLP model that is suitable for edge-AI devices is generated.

Table 1. MLP model configurations.

Dataset	MLP Configurations	# of Parameters
Iris	4-4-3	35
Heart Disease	13-10-2	162
Breast Cancer Wisconsin	30-10-2	332
Credit Card Fraud Detection	29-15-2	482
Fetal Health	21-21-3	528

4 Experimental Results

In this section, we discuss the experimental flow and findings to evaluate our proposed method. We evaluate our proposed method using five well-known classification datasets, i.e., Iris [5], Heart Disease [3], Breast Cancer Wisconsin (Diagnostic) [19], Credit Card Fraud Detection [2], and Fetal Health Classification [1],

respectively. The experiment flow for this work consists of training the classification datasets using contemporary methods by randomly dividing the training and testing data in an 80:20 ratio to generate the MLP model. In the proposed work, we train all the datasets using a single hidden layer MLP. Once the MLP model is generated, we use the same parameters as the pre-trained MLP model along with the dataset as an input to the custom-modified SA algorithm. After running the SA algorithm for several iterations, the optimized version of the MLP model is obtained.

We evaluate our hardware MLP model inference architecture based on an estimation of the hardware resources utilized by a single unit of IEEE-754 single-precision FP32 multiplier and adder circuit architecture. The Xilinx Vivado v.2019.2 tools is used to synthesize the single unit of an IEEE-754 single-precision FP32 multiplier and adder for the Virtex®-7vx485tffg1157-1 FPGA. The resource consumption of a single unit of a multiplier and adder is 60 LUTs and 51 LUTs, respectively. We also evaluate the accuracy of the SA-optimized MLP model using both training and testing datasets.

The model configuration of single hidden layered MLP for five different classification datasets is shown in Table 1. The Iris dataset consists of 150 data instances. The MLP configuration for the Iris dataset [5] consists of 4 input layer units, 4 hidden layer units, and 3 output layer units. The Heart Disease dataset [3] consists of 1025 instances, and its MLP configuration is 13 input layer units, 10 hidden layer units, and 2 output layer units. The Breast Cancer Wisconsin (Diagnostic) [19] consists of 569 instances and its MLP configuration is 30 input layer units, 10 hidden layer units, and 2 output layer units. The Credit Card Fraud Detection dataset [2] consists of 284,807 instances, and its MLP configuration is 29 input layer units, 15 hidden layer units, and 2 output layer units. Similarly, the Fetal Health Classification dataset [1] consists of 2,126 instances, and its MLP configuration is 21 input layer units, 21 hidden layer units, and 3 output layer units.

Table 2. Optimized MLP model configurations for the best case solution with $p = 10\%$ and $N = 10,000$

Dataset	# of Parameters	# of parameters rounded to integers	(%) of parameters rounded
Iris	28	7	20%
Heart Disease	136	26	16%
Breast Cancer Wisconsin	271	61	18%
Credit Card Fraud Detection	370	112	23%
Fetal Health	417	111	21%

We conduct extensive experiments to evaluate the efficacy of our proposed methodology by comparing our SA-optimized MLP model with the regular MLP model. The evaluation of the optimized MLP model is based on the reduced number of LUTs and FFs required as compared to the regular MLP model. We perform a total of 12 experiments by making variations in the perturbation amount p of the hidden layer neurons' weight parameter along with the number of iterations N to execute the custom-modified SA algorithm for generating the optimized model that is suitable for resource-constrained environments. The temperature reduction function, α is kept at 0.95 for all the experiments. The perturbation amounts p used in this experiment are 5%, 10%, 15%, and 20%, respectively. For each p , we execute the SA algorithm for 100, 1,000, and 10,000 iterations, respectively. Figures 3 and 4 compare savings (%) in terms of LUTs and FFs between the regular and the SA optimized model with variation in p and N .

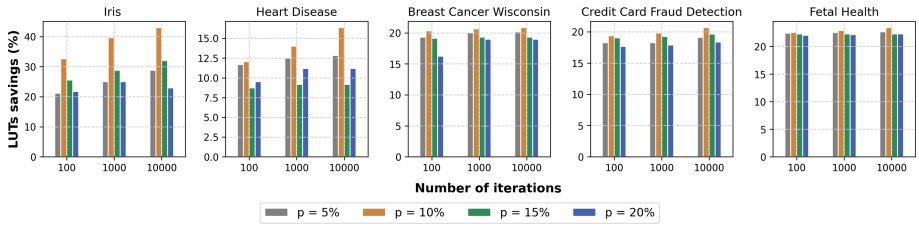


Fig. 3. Comparison of LUTs savings (%) vs. N , varying the p between regular model and SA-optimized model for different datasets.

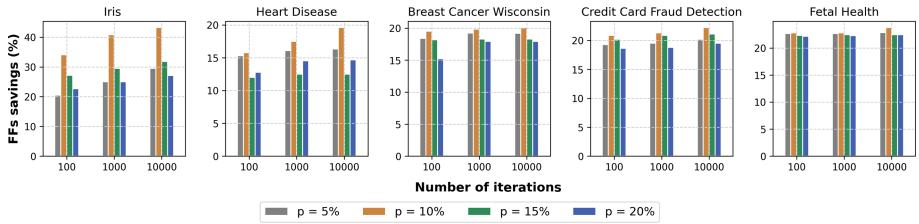


Fig. 4. Comparison of FFs savings (%) vs. N , varying the p between regular model and SA-optimized model for different datasets.

After thorough experimentation on five classification datasets using our proposed method for generating optimal MLP models, we note the following findings: We observe that increasing p does not aid in generating the optimal MLP model after a certain point. We find the best optimal solution for each of the five classification datasets with $p = 10\%$ and $N = 10,000$. The best case savings of LUTs and FFs observed with $p = 10\%$ and $N = 10,000$ are presented in Fig. 5(a) and 5(b), respectively. We observe savings of 42.94% (LUTs) and 43.19% (FFs) for the Iris dataset. For the Heart Disease dataset, the savings are 16.34% (LUTs)

and 19.58% (FFs). For the Breast Cancer Wisconsin (Diagnostic), we observe savings of 20.83% (LUTs) and 20.08% (FFs). For the Credit Card Fraud Detection dataset, we record savings of 20.65% (LUTs) and 22.21% (FFs). Lastly, for the Fetal Health dataset, we achieve savings of 23.49% (LUTs) and 23.77% (FFs). Furthermore, we also observe that the savings in terms of LUTs and FFs increase with the increase in N in our proposed approach. Although it takes more time to run the experiment with a higher N , it generates an optimal solution. Figure 5(c) shows the plot of time vs. N , with $p = 10\%$. Table 2 shows the optimized MLP model configurations for all the datasets when $p = 10\%$ and $N = 10,000$ (best case). We also analyze the accuracy of each of the MLP models generated using our proposed approach. Table 3, 4 and 5 presents the accuracy comparison between the regular and SA optimized models with $\alpha = 0.95$ and $p = 10\%$. We find that the variation in accuracy of the best solution produced using our proposed methodology is on average -1.64% as compared to the regular MLP model’s accuracy.

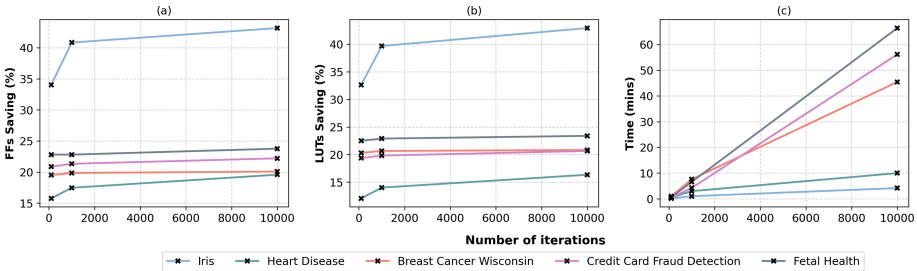


Fig. 5. Best case plots: when $p = 10\%$ (a) FFs saving vs. N . (b) LUTs saving vs. N . (c) Time vs. N .

Table 3. Accuracy comparison between the regular & SA optimized models with $\alpha = 0.95$, $p = 10\%$, and $N = 100$.

Dataset	Accuracy (%) Regular model	N = 100	
		Accuracy (%) SA optimized model	Gain/Loss (%) in accuracy
Iris	96.67	95.48	-1.19
Heart Disease	98.83	96.09	-2.74
Breast Cancer Wisconsin	98.70	95.88	-2.82
Credit Card Fraud Detection	96.82	95.98	-0.84
Fetal Health	96.06	94.78	-1.28
Average			-1.77

Our experimental findings show promising results as compared to the regular MLP model. This shows that our proof-of-concept proposed SA algorithm-based MLP optimization approach might be suitable for generating optimized MLP models. As the generated MLP models contain fewer parameters than the regular MLP model, they are lightweight, compact, and energy efficient. Hence, they might be well suited for resource-constrained environments such as IoT edge devices.

Table 4. Accuracy comparison between the regular & SA optimized models with $\alpha = 0.95$, $p = 10\%$ $N = 1,000$.

		N = 1,000	
Dataset	Accuracy (%) Regular model	Accuracy (%) SA optimized model	Gain/Loss (%) in accuracy
Iris	96.67	94.22	-2.45
Heart Disease	98.83	96.11	-2.72
Breast Cancer Wisconsin	98.70	95.55	-3.15
Credit Card Fraud Detection	96.82	95.39	-1.43
Fetal Health	96.06	95.12	-0.94
Average			-2.14

Table 5. Accuracy comparison between the regular & SA optimized models with $\alpha = 0.95$, $p = 10\%$ $N = 10,000$.

		N = 10,00	
Dataset	Accuracy (%) Regular model	Accuracy (%) SA optimized model	Gain/Loss (%) in accuracy
Iris	96.67	95.81	-0.86
Heart Disease	98.83	95.67	-3.16
Breast Cancer Wisconsin	98.70	95.43	-3.27
Credit Card Fraud Detection	96.82	96.14	-0.68
Fetal Health	96.06	95.83	-0.23
Average			-1.64

5 Conclusions

In this work, we propose an SA algorithm-based MLP optimization approach to build a lightweight, energy-efficient, and compressed hardware MLP model. We finetune the micro-architectural parameters (weights) of the single hidden layer of MLP to generate the optimized model. The hardware optimization techniques are further employed to generate the optimum compressed MLP hardware model. Utilizing five well-known datasets, we conduct comprehensive experiments to confirm our proposed methodology. Experimental observations demonstrate that our proposed method produces superior results in terms of hardware resource (LUTs and FFs) reductions when compared to the regular NN model. For all the datasets, we find the best optimal solution when the $p = 10\%$ and $N = 10,000$. The purpose of this research work is to validate our proof-of-concept. In subsequent research work, we will investigate generating highly optimized compressed NN models with the SA algorithm by varying both the weight and bias parameters of the NN. We will employ a more complex NN, consisting of a large number of hidden layers, to produce optimized compressed NN models on practical hardware platforms.

References

1. Ayres-de Campos, D., Bernardes, J., Garrido, A., Marques-de Sa, J., Pereira-Leite, L.: Sisporto 2.0: a program for automated analysis of cardiocograms. *J. Maternal-Fetal Med.* **9**(5), 311–318 (2000)
2. Dal Pozzolo, A.: Adaptive Machine Learning for Credit Card Fraud detection (2015)
3. Detrano, R.: UCI Machine Learning Repository: Heart Disease Data Set (2019)
4. Fausett, L.V.: Fundamentals of Neural Networks: Architectures, Algorithms and Applications. Pearson Education India, Noida (2006)
5. Fisher, R.A.: UCI Machine Learning Repository: Iris Data Set (2011)
6. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630* (2021)
7. Hu, P., Peng, X., Zhu, H., Aly, M.M.S., Lin, J.: OPQ: compressing deep neural networks with one-shot pruning-quantization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 7780–7788 (2021)
8. Joshi, R., Zaman, M.A., Katkoori, S.: Novel bit-sliced near-memory computing based VLSI architecture for fast sobel edge detection in IoT devices. In: 2020 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS), pp. 291–296. IEEE (2020)
9. Joshi, R., Kalyanam, L.K., Katkoori, S.: Simulated annealing based integerization of hidden weights for area-efficient IoT edge intelligence. In: 2022 IEEE International Symposium on Smart Electronic Systems (iSES), pp. 427–432 (2022). <https://doi.org/10.1109/iSES54909.2022.00093>
10. Joshi, R., Kalyanam, L.K., Katkoori, S.: Area efficient VLSI ASIC implementation of multilayer perceptrons. In: 2023 International VLSI Symposium on Technology, Systems and Applications (VLSI-TSA/VLSI-DAT), pp. 1–4. IEEE (2023)

11. Joshi, R., Zaman, M.A., Katkooi, S.: Fast Sobel edge detection for IoT edge devices. *SN Comput. Sci.* **3**(4), 302 (2022)
12. Kaiming, H., Xiangyu, Z., Shaoqing, R., Jian, S.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
13. Kalyanam, L.K., Joshi, R., Katkooi, S.: Range based hardware optimization of multilayer perceptrons with RELUs. In: *2022 IEEE International Symposium on Smart Electronic Systems (iSES)*, pp. 421–426 (2022). <https://doi.org/10.1109/ISES54909.2022.00092>
14. Kirkpatrick, S., Gelatt, C.D., Jr., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
15. Lee, N., Ajanthan, T., Torr, P.H.: Snip: single-shot network pruning based on connection sensitivity. arXiv preprint [arXiv:1810.02340](https://arxiv.org/abs/1810.02340) (2018)
16. Lin, S., Ji, R., Li, Y., Wu, Y., Huang, F., Zhang, B.: Accelerating convolutional networks via global & dynamic filter pruning. In: *IJCAI*, vol. 2, p. 8. Stockholm (2018)
17. Mamalet, F., Garcia, C.: Simplifying ConvNets for fast learning. In: Villa, A.E.P., Duch, W., Érdi, P., Masulli, F., Palm, G. (eds.) *ICANN 2012. LNCS*, vol. 7553, pp. 58–65. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33266-1_8
18. Park, S., Lee, J., Mo, S., Shin, J.: Lookahead: a far-sighted alternative of magnitude-based pruning. arXiv preprint [arXiv:2002.04809](https://arxiv.org/abs/2002.04809) (2020)
19. Wolberg, W., Street, W., Mangasarian, O.: Breast cancer wisconsin (diagnostic). UCI Machine Learning Repository (1995)
20. Wu, B., et al.: Shift: a zero flop, zero parameter alternative to spatial convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9127–9135 (2018)
21. Yu, R., et al.: NISP: pruning networks using neuron importance score propagation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9194–9203 (2018)



Machine Learning-Based Multi-stratum Channel Coordinator for Resilient Internet of Space Things

Md Tajul Islam^(✉), Sejun Song, and Baek-Young Choi

School of Science and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA
{mi8rd,songsej,choiby}@umsystem.edu

Abstract. Sensing and transferring data are critical and challenging tasks for space missions, especially in the presence of extreme environments. Unlike terrestrial environments, space poses unprecedented reliability challenges to wireless communication channels due to electromagnetic interference and radiation. The determination of a dependable channel for exchanging critical data in a highly temperate environment is crucial for the success of space missions. This paper proposes a unique Machine Learning (ML)-based multi-stratum channel coordinator in building the Resilient Internet of Space Things (ResIST). ResIST channel coordinator accommodates a lightweight software-defined wireless communication topology that allows dynamic selection of the most trustworthy channel(s) from a set of disparate frequency channels by utilizing ML technologies. We build a tool that simulates the space communication channel environments and then evaluate several prediction models to predict the bandwidths across a set of channels that experience the influence of radiation and interference. The experimental results show that ML-prediction technologies can be used efficiently for the determination of reliable channel(s) in extreme environments. Our observations from the heatmap and error analysis on the various ML-based methods show that Feed-Forward Neural Network (FFNN) drastically outperforms other ML methods as well as the simple prediction baseline method.

Keywords: Network Management · Machine Learning · Internet of Things · Reliability · Software-defined Wireless Communication

1 Introduction

Sensing is the basis for the monitoring and operation of various space vehicles, satellites, payloads, CubeSats, surface exploration systems, ground testing systems, space habitats examinations, etc. The necessity lies in a wide range including i) physical sensing for temperature, humidity, and pressure inside or around rocket modules, ii) chemical sensing of crew capsule or space habitats, mold, mildew, or other airborne bacteria, iii) crew health monitoring on vital signs and sleep behavior, etc. So, this sensing and transmitting data reliably and efficiently are integral parts of the National Aeronautics and Space Administration (NASA)'s recent vision for private astronaut missions, the

commercial destination to lower-earth orbit, sustainable demand of in-space manufacturing, production, or development of a commercial application, etc. [17]. But, the space environment poses unique and extreme challenges such as radiations from solar events and cosmic rays, extreme temperatures (both hot and cold according to the location relative to the Sun), and the absence of the insulating atmosphere of the earth which could lead this detection and transmission work more unreliable and challenging. Due to this reliability concerns over the harsh operational environment [6,8], most of the data transfer in space-related projects is mainly transferred through heavy and bulky wire-line communication as shown in Fig. 1. Those wire-line communications lead to many issues such as heavy and spacious spacecraft, inflexible sensor placement, and high project costs.

Recently, many researchers [6,8,18] have been working on wireless communication and sensor networking technologies in space subsystems. In addition, recent progress in communication technology such as effective channel coding and modulation techniques, more significant storage ability, reduced size and expenditure of devices, and ultra-fast processing abilities broaden the scope of using wireless technology in outer space area [7]. When the wireless technology's resilience would be enhanced and ensured, the benefits of wireless sensors and communication networks in space applications will be tremendous including reduced weight and cost of spacecraft, easy deployment of sensors with better area usage, flexible sensor placement along with data gathering from challenging areas and simplification of the Assembly, Integration, and Testing (AIT) process.

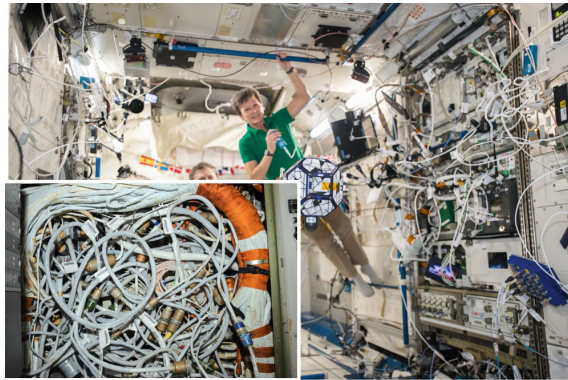


Fig. 1. Wire-lines inside of International Space Station (ISS)-Courtesy of NASA [2]

For facilitating the benefits of transferring sensor data in space environment through wireless medium, this paper proposes a reliable wireless communication mechanism named the Resilient Internet of Space Things (ResIST) for space sensor and data transfer applications. We build a unique Machine Learning (ML)-based multi-stratum channel coordinator, which employs a software-defined wireless communication approach, allowing dynamic selection of the most reliable channel(s) from a set of divergent frequency channels for direct transmission. Expanding the traditional Radio Frequency

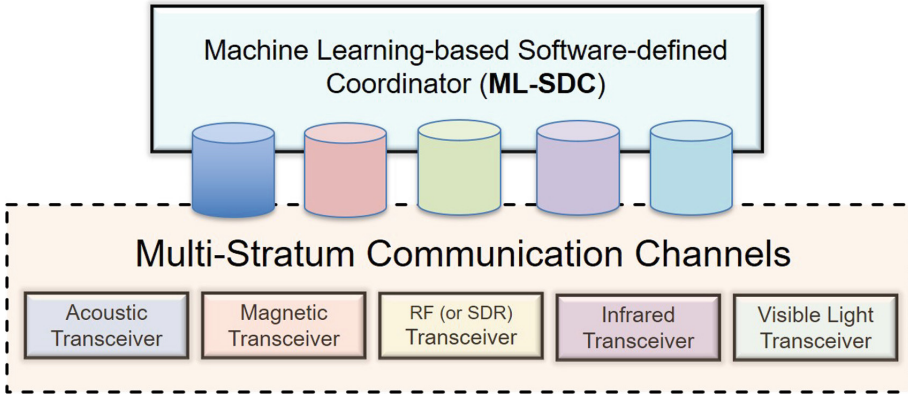


Fig. 2. ML-SDC in ResIST framework

(RF) spectrum levels of wireless communication, we propose to exploit multi-stratum electromagnetic (EM) spectra channels that exhibit different physical characteristics such as RF, infrared (IR), visible lights (VL), etc. Instead of transmitting multiple channels simultaneously (redundancy-based resilience), we explore ML technologies to track many potential channels and accurately decide on the reliable channel(s). As illustrated in Fig. 2, a ML-based Software-defined Coordinator (ML-SDC) monitors the quality of a large number of adjacent electromagnetic frequency channels and dynamically selects the best medium for direct reliable data transmission. We investigate several prediction models, including Exponentially Weighted Moving Average (EWMA) - the baseline, Feed-Forward Neural Network (FFNN), AlexNet, and Convolutional Neural Networks (CNN or ConvNet), to substantiate the coordination-based dynamic selection method's feasibility in space communication. To the best of our knowledge, this study is the first work that dynamically uses diverse EM spectrum bands of different characteristics beyond RF. We first build a simulator that generalizes and generates wireless communication channel conditions by introducing various electromagnetic interference according to temporal and spectral burstiness of radiations in space environments. Then, we implement various ML methods to choose reliable and high bandwidth channels according to the simulated channel conditions. We evaluate the accuracy of ML methods along with a baseline for predictions of reliable channels. Our goal is to explore the feasibility of ML for wireless space communication by finding fast and effective ways to yield high-accuracy predictions of transmission capacity even in extreme, high-interference environments where unpredictable, seemingly random processes are involved. Our experimental results show that ML-prediction technologies in wireless space communication are feasible and we can improve the reliability and scalability of the wireless communication system in the presence of high interference by dynamically switching communication channels in the basis of the ability to render bandwidth availability predictions across all channels. Our observations from the heatmap and error analysis of the implemented methods show that FFNN performs bet-

ter than the baseline method, EWMA, and two other MLDL-based (i.e., AlexNet and ConvNet).

The rest of the paper is organized as follows. Section 2 discusses related works in this subject. We have provided a brief idea of ML and DL methods that have been used in this work in Sect. 3. Our observation and evaluation results are described in Sect. 4. Finally, Sect. 5 concludes the work.

2 Related Work

We discuss some notable work related to ML-based wireless communication in this section as Machine Learning (ML) or Deep Learning (DL) methods have gained significant attention in terrestrial wireless communication for their data-driven pattern recognition capability in physical layer operations, including channel estimation and detection, channel encoding and decoding, modulation recognition, etc. [23].

Sumathi *et al.* represent the channel selection method for cognitive radio networks in [20], where they utilize the machine learning algorithm the support vector machine - SVM for best possible channel selection for the secondary user with maximum throughput using transmitted and received power, the data rate, the maximum vacancy time and the service time. A deep learning-based partially overlapped channel assignment (POCA) method for wireless SDN-IoT system is proposed in [22]. Bitar *et al.* in [5] mention the identification method of sensing wireless ISM band in a heterogeneous environment with a deep convolutional neural network with improved accuracy. Tang *et al.* in [21] propose a deep learning-based prediction algorithm to forecast future traffic load and congestion in a network with a partial channel assignment algorithm and intelligently allocate channels to each link in the SDN-IoT network. Authors in the article [14, 26] proposed a dynamic resource allocation and multiple access control (MAC) protocol method for a heterogeneous network with deep reinforcement learning. As for wireless resource management for space communications. Authors in [25] propose a deep learning technique called long short-term memory (LSTM) network for anticipating the spectrum accessibility of cognitive aerospace communication networks without prior knowledge of the user activities. Ferreira *et al.* in [9] propose a hybrid algorithm for radio resource management for cognitive space communication network where they integrate multi-objective reinforcement learning, and a deep neural network, which significantly reduces the cost of spending time and resources on learning action-performance mapping. Kato *et al.* in [13] proposed an ML-based solution to address different problems on a Space-Air-Ground Integrated Network (GIN) and evaluated a deep learning-based method to improve traffic control performance. Authors in [24] presented the performance of deep learning model for the channel estimation and signal detection in orthogonal frequency-division multiplexing (OFDM) systems, which can perform better for addressing channel distortion and identifying the transmitted symbols rather than conventional methods.

There have been several studies on channel selections for wireless cellular network related. Kato *et al.* in [13] proposed an ML-based solution to address different problems on a Space-Air-Ground Integrated Network (GIN) and evaluated a deep

learning-based method to improve traffic control performance. Studies in [11, 16] consider resource handover issues for multi Radio Access Technology (multi-RAT) networks, traffic flow prediction, and enhance the results of network planning tools by utilizing various deep learning methods and their results demonstrate that deep learning can be promising aspects on those aspects. Authors in [12] proposed a deep learning-based channel estimation technique where they came about a channel learning scheme using deep auto-encoder, which learned the channel state information (CSI) and minimize the mean square error (MSE) of the channel estimation for wireless energy transfer. Authors in [19] formulate channel rank measurement (CRM) metric dataset for normal-equation-based supervised machine learning algorithm (NEC algorithm) which performs channel rank estimation (CRE) of any channel based on average packet receive and instantaneous values of RSSI. Bai *et al.* in [4] propose a channel characteristics prediction mechanism based on machine learning (ML) algorithm and convolutional neural network (CNN) for three-dimensional (3D) millimeter wave (mmWave) massive multiple inputs multiple outputs (MIMO) indoor channels. Authors in [15, 27] proposed a learning-based channel state information (CSI) prediction method for 5G wireless communication where they pointed out several important features like frequency band, location, time, humidity, weather, and temperature, etc. that affect the CSI and after that, they designed a learning mechanism with the integration of CNN (convolutional neural network). Huang *et al.* in [10] proposed a big data and machine learning-enabled wireless channel model framework based on artificial neural networks (ANNs), including feed-forward neural network (FNN) and radial basis function neural network (RBF-NN). They leveraged the input parameters like transmitter (Tx) and receiver (Rx) coordinates, Tx-Rx distance, and carrier frequency, while the output parameters are channel statistical properties, including the received power, root mean square (RMS) delay spread (DS), and RMS angle spreads (ASs).

Our work differs from the above-mentioned work as we have focused on generating channel conditions with electromagnetic interference similar to the space or harsh environment. We have leveraged ML techniques to predict reliable and high bandwidth channels among the multi-dimensional and heterogeneous channel conditions with a software-defined approach. Our model is not dependent on any specific protocol, medium, or channel, rather it can predict or choose the most effective and feasible one through available bandwidth prediction of the channel. By focusing on the physical channel condition rather than protocol or technology, our model ensures the reliability and effectiveness of the channel selection/prediction process in extreme environmental conditions dynamically.

3 Prediction Methods in ML-SDC

This section presents our ML-based Software-defined Coordinator (ML-SDC) architecture and describes the channel prediction models used for this work. ML-SDC is a middleware residing below the wireless protocol layer and above multi-stratum communication channels to coordinate channel registration as a software-defined module. ML-SDC also maintains the registered channel's real-time status and selects the best-performing channel dynamically to the spatiotemporal dimension. We investigate several prediction models including Exponentially Weighted Moving Average- the baseline

algorithm, FFNN, AlexNet, and CNN, for supporting the coordination-based dynamic selection method's feasibility in space communication. ML-SDC's coordination methods coexist in the middleware as running models and can be picked dynamically based on performance. We will first go with the prediction approach of EWMA and then go through the mathematical and parametric definitions of different layers and their control function. Table 1 summarizes the various notations used in the mathematical formulations of this paper.

Table 1. Summary of Notations

Symbols	Descriptions
S_t	Newly observed sample value
α	Sample Weight parameter
w	Weight matrix
x	Input activity
a	Output activity
f	Activation function
E_k	A simple loss function
s	The number of pixels moved upon each application
p	the number of zeroes used to pad the borders
m, n, u, v, g, h	Dimension of Matrices
γ, β	Additional parameters for normalized input

EWMA is a dynamic approach that averages each channel's sequence of measurements through iterative, single-time step updates. It can identify the 'prediction' for each channel using its previous prediction ($EWMA_t$), a newly observed sample value (S_t), and a weight parameter α ($= 0.125$ as this can be user defined) as shown in Eq. (1).

$$EWMA_{t+1} = (1 - \alpha)EWMA_t + \alpha S_t \quad (1)$$

ML-SDC adopts each ML-based model, including FFNN, AlexNet, and CNN (or ConvNet) using different numbers, combinations, and layers in the form of objects imported from Keras framework [1]. Each layer of neural network have a specific mathematical structure, the corresponding gradient to train the parameters, the theoretical advantages and pitfalls. The layer types are dense or fully-connected, two-dimensional convolution, max pooling, and batch normalization. We do not alter bias terms for the benefits of simplicity.

A dense or fully-connected layer refers to a layer consisting of a weight matrix $\mathbf{w} \in \mathbb{R}^{m \times n}$, which is multiplied by an input vector $\mathbf{x} \in \mathbb{R}^m$ before a nonlinear activation function f is applied to yield an output $\mathbf{a} \in \mathbb{R}^n$. Multidimensional input is flattened by a concatenating operation, which is a straightforward operation. The activation function used for each fully-connected layer in all neural networks tested was the widely-used Rectified Linear Unit (ReLU), defined as $f_{\text{ReLU}}(z) = \max(0, z)$. If we let $\mathbf{w}^{(i)}$ denote the

weight matrix of the i^{th} dense layer, then we may write the activity $\mathbf{a}_k^{(i)}$ of the k^{th} neuron of this layer as presented in Eq. (2).

$$\mathbf{a}_k^{(i)} = f_{\text{ReLU}}(\mathbf{z}_k^{(i)}), \quad \mathbf{z}_k^{(i)} = \sum_j^m \mathbf{w}_{jk}^{(i)} \mathbf{x}_j^{(i)} \quad (2)$$

For demonstration purposes, if we apply the chain rule for partial derivatives and assume a simple loss function of $\mathbf{E}_k = 0.5(\mathbf{a}_k^{(l)} - \mathbf{t}_k)^2$, where $\mathbf{a}_k^{(l)}$ is the activity of the k^{th} neuron in the last (l^{th}) layer and \mathbf{t}_k is the corresponding target value. We can obtain the gradients for this layer as shown in Eq. (3).

$$\frac{\partial \mathbf{E}_k}{\partial \mathbf{w}_{jk}^{(l)}} = \frac{\partial \mathbf{E}_k}{\partial \mathbf{a}_k^{(l)}} \frac{\partial \mathbf{a}_k^{(l)}}{\partial \mathbf{z}_k^{(l)}} \frac{\partial \mathbf{z}_k^{(l)}}{\partial \mathbf{w}_{jk}^{(l)}} = \begin{cases} (\mathbf{a}_k^{(l)} - \mathbf{t}_k) \mathbf{x}_j^{(l)} & \text{if } \mathbf{z}_k^{(l)} > 0, \\ 0 & \text{if } \mathbf{z}_k^{(l)} \leq 0 \end{cases} \quad (3)$$

A two-dimensional convolution (Conv2D) layer refers to a layer consisting of an array of two-dimensional kernels, or filters, each denoted $\mathbf{w}^{(i)} \in \mathbb{R}^{u \times v}$ where u and v are the dimensions chosen for the filters and i is the index of the filter. Filters typically operate on input images consisting of multiple frequency channels, giving them an additional dimension with a size equal to the number of channels. However, since our two-dimensional input images only contain one color channel, this additional dimension has been omitted for simplicity of presentation. These filters are first convolved or multiplied at each index of the two-dimensional input image $\mathbf{x} \in \mathbb{R}^{g \times h}$, where g and h are the width and height of fully connected resolution in pixels before a nonlinear activation function is applied. Again, the activation function used for each convolution layer in all networks tested is f_{ReLU} . The activity of the entry in the j -th row and k -th column of the feature map resulting from the application of the i^{th} filter at index (l, m) of the input image is presented in Eq. (4).

$$\mathbf{a}_{jk}^{(i)} = f_{\text{ReLU}}(\mathbf{z}_{jk}^{(i)}), \quad \mathbf{z}_{jk}^{(i)} = \sum_q^u \sum_r^v \mathbf{w}_{qr}^{(i)} \mathbf{x}_{l+q-1, m+r-1} \quad (4)$$

A max-pooling layer refers to a layer with a pool size (u, v) specifying the dimensions of a window over which the max function is applied to extract the most relevant features. The activity of the entry in the j^{th} row and k^{th} column of the output feature map (\mathbf{a}_{jk}) resulting from the application of the max function over a pooling window at index (l, m) of the two-dimensional input feature map $\mathbf{x} \in \mathbb{R}^{g \times h}$ is shown in Eq. (5).

$$\mathbf{a}_{j,k} = \max \{ \mathbf{x}_{l,m}, \mathbf{x}_{l,m+1}, \mathbf{x}_{l+1,m}, \dots, \mathbf{x}_{l+u-1, m+v-1} \} \quad (5)$$

Both convolution and max-pooling layers have variable stride (s , the number of pixels moved upon each application of a filter) and zero-padding (p , the number of zeroes used to pad the borders of the image before filters are applied). They determine the size and shape of the output layer depending on the input's size and shape. Given an input feature map $\mathbf{x} \in \mathbb{R}^{g \times h}$, a kernel / pool size (u, v) , we define the size (q, r) of the resulting output feature map $\mathbf{y} \in \mathbb{R}^{q \times r}$ as shown in Eq. (6).

$$q = \frac{g - u + p_{\text{left}} + p_{\text{right}}}{\text{Horizontal}} + 1, \quad r = \frac{h - v + p_{\text{top}} + p_{\text{bottom}}}{\text{Vertical}} + 1 \quad (6)$$

We use a batch normalization layer, batch norm, to fix each scalar input's means and variances to the next layer independently via a normalization step across a given mini-batch of training inputs. We accelerate and regularize the training of deep neural networks by reducing internal covariate shift. During training, the batch norm step is immediately followed by scaling and shifting operation with two additional parameters γ and β for each normalized input to approximate the mean and variance, respectively.

FFNN can be distinguished from other neural networks like AlexNet and ConvNet, as there is no cycle between the nodes. Values are straightly propagated forward through a set of layers. FFNN's input is 20-time steps over 200 channels, and it results in a single prediction for each channel's next bandwidth measurement. FFNN has three layers, including output, with 4000, 500, and 200 neurons in order. There is a batch norm layer between the second and third layers. Each layer uses a ReLU as its activation and zeros as its bias initializer. ReLU, defined as $f_{\text{ReLU}}(z) = \max(0, z)$, is useful as it only gives non-zero outputs if the input is greater than zero. Each layer's weights use a He Normal (He-et-al) initializer [3], which draws samples from a normal distribution centered on zero. FFNN has trained three epochs at a time.

4 Evaluations

We evaluate the effectiveness of ML-prediction technologies which can improve wireless communications system's reliability in the presence of high interference. For that, system will dynamically switch communication channels by using the ability to render bandwidth availability predictions across all channels. We have conducted the heat map and error analysis of the implemented methods for these objectives.

First, we build a wireless communication channel simulator for space environments that have large scale spectrum channels and with flexible channel interference models. Using the simulator, we create a data-generation model with a specified set of assumptions, which provide a new bandwidth measurement for each channel on-demand. These assumptions are defined in Table 2. We define another set of assumptions for modeling the inference to generate a pulse of interference to propagate through the measurement model's time-steps and channels. We consider the maximum strength of the interference as uniform random between 0 to 1 and the number of time-steps for which the interference pulse lasts as uniform random between 1 and 500. We have considered the drop-off the rate at which the interference decreases as it propagates across neighboring channels is 0.8. These assumptions are defined in Table 3.

Second, EWMA, FFNN, AlexNet, and ConvNet are employed to analyze and compare several predictive models' performance properly. We have implemented those predictive models with the input signal shown in Fig. 3a, which shows how the radiation impacts bands of spectrum. It presents the heatmap representation of the actual generated input signals for 200 sample channel measurements (in Y) over simulation time (in X). The channel reliability is presented in the color range from green to black (i.e., Greener means better signal with less error, and darker means low-reliability channels). Figure 3b shows the heatmap representation of the output predicted signal from EWMA model with $\alpha = 0.125$ for the given input signal. We can observe that the predicted output channels have less similarity to the actual input signal level. However, one of the

Table 2. Assumption Parameters for Data Modeling

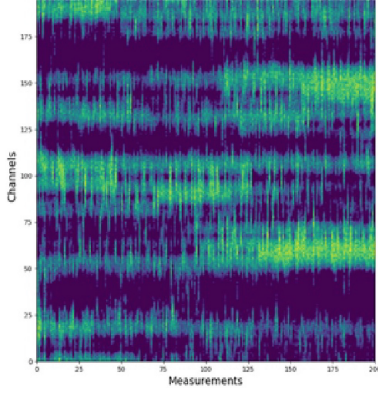
Assumption Parameters	Values	Notes
The number of channels	200	–
The maximum capacity of each channel	100	–
Background interference	0.2	the uniform-random background noise of each channel
Observation window size	200	the number of previous time-steps to keep in memory
The maximum duration of interference	500 measurements	–
Measurement frequency	1 for sequential time steps and at least $window_size + maxduration + 1$ to ensure independent and identically distributed observation windows	the number of new measurements to take upon each call
Interference probability	0.25	a fixed probability that a new pulse of interference begins upon a given measurement
Dropoff	$adjacent = 0.8 * original$	the factor by which interference pulses propagate geometrically from given channel to its adjacent channel(s)

Table 3. Assumption Parameters for Interference Modeling

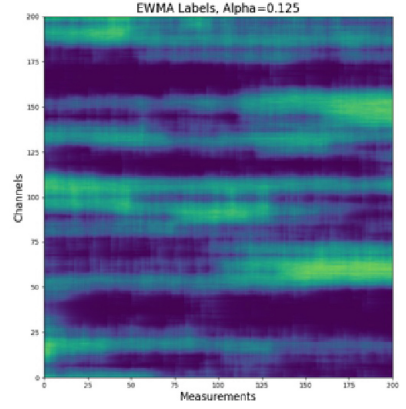
Assumption Parameters	Values	Notes
Amplitude	uniform random between 0 and 1	the maximum strength of the interference
Duration	uniform random between 1 and 500	the number of timesteps for which the interference pulse lasts
Dropoff	0.8	the rate at which the interference decreases as it propagates across neighboring channels

ML-based models, the FFNN in Fig. 3c, exhibits that the output signal pattern highly matches the fundamental input signal level. AlexNet in Fig. 3d and ConvNet in Fig. 3e show better matching patterns than the EWMA model but less accurate results than FFNN with revealing different detection capabilities. For example, ConvNet can accurately predict the signal level, whereas AlexNet can detect the edge of some interference bursts but shows significant errors on the actual signal level.

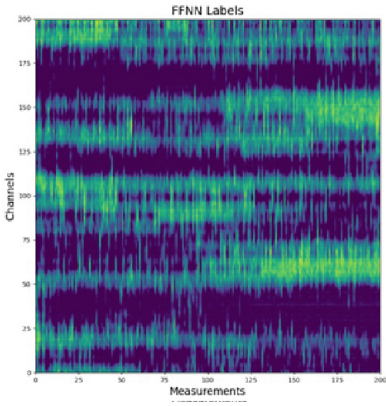
Third, we conduct a Mean Absolute Error (MAE) calculation using the same generated test set. We calculate MAE by summing the absolute value of the difference between each channel’s predicted bandwidth and the same channel’s actual bandwidth at each point in time. The sum is then divided by the total number of predictions, which can be calculated by the number of channels multiplied by the number of time-steps in the test set. The models with a lower MAE are considered to have better performance.



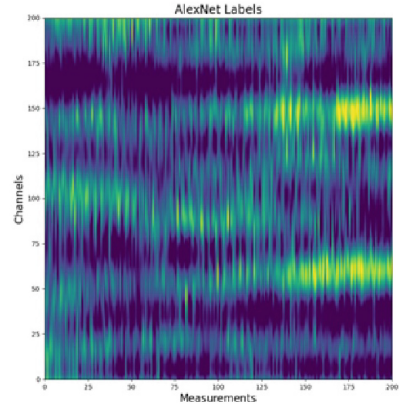
(a) Actual input



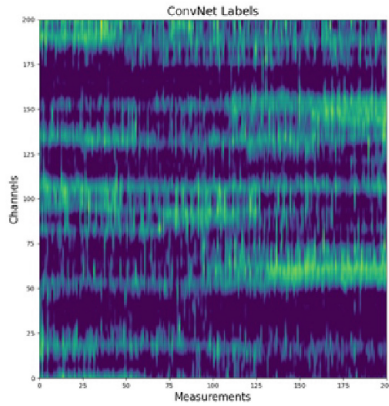
(b) EWMA output



(c) FFNN output



(d) AlexNet output



(e) ConvNet output

Fig. 3. Heatmap Representation of Signals: 200 consecutive samples

As summarized in Table 4, the simulation results show that FFNN has the least MAE (3.12) followed by ConvNet’s MAE (5.08), which is much better than other approaches. The MAE results align with the previous heatmap representations.

Table 4. Model performance for 2K consecutive time steps across 200 channels(20K training examples, where applicable)

Estimation Model	MAE	Testing Time	Training Time (s)
EWMA (Alpha = 0.125)	10.82	0.69 s	N/A
FFNN	3.12	0.27 s	~3 per epoch
ConvNet	5.08	0.71 s	~20 per epoch
AlexNet	16.24	0.91 s	~44 per epoch

Forth, we also present each model’s prediction performance in MAE using Probability Distribution Function (PDF) and Cumulative Distribution Function (CDF). The PDF is generated by sampling each model’s predictions’ absolute error floor (integer-valued) across the entire test set. Then, it counts the frequency of each integer value in each model’s sampled errors. These results are L1-normalized (by dividing the frequency of each integer-valued error by the total number of predictions) to produce proper probability distributions representing each error’s relative frequency. When a PDF of a model concentrates on the lower error probability, it performs better than a model with a PDF where most observations are either distributed evenly or focused away from zero error. We also generate the CDF by iterating through the PDF (sorted from zero to max absolute error) and replacing each relative frequency with the sum of relative frequencies. The CDF visualizes and calculates the probability that the absolute error falls within a specific range of values. A model with a CDF with a high positive slope towards zero error and a vanishing slope further away from zero error is the better-performing model. Additionally, if the slopes are compared between the two models, the higher CDF values model is better. We have trained the prediction models with an extensive training dataset (20K independent and identically distributed training examples) and ran many sets of tests and validations. As shown in Fig. 4a, the MAE results show that FFNN, followed by the ConvNet, performs 3 to 5 times better than other models and the error is half less than the EWMA (with alpha=0.125). Our test results show that AlexNet performs the worst. Additionally, As presented in Fig. 4b, the CDF of both the ConvNet and FFNN had the steepest slopes near zero, but FFNN has the higher values overall. One significant difference between ConvNet and FFNN is that the latter had a much smaller maximum absolute error than the former. It is consistently generated predictions with a calculated MAE, which is larger than the EWMA by about 45%, and over four times higher MAE than FFNN. Overall, our experimental results show that AlexNet is the worst performer according to the PDF and CDF analysis.

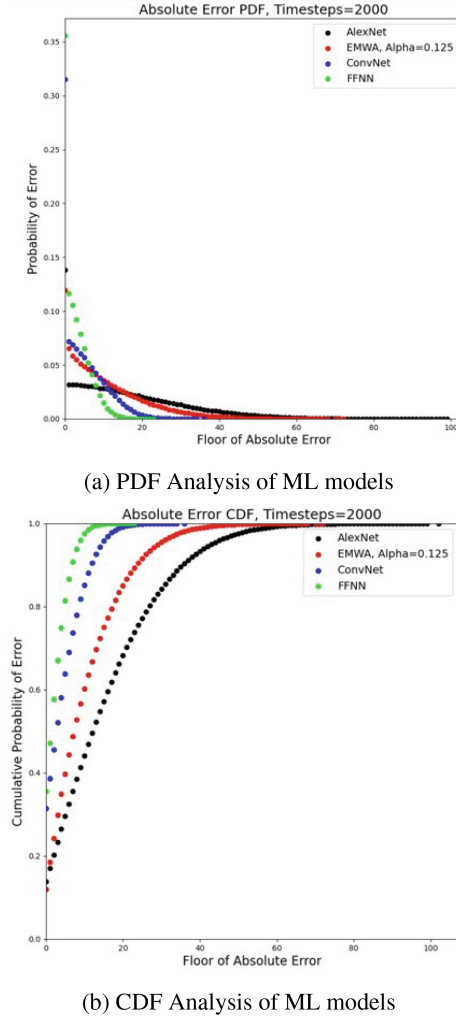


Fig. 4. Error Analysis for 2K time steps across 200 channels

5 Conclusion

We have introduced the Resilient Internet of Space Things (ResIST) by building a Machine Learning (ML)-based software-defined coordinator (ML-SDC), which accommodates a dynamic selection of the most reliable channel(s) from a set of divergent frequency channels. We have conducted a feasibility study of using ML technologies for wireless space communication in extreme or unreliable environments. We have explored several channel prediction models (such as EWMA, FFNN, AlexNet, and ConvNet) and showed a comparative study for the effectiveness of those models. The experimental results show that ML-prediction technologies can improve wireless communi-

cations' reliability and effectiveness in imprinting environments. As our ML design is employed at the physical EM layer, the proposed ML-SDC is not dependent on specific protocols or communication technology and can work in an efficient, effective, and most of all reliable manner. Our observations from the heatmap and error analysis show that FFNN drastically outperforms the baseline method (EWMA), AlexNet, and ConvNet. This work will shed light on wireless communication capability in space to reduce bulky and heavy wirelines, thus significantly decreasing the spacecraft's weight while facilitating reliable communication among many onboard sensors and devices.

Acknowledgements. We express our gratitude to NASA-Missouri Space Grant Consortium (MOSGC) for the grant to facilitate this research work. This work is also supported by the Korea Institute for Advancement of Technology (KIAT) grant that is funded by the Ministry of Trade, Industry and Energy (MTIE) (No. P0019809: Building Enablers for Multi-Industry Sectors Collaborative Federated Testbeds as a Foundation (Distributed Open Platform) for Cross-Industry End-to-End Services Innovation and Delivery Agility in the 5G & Beyond). We want to thank Mark Ekis and Tyler Wheaton who contributed to the early stage of the work.

References

1. Keras Framework. <https://keras.io/>
2. Recap From the Expedition Lead Scientist. https://www.nasa.gov/mission_pages/station/research
3. Xavier and He Normal (He-et-al) Initialization. <https://bit.ly/3qkeyvv>
4. Bai, L., et al.: Predicting wireless mmWave massive MIMO channel characteristics using machine learning algorithms. *Wirel. Commun. Mob. Comput.* **2018** (2018)
5. Bitar, N., Muhammad, S., Refai, H.H.: Wireless technology identification using deep convolutional neural networks. In: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 1–6. IEEE (2017)
6. Choi, B.Y., Boyd, D., Wilkerson, D.: Toward reliable and energy efficient wireless sensing for space and extreme environments. In: 2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), pp. 156–157. IEEE (2017)
7. De Cola, T., Marchese, M.: Reliable data delivery over deep space networks: benefits of long erasure codes over ARQ strategies. *IEEE Wirel. Commun.* **17**(2), 57–65 (2010)
8. Drobczyk, M., Lübken, A.: Novel wireless protocol architecture for intra-spacecraft wireless sensor networks (inspawsn). In: 2018 6th IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), pp. 89–94. IEEE (2018)
9. Ferreira, P.V.R., Paffenroth, R., Wyglinski, A.M., Hackett, T.M., Bilén, S.G., Reinhart, R.C., Mortensen, D.J.: Multi-objective reinforcement learning-based deep neural networks for cognitive space communications. In: 2017 Cognitive Communications for Aerospace Applications Workshop (CCAA), pp. 1–8. IEEE (2017)
10. Huang, J., et al.: A big data enabled channel model for 5G wireless communication systems. *IEEE Trans. Big Data* (2018)
11. Huang, W., Song, G., Hong, H., Xie, K.: Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Trans. Intell. Transp. Syst.* **15**(5), 2191–2201 (2014)
12. Kang, J.M., Chun, C.J., Kim, I.M.: Deep-learning-based channel estimation for wireless energy transfer. *IEEE Commun. Lett.* **22**(11), 2310–2313 (2018)
13. Kato, N., et al.: Optimizing space-air-ground integrated networks by artificial intelligence. *IEEE Wirel. Commun.* **26**(4), 140–147 (2019)

14. Li, J., Zhang, X., Zhang, J., Wu, J., Sun, Q., Xie, Y.: Deep reinforcement learning-based mobility-aware robust proactive resource allocation in heterogeneous networks. *IEEE Trans. Cogn. Commun. Network.* **6**(1), 408–421 (2019)
15. Luo, C., Ji, J., Wang, Q., Chen, X., Li, P.: Channel state information prediction for 5G wireless communications: a deep learning approach. *IEEE Trans. Netw. Sci. Eng.* (2018)
16. Mroue, M., Prevotet, J.C., Nouvel, F., Mohanna, Y., et al.: A neural network based handover for multi-rat heterogeneous networks with learning agent. In: 2018 13th International Symposium on Reconfigurable Communication-Centric Systems-on-Chip (ReCoSoC), pp. 1–6. IEEE (2018)
17. Northon, K.: Nasa opens international space station to new commercial opportunities, private astronauts, June 2020. <https://go.nasa.gov/2CUqekw>
18. Pelissou, P.: Building blocks for an intra-spacecraft wireless communication. In: 2015 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), pp. 1–6. IEEE (2015)
19. Rehan, W., Fischer, S., Rehan, M.: Machine-learning based channel quality and stability estimation for stream-based multichannel wireless sensor networks. *Sensors* **16**(9), 1476 (2016)
20. Sumathi, D., Manivannan, S.: Machine learning-based algorithm for channel selection utilizing preemptive resume priority in cognitive radio networks validated by ns-2. *Circuits Syst. Sig. Process.* **39**(2), 1038–1058 (2020)
21. Tang, F., Fadlullah, Z.M., Mao, B., Kato, N.: An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: a deep learning approach. *IEEE Internet Things J.* **5**(6), 5141–5154 (2018)
22. Tang, F., Mao, B., Fadlullah, Z.M., Kato, N.: On a novel deep-learning-based intelligent partially overlapping channel assignment in SDN-IoT. *IEEE Commun. Mag.* **56**(9), 80–86 (2018)
23. Wang, T., Wen, C.K., Wang, H., Gao, F., Jiang, T., Jin, S.: Deep learning for wireless physical layer: opportunities and challenges. *China Commun.* **14**(11), 92–111 (2017)
24. Ye, H., Li, G.Y., Juang, B.H.: Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wirel. Commun. Lett.* **7**(1), 114–117 (2017)
25. Yu, L., Wang, Q., Guo, Y., Li, P.: Spectrum availability prediction in cognitive aerospace communications: a deep learning perspective. In: 2017 Cognitive Communications for Aerospace Applications Workshop (CCAA), pp. 1–4. IEEE (2017)
26. Yu, Y., Wang, T., Liew, S.C.: Deep-reinforcement learning multiple access for heterogeneous wireless networks. *IEEE J. Sel. Areas Commun.* **37**(6), 1277–1290 (2019)
27. Yuan, J., Ngo, H.Q., Matthaiou, M.: Machine learning-based channel prediction in massive MIMO with channel aging. *IEEE Trans. Wirel. Commun.* **19**(5), 2960–2973 (2020)



A Schedule of Duties in the Cloud Space Using a Modified Salp Swarm Algorithm

Hossein Jamali, Ponkoj Chandra Shill, David Feil-Seifer, Frederick C. Harris Jr.,
and Sergiu M. Dascalu^(✉)

Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV, USA
{hossein.jamali,ponkoj}@nevada.unr.edu, {dave,fred.harris,
dascalus}@cse.unr.edu

Abstract. Cloud computing is a concept introduced in the information technology era, with the main components being the grid, distributed, and valuable computing. The cloud is being developed continuously and, naturally, comes up with many challenges, one of which is scheduling. A schedule or timeline is a mechanism used to optimize the time for performing a duty or set of duties. A scheduling process is accountable for choosing the best resources for performing a duty. The main goal of a scheduling algorithm is to improve the efficiency and quality of the service while at the same time ensuring the acceptability and effectiveness of the targets. The task scheduling problem is one of the most important NP-hard issues in the cloud domain and, so far, many techniques have been proposed as solutions, including using genetic algorithms (GAs), particle swarm optimization, (PSO), and ant colony optimization (ACO). To address this problem, in this paper one of the collective intelligence algorithms, called the Salp Swarm Algorithm (SSA), has been expanded, improved, and applied. The performance of the proposed algorithm has been compared with that of GAs, PSO, continuous ACO, and the basic SSA. The results show that our algorithm has generally higher performance than the other algorithms. For example, compared to the basic SSA, the proposed method has an average reduction of approximately 21% in makespan.

Keywords: Cloud Computing · Task Scheduling · Salp Swarm Algorithm

1 Introduction

Today, modern computing methods have attracted the attention of researchers in many fields such as cloud computing, artificial intelligence, and machine learning by using techniques including artificial neural networks in building air quality prediction models that can estimate the impact of climate change on future summer trends [1]. A computational science algorithm is used in this article to determine the schedule of duties in the cloud.

Cloud computing has brought about the availability of tools that provide extensive computing resources on the internet platform. Users can submit their requests for various resources, such as CPU, memory, disk, and applications, to the cloud provider. The

provider then offers the most suitable resources, which meet the user's requirements and offer benefits to the resource owners, based on the price that they can afford to pay [2]. In cloud computing, the main entities are users, resource providers, and a scheduling system whose main body has been proposed for the users' tasks and timeline strategy [3].

Cloud computing consumers rent infrastructure from third-party providers instead of owning it. They opt for this to avoid extra costs. Providers typically use a "pay-as-you-go" model, allowing customers to meet short-term needs without long-term contracts, thus reducing costs [4].

Behind the numerous benefits of cloud computing, there are many challenges too. The most important is the task scheduling problem or resource allocation to the users' requests. The targets of task scheduling in cloud computing are to provide operating power, the optimal timeline for users, and service quality simultaneously. The specific targets related to scheduling are load balance, service quality, economic principles, the best execution time, and the operating power of the system [5]. Cloud computing has three timelines: resources, workflow, and tasks. Resource scheduling involves mapping virtual resources to physical machines. Workflow scheduling ensures the orderly flow of work. Task scheduling assigns tasks to virtual resources. Task scheduling methods can be concentrated or distributed, homogeneous or heterogeneous, and performed on dependent or independent tasks.

Task scheduling in cloud computing has two types based on the characteristic of the tasks:

- *Static*: In static scheduling, the tasks reach the processor simultaneously and are scheduled on accessible resources. The scheduling decisions are made before reaching the tasks and the processing time after doing the entire run of duty is updated. This type of scheduling is mostly employed for tasks that are sent continuously [6]; and
- *Dynamic*: In dynamic scheduling, the number of tasks, the location of the virtual machines, and the method for resource allocation are not constant, and the input time of tasks before sending them is unknown [6].

Scheduling the mechanism of dynamic algorithms compared to static algorithms is better but the overhead of the dynamic algorithm is quite significant [7]. Dynamic scheduling can be done in two ways; in batch and online modes. In batch mode, the tasks are lying in a line, gathered in a set, and after a certain time, scheduled. In the online mode, when the tasks reach the system, they are scheduled [6].

The task scheduling problem in cloud computing focuses on efficiently distributing tasks among machines to minimize completion time [8]. Proper task arrangement has numerous benefits, including reduced energy consumption, increased productivity, improved distribution across machines, shorter task waiting times, decreased delay penalties, and overall faster task completion [9].

The task scheduler plays a crucial role in efficiently scheduling computing actions and logically allocating computing resources in IaaS cloud computing. Its objective is to assign tasks to the most suitable resources to achieve specific goals. Selecting an appropriate scheduling algorithm is essential to enhance resource productivity while maintaining a high quality of service (QoS). Task scheduling involves optimizing the allocation

of subtasks to virtual servers in order to accomplish the task schedule's objective. This area of research continues to receive significant attention [10].

Efficient task planning in cloud computing is essential to minimize fetch time, waiting time, computing time, and resource usage. Task scheduling is crucial for maximizing cloud productivity, meeting user needs, and enhancing overall performance. Its primary goal is to manage and prioritize tasks, reducing time and preventing work failures while meeting deadlines. Task scheduling optimizes the cloud computing system for improved calculation benefits, high performance, and optimal machine output. The scheduling algorithm distributes work among processors to maximize efficiency and minimize workflow time [11].

The rest of this paper is organized as follows: Sect. 2 covers related work; Sect. 3 provides details of the SDSA optimization algorithm; Sect. 4 describes our proposed method, including the expansion and improvement of the salp algorithm; Sect. 5 focuses on the algorithm's target, the fitness function; Sect. 6 presents the results of our simulation; and, finally, Sect. 7 contains the conclusions of our work.

2 Related Works

Ghazipour et al. [12] have proposed a task scheduling algorithm so the tasks existing in the grid are allocated to accessible resources. This algorithm is based on the ACO algorithm, which is mixed with the scheduling algorithm right to choose so that its results are used in the ACO algorithm. The main goal of their article is to minimize the total finish time (makespan) for setting up tasks that have been given [12].

In their research on task scheduling in cloud computing, Sharma and Tyagi [13] examined nine heuristic algorithms. They conducted comparative analyses based on scheduling parameters, simulation tools, observation domain, and limitations. The results indicated the existence of a heuristic approach that satisfies all the required parameters. However, considering specific parameters such as waiting time, resource utilization, or makespan for each task or workflow individually can lead to improved performance. [13].

In 2019, Mapetu et al. [14] researched the "binary PSO algorithm for scheduling the tasks and load power in cloud computing". They introduced a binary version of the PSO algorithm named BPSO with lower complexity and cost for scheduling the tasks and load power in cloud computing, to minimize waiting time, and imbalance degree while minimizing resource use. The results showed that the proposed algorithm presents greater task scheduling and load power than existing heuristic algorithms [14].

Saeedi et al. [15] studied the development of the multi-target model of PSO for scheduling the workflow in the cloud areas. They proposed an approach for solving the scheduling problem considering four contrasting goals (i.e., minimizing the cost, waiting time, energy consumption, and maximizing reliability). The results showed that the proposed approach had a better performance compared to LEAF and EMS-C algorithms [15].

Zubair et al. [10] presented an optimal task scheduling method using the modified symbiotic organisms search algorithm (G_SOS) and aimed to minimize the makespan of the tasks, costs, response time, and imbalance degree, and improve the convergence

speed. The performance of the proposed method using CloudSim (a simulator tool) was evaluated and according to the simulation results, the proposed technique has better performance than the SOS and PSO-Simulated Annealing (PSO-SA) in terms of the convergence speed, cost, imbalance degree, response time, and makespan. The findings confirm the suggested G_SOS approach [10].

Rajagopalan et al. [16] introduced an optimal task-scheduling method that combines the firefly optimization algorithm with a genetics-based evolutionary algorithm. This hybrid algorithm creates a powerful collective intelligence search algorithm. The proposed method excels in minimizing the makespan for all tasks and quickly converges to near-optimal solutions. The results demonstrated that this hybrid algorithm outperformed traditional algorithms like First in, First Out (FIFO) and genetics. However, a potential drawback of this method is the increased overload resulting from the sequential use of two algorithms [16].

3 SSA Optimization Algorithm

This section briefly describes the SSA optimization algorithm proposed by Mirjalini Al which is an extension of the standard SSA algorithm [17]. salps are a type of Salpidae family and have a transparent and barrel-shaped body. Their bodies are very similar to jellyfish. They still move the same as jellyfish, and water is pumped from the middle of the body as a motive force to move forward [17]. The shape of salp is shown in Fig. 1(a).

The biological study of these animals is just starting because it is so difficult to capture them and maintain them in laboratories. One of the most intriguing habits of salps is their tendency to swarm. The salps commonly form a chain in the deep oceans. This chain is shown in Fig. 1(b). Although the primary cause of this behavior is unknown, some researchers think that it is carried out through quick coordinated movements and searches to achieve improved movement [17].

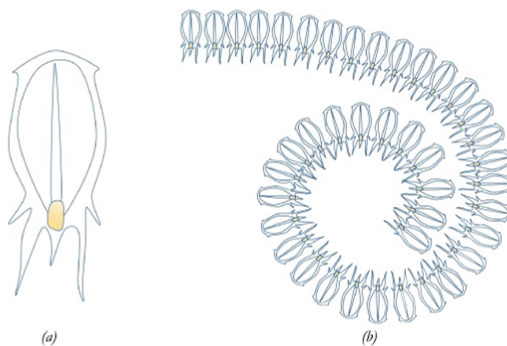


Fig. 1. (A) Illustration of a salp. (B) Salp chain structure. [17].

To model mathematically the salp chains, first, the population is divided into two groups: leaders and followers. The leader is in front of the chain, while the remaining

are considered the followers. As seen from their names, the leader guides the group and the followers follow each other [17].

Like other techniques based on the swarm, the location of salps in a search space is n -dimensional, where n is the number of variables in a problem and known; therefore, the location of all salps is stored in the two-dimensional matrix x . Also, it is assumed that a food source, F , exists in the search space as a swarm target [17].

Equation 1 has been proposed for updating the location of the leader as follows:

$$x_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 \geq 0 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j) & c_3 < 0 \end{cases} \quad (1)$$

where x_j^1 shows the location of the first salp (leader) in the j dimension, F_j is the location of the food source in the j dimension, ub_j identifies the upper boundary of the j dimension, lb_j identifies the lower boundary of the j dimension, and c_1 , c_2 and c_3 are random numbers (between 0,1) [17].

Equation 1 shows that the leader just updates its location according to the food source. The c_1 the constant is the most important parameter in the SSA because it creates a balance between exploration and detection and is defined as Eq. 2:

$$c_1 = 2e^{-\left(\frac{4t}{L}\right)^2} \quad (2)$$

Here, t is the current iteration and L is the maximum iteration.

The parameters of c_2 and c_3 are the random numbers which are uniformly produced in the range [0,1]. They determine if the latter location in the j dimension should be infinite positive or infinite negative, as well as determine the step size.

To update the followers' location, Eq. 3 is used (Newton's law of motion):

$$x_j^i = \frac{1}{2}at^2 + v_0t \quad (3)$$

If $i \geq 2$, x_j^i shows a salp follows the i location in the j dimension, t is the time, and v_0 is the initial velocity; $a = \frac{v_{final}}{v_0}$ and $v = \frac{x-x_0}{t}$. Since the time is iterated in the optimization, the difference between the iterations is equal to 1 and, considering $v_0 = 0$, this relation is expressed as Eq. 4.

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \quad (4)$$

Here, $i \geq 2$ and x_j^i shows a salp follows the i location in the j dimension.

The salp chains can be simulated by Eqs. 1 and 4. In the SSA model, the followers follow the salp leader. The leader also moves towards the food source; therefore, if the food source is substituted for the global optimization, the salp chain automatically moves towards it. However, there is a problem that global optimization is unknown in the optimization problems. In this way, it is assumed that the best solution obtained so far is the global optimum, which is assumed as a food source for following the salp chain.

The pseudo-code for the SSA algorithm is shown in Fig. 2 [17]. This figure shows that the SSA algorithm begins the global optimum by starting several salps at random locations. Then, each fitting related to the salps are calculated and the location where they have acquired the best fitting is allocated to the variable F as a food source followed by the salp chain. Meanwhile, the value of c_1 constant is updated by Eq. 2. For every dimension, the location of the leader is updated by relation 1 and that of the followers by Eq. 4. If each salp goes out of the search space, they are returned to the border again. All the mentioned stages except for the initial value are iterated till consent is obtained.

The computing complexity of the SSA algorithm is considered as $O(t(d * n + Cof * n))$ where t shows the number of iterations, d is that of variables (dimension), n is that of solutions, and Cof is a target cost of the function.

```

Initializes the salp population  $x_i$  ( $i = 1, 2, \dots, n$ ) considering  $ub$  and  $lb$ 
While (end condition is not satisfied)
  Calculate the fitness of each search agent(salp)
   $F =$  the best search agent(salp)
  Update  $c_1$  by Eq. (2)
  for each salp ( $x_i$ )
    If ( $i == 1$ )
      Update the position of the leading salp by Eq. (1)
    else
      Update the position of the, follower salp by Eq. (4)
    end
  end
  Amend the salps based on the upper and lower bounds of variables.
end
return  $F$ 

```

Fig. 2. Pseudo-code of the salp swarm algorithm. [17].

4 Proposed Method

Our proposed method for scheduling the tasks of the virtual machines in the cloud computing area uses an optimized SSA based on the fitness function. First, a set of random answers created is assigned as the initial population. Each member of this set is called a salp. In the first stage, the fitness of salps produced randomly is calculated by the target function and the best salp is chosen among all salps and its location is determined by the location of the food source. In the following, the salps move towards the food source until they achieve the best food source (i.e., solution). In this algorithm, each salp is represented as a solution that moves for searching based on a mechanism in the problem space. In the suggested method, the salps are divided into two groups, the leaders and the followers. One group of salps named leader salps updates its location according to the food source and tries to move towards the existing food source and discover a better solution. If they find a better solution than the existing food source, the location of the leader salp is considered as its new location. The group salps follow each other, and if they discover a better solution for the food source location, the location of the salp follower is considered the new location of the food source.

4.1 The Task Scheduling Problem in the Cloud Area

The task scheduling problem in the cloud is allocating the settings of tasks to a set of sources. We have assumed a set of n tasks, $T = (T_1.T_2.T_3. \dots .T_n)$, and of m sources, which are virtual machines in targeted source research, $V = (V_1.V_2.V_3. \dots .V_m)$. The set of T includes the tasks which should be scheduled. Each task should be processed by virtual machines so that the completion time of all tasks is minimized as much as possible.

The main goal of task scheduling is to allocate optimally to the sources so that the lowest completion time of the tasks (i.e., makespan) and the lowest cost is obtained. The makespan shows the total required time for implementing all the tasks. The main goal of our research is to minimize the makespan using the modified SSA.

4.2 The Proposed Coding Method

Assume that an array of 200 tasks exists and each task has a value between 1–15. For example, if the second value of this array is 4, it shows that task 2 has been allocated to the virtual machine 4 and, if the seventh value of the array is 14, it means that, task 7 has been allocated to the virtual machine 14. Similarly, all the tasks T_1 to T_{200} are allocated to virtual machines V_1 – V_{15} . In Fig. 3, an example of allocating tasks to virtual machines is depicted.

T_1	T_2	T_3	...	T_i	...	T_{200}
V_2	V_4	V_{14}	...	V_1	...	V_7

Fig. 3. Allocation of tasks to virtual machines.

In the suggested algorithm, solutions are shown by a salp chains. Each solution of the suggested algorithm is shown by an array of natural numbers. The locations of all salps are stored in a 2-dimensional matrix named x . For instance, in a problem with n tasks and m virtual machines, the rows of a two-dimension matrix are considered as the number of the salp population. It means that the location of each salp is restored in a row of a matrix. The columns of the matrix are equal to n . Also, the content of each cell of the array shows the virtual machine number, which can be a number between 1 to m . Figure 4 shows an example of a salp.

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
↓	↓	↓	↓	↓	↓	↓	↓
4	2	3	4	5	2	1	3

Fig. 4. An example of a salp.

To begin the work, this salp can be produced as a desired number where this number is the same as the primary population of the algorithm that is adjusted. First, the population

is randomly generated and stored in a two-dimensional matrix where its rows are identical to the number of salps and its columns equal to those of tasks identified for the scheduling.

After generating the primary population of salps in the range of the problem answer, the fitness of all salps is assessed by all salps and the salp with the best fitness is determined. In this algorithm, it is assumed that a food source named F exists in a search space as a swarm target that all salps try to move towards it.

In the first stage of this algorithm, the location of the best generated salp (the best solution) is considered as the food source.

In the next stage of this algorithm, the salps are divided into two groups of leaders and followers. The number of salps is considered as the leader salp group and the remaining as the follower one. In the proposed algorithm, 50% of salps are considered as the leader group and the remaining 50% as followers. The location of the leader group is updated by Eq. 5.

$$x_j^i = F_j + \alpha \text{Randn}(\) \tag{5}$$

where x_j^i is the location of the leader salp, i , F_j the location of the food source in the j dimension, α the constant of the random moving step in the range of [0, 1] that is adjusted by the targeted problem, and $\text{Randn}(\)$ a random number with a normal distribution and determines a random step with a normal distribution for the leader group. Equation 6 updates the location of the follower group.

$$x_j^i = \frac{1}{2} (x_j^i + x_j^{i-1}) + c_1 \text{Randn}(\) \tag{6}$$

where x_j^i is the location of the follower salp i in the j dimension. The constant c_1 creates a balance between the exploration and discovery by generating an adaptive step, and this constant decrease consistently during the iterations; so, it leads to higher discovery in the first iterations and higher exploration in the end iterations if the algorithm, $\text{Randn}(\)$ is a random number with a normal distribution and determines a random step with this distribution for the leader group. The parameter c_1 is defined in Eq. 7 and is updated in each iteration.

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \tag{7}$$

Here, l is the current iteration and L the maximum of iterations.

In each iteration of the algorithm, after updating, first, the location of each salp is explored; if each salp goes out of the search space, it returns to the borders. Next, its fitness has been assessed based on the target function; if its fitness has been better than that of the food source, the location of the desired salp has been substituted for that of the food source.

It is noted that in the substitution of the salp location for the food source, there is a difference between the leader group and the follower group when swapping. In the case of the leader group, even if the fitness of the leader salp and food source are identical, the location of the leader salp is substituted for the food source, because the salps with equal fitness have different locations, and this mechanism is an effective alternative for

diversifying a search space, releasing from the local optimum, as well as discovering accurately surrounding the existing food source.

Based on this, the population of the leader group updated its location using the location of the food source. When the location of each leader salp group is substituted for that of the food source, the latter group has updated its location using the new location of the food source. Figure 5 depicts the algorithm’s pseudo-code of the optimized SSA.

The stages of the algorithm until reaching the end are continued. In the proposed algorithm, the condition for finishing the algorithm is the number of iterations.

```

Initializes the salp population  $x_i(i = 1, 2, \dots, n)$  considering  $ub$  and  $lb$ 
Calculate the fitness of each search agent(salp) from the fitness function.
 $F =$  the best search agent(salp)
Initialize  $\alpha$ 
While (end condition is not satisfied)
    Update  $c_1$  by Eq. (7)
    For each salp ( $x_i$ )
        If ( $r <= N * 0.5$ )
            Update the position of the leading salp by Eq. (5)
            Amend the sales based on the upper and lower bounds of variables.
            Calculate the fitness of the leading salp from the fitness function.
            If (the fitness of the leading salp  $<=$  the fitness of the  $F$ )
                 $F =$  position of the leading salp
            End If
        else
            Update the position of the follower salp by Eq. (6)
            Amend the salps based on the upper and lower bounds of variables.
            Calculate the fitness of the follower salp from the fitness function.
            If (the fitness of the follower salp  $<$  the fitness of the  $F$ )
                 $F =$  position of the follower salp
            End If
        End If
    End For
End While
return  $F$ 
    
```

Fig. 5. The pseudo-code of the modified SSA.

5 Fitness Function

The main goal of this research is to minimize the makespan, one of the most important targets for the task scheduling problem in the cloud areas. An example of task samples and task sizes is given in Table 1 and another is shown in Table 2 for virtual machines and the processor speed as individual values.

Table 1. An example of the tasks and their size.

Tasks	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}
Size	18	15	19	24	33	41	22	12	30	16	13	32

We aim to reduce the completion time of tasks in this research. This time duration is the longest completion time among virtual machines. If we consider T_i as the task size

Table 2. An example of virtual machines and their speed.

Virtual machine number	1	2	3	4	5
Processor speed	3.4	2.4	3.2	1.8	2.2

of i and C_j as the processor speed of the virtual machine j , we can obtain the makespan i from Eq. 8.

$$t_{exe}(i,j) = T_i/C_j \quad (8)$$

According to the allocated tasks for each resource and the length of desired tasks, there has been a completion time for tasks relative to the processor speed of the virtual machine for each of them.

For instance, assume that the tasks T_3, T_6, T_{10}, T_8 are allocated to virtual machine 2, the makespan of each task delivered to virtual machine 2 can be calculated as follows:

$$t_{exe}(3.2) = \frac{19}{2.4} = 7.9 \quad t_{exe}(6.2) = \frac{41}{2.4} = 17.1$$

$$t_{exe}(10.2) = \frac{16}{2.4} = 6.7 \quad t_{exe}(8.2) = \frac{12}{2.4} = 5$$

So, the completion time of tasks calculated on virtual machine 2 is:

$$t_{complete}(2) = 7.9 + 17.1 + 6.7 + 5 = 36.7$$

Similarly, the times for all virtual machines can be computed from the assigned tasks. The longest completion time of tasks amongst that for all virtual machines is calculated by Eq. 9:

$$Makespan = \text{Max}\{t_{complete}(j)\} \quad 1 \leq j \leq m \quad (9)$$

In Eq. 9, $t_{complete}(j)$ shows the completion time of tasks allocated to the virtual machine j . Minimizing Eq. 9 (i.e., the completion time of all tasks (makespan)) is the main target of this research.

6 Simulation and Results

In this section, the performance of the proposed algorithm (Modified salp Swarm Algorithm) is evaluated for solving the task scheduling problem in the cloud area and compared with other algorithms such as Standard salp Swarm Algorithm (SSA), Ant Colony Optimization (ACOr), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) in multiple scenarios [17]. MATLAB software has been used for simulation. The parameters and their initial values of the compared algorithms have been given in Table 3 and their description in Table 4. The simulation was run for four scenarios with parameters shown in Table 5 and the findings of each scenario are depicted in Fig. 6 using associated the chart.

Table 3. Parameters and the initial values of the compared algorithms.

Algorithm	Parameters and the initial values of the algorithms
GA	nPop = 40, MaxIt = 500, pc = 0.8, pm = 0.3, mu = 0.02, nc = 32, nm = 12, beta = 8, RWS = 0
PSO	nPop = 40, MaxIt = 500, C1 = 2, C2 = 2, w = 0.7
ACO	nPop = 40, MaxIt = 500, nSample = 40, q = 0.9, zeta = 0.1
SSA	nPop = 40, MaxIt = 500
Modified SSA	nPop = 40, MaxIt = 500, $\alpha = 0.19$

Table 4. Description of parameters used for comparing the algorithms.

For all	MaxIt = Maximum Number of Iterations nPop = Population Size
GA	Pc = Crossover Percentage nc = Number of Offsprings (Parnets) pm = Mutation Percentage nm = Number of Mutants mu = Mutation Rate beta = Roulette Wheel Selection (RWS) Pressure RWS = 0 or 1
PSO	c1 = Personal Learning Coefficient w = Inertia Weight c2 = Global Learning Coefficient
ACOR	nSample = Archive Size, q = Intensification Factor (Selection Pressure) zeta = Deviation-Distance Ratio
Modified SSA	α = Random step coefficient

Table 5. Parameters of the scenarios.

Scenario	The number of virtual machines	The number of tasks
First	10	150-200-250-300
Second	15	150-200-250-300
Third	20	150-200-250-300
Fourth	25	150-200-250-300

In the experiments, all algorithms used a number of 40 primary populations and a maximum of 500 iterations. Each scenario was run 20 times to obtain the results. The primary objective was to examine and minimize the makespan measure across different scenarios.

The results of our simulation study using a Modified Salp Swarm Algorithm (MSSA) for scheduling cloud computing tasks have been analyzed and compared with other well-known optimization algorithms, specifically the Standard salp Swarm Algorithm (SSA), (ACOr), (PSO), and (GA). The simulation results demonstrate that the proposed MSSA algorithm outperforms other algorithms in terms of task completion time.

Table 6. Data results of the first scenario.

No. of Tasks Algorithm	300	250	200	150
SSA	308.00	258.34	212.50	156.15
ACOr	282.69	236.85	192.44	144.69
PSO	275.05	230.64	186.71	139.91
GA	271.71	226.82	184.80	138.48
Average	284.36	238.16	194.61	144.80
STD	16.4148	14.07	12.68	8.014
MSSA	269.80	225.39	182.41	136.09
Average Improvement in MSSA	5.40%	5.66%	6.68%	6.40%

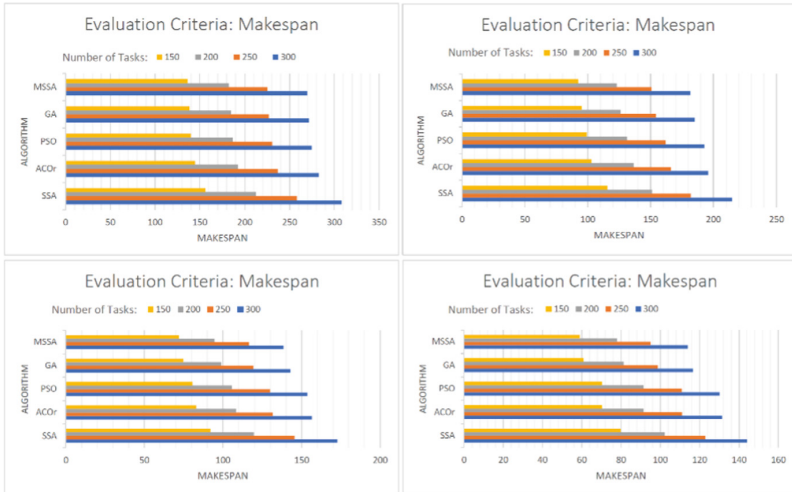


Fig. 6. Performance output for the four scenarios, comparing MSSA with other algorithms; MSSA shows lower calculation amount, which is desirable as lower values indicate improved efficiency in minimizing makespan for cloud computing task scheduling.

As shown in Table 6, the MSSA algorithm achieved an average completion time that was 5.40%, 5.66%, 6.68%, and 6.40% better than the average completion time of SSA, ACOr, PSO, and GA, respectively. Furthermore, the standard deviation of the MSSA algorithm was lower than that of other algorithms, indicating more consistent performance. The findings of this study provide valuable insights into the efficiency of different optimization algorithms for scheduling cloud computing tasks. The MSSA algorithm has shown substantial potential in reducing task completion time and improving the overall performance of cloud computing systems. Therefore, it can be concluded

that the MSSA algorithm can be a useful tool for scheduling cloud computing tasks in real-world scenarios.

7 Conclusion

The results from the stated scenarios show that the proposed algorithm had better performance compared to the other algorithms to solve the task scheduling problem in all four scenarios of cloud computing.

The results show that the makespan is reduced by increasing the number of virtual machines and vice versa. They also indicate that the optimized salp swarm algorithm has increased performance compared to the basic one. The outputs of all scenarios were similar and the MSSA is better in all case. As a result, the suggested method has shown better performance in all scenarios to solve the task scheduling problem in the cloud computing domain.

In addition, the findings of this study provide valuable insights into the efficiency of different optimization algorithms for scheduling cloud computing tasks. The MSSA algorithm has shown substantial potential in reducing task completion time and improving the overall performance of cloud computing systems. Therefore, it can be concluded that the MSSA algorithm can be a useful tool for scheduling cloud computing tasks in real-world scenarios.

Overall, while the study's results demonstrate the effectiveness of the MSSA algorithm in reducing task completion time and improving the overall performance of cloud computing systems, it is important to consider the limitations and scope of the study's findings. Future work could explore alternative performance metrics, evaluate the algorithm's robustness and scalability, and investigate its suitability for different cloud computing scenarios.

Acknowledgment. This material is based in part upon work supported by the National Science Foundation under grant #DUE-2142360. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.




References

1. Mosadegh, E., Ashrafi, K., Motlagh, M.S., Babaeian, I.: Modeling the regional effects of climate change on future urban ozone air quality in Tehran, Iran. arXiv: [abs/2109.04644](https://arxiv.org/abs/2109.04644) (2021)
2. Jamali, H., Karimi, A., Haghhighizadeh, M.: A new method of cloud-based computation model for mobile devices: energy consumption optimization in mobile-to-mobile computation offloading. In: Proceedings of the 6th International Conference on Communications and Broadband Networking, pp. 32–37. Presented at Singapore (2018). <https://doi.org/10.1145/3193092.3193103>
3. Chen, H., Wang, F.Z., Helian, N., Akanmu, G.: User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In: 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH), pp. 1–8 (2013)

4. Sehgal, N.K., Bhatt, P.C.P.: *Cloud Computing: Concepts and Practices*. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-77839-6>
5. Sun, H., Chen, S.-P., Jin, C., Guo, K.: Research and simulation of task scheduling algorithm in cloud computing. *TELKOMNIKA Indonesian J. Electr. Eng.* **11**, 6664–6672 (2013). <https://doi.org/10.11591/telkonnika.v11i11.3513>
6. Akilandeswari, P., Srimathi, H.: Survey and analysis on task scheduling in cloud environment. *Indian J. Sci. Technol.* **9**(37), 1–6 (2016). <https://doi.org/10.17485/ijst/2016/v9i37/102058>
7. Singh, A.B., Bhat, S., Raju, R., D’Souza, R.: A comparative study of various scheduling algorithms in cloud computing. *Am. J. Intell. Syst.* **7**(3), 68–72 (2017). <https://doi.org/10.5923/j.ajis.20170703.06>
8. Lavanya, M., Shanthi, B., Saravanan, S.: Multi objective task scheduling algorithm based on SLA and processing time suitable for cloud environment. *Comput. Commun.* **151**, 183–195 (2020)
9. Mansouri, N., Javidi, M.M.: Cost-based job scheduling strategy in cloud computing environments. *Distrib. Parallel Databases* **38**, 365–400 (2020). <https://doi.org/10.1007/s10619-019-07273-y>
10. Zubair, A.A., et al.: A cloud computing-based modified symbiotic organisms search algorithm (AI) for optimal task scheduling. *Sensors* **22**(4), 1674 (2022). <https://doi.org/10.3390/s22041674>
11. Rajakumari, K., Kumar, M.V., Verma, G., Balu, S., Sharma, D.K., Sengan, S.: Fuzzy based Ant Colony Optimization scheduling in cloud computing. *Comput. Syst. Sci. Eng.* **40**(2), 581–592 (2022)
12. Ghazipour, F., Mirabedini, S.J., Harounabadi, A.: Proposing a new job scheduling algorithm in grid environment using a combination of Ant Colony Optimization Algorithm (ACO) and Suffrage. *Int. J. Comput. Appl. Technol. Res.* **5**(1), 20–25 (2016)
13. Sharma, S., Tyagi, S.: A survey on heuristic approach for task scheduling in cloud computing. *Int. J. Adv. Res. Comput. Sci.* **8**, 1089–1092 (2017)
14. Mapetu, J.P., Chen, Z., Kong, L.: Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing. *Appl. Intell.* **49**, 3308–3330 (2019)
15. Saeedi, S., Khorsand, R., Ghandi Bidgoli, S., Ramezanpour, M.: Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing. *Comput. Ind. Eng.* **147**, 159–187 (2020)
16. Rajagopalan, A., Modale, D.R., Senthilkumar, R.: Optimal scheduling of tasks in cloud computing using hybrid firefly-genetic algorithm. In: Satapathy, S.C., Raju, K.S., Shyamala, K., Krishna, D.R., Favorskaya, M.N. (eds.) *ICETE 2019. LAIS*, vol. 4, pp. 678–687. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-24318-0_77
17. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **114**, 163–191 (2017). <https://doi.org/10.1016/j.advengsoft.2017.07.002>



Layer-Wise Filter Thresholding Based CNN Pruning for Efficient IoT Edge Implementations

Lakshmi Kavya Kalyanam^(✉) , Rajeev Joshi , and Srinivas Katkoori 

University of South Florida, Tampa, FL, USA
lakshmikavya@usf.edu

Abstract. This paper presents a novel approach for efficiently running convolutional neural networks (CNNs) on Internet of Things (IoT) edge devices. The proposed method utilizes threshold-based pruning to optimize pre-trained CNN models, enabling inference on resource-constrained IoT and edge devices. The pruning thresholds for each layer are iteratively adjusted using a range-based threshold pruning technique. The pre-trained network evaluates the accuracy of the pruned model and dynamically adjusts the pruning thresholds to maximize accuracy. The effectiveness of the proposed approach is validated on the widely-used LeNet benchmark network, with MNIST, Fashion-MNIST, and SVHN datasets. Our experimental results show that for the MNIST dataset, we can prune 62–64% of weights for an accuracy loss of 1–4%. Similarly, for Fashion-MNIST, we can prune around 64% for an accuracy loss of around 2.92%, and for the SVHN dataset, we can prune around 55% of weights for an accuracy loss of 1.7% on average, saving resources.

Keywords: Convolutional Neural Networks · IoT edge devices · Threshold-based pruning · Resource optimization

1 Introduction

In recent years, the field of artificial intelligence (AI) has made remarkable strides, with deep learning techniques, particularly convolutional neural networks (CNNs), leading the way. CNNs have played a pivotal role in achieving state-of-the-art results in image classification, object detection, and computer vision tasks. However, their computational complexity and memory requirements present challenges for deploying them on resource-constrained devices like mobile phones and embedded systems.

Optimizing CNNs for hardware has become increasingly crucial as these networks grow in complexity and size. The substantial computational and memory demands of CNNs make it difficult to deploy them on devices with limited resources such as smartphones, embedded systems, and IoT edge devices. Hardware optimization techniques can significantly reduce the computational requirements and memory usage of CNNs, making them more viable for deployment on resource-constrained devices.

© IFIP International Federation for Information Processing 2023

Published by Springer Nature Switzerland AG 2023

D. Puthal et al. (Eds.): IFIPIoT 2023, IFIP AICT 683, pp. 76–93, 2023.

https://doi.org/10.1007/978-3-031-45878-1_6

The benefits of optimizing CNNs for hardware are manifold, including faster inference times and lower energy consumption. Faster inference times are particularly important for real-time applications like autonomous vehicles and robotics, where rapid decision-making is critical. Similarly, lower energy consumption is desirable for such applications as it can extend battery life and reduce overall ownership costs.

Optimizing CNNs for hardware is crucial for making deep learning more accessible and feasible for a wider range of applications and devices. By reducing computational requirements, improving energy efficiency, and reducing costs, optimized CNNs have the potential to enable new possibilities and benefits for a range of fields from healthcare to education to entertainment. Ongoing research endeavors are required to devise inventive techniques that maximize CNN performance on resource-constrained devices, driving advancements in hardware optimization and facilitating a broader deployment of CNNs in real-world scenarios [8, 11].

Another important aspect of optimizing CNNs for hardware is the need for efficient memory management. In addition to reducing the size of the network, memory management techniques can be applied to optimize memory usage during the training and inference process. For example, techniques such as data augmentation and batch normalization can reduce the memory usage of the CNN during training. During inference, techniques such as model compression can be used to further reduce the memory requirements of the network [4].

Our proposed approach presents a novel technique called range-based threshold pruning for optimizing convolutional neural networks (CNNs). Traditional pruning methods rely on fixed threshold values, which may not effectively capture the weight distribution within each layer. In contrast, our approach leverages the weight ranges present in each layer to determine the pruning threshold dynamically. By profiling the weight matrix of a pre-trained CNN, we calculate the maximum weight value for each layer. This maximum weight serves as a reference for normalizing the pruning threshold. We then iteratively adjust the threshold for each layer based on its weight range, selectively pruning weights below the threshold. We have used profiling on multilayer perceptron (MLP) networks to find the least sensitive neurons to prune in [7]. The key advantage of range-based threshold pruning is its fine-grained control over the pruning process. By considering the weight ranges, we can preserve important network information while removing unnecessary weights. This leads to improved network efficiency and reduced memory footprint, making CNN models more suitable for deployment on resource-constrained devices. Our experimental results demonstrate the effectiveness of range-based threshold pruning. We evaluate our approach on the widely used LeNet5 network, pruning each layer individually. The results show that our method achieves significant weight reduction while maintaining high accuracy on three benchmark datasets, MNIST, Fashion-MNIST, and SVHN datasets.

Fine-grained pruning refers to the ability to selectively prune individual weights based on their importance or contribution to the network. Instead of

using a fixed threshold that applies uniformly to all weights, our approach takes into account the weight ranges to determine the optimal pruning threshold for each layer. This allows us to remove less important or redundant weights while preserving the critical ones. This makes the network more efficient and suitable for deployment on devices with limited resources, such as mobile phones or IoT edge devices, where computational power and memory capacity are constrained.

2 Background and Related Work

The background section provides a comprehensive introduction to several key aspects related to our study. We begin by discussing Convolutional Neural Networks (CNNs), which are widely utilized in various computer vision tasks. Next, we delve into different pruning methods, which aim to optimize CNNs by removing redundant weights or connections. We then focus on the LeNet-5 network, a well-known CNN architecture often used as a benchmark in pruning research. Finally, we highlight previous works that specifically address pruning techniques applied to LeNet-5 for the purpose of enhancing its efficiency and inference performance. This background information sets the foundation for our proposed approach of pruning and optimizing LeNet-5 for efficient inference.

2.1 Convolutional Neural Networks

CNNs (Convolutional Neural Networks) are a form of neural network utilized frequently for image and video recognition tasks. CNNs are composed of numerous layers, each executing a distinct operation on the input data. Figure 1 shows the architecture of a generic CNN with 8 layers. The first layer is a convolutional layer that applies filters to the input image to generate a set of feature maps. The following layers may include pooling layers, which downsample the feature maps, and fully connected layers, which classify or implement regression on the features. Backpropagation is commonly used to train Convolutional Neural Networks (CNNs), which modifies the network’s weights to minimize the difference

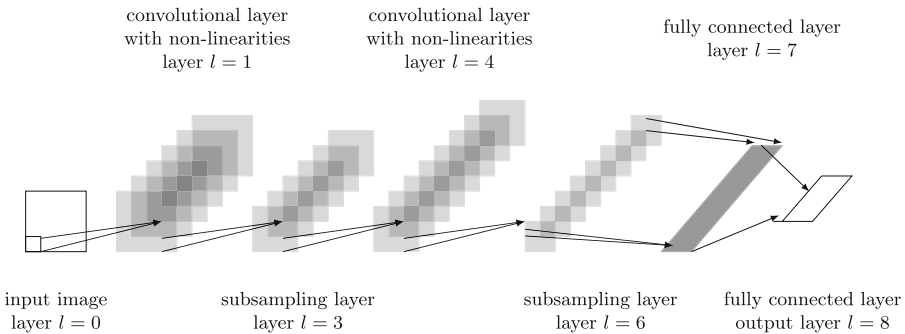


Fig. 1. Architecture of a traditional convolutional neural network.

between predicted outputs and actual labels. This training process typically utilizes massive image datasets with labels. CNNs have applications in domains such as natural language processing, speech recognition, recommendation systems, and biomedical image analysis, in addition to computer vision. In tasks such as sentiment analysis, language translation, speech recognition, and biomedical image classification, CNNs have demonstrated promising findings. CNNs' ability to automatically acquire hierarchical representations from input data makes them applicable to tasks and domains beyond computer vision. In addition to computer vision, CNNs have applications in disciplines such as natural language processing and drug discovery. In Sect. 1, we discussed the significance of pruning CNNs as a hardware optimization technique. However, additional techniques exist to improve the hardware performance of CNNs, and we will examine these techniques in the following sections.

- Quantization: This involves reducing the precision of the weights and activations of the network. For example, using 8-bit integers instead of 32-bit floating point numbers can significantly reduce the memory requirements and improve the speed of the network [2, 16, 21].
- Pruning: Pruning convolutional neural networks (CNNs) involves removing redundant weights from the neural network, which can significantly reduce the network's size and computational requirements. Pruning is one of several techniques used to optimize CNNs for hardware implementation, and it has been shown to be effective in reducing the hardware requirements of CNNs while maintaining their accuracy and performance [6].
- Weight sharing: This method reduces the memory requirements of convolutional neural networks (CNNs) by sharing the weights of multiple neurons. By reducing the number of unique weights stored in memory, weight sharing can considerably reduce the memory bandwidth requirements of CNNs, allowing them to be implemented more efficiently on resource-constrained devices [3].
- Hardware-specific optimizations: The network's architecture and algorithms require being configured to the hardware platform it will be implemented on. For instance, the network's speed and efficiency can be substantially increased by employing specifically designed hardware for convolutional operations [5].

Efficient memory management is also crucial when optimizing CNNs for hardware. Techniques such as data augmentation and batch normalization can be used to reduce memory usage during training, and model compression can be utilized during inference to further reduce memory requirements [4]. Overall, these methods can significantly improve the efficiency and performance of CNNs when implemented on hardware devices with limited resources, such as embedded systems or mobile devices.

2.2 Pruning CNNs

Neuron pruning is a technique in deep learning that aims to enhance the efficiency of a neural network by reducing the number of neurons. The aim is to

enhance the network’s resource efficiency to enable its operation on hardware that uses fewer processing units.

There are several methods for pruning neurons, including pruning by weight, pruning by connections, and pruning by neurons themselves. Weight pruning entails eliminating low-weight neurons. Connection pruning entails the complete elimination of neuronal connections. Neuron pruning entails the elimination of complete neurons from the network. Neuron pruning offers numerous advantages for hardware optimization by reducing the number of neurons in a network can decrease its computational cost, enabling it to operate more efficiently on hardware with restricted resources. Efficient resource utilization is crucial in IoT devices and edge computing due to limited hardware resources. Neuron pruning can enhance model accuracy by removing noise and overfitting, while also reducing computational expenses. Pruning neurons that have minimal contribution to the model’s accuracy could improve the accuracy by allowing the remaining neurons to solely concentrate on the crucial features of the input data. It enables the development of optimized and precise models that can operate efficiently on resource-constrained hardware. Structured pruning and unstructured pruning are two common methods for pruning CNNs. Here is an overview of each method:

- Structured pruning involves removing entire neurons or groups of neurons from the network. This method is called “structured” because the weights being pruned are part of a larger structure, such as a filter or a channel. By removing entire structures from the network, structured pruning can significantly reduce the network’s size and computational requirements [1].
- Unstructured pruning involves removing individual weights from the network. This method is called “unstructured” because the weights being pruned are not part of a larger structure. By removing individual weights from the network, unstructured pruning can achieve a higher degree of compression than structured pruning [13].

It is possible to further divide structured pruning into three distinct approaches:

1. Channel Pruning is the process of removing channels from a convolutional layer based on their importance scores. Channel pruning can be especially efficient because it can eliminate a large number of parameters with minimal or no loss of precision.
2. Filter pruning entails deleting whole filters, which are clusters of weights that each correspond to a particular output feature map. It is possible for filter pruning to be more precise than channel pruning, but this may necessitate more granular importance scoring.
3. Layer Pruning is the removal of entire layers from a network, which can be effective in lowering the computational cost of deep networks. However, layer pruning can be hazardous, as removing crucial layers can significantly degrade network performance.

The choice of approach depends on the particular application and hardware limitations, both structured and unstructured pruning can reduce the size and computational cost of CNNs. Unstructured pruning is better suited for cases where

the network has a large number of redundant weights that can be removed without affecting the network’s performance. This is because unstructured pruning allows for greater flexibility in removing individual weights, while structured pruning involves removing entire neurons or groups of neurons from the network. Structured pruning is better suited for cases where the network has a large number of redundant structures, such as filters or channels [1, 10].

2.3 LeNet-5

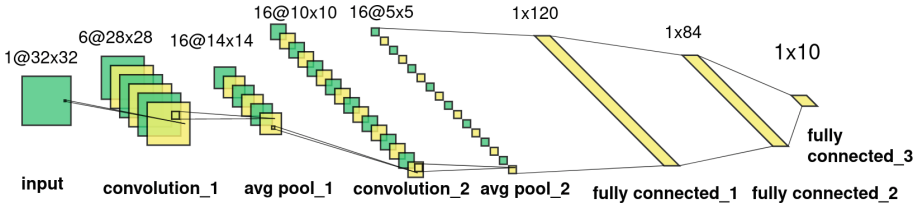


Fig. 2. LeNet5 architecture.

LeNet-5 is a historic CNN architecture that played a pivotal role in the development of deep learning and the modern artificial intelligence landscape. While it may not be the most advanced CNN architecture available today, it remains an important benchmark in the field and a testament to the power of convolutional neural networks for image-processing tasks [9]. Figure 2 shows the architecture of the network. The LeNet-5 architecture consists of seven layers:

1. **Input layer:** This layer receives the input image, which is typically a 32×32 grayscale image.
2. **Convolutional layer:** This layer applies six 5×5 filters to the input image, producing six feature maps. The filters have a stride of 1 and are padded with zeros to preserve the spatial dimensions of the input.
3. **Pooling layer:** This layer performs subsampling on each of the six feature maps produced by the previous layer. It uses 2×2 filters with a stride of 2, reducing the spatial dimensions of each feature map by a factor of 2.
4. **Convolutional layer2:** This layer applies 16 5×5 filters to the feature maps produced by the previous layer, producing 16 new feature maps.
5. **Pooling layer2:** This layer performs subsampling on each of the 16 feature maps produced by the previous layer, using the same 2×2 filters with a stride of 2.
6. **Fully connected layer (FC):** This layer flattens the output of the previous layer into a 120-dimensional vector and applies a fully connected neural network with 120 hidden units.
7. **Fully connected layer (FC):** This layer flattens the output of the previous layer into an 84-dimensional vector and applies a fully connected neural network with 84 hidden units.

The LeNet-5 architecture has been the subject of extensive research and experimentation over the years, with many researchers exploring new and innovative ways to optimize its performance for various tasks and hardware platforms. In addition, researchers have explored the use of different activation functions, such as ReLU and sigmoid, to improve the performance of the network. Some researchers have even experimented with the use of hybrid activation functions, which combine the strengths of multiple activation functions to achieve better results. We use the ReLU activation function in our network. LeNet-5 is extensively utilized in numerous applications, including handwritten digit recognition, object detection, and facial recognition. A notable application of LeNet-5 is the recognition of handwritten characters, such as postal codes on letters and checks dataset [14]. The architecture of LeNet-5 served as the basis for many other CNNs, including AlexNet, VGGNet, and ResNet, and has had a significant impact on the evolution of modern CNNs. LeNet-5 can be implemented in the Internet of Things applications requiring efficient and accurate image recognition.

LeNet-5 can accurately classify and analyze medical images, such as X-rays and MRI scans, for quicker and more precise diagnosis. By using the power of deep learning, LeNet-5 can solve difficult image recognition tasks quickly and accurately [9].

2.4 Related Work

In this section, we provide an overview of the related work in the field of optimizing convolutional neural networks (CNNs). Numerous techniques, including model compression, weight pruning, and quantization, have been proposed to boost the performance of CNNs. ADMM-NN is a framework for joint weight pruning and quantization of deep neural networks that use ADMM, a technique to solve non-convex optimization problems with combinatorial constraints. It achieves significantly higher pruning ratios than the state-of-the-art, with $85\times$ and $24\times$ pruning on LeNet-5 and AlexNet models, respectively, and $1,910\times$ and $231\times$ reductions in overall model size on these two benchmarks when combining weight pruning and quantization. The framework also prioritizes convolutional layer compression and accounts for hardware performance overhead. Additionally, ADMM-NN performs hardware-aware DNN optimizations, which take into account computation reduction, energy efficiency improvement, and hardware performance overhead due to irregular sparsity. This is achieved through a concept of break-even pruning ratio, which is the minimum pruning ratio of a specific layer that results in no hardware performance degradation. Furthermore, ADMM-NN achieves these results without accuracy loss and has shown highly promising results on other representative DNNs such as VGGNet and ResNet-50. Overall, ADMM-NN presents a robust and efficient solution for DNN model compression for hardware implementation [17].

You *et al.* [20] propose a Reconfigurable Sparse convolutional Neural Network accelerator design (RSNN) that combines software and hardware optimizations

for sparse CNN computation on FPGAs. The proposed design includes an efficient sparse dataflow for convolution, a load balance-aware pruning method, a kernel merging technique, and an efficient reconfigurable hardware accelerator design. RSNN achieves high performance on Xilinx Zynq ZC706, with 87.7 GOPS for AlexNet and 112.8 GOPS for VGG16. It is significantly more efficient than previous dense FPGA accelerator designs and even more efficient than sparse accelerators like NullHop. Salama *et al.* [18] proposed a pruning method to compress neural networks by removing entire filters and neurons according to a global threshold across the network without pre-calculation of layer sensitivity. The method has been proven viable by producing highly compressed models, including VGG-16, ResNet-56, and ResNet-110 on CIFAR10 and ResNet-34, and ResNet-50 on ImageNet. Additionally, the method compresses more than 56% and 97% of AlexNet and LeNet-5 respectively. They prune the network and retrain iteratively to reach the pruned model, and they find that it results in a network with more activated neurons.

3 Proposed Method

On a pre-trained CNN network, we propose a threshold-based optimization. The seven layers of the LeNet network are described in Sect. 2. The methodology involves a systematic process for determining the optimal pruning threshold and selectively pruning weights based on their magnitude. The pre-trained network evaluates the precision of the pruned model on the test set and adjusts the pruning threshold. We profile the weight matrix of a pre-trained CNN to identify the maximum weight value for each layer. This maximum weight value serves as a reference point for normalizing the pruning threshold. We calculate the normalized threshold by dividing a threshold value by the layer’s weights maximal weight value as shown in Eq. 1.

$$Threshold_{normalized} = \frac{Threshold}{Max(weight\ matrix_{layer})} \quad (1)$$

To identify the optimal threshold, we profile the weight matrix of a pre-trained CNN and calculate the weight range within each layer. By analyzing the weight distribution, we iteratively adjust the threshold, gradually increasing it until the desired level of accuracy is achieved or a heuristically set threshold value is reached. The absolute values of the weights below the threshold are set to zero, effectively pruning them. This iterative process is performed layer by layer, in a top-down fashion, allowing fine-grained control over the pruning procedure. To evaluate the effectiveness of our approach, we measure the sparsity achieved through pruning. Sparsity refers to the percentage of pruned weights compared to the total number of weights in the network. A higher sparsity value indicates a greater reduction in network size and computational requirements. Through experimental evaluations on benchmark datasets, MNIST, Fashion-MNIST, and SVHN, we demonstrate the efficacy of range-based threshold pruning in achieving significant weight reduction with low loss in accuracy. The Eq. 2 calculates

the sparsity of a pruned neural network, which represents the percentage of weights that have been pruned or set to zero compared to the total number of weights in the network. The LeNet contains two convolution layers and three fully connected layers with prunable parameters.

$$Sparsity = \frac{Number\ of\ Pruned\ Weights}{Total\ Number\ of\ Weights} * 100\% \quad (2)$$

Figure 3 depicts the steps involved in the pruning procedure. We load the pre-trained model and assign a set of initial threshold values to each layer of the model. Set a maximum weight value for each layer and normalize the threshold by dividing it by the maximum weight. While the accuracy of the pruned model on the test set is greater than a predetermined stop accuracy threshold, carry out the steps below for each layer:

1. Load the pre-trained model.
2. Profile the weight matrix of the model.
3. Calculate the optimal threshold for pruning by normalizing the threshold value.
4. Iterate through each layer of the model:
 - a Prune weights below the normalized threshold by setting them to zero.
 - b Evaluate the accuracy of the pruned model on a test set.
5. Check if the accuracy falls below a predetermined stop accuracy.
 - a If yes, stop pruning for the current layer freeze the pruning threshold for the current layer and proceed to the next layer.
 - b If no, continue pruning for the current layer.
6. If the current layer is the final layer in the network, cease pruning and proceed to the next layer. If not, proceed to prune the current layer.
7. Increase the threshold value and repeat the pruning process for the current layer.
8. Once all layers have been pruned, return the pruned model and the optimal threshold values for each layer.
9. Evaluate the accuracy of the optimized model and compare it to the accuracy of the original model.

The novelty of the proposed method lies in the approach of range-based threshold pruning for optimizing convolutional neural networks (CNNs). Traditional pruning approaches often use fixed threshold values, which may not represent the weight distribution inside each layer properly [12]. The proposed method, on the other hand, dynamically sets the pruning threshold based on the weight ranges inside each layer. By profiling the weight matrix and calculating the optimal threshold for each layer, the method achieves fine-grained pruning and maintains important network information while removing unnecessary weights. In addition, the iterative adjustment of the pruning threshold and the use of a stopping condition ensure that the pruning process is optimized to strike a balance between model size reduction and preservation of accuracy. Overall, the novelty of the proposed method lies in its adaptive and data-driven approach to

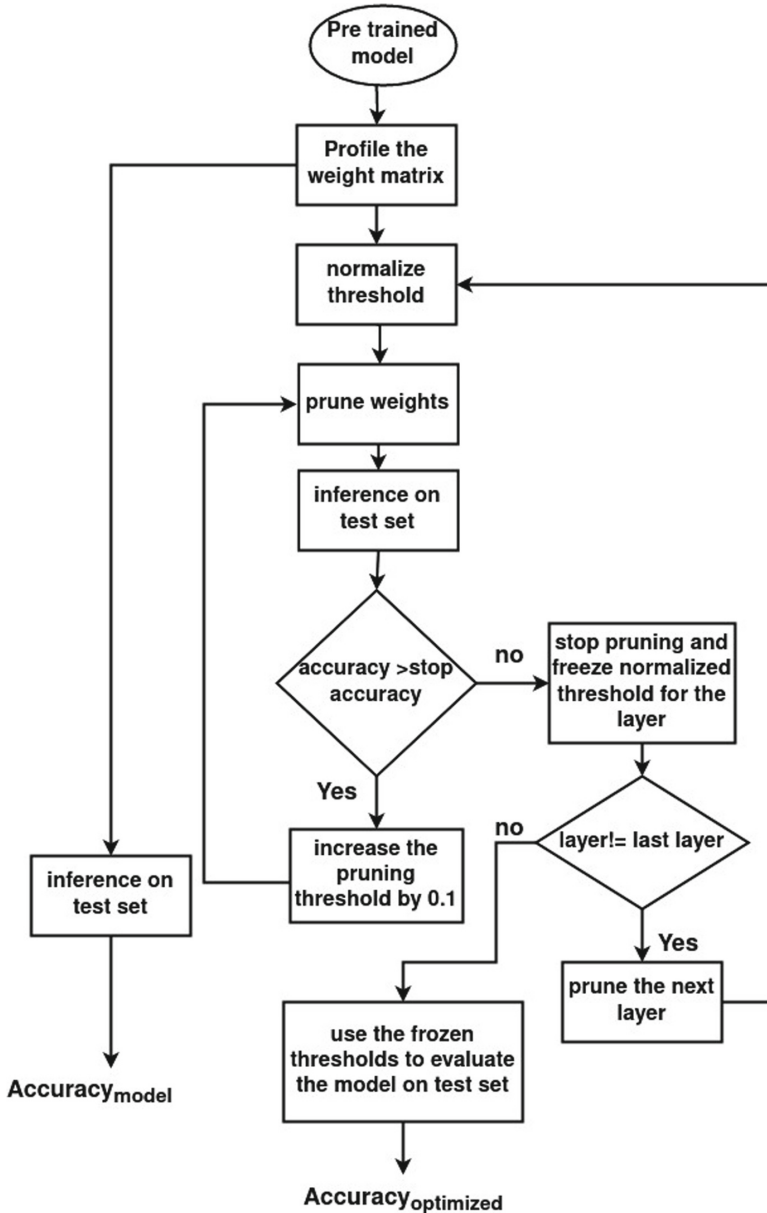


Fig. 3. Flowchart for proposed model based on range-based optimization.

threshold-based pruning, which effectively exploits the weight ranges within each layer and provides a systematic method for optimizing the efficiency of CNNs while keeping the loss in accuracy minimal. The goal of profiling is to understand

the characteristics of the weights in each layer, which can aid in determining an appropriate pruning threshold. By analyzing the weight range and distribution, it becomes possible to set a threshold that effectively prunes unnecessary or less important weights while preserving important network information. The profiling step helps in making informed decisions about the pruning threshold and contributes to the overall effectiveness of the pruning process.

4 Experimental Results

In this section, we present the experimental setup and results of our study. Our experimentation focuses on evaluating the effectiveness of the threshold-based optimization technique on the LeNet-5 network using three different datasets: MNIST, CIFAR10, and SVHN. The pretrained network is implemented and trained using the PyTorch framework [15] on a cloud computing platform. The pretrained network is the baseline model we make comparisons against.

To assess the performance of the proposed approach, we conduct multiple experiments where the network is trained independently on each dataset. The threshold-based optimization process is applied to each baseline model, and the resulting models are evaluated based on their accuracy and network sparsity. Sparsity represents the ratio of pruned weights to the total number of weights in the network. Higher sparsity values indicate a more substantial reduction in network size and computational demands.

4.1 MNIST (Modified National Institute of Standards and Technology) Dataset

The MNIST dataset is a widely used benchmark for evaluating machine learning algorithms, particularly those focused on image recognition and classification. It consists of 60,000 training images and 10,000 testing images of handwritten digits from 0 to 9. Each image is grayscale and has a resolution of 28×28 pixels. CNNs have achieved remarkable performance on the MNIST dataset, with some models achieving error rates as low as 0.23%. The MNIST dataset is a useful benchmark for evaluating machine learning algorithms new deep-learning architectures and optimization techniques [5, 9].

Table 1, shows the results of optimizing the LeNet-5 network on the MNIST dataset for five different models trained. The network has 61,706 trainable parameters. We observe that for the average optimal threshold, we can prune at least 60% of the prunable parameters, which are weights for an accuracy loss of 1–5%. Table 2, shows the optimization results for one model in detail, we can see the number of weights pruned for each layer and compare them to the number in the baseline model. The results highlight the importance of finding an optimal pruning threshold that strikes a balance between model size reduction and maintaining acceptable accuracy. Table 1 provides information about the pruning threshold, the number of weights in the baseline model, and the number of weights pruned for each layer of a neural network. In the fully connected

Table 1. Optimization results for Lenet with MNIST dataset.

Exp #	# weights	% weights pruned	accuracy (%) in regular model	avg threshold pruned model	accuracy (%) in pruned model	loss (%) accuracy
1	39,073	63.32	98.95	0.5	95.61	3.34
2	39,620	64.21	98.90	0.5	97.87	1.03
3	39,252	63.61	99.15	0.5	98.08	1.07
4	38,654	62.64	98.77	0.5	97.16	1.61
5	39,879	64.63	98.73	0.51	95.46	4.24

layer 1 (FC1) layer, 32,160 weights are pruned of 48,000, resulting in a sparsity of approximately 67%. This means that around 67% of the weights in the FC1 layer have been pruned, leading to a sparse network representation. We use the optimal pruning thresholds to calculate the average optimal threshold, and a total of 39,973 weights are pruned out of the initial 61,706 weights, demonstrating a significantly sparser matrix. Figure 4 showcases the impact of pruning on accuracy for different layers of the MNIST dataset. It compares the accuracy of the original model with the pruned model as each layer undergoes pruning. The plot demonstrates that applying dynamic pruning to the CNN model leads to accuracy changes in each layer. It is evident that the pruning thresholds have a more pronounced impact on the fully-connected layers (FC1, FC2, and FC3) compared to the convolutional layers (Conv1 and Conv2). The plot depicts a steeper decline in accuracy for the fully-connected layers as the pruning threshold increases, signifying their higher sensitivity to pruning. Conversely, the convolutional layers demonstrate a more gradual decrease in accuracy, indicating their relative resilience to changes in the pruning threshold.

Table 2. Optimization results for Lenet with MNIST dataset for one model.

layer name	optimal pruning threshold	# weights in baseline model	# weights pruned
conv1	0.67	150	100
conv2	0.67	2,400	1,608
fc1	0.67	48,000	32,160
fc2	0.30	10,080	5,541
fc3	0.30	840	564
Total	avg = 0.48	61,706	39,973

4.2 FASHION-MNIST

The Fashion MNIST dataset is a collection of images of clothing items, designed as a replacement for the traditional MNIST dataset. It consists of 70,000

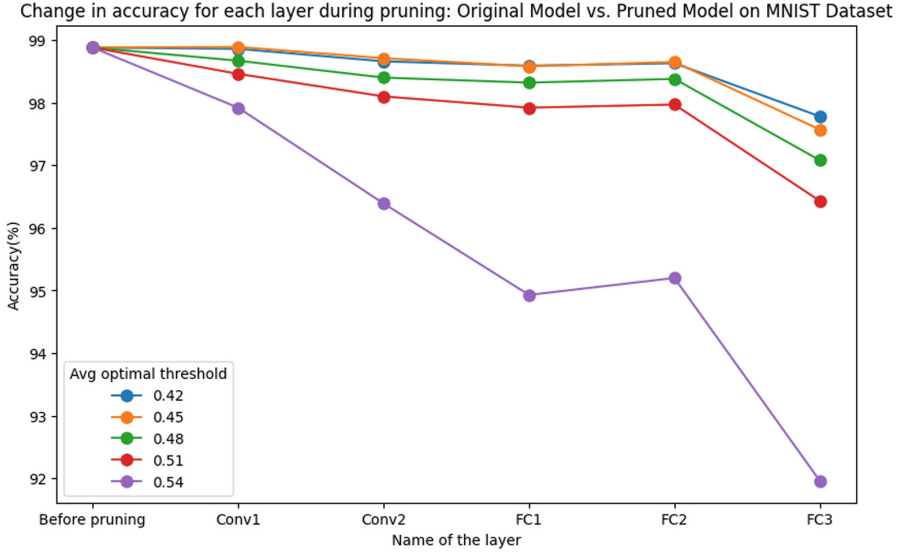


Fig. 4. Change in accuracy for each layer during pruning: comparison between the original model and the pruned model on the MNIST dataset.

grayscale images of 10 different categories of clothing, including t-shirts, dresses, and shoes. The images are 28×28 pixels and are divided into a training set of 60,000 images and a test set of 10,000 images. The Fashion MNIST dataset has also been used as a testbench for evaluating new techniques for data augmentation, network architecture, and training algorithms [19].

Table 3. Optimization results for Lenet with FASHION-MNIST dataset.

Exp #	# weights	% weights pruned	accuracy (%) in pruned model	threshold regular model	accuracy (%) in proposed model	loss (%) accuracy
1	34,160	70.90	89.63	0.44	87.39	2.24
2	28,907	65.07	89.67	0.41	87.24	2.43
3	28,237	63.56	89.15	0.38	87.11	2.04
4	30,218	68.02	89.98	0.41	86.76	3.22
5	25,976	58.47	89.02	0.32	85.34	3.68

Table 3 presents the results of optimizing the LeNet-5 network on the Fashion-MNIST dataset for five different models. The network has a total of 62006 trainable parameters, and we can see that the optimal pruning thresholds for each model differ. We observe that, for the average optimal threshold, we can prune an average of 65% of parameters, which are the network weights. The corresponding loss percentages range from 2.04% to 3.68%, suggesting a slight trade-off between accuracy and loss.

In Table 4, we present the optimization results for one model in detail, including the number of weights pruned for each layer and a comparison to the baseline model. We observe that the weights pruned for each layer vary, indicating that the optimal pruning threshold for each layer is different. This observation is supported by the results presented in Fig. 5, where we see the change in accuracy for each layer of the model for different average optimal threshold values.

To calculate the average optimal pruning thresholds, we use the values obtained for all models, and the results are presented in Table 3. Figure 5 illustrates the change in accuracy for each layer during the pruning process of the proposed model on the Fashion-MNIST dataset. It is worth noting that, with the highest average optimal threshold value of 0.48, the accuracy drops drastically at the first convolution layer. The analysis of the graph shows that as the pruning threshold increases, there is a gradual decrease in accuracy for each layer.

Table 4. Optimization results for Lenet with FASHION-MNIST dataset for one model.

layer name	optimal pruning threshold	# weights pruned	# weights baseline model
conv1	0.4	79	150
conv2	0.4	1,272	2,400
fc1	0.4	16,282	30,720
fc2	0.35	5,147	10,080
fc3	0.15	150	840
total		34,160	44,426
average threshold			0.48

4.3 SVHN Dataset

The Street View House Numbers (SVHN) dataset is a large-scale dataset of house numbers in Google Street View images. It consists of over 600,000 digit images, each of which is a cropped and resized section of a larger image. The digits range from 0 to 9 and are presented in a variety of fonts, styles, and sizes [14]. The SVHN dataset has become a popular benchmark for evaluating machine learning models for image recognition tasks, particularly those focused on digit recognition. It is often used as a more challenging alternative to the MNIST dataset, as it contains more variability in terms of digit appearance, background complexity, and image quality. SVHN is a real-world image dataset that is ideal for developing machine learning and object recognition algorithms. It requires minimal data preprocessing and formatting.

The dataset is available in two formats:

1. Original images with character-level bounding boxes.
2. Similar to MNIST, 32×32 images centered around a single character.

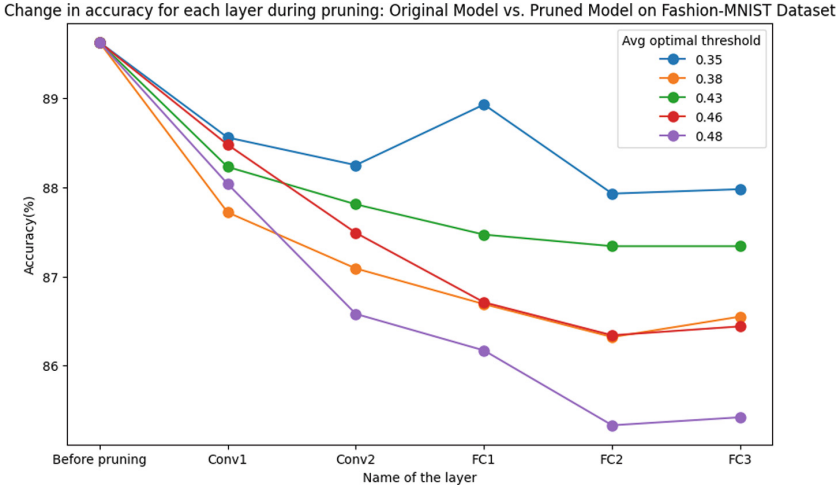


Fig. 5. Change in accuracy for each layer during pruning: comparison between the original model and the pruned model on the Fashion-MNIST dataset.

Table 5. Optimization results for Lenet with SVHN dataset.

Exp #	# weights	% weights pruned	accuracy (%) in pruned model	avg threshold model	accuracy (%) in proposed model	loss (%) accuracy
1	23,106	52.01	88.37	0.4	87.0	1.37
2	33,783	54.48	88.34	0.4	85.46	2.88
3	31,351	50.56	88.69	0.4	86.72	1.97
4	33,779	54.48	88.37	0.35	87.08	1.29
5	36,166	58.33	87.54	0.4	86.34	1.2

We use the format in 32×32 image sizes because LeNet-5 has an input size of 32×32 .

Table 5 shows the results of optimizing the LeNet-5 network on the SVHN dataset for five different models. The network has 62006 trainable parameters. We observe that, for the average optimal threshold, we can prune at least 50% of the prunable parameters (which are weights) for an accuracy loss of 1–3%. Table 6 shows the optimization results for one model in detail, including the number of weights pruned for each layer and a comparison to the baseline model. We use the optimal pruning thresholds of all models to calculate the average, which is presented in Table 5. In Fig. 6, we see the change in accuracy for each layer of the model for different average optimal threshold values. We observe that, for the highest average optimal threshold of 0.42, accuracy drops drastically at the first convolution layer.

Figure 6 illustrates the relationship between the pruning threshold and the accuracy of the pruned model for different layers in the SVHN dataset. Some

layers exhibit a more gradual decline in accuracy, while others show a steeper decline. This suggests that different layers contribute differently to the overall model performance and have varying degrees of sensitivity to weight pruning.

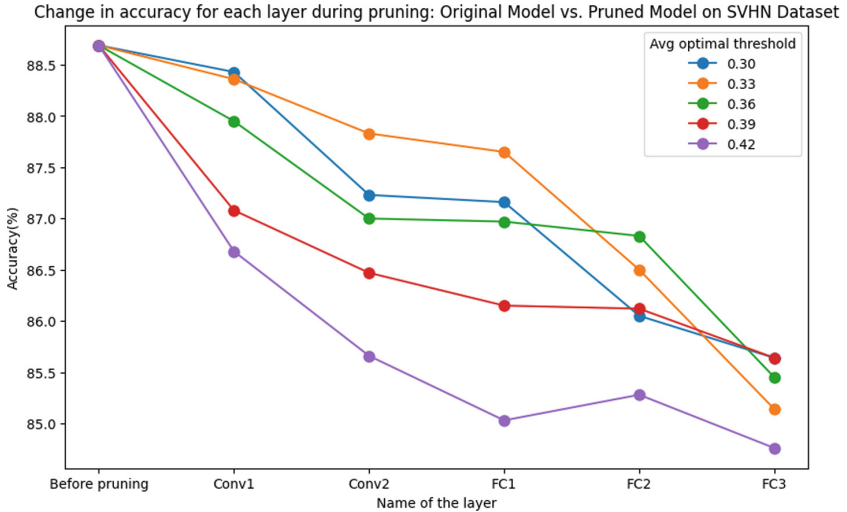


Fig. 6. Change in accuracy for each layer during pruning: comparison between the original model and the pruned model on the SVHN dataset.

Table 6. Optimization results for Lenet with SVHN dataset for one model.

layer name	optimal pruning threshold	# weights pruned	# weights baseline model
conv1	0.4	180	450
conv2	0.4	960	2400
fc1	0.4	19,200	48000
fc2	0.35	7,308	10080
fc3	0.15	166	840
total		36,166	62006
average threshold			0.34

In the experimental section, we evaluated the proposed pruning approach on the MNIST and Fashion-MNIST datasets using LeNet-5 architecture. Through the iterative pruning process, we achieved significant reduction in the number of weights while maintaining relatively high accuracy. The results demonstrate the effectiveness of the proposed method in reducing model complexity and computational requirements without sacrificing performance.

5 Conclusions and Future Work

For resource optimization of CNNs on hardware for IoT and edge applications, we proposed a range-based threshold pruning approach on LeNet and conducted experiments to compare the results on three different datasets: MNIST, Fashion-MNIST, and SVHN. Through our experiments, we observed that we can prune a significant portion of the weights while still maintaining a relatively small accuracy loss. Specifically, for the MNIST dataset, we were able to prune approximately 62–64% of the weights while only incurring an accuracy loss of 1–4%. Similarly, for Fashion-MNIST, we found that we could prune around 55% of the weights while only experiencing an accuracy loss of around 1.7%. The results of the SVHN dataset were also promising, as we were able to prune approximately 53% of the weights while incurring an average accuracy loss of only 1.7%. The experimental results on the three datasets demonstrated the effectiveness of the proposed method in achieving high sparsity levels while maintaining relatively high accuracy. These findings suggest that our proposed approach could be useful for reducing the computational resources required for deploying deep learning models in resource-constrained environments, while still achieving a satisfactory level of accuracy.

References

1. Anwar, S., Hwang, K., Sung, W.: Structured pruning of deep convolutional neural networks. CoRR abs/1512.08571 (2015). <http://arxiv.org/abs/1512.08571>
2. Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K.: A survey of quantization methods for efficient neural network inference. CoRR abs/2103.13630 (2021). <https://arxiv.org/abs/2103.13630>
3. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural network with pruning, trained quantization and Huffman coding. In: Bengio, Y., LeCun, Y. (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016, Conference Track Proceedings (2016). <http://arxiv.org/abs/1510.00149>
4. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems, pp. 1135–1143 (2015)
5. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861 (2017). <http://arxiv.org/abs/1704.04861>
6. Ide, H., Kobayashi, T., Watanabe, K., Kurita, T.: Robust pruning for efficient CNNs. Pattern Recogn. Lett. **135**, 90–98 (2020). <https://doi.org/10.1016/j.patrec.2020.03.034>, <https://www.sciencedirect.com/science/article/pii/S0167865520301185>
7. Kalyanam, L.K., Joshi, R., Katkooori, S.: Range based hardware optimization of multilayer perceptrons with ReLUs. In: 2022 IEEE International Symposium on Smart Electronic Systems (iSES), pp. 421–426 (2022). <https://doi.org/10.1109/iSES54909.2022.00092>

8. Kalyanam, L.K., Ramnath, V.L., Katkooari, S., Zheng, H.: A distributed framework for real time object detection at low frame rates with IoT edge nodes. In: 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), pp. 285–290 (2020). <https://doi.org/10.1109/iSES50453.2020.00070>
9. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998). <https://doi.org/10.1109/5.726791>
10. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. *CoRR abs/1608.08710* (2016). <http://arxiv.org/abs/1608.08710>
11. Li, J., Zhang, S., Liu, Y., Cheng, H., Wang, X.: Efficient convolutional neural network pruning based on batch normalization. *IEEE Access* **8**, 24010–24018 (2020)
12. Liu, J., Tripathi, S., Kurup, U., Shah, M.: Pruning algorithms to accelerate convolutional neural networks for edge applications: a survey. *CoRR abs/2005.04275* (2020). <https://arxiv.org/abs/2005.04275>
13. Liu, X., Zhang, X., Wang, H., Lu, J., Liu, Y.: Pruning convolutional neural networks for efficient inference on IoT edge devices. *IEEE Internet Things J.* **5**(5), 3664–3675 (2018)
14. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning* (2011)
15. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems*, vol. 32, pp. 8024–8035. Curran Associates, Inc. (2019). <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
16. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: imagenet classification using binary convolutional neural networks. *CoRR abs/1603.05279* (2016). <http://arxiv.org/abs/1603.05279>
17. Ren, A., et al.: Admm-nn: an algorithm-hardware co-design framework of DNNs using alternating direction methods of multipliers. In: *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. ASPLOS '19*, pp. 925–938. Association for Computing Machinery, New York (2019). <https://doi.org/10.1145/3297858.3304076>
18. Salama, A., Ostapenko, O., Klein, T., Nabi, M.: Pruning at a glance: global neural pruning for model compression. *CoRR abs/1912.00200* (2019). <http://arxiv.org/abs/1912.00200>
19. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017)
20. You, W., Wu, C.: RSNN: a software/hardware co-optimized framework for sparse convolutional neural networks on FPGAS. *IEEE Access* **9**, 949–960 (2021). <https://doi.org/10.1109/ACCESS.2020.3047144>
21. Zhou, A., Yao, A., Guo, Y., Xu, L., Chen, Y.: Incremental network quantization: Towards lossless CNNs with low-precision weights. *CoRR abs/1702.03044* (2017). <http://arxiv.org/abs/1702.03044>

Energy-Aware Security for IoT (EAS)



Shrew Distributed Denial-of-Service (DDoS) Attack in IoT Applications: A Survey

Harshdeep Singh, Vishnu Vardhan Baligodugula, and Fathi Amsaad^(✉)

Wright State University, 3640 Colonel Glenn Hwy, Dayton, OH 45435, USA
{singh.276,baligodugula.2,fathi.amsaad}@wright.edu

Abstract. With the rise of IoT and cloud computing, DDoS attacks have become increasingly harmful. This paper presents a survey of techniques for detecting and preventing DDoS attacks, specifically focusing on Shrew DDoS or low-rate DDoS attacks. We explore the use of machine learning for DDoS detection and prevention and introduce a new potential technique that simplifies the process of detecting and preventing DDoS attacks originating from multiple infected machines, typically known as zombie machines. As a future direction, we discuss a new technique to simplify the detection and prevention of shrew DoS attacks originating from multiple infected machines, commonly known as botnets. The insights presented in this paper will be valuable for researchers and practitioners in cybersecurity.

Keywords: Shrew DoS · Low-rate DDoS · LDDoS Attack · IOT · Communication security · Ns-2

1 Introduction

The Internet of Things (IoT) represents the physical thing (i.e. devices, objects, etc.) connected through the internet. It utilizes embedded systems, smart sensors, software, and other technologies to exchange data with each other over the internet. This survey explores Shrew distributed denial-of-service attacks to summarize the recent studies in secure communication of IoT applications. First, the survey explains denial-of-service attacks and the different prevention methods. It includes papers using the NS-2 simulator to generate a data set and detect the impact on communication devices. We also explore the security constraints of the victim network. For that, DDoS is classified into different parameters. We also discuss novel research that can protect IoT applications against DDoS attacks. For that, we explain the future work with inference from a real-world example by detecting the number of machines performing the DoS attack. The article is organized into different parts which are: a) Literature review: Review of previously published works b) Research challenges: Identification and examination of difficulties encountered in research c) Related Research: Investigation

© IFIP International Federation for Information Processing 2023

Published by Springer Nature Switzerland AG 2023

D. Puthal et al. (Eds.): IFIPIoT 2023, IFIP AICT 683, pp. 97–103, 2023.

https://doi.org/10.1007/978-3-031-45878-1_7

of studies that are related to the topic d) Discussion and Improvement: Conversation and enhancement of findings e) Future work: Plans for future research endeavors (Fig. 1).

2 Literature Review

Distributed denial-of-service (DDoS) attacks are designed to disrupt the normal traffic of a server or network by flooding it with malicious packets. These attacks are initiated by single or multiple infected systems within a network, making them robust. To address this issue, researchers have developed various preventive measures, and we discuss how to detect and counter multiple machines performing a DDoS attack on a single system. Many researchers examined DDoS attacks, including UDP flood, SYN flood, HTTP flood, protocol-exploitation flooding, reflection-based flooding, and amplification-based flooding. Using IoT as an example, we better understand how these different flooding and exploitation attacks are performed. The perception, network, and application layers are the three standard levels of IoT architecture.

Shrew DDoS attacks are a specific type of DDoS attack that involves continuously sending packets to a network at a low rate. Unlike flooding DDoS attacks, shrew attacks occur periodically, which makes them unique. However, they can still significantly reduce the quality of service on the victim’s system. To counter this type of attack, researchers have proposed various approaches, including the spectral template matching approach [1]. Researchers are utilizing the NS simulator to create a large-scale network simulation test-bed to simulate an attack flow over background traffic that exhibits self-similarity. TCL scripts are used to create simulations, and the GREP command-line interface is a helpful tool for writing the code. The OTCL configuration file, known as the “TCL script,” contains information on topology creation, node creation, link setup, and other parameters [2].

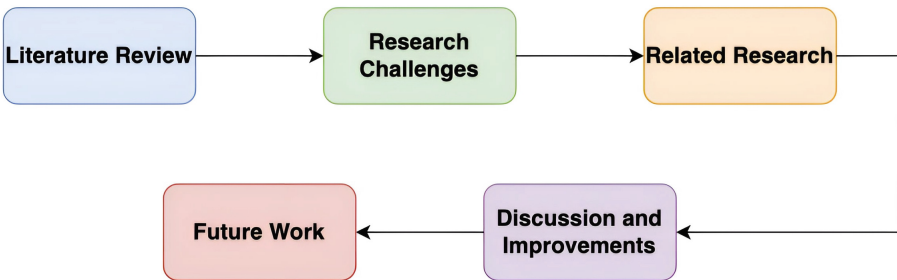


Fig. 1. Denial of service attacks of sections follows.

3 Research Challenges

When smart home appliances are connected to the internet, the likelihood of malicious attacks increases, posing significant security concerns for the home systems if any critical security area is breached. Additionally, the need for firmware updates for older IoT devices is a challenge. New devices are introduced to the market without considering legacy devices already paired with users' networks, leading to vulnerabilities that attackers can exploit.

Furthermore, privacy concerns arise with the developing new technology, including smart home appliances. It is essential to ensure that consumers' privacy is protected and their personal data is not compromised.

Distributed denial-of-service (DDoS) attacks can cause significant damage to companies by disrupting their services and resulting in financial losses [3]. The authors propose a machine-learning solution that involves building a Z-wave network on a Raspberry Pi gateway and using testbed equipment to carry out a DDoS attack by flooding the gateway with packets to address this issue. Wireshark is used to capture network traffic during the attack and employs various machine learning models such as logistic regression, decision trees, random forests, support vector machines, and deep learning models to assess the effectiveness of the proposed intrusion detection systems.

Chen Yu discusses shrew DDoS attacks, which exploit a system's temporary behavior to gradually reduce its capacity or service quality. These attacks are also known as "pulsing DDoS attacks" or "reduction of quality attacks." The paper proposes a distributed system for collaborative detection and filtering (CDF) that distinguishes shrew attack flow from legitimate TCP and UDP traffic flows by detecting a traffic stream with higher energy in the low-frequency band.

Agarwal studied using support vector machines to detect zombies carrying out DDoS attacks [5]. The paper used radial basis functions and polynomial functions for the training set and employed the structural risk minimization principle to generalize performance under small datasets. The training dataset was obtained from the ns-2 simulator and generated using the Gt-IMT topology generator. The study found that with an increase in zombie machines, the deviation in entropy increased using both the training and test data.

A new approach based on IoT sensors is proposed to create resistance against DDoS attacks [6]. The proposed solution involves using an Apache server as a load balancer, a Python script for traffic filtering, and the Wireshark network protocol analyzer for analyzing traffic.

4 Related Research

There are only a few researchers have explored utilizing machine learning algorithms to detect the number of machines involved in a DoS attack. Therefore, it is imperative to investigate and develop effective methods to detect the number of machines participating in a DoS attack using machine learning techniques. Practical implementation has also highlighted the importance of identifying the

number of machines involved in a DoS flood attack, as it is crucial to mitigate the situation efficiently [5,13] (Fig. 2).

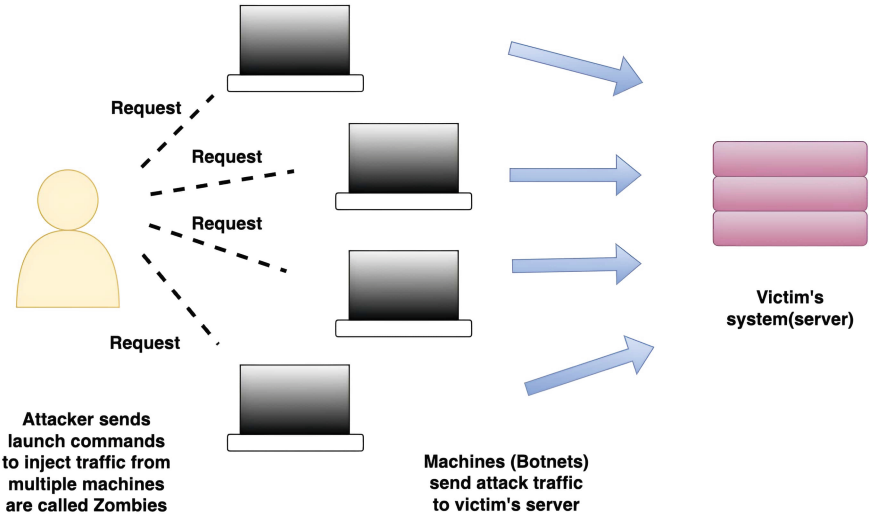


Fig. 2. Multiple machines performing DOS.

Shrew DDoS attacks aim to disrupt a system’s service through low-rate packet mugging techniques. It proposes a comprehensive survey analysis and offers effective means to prevent multiple machines or zombies from carrying out shrew DDoS attacks on a system, as outlined in [5].

A new method is proposed for optimizing the pulse rate in low and slow DDoS (LDDoS) attacks, assuming that the target bottleneck link’s bandwidth and buffer size are unknown. This proposed approach adopts an exploratory approach to determine the optimal pulse rate required to reduce the target’s TCP communication quality to achieve the desired attack effect in a limited attack scenario and environment. The approach involves intentionally limiting the throughput of the target network’s TCP flow to a specific level to decrease its quality. The strategy relies on a feedback mechanism that employs an observer, a bot node placed within the target network, to estimate the attack’s impact. Based on this feedback, the attack pulse rate is then iteratively increased.

The Control Attack function, executed by the master, an attack control node, performs several tasks at regular intervals during an observation window of W seconds. These tasks include determining the pulse parameters for each active attack node based on the parameters described. It determines the value of increment C , which represents the incremental increase in the number of active attack nodes c based on the estimated current attack effect. The Send Attack Pulse function sends a command to each active attack node to execute an LDDoS attack for W seconds. Each active attack node sends an attack pulse with parameters R , T , and L , where ΔR represents pulse rate, L represents duration,

and T represents interval. These parameters are aggregated in the bottleneck link to form $R = R_c, T$, and L . The function can estimate the characteristics of the bottleneck link and reduce the size of the aggregated attack pulse R to maintain the attack's stealth after achieving the target attack effect. New research presented a different perspective on mitigating DoS attacks is proposed [13]. The paper discusses a device called the Arduino Opla Device, which adds connectivity to devices in the home or workplace. The device comes with eight Internet of Things self-assembly projects that show how to turn everyday appliances into 'smart appliances' and build custom-connected devices that can be controlled with a mobile phone. One of the critical features of the Opla Kit is its companion app, which allows easy configuration and control of the board without requiring any coding. The Kit includes an Arduino MKR Wi-Fi 1010 board, sensors, and actuators. These sensors and actuators enable measuring and controlling parameters such as temperature, humidity, light, and motion. The author applies four machine learning algorithms to the dataset, including minimum, maximum, and average packet transit times over the network. The algorithms used are Naïve Bayes, Decision Tree, Support Vector Machine, and Multilayer Perceptron. After calculating the accuracy and F1 score, it was found that the J48 algorithm with 5-fold cross-validation performed better in detecting DoS attacks (Fig. 3).

5 Discussion and Improvements

Machine learning algorithms are commonly used to detect DoS attacks on networks by analyzing features such as packet count, segment size, flow duration, and acknowledgment flag count. Random forest is found to be the most effective algorithm in detecting attacks, while deep neural networks offer highly accurate results. In situations where physical hardware is unavailable, the NS-2 simulator has been used by some researchers to generate datasets for training their models. A new methodology is proposed to combat screw DDoS attacks from multiple machines based on survey results. This method involves detecting the pulse rate of the attack from each infected system and combining the different frequency pulses into one amplified frequency within the same time interval. This technique can be utilized to detect DOS attacks from multiple machines by generating concatenated output pulses in the form of readings. The attack packet frequency can then be calculated by adding up the pulse rate that detects the number of malicious packets within one wavelet, which serves as input for a complete attack scenario to prevent the DoS attack. Although this model is still conceptual, future research can expand on the present findings and develop a technical paper.

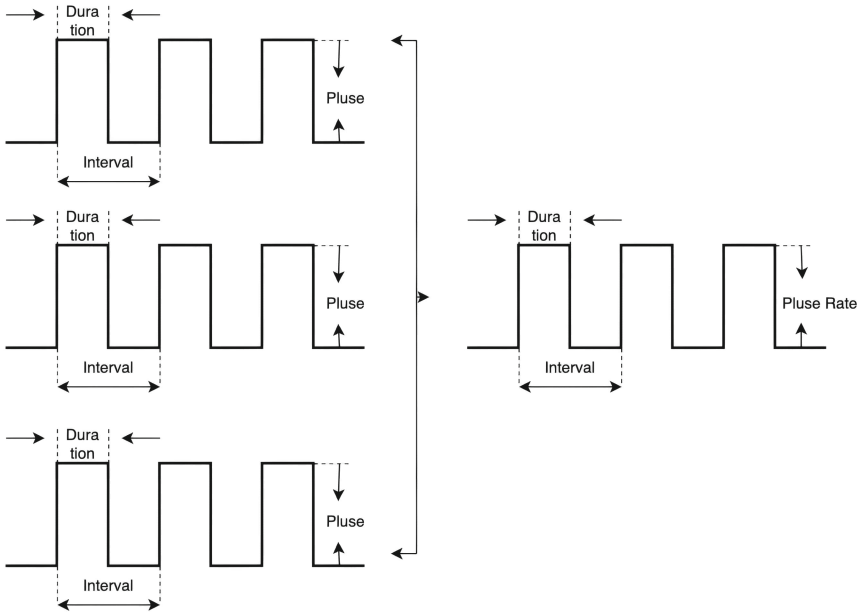


Fig. 3. Methodology on tackling screw DOS attack coming from multiple sources.

6 Future Work

As a further development to the research above, we aim to conduct a technical investigation focused on generating a single wavelet using multiple pulse rate frequencies obtained from various zombie machines during a low-rate DDoS attack. This wavelet will encompass all relevant pulse rate data, including the time intervals of maximum and minimum rates and the duration of the pulse rate during which the infected packets congest the network line.

References

1. Chen, Yu., Hwang, K.: Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. *J. Parallel Distrib. Comput.* **66**(9), 1137–1151 (2006)
2. Li, L., Lee, G.: DDoS attack detection and wavelets. *Telecommun. Syst.* **28**, 435–451 (2005)
3. Toutsof, O., Das, S., Kornegay, K.: Exploring the security issues in home-based IoT devices through denial of service attacks. In: 2021 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI). IEEE (2021)
4. Takahashi, Y., Inamura, H., Nakamura, Y.: A low-rate DDoS strategy for unknown bottleneck link characteristics. In: 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). IEEE (2021)

5. Agrawal, P.K., Gupta, B.B., Jain, S.: SVM based scheme for predicting number of zombies in a DDoS attack. In: 2011 European Intelligence and Security Informatics Conference. IEEE (2011)
6. Huraj, L., Šimon, M., Horák, T.: Resistance of IoT sensors against DDoS attack in smart home environment. *Sensors* **20**(18), 5298 (2020)
7. Sinha, M., et al.: Securing an accelerator-rich system from flooding-based denial-of-service attacks. *IEEE Trans. Emerg. Top. Comput.* **10**(2), 855–869 (2021)
8. Araki, R., Hsu, Y.-F., Matsuoka, M.: Early detection of campus network DDoS attacks using predictive models. In: GLOBECOM 2022–2022 IEEE Global Communications Conference. IEEE (2022)
9. Ali, J., et al.: A machine learning framework for prevention of software-defined networking controller from DDoS attacks and dimensionality reduction of big data. In: 2020 International Conference on Information and Communication Technology Convergence (ICTC). IEEE (2020)
10. Ibrahim, R.F., Al-Haija, Q.A., Ahmad, A.: DDoS attack prevention for internet of thing devices using ethereum blockchain technology. *Sensors* **22**(18), 6806 (2022)
11. Rani, S.V.J., et al.: Detection of DDoS attacks in D2D communications using machine learning approach. *Comput. Commun.* **198**,32–51 (2023)
12. Shen, Y., Fei, M., Du, D., Zhang, W., Stanković, S., Rakić, A.: Cyber security against denial of service of attacks on load frequency control of multi-area power systems. In: Li, K., Xue, Y., Cui, S., Niu, Q., Yang, Z., Luk, P. (eds.) LSMS/ICSEE -2017. CCIS, vol. 763, pp. 439–449. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-6364-0_44
13. Al-Maani, M., et al.: A classifier to detect number of machines performing DoS attack against arduino Oplà device in IoT environment. In: 2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet). IEEE (2022)
14. Rustam, F., et al.: Denial of service attack classification using machine learning with multi-features. *Electronics* **11**(22), 3817 (2022)
15. Biron, Z.A., Dey, S., Pisu, P.: Real-time detection and estimation of denial of service attack in connected vehicle systems. *IEEE Trans. Intelli. Transport. Syst.* **19**(12), 3893–3902 (2018)
16. Idhammad, M., Afdel, K., Belouch, M.: Detection system of HTTP DDoS attacks in a cloud environment based on information theoretic entropy and random forest. In: Security and Communication Networks 2018 (2018)



Honeypot Detection and Classification Using Xgboost Algorithm for Hyper Tuning System Performance

Vinayak Musale³, Pranav Mandke³, Debajyoti Mukhopadhyay¹, Swapnoneel Roy²(✉), and Aniket Singh³

¹ WIDiCoReL Research Lab, Mumbai, India

² University of North Florida, Jacksonville, FL 32246, USA
s.roy@unf.edu

³ Dr. Vishwanath Karad MIT World Peace University, Pune, India

Abstract. The purpose of this research paper is to detect and classify the hidden honeypots in Ethereum smart contracts. The novelty of the work is in hypertuning of parameters, which is the unique addition along with classification. Nowadays, blockchain technologies are the grooming technologies. In the current trend, the attackers are implementing a new strategy that is much more proactive. The attackers attempt to dupe the victims by sending seemingly vulnerable contracts containing hidden traps. Such a seemingly vulnerable contract is called a *honeypot*. This work aims to detect such deployed honeypots. A tool named Honeybadger has been presented. It is a tool that uses symbolic execution to detect honeypots by analyzing contract bytecode. In this system, we consider different cases such as fund movement between the contractor and contract, the transaction between sender and participant, and several other contract features in terms of source code length and compilation information. In the methodology used, the features are then trained and classified using a machine learning algorithm (XGBoost and gradient boosting with hyper tuning) into Balance Disorder, Hidden State Update, Hidden Transfer, Inheritance Disorder, Skip Empty String Literal, Straw Man Contract, Type Deduction Overflow, and Uninitialized Struct. Through this algorithm, we developed a machine-learning model that detects and classifies the hidden honeypots in Ethereum smart contracts. Hypertuning of parameters is the unique addition along with classification that separates the rest of the studies done in this area.

Keywords: Blockchain · Ethereum · Firewall · Honeypot

1 Introduction

A smart contract is a simple computer program designed to automate certain things or events under an agreement or contract. The first public blockchain to enable smart contracts is Ethereum. However, the contract design in Ethereum exhibits obvious flaws such as when a user sends some funds to a contract,

then that same user can withdraw the funds that were just sent. Due to these flaws, many attempts are to poach the vulnerabilities and target naive users. To prevent this from happening such a detection system's existence is crucial. When someone attempts to exploit this flaw, they are unable to recover the money initially sent. In this case, the malicious attacker is the one who creates the contract. On the other hand, the one who falls for the trap is known as the victim. Honeypots take advantage of the naivete of the victim by preying on the victim's ignorance of hidden attributes. There are hundreds of honeypots in Ethereum that have been discovered, and depending on the various techniques used, they are divided into many types. Malicious users creating smart contracts on the blockchain mainly benefit from such honeypots. Thus, there is a higher emphasis on security when public smart contracts are concerned.

Ethereum smart contracts have gained immense popularity from many sources, such as the media and industry, in the past few years. With the increase in popularity, there has been a rise in malicious users capitalizing on this situation, trying to find new opportunities to deceive newcomers and profit from them. Consequently, such lucrative individuals started luring others into contracts that seem exploitable, but they are a honeypot with a hidden trap that in turn benefits the one who creates the contract. The detection of this honeypot is a challenging task. Hence, this system is developed to detect and classify honeypots based on their features in the Ethereum blockchain platform.

1.1 Our Contributions

The research objective is to survey different approaches applicable to honeypot detection in smart contracts, classify them and select appropriate algorithms for the design of the system along with to evaluate the performance of the system. The paper describes the previous methods implemented in the form of a literature review followed by an architecture of the system and research methodology and accompanied by experimentations, results, and observations.

2 Related Work

This section focuses on the important work done and research carried out relevant to this topic. It also provides comprehensive information such as the authors, the methodology adopted, and the research gap identified. Research on honeypots focuses on measurement and detection. Torres et al. [3] first defined the honeypot in *The Art of Scam*. They used a variety of methods to measure the number of smart contracts and honeypots. They found that there were 150,000 smart contracts and 282 honeypots and that these numbers are growing each day. In the past, Honeypot systems were used neither detecting intrusion detection systems nor the firewall for a direct specific problem. Honeypots now are used as a part of security systems and what kind of problem they will offer a solution is depends on the design and usage purposes. Hence contrary to other information security equipment is not to be able to mention a honeypot that can give a general answer to every problem solution [3].

Riboldi et al. [22] created a low-interaction honeypot system to keep an eye on unlawful activity on VoIP systems. They acquired 3502 SIP Protocol-related events over a period of 92 days. They have viewed their system as being accessible as a VOIP environment with a firewall and intrusion detection system. The final results could also be embedded in suites of software used in active or passive network defense.

In their research, Shukla et al. [21] used a honeypot system to find malicious web URLs. On the client side, the system that was created with the Python programming language is used. The URL addresses are collected using a crawler on the client side, and if a visit is required, websites are then accessed. A trigger is set off by the signature-based intrusion detection system if certain URLs are malicious or contain vulnerabilities. As a result, security is available, and the dangerous URL addresses are preserved in the blacklist [21]. One of the observations is that the network security can be improved when such technologies are combined with the honeypot system.

For the study and visualization of harmful activities and connections, Koniaris et al. [11] have deployed honeypot systems. Two separate search honeypots have been set up. The first of these was designed to collect malicious software and often features a self-propagation option, while the second was designed as a trap system to collect harmful activity [11].

It has been discussed by Xiangfeng Suo et al. [23] how to use honeypot technology in intrusion detection systems. They have proposed using honeypot systems to fix issues with intrusion detection systems as part of their study [23]. In future works, this system can be evaluated on a range of synthetically generated polymorphic worms for accuracy, efficiency, and effectiveness.

A honeypot-based signature generator for computer network security has been carried out by Paul et al. [14]. The created technique has been utilized mostly to defend against attacks from polymorphic worms. The created system is also capable of isolating suspicious traffic and gathering a wealth of information on harmful traffic and worm assaults [14].

Markert J. et al. [12] have presented an effective analysis of a honeypot for WSN and show detection capabilities in the categories of known and unknown attacks in their paper [12].

Musca C. et al. [13] have presented methods for isolating malicious traffic by using a honeypot system and analyzing it to generate attack signatures automatically for the SNORT intrusion detection/prevention system in their study [13].

In their study, Haltas F. et al. [9] introduce BFH (BotFinder via Honeypots), an innovative automated bot-infected machine detection system based on BotFinder that locates the infected hosts in a real corporate network by using a learning technique [9].

Bashir U. et al. [1] have surveyed the overall progress of intrusion detection systems in their paper. They survey the literature's existing types, techniques, and architectures of Intrusion Detection Systems. Finally, they outline the present research challenges and issues [1].

A novel idea of proactive IDS was proposed in paper authored by Benmoussa H. et al. [2], who offered a survey of distributed intrusion detection systems based on intelligent and mobile agents. At first, they introduced the topic, then presented the limitations of classical IDSs. Furthermore, they presented the technologies of agent and multi-agent systems along with the benefits of using them to address the shortcomings of classical IDSs [2].

3 System Architecture and Research Methodology

This section explains the system structure and flow along with the methodology used. A gradient-boosting decision tree ensemble learning algorithm is a type of decision tree that can be used for classification and regression. It works a little bit like the forest algorithm but is better at finding better solutions. To create a more accurate model, ensemble learning algorithms mix different machine learning techniques. Both GBDT and random forest create models made up of several decision trees. The way the trees are constructed and joined makes a difference. Extreme Gradient Boosting (XGBoost) is a distributed, scalable gradient-boosted decision tree (GBDT) machine learning framework. The top machine-learning library for regression, classification, and ranking issues, it offers parallel tree boosting. The following Fig. 1 presents system structure having functional blocks such as dataset splitting, preprocessing, training using machine learning algorithms, etc. It explains the structure of the system in a sequential manner (Fig. 1).

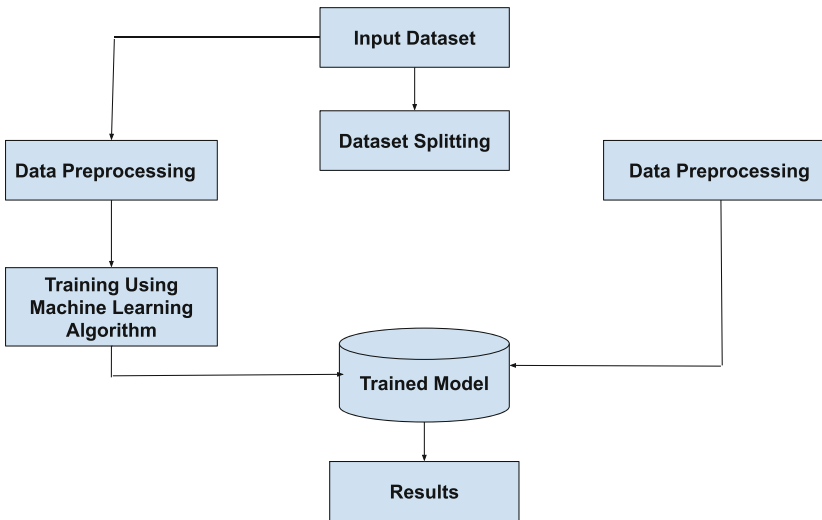


Fig. 1. System Structure for Data Preprocessing and Training

In this model, we are using an ensemble of multiple algorithms like decision trees, SVM, KNN, and XG-BOOST. Through the hyper-tuning of parameters like `max_depth`, `n_estimators`, `n_jobs`, etc., we were able to achieve higher classification accuracy than the earlier model. XGBoost is a powerful gradient-boosting solution that can help improve the performance and speed of machine learning models (Fig. 2). With XGBoost, trees are created concurrently instead of sequentially like with GBDT. The algorithm uses a level-wise strategy to evaluate the quality of splits at every possible split in the training set.

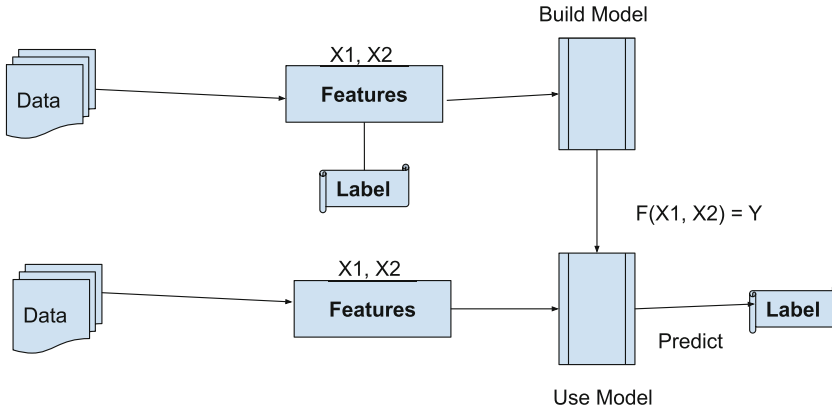


Fig. 2. Processing steps of XGBoost algorithm

4 Experimentation and System Modeling

The dataset is split into 10 parts. In each iteration nine-tenth of the dataset is used to train the model and the remaining one-tenth part is used for testing the accuracy. In every iteration, the one-tenth testing portion keeps on changing. With the help of such parameters and many others being hyper-tuned, the model is able to increase its accuracy by a combination of iterations mentioned in the results section. This makes the model more robust than what we have seen in the past in the form of literature review and gap identification (Fig. 3). A detailed explanation of the same is presented in the results and discussions section.

5 Results and Discussions

Additional data cleaning process is carried out before applying machine learning algorithms to conduct experiments. Mainly, the missing values are managed in our dataset by doing feature extraction. Because all of the transactions in the contract have errors, it is impossible to compute the aggregated characteristics

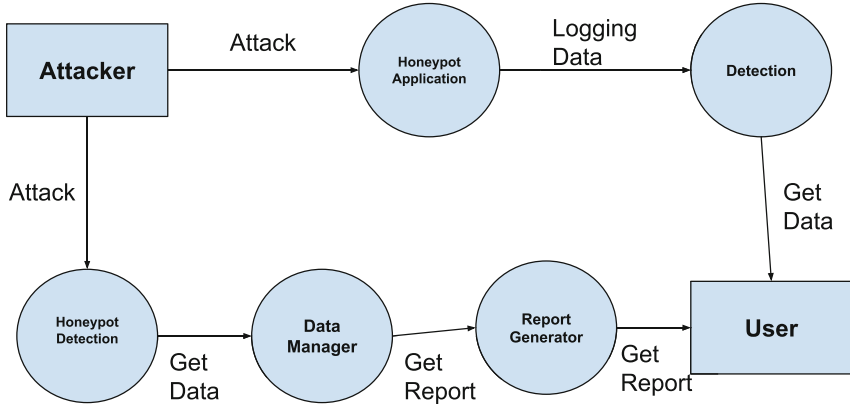


Fig. 3. Data Flow Diagram – DFD 1

of any type of transaction. When a contract has only one normal transaction, it is difficult to measure the difference in time or block number between subsequent normal transactions. Since all transactions have at least one normal outcome, any remaining characteristics of a normal transaction are always defined. Sometimes, we cannot find the right answer to a question. But even when we cannot find the answer, still used zeros to represent it.

The system uses a machine learning model to predict whether a contract is a honeypot or not. We use k -fold cross-validation to make sure that the model is effective at predicting data that has not been seen before. The data is divided into 10 different groups, each of which is used to test the algorithm. Each group of ten people will be used for testing once. We used a stratified k -fold cross-validation procedure to ensure that the data is representative of the population. There are more non-honeypot contracts in our dataset than honeypot contracts. This makes it hard for computer algorithms to learn how to classify contracts.

The XGBoost algorithm is set to use a scaling factor for the positive class so it would learn better. This model also measures how much power each fold has, the AUROC technology. AUROC stands for the *Area Under the Receiver Operating Characteristics* graph. The ROC curve shows how well a model can correctly identify different categories. The area under the curve shows how many times the model correctly classified a particular item as belonging to one of the categories. The performance of the XGBoost classifier on the selected dataset is shown below along with the results of the Honeypot classification model.

Iteration 1:	train ROC AUC 0.996 TN 141700 FP 1012 FN 0 TP 264
	test ROC AUC 0.980 TN 15729 FP 127 FN 1 TP 30
	train score - test score = 0.017
Iteration 2:	train ROC AUC 0.996 TN 141666 FP 1034 FN 0 TP 276
	test ROC AUC 0.997 TN 15757 FP 111 FN 0 TP 19
	train score - test score = -0.000

Iteration 3:	train ROC AUC 0.997 TN 141719 FP 991 FN 0 TP 266 test ROC AUC 0.962 TN 15744 FP 114 FN 2 TP 27 train score - test score = 0.035
Iteration 4:	train ROC AUC 0.996 TN 141528 FP 1190 FN 0 TP 259 test ROC AUC 0.967 TN 15688 FP 162 FN 2 TP 34 train score - test score = 0.029
Iteration 5:	train ROC AUC 0.996 TN 141607 FP 1109 FN 0 TP 261 test ROC AUC 0.981 TN 15724 FP 128 FN 1 TP 33 train score - test score = 0.015
Iteration 6:	train ROC AUC 0.996 TN 141620 FP 1096 FN 0 TP 261 test ROC AUC 0.967 TN 15754 FP 98 FN 2 TP 32 train score - test score = 0.029
Iteration 7:	train ROC AUC 0.996 TN 141566 FP 1146 FN 0 TP 265 test ROC AUC 0.963 TN 15735 FP 121 FN 2 TP 28 train score - test score = 0.033
Iteration 8:	train ROC AUC 0.997 TN 141732 FP 981 FN 0 TP 264 test ROC AUC 0.964 TN 15735 FP 120 FN 2 TP 29 train score - test score = 0.033
Iteration 9:	train ROC AUC 0.996 TN 141510 FP 1199 FN 0 TP 268 test ROC AUC 0.959 TN 15731 FP 128 FN 2 TP 25 train score - test score = 0.037
Iteration 10:	train ROC AUC 0.997 TN 141719 FP 987 FN 0 TP 271 test ROC AUC 0.955 TN 15765 FP 97 FN 2 TP 22 train score - test score = 0.041

Results: train: 0.996 ± 0.000 test: 0.970 ± 0.012

The following Table 1 represents comparative results. This system achieved a training accuracy of 0.996 and for testing accuracy of 0.96.

Table 1. Comparative results

Sr. No.	Features	Accuracy	Accuracy
1.	All Features such as symbols (fund flow), normal_transaction_value, contract_compiler_patch, etc	0.985 ± 0.002	0.968 ± 0.015
2.	Only Transactions	0.966 ± 0.004	0.954 ± 0.030
3.	Only Source Code	0.953 ± 0.002	0.942 ± 0.025
4.	Only Fund Flow	0.952 ± 0.002	0.938 ± 0.023
5.	Proposed Method (all features)	0.996 ± 0.000	0.962 ± 0.030

This section describes the readings of a particular traffic signal at one of the chosen squares. From Table 2, Signal No. (i) depicts the iteration number (1 to

10). N and Traffic Level used in Table 2 represents the count of vehicles and traffic density at that particular traffic signal respectively.

Table 2. Importance of features

Sr. No.	Feature	Accuracy
1.	symbol_83	0.683355
2.	normal_transaction_value_mean	0.105948
3.	contract_num_source_code_lines	0.058222
4.	normal_transaction_gas_mean	0.036259
5.	contract_compiler_patch_137	0.021729
6.	normal_transaction_gas_used_mean	0.015829
7.	normal_transaction_value_std	0.014596
8.	contract_compiler_patch_125	0.009466
9.	normal_transaction_block_span	0.00773
10.	symbol_73	0.007420

6 Limitations of Study and Future Scope

The model needs sufficiently large dataset as an input with similar data points to create a model that is capable of discerning honeypots from legitimate smart contracts. The proposed system can be extended for different datasets of honeypots. The system can be further improved by deep learning algorithms like 1D CNN.

7 Conclusion

To detect honeypots, the implemented system offered a step-by-step technique in for obtaining, processing, and analyzing Ethereum contracts. With the use of real data, we demonstrated how assumptions and theories regarding honeypot behavior may be compared, and we developed features for classification models. Even after removing all the contracts associated with a single honeypot approach from training, the machine learning models continued to perform well in terms of generalization. Most crucially, we demonstrated that our method discovered honeypots from two novel methodologies, which would not have been achievable using byte code analysis without the creation of new detection criteria manually. Of particular interests would be to assess honeypot detection systems with regards security (e.g. [7, 8, 10, 15–18]) and improvements in the sustainability (e.g. energy consumption [4–6, 19, 20]) of such detection systems.

Acknowledgement. The research has been supported in part by Fidelity National Financial Distinguished Professorship in CIS grant of S. Roy, 0583-5504-51.

References

1. Bashir, U., Chachoo, M.: Intrusion detection and prevention system: challenges & opportunities. In: 2014 International Conference on Computing for Sustainable Global Development (INDIACom), pp. 806–809. IEEE (2014)
2. Benmoussa, H., Abou El Kalam, A., Ouahman, A.A.: Towards a new intelligent generation of intrusion detection system. In: Proceedings of the 4th Edition of National Security Days (JNS4), pp. 1–5. IEEE (2014)
3. Camino, R., Torres, C.F., Baden, M., State, R.: A data science approach for detecting honeypots in ethereum. In: 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). pp. 1–9. IEEE (2020)
4. Castellon, C., Roy, S., Kreidl, P., Dutta, A., Bölöni, L.: Energy efficient Merkle trees for blockchains. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1093–1099. IEEE (2021)
5. Castellon, C.E., Roy, S., Kreidl, O.P., Dutta, A., Bölöni, L.: Towards an energy-efficient hash-based message authentication code (hmac). In: 2022 IEEE 13th International Green and Sustainable Computing Conference (IGSC), pp. 1–7. IEEE (2022)
6. Escobar, C.C., Roy, S., Kreidl, O.P., Dutta, A., Bölöni, L.: Toward a green blockchain: Engineering Merkle tree and proof of work for energy optimization. *IEEE Trans. Netw. Serv. Manag.* **19**(4), 3847–3857 (2022)
7. Garrett, K., Talluri, S.R., Roy, S.: On vulnerability analysis of several password authentication protocols. *Innov. Syst. Softw. Eng.* **11**, 167–176 (2015)
8. Gouge, J., Seetharam, A., Roy, S.: On the scalability and effectiveness of a cache pollution based dos attack in information centric networks. In: 2016 International Conference on Computing, Networking and Communications (ICNC), pp. 1–5. IEEE (2016)
9. Haltaş, F., Uzun, E., Şişeci, N., Poşul, A., Emre, B.: An automated bot detection system through honeypots for large-scale. In: 2014 6th International Conference On Cyber Conflict (CyCon 2014), pp. 255–270. IEEE (2014)
10. Khatwani, C., Roy, S.: Security analysis of ECC based authentication protocols. In: 2015 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 1167–1172. IEEE (2015)
11. Koniaris, I., Papadimitriou, G., Nicopolitidis, P., Obaidat, M.: Honeypots deployment for the analysis and visualization of malware activity and malicious connections. In: 2014 IEEE International Conference on Communications (ICC), pp. 1819–1824. IEEE (2014)
12. Markert, J., Massoth, M.: Honeypot effectiveness in different categories of attacks on wireless sensor networks. In: 2014 25th International Workshop on Database and Expert Systems Applications, pp. 331–335. IEEE (2014)
13. Musca, C., Mirica, E., Deaconescu, R.: Detecting and analyzing zero-day attacks using honeypots. In: 2013 19th International Conference on Control Systems and Computer Science, pp. 543–548. IEEE (2013)
14. Paul, S., Mishra, B.K.: Honeypot based signature generation for defense against polymorphic worm attacks in networks. In: 2013 3rd IEEE International Advance Computing Conference (IACC), pp. 159–163. IEEE (2013)
15. Roy, S.: Denial of service attack on protocols for smart grid communications. In: Research Anthology on Combating Denial-of-Service Attacks, pp. 560–578. IGI Global (2021)

16. Roy, S., Das, A.K., Li, Y.: Cryptanalysis and security enhancement of an advanced authentication scheme using smart cards, and a key agreement scheme for two-party communication. In: 30th IEEE International Performance Computing and Communications Conference, pp. 1–7. IEEE (2011)
17. Roy, S., Khatwani, C.: Cryptanalysis and improvement of ECC based authentication and key exchanging protocols. *Cryptography* **1**(1), 9 (2017)
18. Roy, S., Morais, F.J.A., Salimitari, M., Chatterjee, M.: Cache attacks on blockchain based information centric networks: an experimental evaluation. In: Proceedings of the 20th International Conference on Distributed Computing and Networking, pp. 134–142 (2019)
19. Roy, S., Rudra, A., Verma, A.: An energy complexity model for algorithms. In: Proceedings of the 4th Conference on Innovations in Theoretical Computer Science, pp. 283–304 (2013)
20. Roy, S., Rudra, A., Verma, A.: Energy aware algorithmic engineering. In: 2014 IEEE 22nd International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 321–330. IEEE (2014)
21. Shukla, R., Singh, M.: Pythonhoneymonkey: detecting malicious web URLs on client side honeypot systems. In: Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization, pp. 1–5. IEEE (2014)
22. da Silva Vargas, I.R.J., Kleinschmidt, J.H.: Capture and analysis of malicious traffic in voip environments using a low interaction honeypot. *IEEE Latin Am. Trans.* **13**(3), 777–783 (2015)
23. Suo, X., Han, X., Gao, Y.: Research on the application of honeypot technology in intrusion detection system. In: 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), pp. 1030–1032. IEEE (2014)



Electromagnetic Fault Injection Attack on ASCON Using ChipShouter

Varun Narayanan and Sriram Sankaran^(✉)

Centre for Cyber Security Systems and Networks, Amrita Vishwa Vidyapeetham,
Amritapuri, Kollam 690525, Kerala, India
amenp2csn21030@am.students.amrita.edu, srirams@am.amrita.edu

Abstract. Electromagnetic fault injection (EMFI) is a deliberate technique used to induce faults in a device by exposing it to electromagnetic interference. ASCON is a lightweight cipher that offers better performance than other ciphers, making it suitable for IoT devices with limited resources. However, the use of lightweight ciphers on hardware devices can pose a significant security risk against EMFI attacks, which can manipulate both the device's behavior and the implemented encryption algorithms. Our research used the ChipShouter, a specialized tool designed specifically for EMFI attacks on electronic devices. During these attacks, we intentionally exposed the M5STACK ESP32 Timer Camera (OV3660) module, on which we implement the ASCON algorithm, to electromagnetic pulses emitted by the ChipShouter. These pulses were directed at the PSRAM of the target device, where essential values such as plaintext, associated data, nonce, key, etc., are stored. Through the introduction of these pulses, we successfully inject faults and demonstrate the vulnerability of ASCON to EMFI attacks. To evaluate the impact, We test with different string sizes for input plaintext, namely 250 Kb, 500 Kb, and 1 MB. The results revealed that the fault injection percentages were as follows: 24% for the 250 Kb string size, 54% for the 500 Kb size, and 90% for the 1 MB size.

Keywords: ASCON · EMFI · ChipShouter · Hardware Security · Vulnerability testing · Side-channel attacks · Fault Injection

1 Introduction

Hardware security involves safeguarding physical components, systems, and devices against theft, manipulation, and unauthorized access. This encompasses protecting not only the hardware but also the software and data that it processes or stores. The significance of hardware security is especially crucial in industries like finance, healthcare, and national security where vital infrastructure and confidential information are at risk. Without adequate hardware security measures, these industries would be vulnerable to attacks that compromise the confidentiality, integrity, or availability of their systems and data. Testing and certification

procedures are employed to ensure that hardware meets strict security requirements and can withstand various attacks. These measures help establish trust with customers and stakeholders by ensuring that devices are trustworthy and secure [4, 16]. Hardware security is a fundamental aspect of overall security in industries dealing with sensitive data or critical equipment. It requires a multi-layered approach that combines various hardware security controls, certification and testing procedures, and supply chain security controls.

We use EMFI as an attack technique in our field it involves intentionally causing faults or glitches in a device by exposing it to electromagnetic interference. This technique is commonly employed in security evaluations to assess the resistance of electronic devices, including micro-controllers and smart cards, to attacks. By exposing a device to electromagnetic pulses of various frequencies and intensities, researchers can simulate real-world electromagnetic interference and study how the device responds. Through EMFI, security vulnerabilities and flaws in a device can be identified and used to develop defenses against electromagnetic attacks. Nevertheless, EMFI is also a possible attack vector that malicious actors could exploit to breach a device's security [12]. Thus, safeguarding the security of electronic devices necessitates awareness and protection against EMFI attacks.

In addition to the definition of EMFI, researchers have employed several methods to carry out this attack. Some common approaches [10, 11] include using a pulse generator with handmade injection probes and an oscilloscope to monitor the EMPs and target devices' status. AES algorithm implementation on FPGA and EMI attack injection onto the target device is another approach taken by researchers. Varying diameters of injection tips comprising ferrite core and copper wire have also been utilized by researchers. Furthermore, optical radiations such as a laser or vibrant white light have been applied in some instances. The ChipShouter, which is used in our work, was also used by the author in [7] for penetrating hardware wallets. This simple EMP generator sold by NewAE Technologies has various functions such as voltage, pulse width, pulse dead-time, pulse repeat, and other status functions. It costs roughly 3.3k USD and comes with four probes of two types: Counter Clockwise and Clockwise, each having 1mm and 4mm diameters.

The ChipShouter is used as an EMP (Electromagnetic Pulses) source to execute an EMFI attack on the M5STACK ESP32 Timer Camera (OV3660) module, which belongs to the ESP32 family. The module features an 8 MB PSRAM and 4 MB flash memory, as well as a camera with a maximum resolution of 2048×1536 pixels and an OV3660 sensor with a resolution of 3 MP. It is equipped with a reset button and an LED status indicator on the board, and its ultra-low power consumption, timing, sleep, and wake-up functions are enabled by the use of RTC (BM8563). To program the module, the Arduino IDE platform is used, and it can be connected to a computer through a USB type C cable. The ASCON for Arduino is programmed and implemented on the module to store input values such as plaintext, key, nonce, associated data, etc., on the PSRAM, where the ChipShouter's injection probe can be targeted. The RS232

terminal “Termite” is utilized to adjust the parameters of the ChipShouter. Our goal is to inject EMPs onto the PSRAM and accomplish the following objectives:

- Report the faults that were injected into the target.
- Determine the number of faults that were injected.
- Analyze the behavior of the device after the EMPs were injected.
- Record the frequency of fault injection that occurred.

Our study intends to evaluate how EMPs affect the PSRAM of the M5STACK ESP32 module. We are particularly interested in identifying the faulty bits and the locations of the subsequent faults, as well as how the behavior of the module changes when it is subjected to EMPs. This module was chosen due to its large flash and RAM capacities to support the ASCON library and its 8 MB PSRAM, which can hold variables of various lengths, including plaintext, keys, nonces, and related data. This allows direct modification of ASCON program values by injecting EMPs. To perform the research, a variable string of plaintext values with sizes of 250 Kb, 500 Kb, and 1 MB is used. The resulting proportion of fault injection is as follows: 24% for a 250 Kb string, 54% for a 500 Kb string, and 90% for a 1 MB string.

2 Background

2.1 Electromagnetic Fault Injection

EMFI assaults can affect any conventional Integrated Chip (IC), including CPUs, SRAM, voltage regulators, microcontrollers, and processors. An EMFI has the power to change an IC’s behavior by applying electromagnetic pulses to it; the resulting electromagnetic interference can create voltage spikes or transient voltage dips, which can interrupt the current flow and lead to an IC’s failure. This interference may affect the circuit’s clock, data, or power supply connections, which might lead to errors or unexpected behavior [1]. By using specialized equipment an attacker can inject EMPs onto the target device/module. This specialized equipment commonly consists of a pulse generator and a magnetic coil. This equipment can be available commercially at decent rates and can support multiple features like varying voltage, pulse width, pulse dead time, number of pulses, etc. via programming. One such device is called the ChipShouter, as shown in Fig. 1 which is used in our work.

2.2 PSRAM

PSRAM, also known as pseudo-static random access memory, is a kind of memory that combines the high-density storage and low cost of dynamic random access memory (DRAM) with the quick access of static random access memory (SRAM). For a variety of applications, including mobile phones, gaming consoles, industrial control systems (ICS), and digital cameras, this component was designed to provide high-speed data access. Applications that require speedy

read-and-write operations but don't require the high capacity or high power consumption of traditional DRAM or flash memory would particularly benefit from it. Like all other integrated circuits (ICs), the PSRAM is vulnerable to electromagnetic interference (EMI), a disturbance that can cause data loss, bit flipping, and data corruption. Knowing these details, we made the decision to launch an EMFI attack on the module's PSRAM because the ASCON algorithm is programmed in such a way that the code stores data like plaintext, associated data, nonce, key, etc. are all stored in the PSRAM. To support the hypothesis that ASCON is vulnerable to EMFI the EMPs from the ChipShouter are injected into the PSRAM.

Figure 2 shows an 8 MB PSRAM.



Fig. 1. ChipShouter by NewAE

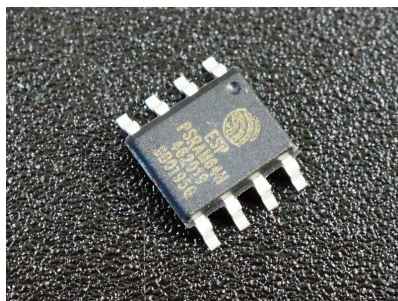


Fig. 2. 8 MB PSRAM

2.3 ASCON

The Austrian Graz University of Technology's research team created ASCON in 2014 [15]. Its design aims to make it lightweight effective, and safe against side-channel attacks. Several variations of the ASCON family are available, each with a different security level and block size, and they may be chosen according to the particular security needs of the application. The permutation function in the method is used to convert the input data into an encrypted output. It is based on a sponge structure. The final encrypted output is created by applying a set of round keys that are created using the encryption key to the input data throughout several rounds. Applying a different key to the input data and the encrypted output results in the generation of the authentication tag. The

National Institute of Standards and Technology (NIST) advises ASCON for use in lightweight cryptography, and the European Telecommunications Standards Institute (ETSI) has standardized it. It is utilized in several applications, including embedded systems, wireless communication protocols, and the Internet of Things (IoT). The ASCON encryption process is divided into 4 parts:

1. Initialization: Starts the state with the key K and the nonce N .
2. Associated Data Processing: This method changes the state with related data blocks. A_i .
3. Plaintext Processing: Introduces plaintext blocks P_i into the state and retrieves ciphertext blocks C_i .
4. Finalization: Adds the key K and subsequently retrieves the tag T for authentication.

After injecting each block except the last plaintext block, the entire state is subjected to the core permutation p^b . During initialization and finalization, a stronger permutation p^a with more rounds is used. The number of rounds for a and b and the speed and capacity of the sponge are determined by the specific version of ASCON (Fig. 3).

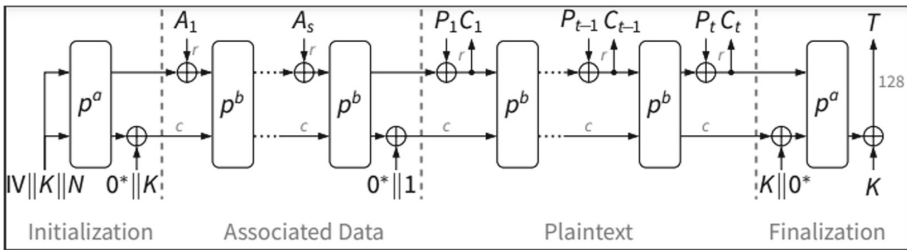


Fig. 3. Duplex Sponge model of ASCON

As a result, we picked ASCON because it has not been utilized in EMFI attacks before and because it is a lightweight cipher that may be used in devices with limited resources that contain ICs and other components that are vulnerable to EMFI attacks. By exploiting this vulnerability we can inject faults in the ASCON to see its resilience [13].

3 Related Work

3.1 Electromagnetic Fault Injection

A 32-bit microcontroller built-in CMOS 130 nm technology underwent an EMFI by the authors of [4] a few years ago. The microcontroller is based on an ARM Cortex-M3 CPU with a 56 MHz operating frequency and no cache memory built in. They apply pulses with an amplitude ranging from -200V to +200V and a pulse

width of 10ns-200ns to the target device using an EMP generator with a magnetic coil. In order to retrieve data from the device they use a Serial Wire Debug (SWD) which is an alternative to the Joint Test Action Group (JTAG). SWD is a 2-pin interface that can be used to communicate with the microcontroller and has the same protocol as that of a JTAG, It employs the bi-directional wire protocol that is a standard for ARM CPUs and was developed by the ARM Debug programmer. It is the author's intention to provide readers with a basic grasp of the vulnerabilities that an EMFI may introduce into embedded software. They were able to show that an EMFI can lead to timing constraint violations during bus transfers from flash memory, which would enable an attacker to circumvent some safeguards against more traditional timing fault-injection techniques like clocks or voltage glitches.

Injecting EMPs into a Solo Key open-source FIDO2 authentication key as well as a Trezor bitcoin wallet using the ChipShouter was done by Colin O'Flynn [7]. The vulnerable code, which also contains the EMFI, is a part of the USB stack. The author was able to obtain the device's private info using this method. The attack used a standard logical flow that almost all USB stacks provide to allow reading up to 64 Kb of data from the device. The author further notes that to successfully execute this attack, the exact timing of the fault injection in reference to the location of the USB transactions is necessary. As a result, the experiment can be supported by an open-source application called PhyWhisperer-USB, which triggers a fault injection platform from a USB message with incredibly high temporal precision.

Beckers, Arthur, et al. [10] explored the impact of EM pulses on the flash memory of an ATmega328p 8-bit microcontroller. Their study aimed to develop a fault model that accurately depicts the properties and consequences of the injected errors. To conduct their EM fault injection experiments, they employed the Langer EM fault injection setup, which includes various probes generating magnetic fields, electric fields, and current pulses, as well as a power generator and a magnetic field pulse source capable of supplying up to 500V to the magnetic probe. In their experiment, the authors tried two different approaches: one included filling the flash memory entirely with zeroes (0×00), while the other involved filling it entirely with ones ($0xff$). They discovered that only the data read from the flash memory can be altered and that the modification only lasts as long as the flash memory values are set to $0xff$. This indicates that the real values in the flash memory are not changed only the process of reading from the flash memory was impacted.

Dehbaoui, Amine, et al. [2] explain the usage of an EM channel to execute active attacks against a hardware AES built in an FPGA and a software AES operating on a CPU. The experiment is carried out on two different platforms, one of which has a smart card emulation board made up of an 8-bit AVR Atmega 128 micro-controller coupled with a 128 Kb flash program memory, 4KBEEP-ROM, and 4KBSRAM. The working voltage and frequency of this microcontroller are 4.5-5.5V and 3.57 MHz, respectively. The second platform uses an FPGA from the Xilinx Spartan 3 series. Both platforms use the same EMP sys-

tem, which is made up of a control PC, a mechanized stage, a pulse generator, and a magnetic probe. The pulse generator's output voltage ranges from 1-100V with pulse widths of 10–100ns, and the probe being used here has a diameter of 500 m. After carefully examining the data from both systems, it was found that the errors were brought on by omitting to carry out an instruction that was meant to be carried out during the EMP. As a consequence, the AES's bytes were faulted independently by altering the injection time. The EMP caused single-bit and multi-bit flaws that influenced the FPGA's AES calculations.

A recent study on EMFI attacks [5] looked at how they are perceived as a real threat to modern security. Their study followed the EMFI principles by examining several techniques for calibrating fault injection benches. By contrasting EMFI to other fault injection techniques, they demonstrate how practical and expensive various fault injection attacks are. In their conclusion, they highlight AI (Artificial Intelligence) techniques that are used in the fault injection area but not in side-channel analysis. Researchers are constantly looking for new ways to improve attack strategies because side channel analysis employs machine learning techniques to launch powerful attacks. The goal of the study is to bridge the knowledge gap between the effectiveness of EMFI assaults and the full range of available defenses against them.

Majeric, Fabien, et al. [3] deal with the injection of EMPs into an SoC (System on Chip). In this work, EMPs are introduced onto the SoC using an 8-bit oscilloscope and an EM probe called LANGER ICR HH-150. The SoC's 32-bit CPU, built using CMOS 40nm technology, has one ARM Cortex-A9 core operating at 1GHz. After injecting the pulses, the authors discovered that 3% of the time the device is in a silent condition, meaning there are no responses from the SoC and a reset is required for the experiment to continue, and 0.3% of the time the device is replying with faulty ciphers. Only one of the three types of incorrect cipher replies that were identified as such had a direct connection to the HW-AES procedure.

3.2 Laser Fault Injection

In [6] the authors propose using a Differential Fault Analysis(DFA) on an Atmel ATXMega16A4U, which has a specification of 2 Kb SRAM, 1 Kb EEPROM, 20 Kb of flash, and a clock speed of up to 32 MHz, to extract a secret key from an AES. They used two platforms for their research: one for a significant search of space using an optical beam-induced current (OBIC) to locate target locations, such as flip-flops, and the other for inducing laser faults using a customized Laser-Fault-Injection Microscope from Opto GmbH. The Device Under Test (DUT) was positioned beneath the laser setup, and after pinpointing the location of the flip-flops, the laser was pumped into these targets. They were able to successfully attack the AES hardware implementation of the target device and extracted the secret key.

In conclusion, this part highlights the research done on EMFI and Laser Fault Injection. These studies have offered significant information on the techniques, weaknesses, and probable effects that may be acquired by fault injection. The

majority of the study has focused on fault injection on FPGA, Raspberry Pi, and other commercially accessible microcontrollers, where they rely on either an EMP generator or equipment capable of producing EMPs to inject faults. The study conducted by [2, 4, 12] aided in situating the injection probe and selecting the range of pulse widths to experiment with. The author of [4] assists in comprehending the basic problems that an EMFI can create in an embedded program. Colin O'Flynn [7] uses the ChipShouter to access data from a Trezor Bitcoin wallet, which aided in understanding the ChipShouter's possible functionality. The study done in [10] discusses the injection of faults into a target module's flash memory, which motivated us to perform the same.

4 Proposed System

The security and reliability of IoT devices needing protection might suffer significantly from ASCON's susceptibility to Electromagnetic Fault Injection (EMFI). Previous attempts [3, 14, 17] used the AES algorithm on FPGAs and microcontrollers, thus the researchers had to precisely point the EM probe onto the target's memory ICs. This strategy is challenging because it necessitates pinpointing the configuration memory, power supply lines, and input/output pins, which are all connected to the microcontroller. This microcontroller might be damaged by introducing faults, which would produce incorrect results. The goal of this part is to provide a thorough understanding of how the experiment was carried out and how the resulting data was analyzed. The proposed system describes the technique used in our research, including test scenarios, data collecting, and experiment analysis which are mentioned below:

1. **Experimental Setup:** The research was carried out using an M5STACK ESP32 Timer Camera module, which has an 8 MB PSRAM and 4 MB flash memory. The encryption and decryption keys, together with data like plaintext, related data, key, nonce, and other variables, were all encoded into the ASCON algorithm to be stored in the PSRAM. To inject EMPs, the ChipShouter was used, and it was placed close to the target module's PSRAM. For parameter control and analysis, a computer was connected to the ChipShouter and the target module.
2. **Test Scenario:** EMPs with varying pulse widths, pulse dead time, number of pulses, and pulse length are injected into the target module's PSRAM. The two sorts of faults that were meant to be created were adding bits and bit flips. The experiment uses four separate probes, each of which is available with the ChipShouter. The probes are classified as clockwise and counterclockwise, with probe diameters of 4mm and 1mm.
3. **Integrating with existing methods:** Placing the EM probe was considered challenging, as it depends on the accuracy of previous research outcomes that employed comparable techniques on various devices like FPGA and Raspberry PI [2, 12]. Unlike prior attempts, our study employs a new target device, the M5STACK ESP32, which allows us to exclusively insert faults into a PSRAM,

a single memory component, without compromising the microcontroller. As a result, this method produces far more exact findings than earlier attempts.

4. **Data Collection:** The suggested method gathers data by displaying the output on the serial monitor of an Arduino IDE installed in the experiment computer. The output consists of encryption and decryption time and prints faulty bits when a fault occurs.
5. **Data Analysis:** The collected data is evaluated by noting all of the errors that occur throughout the encryption and decryption processes. Each error is examined separately to see if bits were added or flipped.

4.1 Termite by Compuphase

The Termite RS232 terminal is simple to set up and use. It offers a user interface similar to those of “messenger” or “chat” apps, with a huge window displaying all incoming data and an edit line for appending text to the broadcast. This program is installed on a PC to modify the ChipShouter’s parameters, which include voltage, pulse width, pulse dead time, pulse repetition, pulse pattern, and so on.

4.2 M5STACK ESP32 Timer Camera Module

The ESP32-based Timer Camera includes 4 MB flash memory and 8 MB of inbuilt PSRAM, an operating frequency of up to 240 MHz, and a camera module capable of recording photos at a maximum resolution of 2048×1536 pixels, with a 3 million (3MP) pixel sensor (ov3660) and DFOV 66.5° . A reset button and a status indication LED are included on the board. Because it provides time, sleep, and wake-up functions, the RTC (BM8563) consumes very little power. We chose this device because of the ASCON library’s memory requirement of at least 2 MB. The module can be programmed in Arduino C by using an Arduino IDE. This module has more flash memory than ordinary market modules. The ASCON is written in Arduino C, and all variables such as plaintext, associated data, nonce, key, and so on are allocated in this module’s PSRAM, where EMPs are introduced to cause faulty bits.

4.3 Design

We analyze the research done on EMFI [4, 10, 12] and came up with a different methodology from them which is mentioned in the previous section. The study’s overall concept entailed using an experimental setup to cause electromagnetic faults in the PSRAM of an M5STACK ESP32 module as shown in Fig. 4. In order to conduct the study, data on the kinds and locations of faults caused by the ChipShouter were gathered and examined. Additionally, consideration was given to the study’s limitations. The design of the study is given in the following:

1. **Experiment Setup:** The study made use of an M5STACK ESP32 Timer Camera module with a 4 MB flash memory and an 8 MB PSRAM. The encryption and decryption keys and data were stored in the PSRAM as part of the

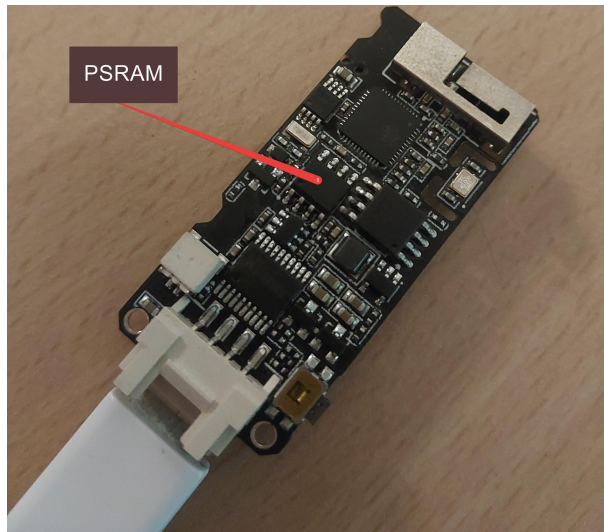


Fig. 4. M5STACK ESP32

device's ASCON algorithm written. The EMPs are injected using a ChipShouter placed close to the module's PSRAM. Both the Chipshouter and the target module are connected to the PC.

2. **EMFI Parameters:** The faults were injected into the PSRAM of the module using a ChipShouter connected to a PC by adjusting settings such as pulse width, pulse dead time, number of pulses, and pulse duration. The position and orientation of the ChipShouter probe were also taken into account.
3. **Data Collection:** The suggested method gathers data by displaying the output on the serial monitor of an Arduino IDE installed in the experiment computer. The output consists of encryption and decryption time and prints faulty bits when a fault occurs.
4. **Data Analysis:** The collected data is evaluated by noting all of the errors that occur throughout the encryption and decryption processes. Each error is examined separately to see if bits were added or flipped.
5. **Limitations:** The investigation is constrained by the size of the experimental set-up and the difficulty of the employed encryption and decryption algorithms. Because most countermeasures in this field of work are standard, the research did not examine potential defenses against EMFI.

The block diagram in Fig. 6 depicts the implementation of the ASCON algorithm on the M5STACK ESP32 module using the Arduino C language. The code starts by initializing setup procedures, which principally include a PSRAM initialization function to test the module's PSRAM functioning. If the initialization fails, an error message indicating "PSRAM initialization failed" is printed. In the event of a successful startup, the allocation of plaintext values, related data, encrypted text, and decrypted text is performed in the PSRAM rather than the

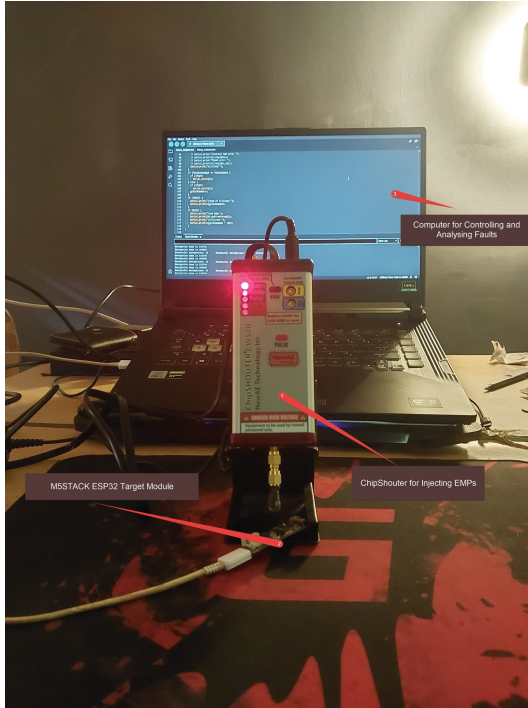


Fig. 5. Experimental Setup

module’s main memory. If the allocation fails, the message “Memory allocation failed” is displayed (see Fig. 5).

Following these preliminary steps, the algorithm starts a loop in which data is placed into the plaintext variable based on the string size defined in the setup function. Letters from ‘A’-‘Z’ are added repeatedly until the necessary string size is reached. Following that, the encryption and decryption operations are carried out, and the timings for each operation are printed.

A condition is introduced after printing the decrypted text to see if it matches the original plaintext. If they are not equal, it signals a problem. In such circumstances, the program is designed to output the glitched values and their related index numbers, allowing the location at which the glitch occurred.

5 Evaluation

This section covers the results and evaluations of the EMFI attack on the ASCON cipher experimentation. The ChipShouter is put to use to target the M5STACK ESP32 module and inject EMPs into its PSRAM. Faults are introduced into the cipher, resulting in outputs that support our contention that the

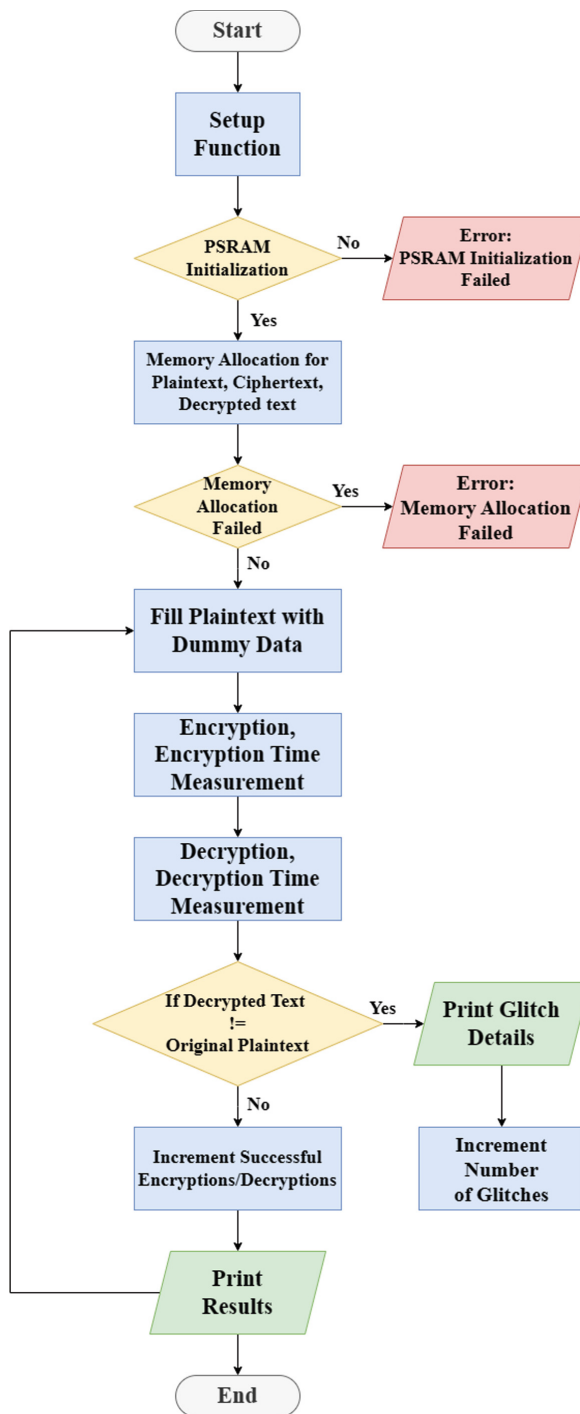


Fig. 6. Block diagram of ASCON algorithm implemented in M5STACK ESP32

ASCON is susceptible to EMFI attacks. The ASCON could only handle a plaintext string size of up to 1 MB; if the amount was increased, the code would not function properly, hence a maximum value of 1 MB was chosen.

We experiment with the insertion of faults using all four injection probes offered by the ChipShouter and uncovered a common fault. The first register of the PSRAM memory, where plaintext information is placed, generates an identical result with all four injection probes. This output sequence implies that the first register of PSRAM memory is impacted in this way (Table 1).

Table 1. First Register of PSRAM Glitched

Glitch Found at Index	Plaintext Around Glitch	Plaintext Binary Values at Glitched Position	Decrypted text Around Glitch	Decrypted text Binary Values at Glitched Position	Found in
0	ABCDEFGHIJK	01000001	Unrecognized Unicode Values	00000000	250 Kb, 500 Kb, 1 MB

The following tables show the outcomes of injecting pulses into the PSRAM using various string values as code input. We perform fault injection on strings of 250 Kb, 500 Kb, and 1 Mb, The tables identify where glitches happen in the plaintext, and the algorithm offers both the places and the accompanying binary values to determine whether bits were flipped or added. The experiment focuses on changing the ChipShouter’s pulse width and pulse repeat settings. The pulse width ranges from 80ns-540ns for all injection tips, while the pulse repeat reflects the number of pulses per trigger, which is mainly set between 3 and 5. The ChipShouter supports pulse widths of up to 960ns and pulse repeat values of up to 1000, we avoid using them to avoid potential harm to the target module. We could not use string sizes less than 250 Kb and greater than 1 MB because injection of faults on values less than 250 Kb is difficult since the memory size of PSRAM is 8 MB and string sizes above 1 MB are not compilable because the algorithm breaks.

Table 2 shows faults found for a 250 Kb string. Certain values, however, are missing because they exist in Unicode format, where they show unknown symbols. The faults found are completely random, involving flipped, added, or swapped bits. These faults are most noticeable after the encryption process. The glitches in Table 3 involve bits being changed into lowercase characters, particularly special characters being added. In Table 4, certain plaintext values are impacted by glitches prior to the encryption process, thus affecting the decrypted values.

The graphical data shows how long the ASCON algorithm takes to encrypt and decrypt plaintext strings and how many faults arise. The horizontal axis shows the number of algorithm cycles, whereas the vertical axis shows encryption and decryption durations in microseconds because seconds would not effectively reflect the fluctuations. Figures 7, 8 and 9 shows the encryption and decryption

times for a 250 Kb, 500 Kb, and 1 Mb plaintext string, as well as the faults that occurred during the operation.

We can see in the graphs that number of faults increases with the increase in plaintext size. This also affects the encryption and decryption time. A plaintext string size of 1 MB takes a longer duration than that of 250 Kb and 500 Kb respectively as the longer the plaintext size the easier to inject faults and the lesser the size the more difficult to inject faults.

The fault injection percentages are calculated by counting the number of faults that occur in plaintext string sizes of 250 Kb, 500 Kb, and 1 Mb. The obtained count is then divided by 50, representing the total number of cycles taken for each observation of string size. Finally, the result is multiplied by 100 to obtain the fault injection percentages. The equation for the calculation of fault percentage is given below:

$$\frac{\text{Number of faults}}{\text{Total number of cycles}} \times 100$$

The calculated fault injection percentages according to the equation are as follows: 24% for the 250 Kb string size, 54% for the 500 Kb size, and 90% for the 1 MB size.

Table 2. For 250 Kb String Value

Glitch Found at Index	Plaintext Around Glitch	Plaintext Binary Values at Glitched Position	Decrypted text Around Glitch	Decrypted text Binary Values at Glitched Position
132632	WXYZABCDEF GHIJKLMNOPQ	10000111	WXYZABCDEF DtKLMNOPQ	01000100
216092	WXYZABCDEF GHIJKLMNOPQ	01000111	WXYZABCD EF_x001D_	00000000
194704	GHIJKLMNOPQ RSTUVWXYZA	01010001	GHIJK LMNOP	00000000
59836	ABCDEF GHIJ	00000000	ABCDEFGH IJKLMNOPQRSTU	01001011
63170	GHIJKLMNOPQ RSTUVWXYZA	01010001	GHIJKLMNOPU UUUUUUUUUU	01010101
120872	OPQRSTUVWXYZ YZABCDEFGHI	01011001	OPQRSTUVWXYZ X\$4DTdt	10010101

During the initial stage of the experiment, where EMPs were injected into the module’s PSRAM, the device frequently became unresponsive and occasionally resulted in memory corruption, and the device obtained responsiveness only after many resets. Memory corruption happened as a result of the injected EMPs influencing the target device’s core and, as a result, affecting the algorithm’s execution.

Table 3. For 500 Kb String Value

Glitch Found at Index	Plaintext Around Glitch	Plaintext Binary Values at Glitched Position	Decrypted text Around Glitch	Decrypted text Binary Values at Glitched Position
80571	NOPQRSTUVWXYZ XYZABCDEFGH	1011000	NOPQRSTUVWXYZe YZABCDEFGH	1100101
208414	OPQRSTUVWXYZ XYZABCDEFGHI	01011001	OPQRSTUVWXYZ \$4DTdt	10000101
477393	XYZABCDEFG GHIJKLMNOPQR	01001000	XYZABCDEFGD IJKLMNOPQR	01000100
227959	HIJKLMNOPQ %5EUWXYZAB	00000101	HIJKLMNOPQR STUVWXYZAB	01010010
154981	LMNOPQRSTU VWXYZABCDEFG	01010110	LMNOPQRSTU FWXYZABCDEFG	01000110
204184	WXYZABCDE FGHIJKLMNOPQ	01000111	WXYZABCDE FHIJKLMNOPQ	01000110

Table 4. For 1 MB String Value

Glitch Found at Index	Plaintext Around Glitch	Plaintext Binary Values at Glitched Position	Decrypted text Around Glitch	Decrypted text Binary Values at Glitched Position
195330	IJKLMNOPQR STUVWXYZABC	1010011	IJKLMNOPQRCT UVWXYZABC	1000011
70726	WXYZABCDE Fdt	1100100	WXYZABCDEF GHIJKLMNOPQ	1000111
928828	UVWXYZABC DEFGHIJKLMNO	1000101	UVWXYZABCD FGHIJKLMNOP	1000110
923933	NOPQRSTUVWXYZ XYZABCDEFGH	1011000	NOPQRSTUVWXYZ xYZABCDEFGH	1111000
36	_x0019_	00011001	_x001D_	00011101

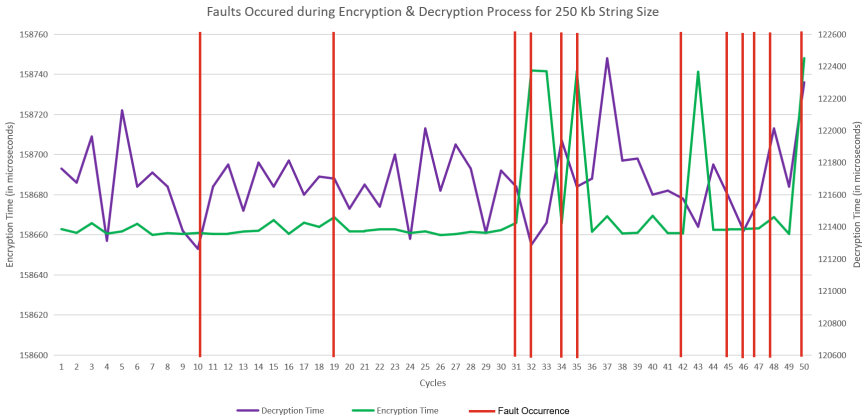


Fig. 7. Fault Occurrence for Encryption & Decryption Processes for a 250 Kb String Value



Fig. 8. Fault Occurrence for Encryption & Decryption Processes for a 500 Kb String Value

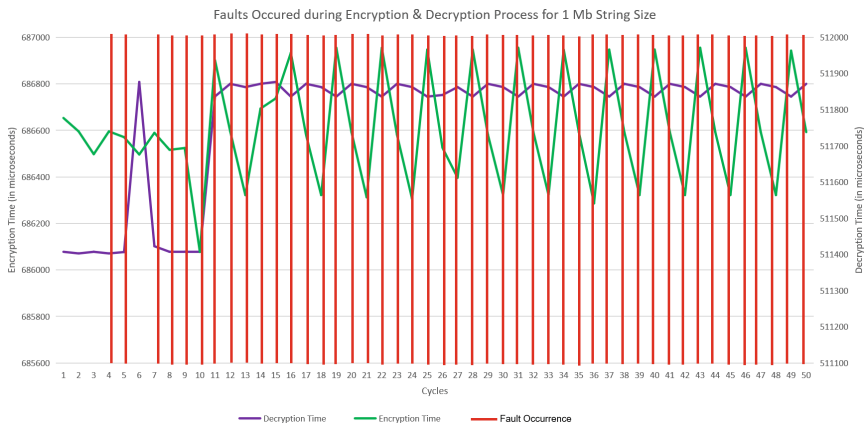


Fig. 9. Fault Occurrence for Encryption & Decryption Processes for a 1 Mb String Value

6 Conclusion

In summary, we conducted an EMFI (Electromagnetic Fault Injection) attack on the ASCON algorithm implemented on an M5STACK ESP32 module using the ChipShouter. This approach proved to be an effective means of assessing the security and robustness of ASCON. EMFI attacks involve introducing faults into a device's operations to exploit vulnerabilities and weaknesses in the system.

By injecting EMPs using the ChipShouter, we successfully introduced faults into the ASCON algorithm and analyzed their nature. This demonstrated that ASCON is indeed susceptible to EMFI attacks. Moreover, the use of ChipShouter provided a controlled and reproducible environment for conducting

fault injection studies. This tool allowed us to precisely control the timing and severity of fault injections, facilitating comprehensive testing of ASCON's resilience against EMFI assaults. Through iterative experimentation and analysis, ASCON's resilience can be continually evaluated.

The nature of the injected faults varied depending on the injection tips provided by the ChipShouter. Some tips introduced new bits, while others modified existing ones. Interestingly, we observed a consistent result in the first register of the memory upon fault injection using all injection tips. To evaluate the impact, we tested different string sizes for input plaintext, namely 250 Kb, 500 Kb, and 1 MB. The results revealed that the fault injection percentages were as follows: 24% for the 250 Kb string size, 54% for the 500 Kb size, and 90% for the 1 MB size.

Further work can expand our research by developing mitigation strategies to defend against such attacks on ASCON. Previous works [8, 9] have explored dynamic key generation for the AES algorithm, employing additional logical and bitwise procedures to generate all keys. This complexity makes it challenging to inject faults into the key, offering potential avenues for enhancing ASCON's resistance to EMFI attacks.

References

1. Ordas, S., Guillaume-Sage, L., Maurine, P.: EM injection: fault model and locality. In: 2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC). IEEE (2015)
2. Dehbaoui, A., et al.: Electromagnetic transient faults injection on a hardware and a software implementations of AES. In: 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography. IEEE (2012)
3. Majéric, F., Bourbao, E., Bossuet, L.: Electromagnetic security tests for SoC. In: 2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE (2016)
4. Moro, N., et al.: Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller. In: 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography. IEEE (2013)
5. Beckers, A., et al.: (Adversarial) electromagnetic disturbance in the industry. IEEE Trans. Comput. **72**, 414–422 (2022)
6. Schellenberg, F., et al.: On the complexity reduction of laser fault injection campaigns using OBIC measurements. In: 2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC). IEEE (2015)
7. O'Flynn, C.: MIN () imum failure: EMFI attacks against USB stacks. In: WOOT@USENIX Security Symposium (2019)
8. Pilla, R., Jain, K.: A new authentication protocol for hardware-based authentication systems in an IoT environment. In: Smys, S., Kamel, K.A., Palanisamy, R. (eds.) *Inventive Computation and Information Technologies. Lecture Notes in Networks and Systems*, vol. 563, pp. 629–640. Springer, Singapore (2023). <https://doi.org/10.1007/978-981-19-7402-1-44>
9. Saran, D.V.G., Jain, K.: An improvised algorithm for a dynamic key generation model. In: Smys, S., Kamel, K.A., Palanisamy, R. (eds.) *Inventive Computation and Information Technologies. Lecture Notes in Networks and Systems*, vol.

- 563, pp. 607–627. Springer, Singapore (2023). https://doi.org/10.1007/978-981-19-7402-1_43
10. Beckers, A., et al.: Characterization of EM faults on atmega328p. In: 2019 Joint International Symposium on Electromagnetic Compatibility, Sapporo and Asia-Pacific International Symposium on Electromagnetic Compatibility (EMC Sapporo/APEMC). IEEE (2019)
 11. Proy, J., et al.: Studying EM pulse effects on superscalar microarchitectures at ISA level. arXiv preprint: [arXiv:1903.02623](https://arxiv.org/abs/1903.02623) (2019)
 12. Troughkine, T., et al.: Electromagnetic fault injection against a system-on-chip, toward new micro-architectural fault models. arXiv preprint: [arXiv:1910.11566](https://arxiv.org/abs/1910.11566) (2019)
 13. Surya, G., Maistri, P., Sankaran, S.: Local clock glitching fault injection with application to the ASCON cipher. In: 2020 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS). IEEE (2020)
 14. Gravelier, J., Dutertre, J.-M., Teglia, Y., Moundi, P.L., Olivier, F.: Remote side-channel attacks on heterogeneous SoC. In: Belaïd, S., Güneysu, T. (eds.) CARDIS 2019. LNCS, vol. 11833, pp. 109–125. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-42068-0_7
 15. Dobraunig, C., et al.: ASCON v1. 2. Submiss. CAESAR Competit. **5**(6), 7 (2016)
 16. Skorobogatov, S.P., Anderson, R.J.: Optical fault induction attacks. In: Kaliski, B.S., Koç, K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 2–12. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36400-5_2
 17. Ordas, S., Guillaume-Sage, L., Maurine, P.: Electromagnetic fault injection: the curse of flip-flops. *J. Cryptogr. Eng.* **7**, 183–197 (2017)

Edge AI for Smart Wearables (EAW)



A Smart Health Application for Real-Time Cardiac Disease Detection and Diagnosis Using Machine Learning on ECG Data

Ucchwas Talukder Utsha^(✉), I Hua Tsai, and Bashir I. Morshed

Department of Computer Science, Texas Tech University, Lubbock, USA
{uutsha, i-hua.tsai, bmorshed}@ttu.edu

Abstract. Cardiac disease, also referred to as cardiovascular disease, is a collection of conditions that affect the heart and blood vessels. Medical professionals typically use a combination of medical history, physical examination, and various diagnostic tests, such as electrocardiograms (ECG/EKG), echocardiograms, and stress tests, to diagnose cardiac diseases. In response to this issue, we are introducing a mobile application that continuously monitors electrocardiogram signals and displays both average and instantaneous heart rates. The aim of this project is to detect and diagnose cardiac diseases so that patients can become informed about their heart health and take appropriate actions based on the results obtained. To identify diseases from real-time ECG data, we used machine learning (ML) classifiers and compared them with offline data to validate the classification. The model we used in our application is pre-trained on the MIT-BIH Arrhythmia Database, which contains a wide range of heart conditions. We used Artificial Neural Network (ANN) as a pre-trained model for multiclass detection as it performed the best among ML models, showing an overall accuracy of 94%. The performance of the model is evaluated by testing it on the application using MIT-BIH test Dataset. On the application, the beat-detecting pre-trained model showed an overall accuracy of 91.178%. The results indicate that the Smart-Health application can accurately classify heart diseases, providing an effective tool for early detection and monitoring of cardiac conditions.

Keywords: Cardiac disease · Electrocardiograms · Pre-Trained Model · Smart-Health Application

1 Introduction

Cardiovascular diseases (CVDs) are critical and common heart diseases that can be detected using electrocardiogram (ECG or EKG) signals. The ECG signals are used to diagnose different types of heart diseases such as heart failure, myocardial infarction (MI), premature ventricular contractions (PVCs), etc. Analyzing the bio-electrical signals of each heartbeat, cardiologists can detect abnormalities in the heart, such as irregular heartbeats or abnormal rhythms. However, manual

scrutiny of continuous ECG signals for long durations for each patient is not practical or feasible [1]. Thus, automated detection using machine learning (ML) models is essential for accurate and efficient diagnosis of heart disease.

With the integration of IoT technology into heart disease monitoring, wearable devices and sHealth applications are gaining popularity [2]. These devices and apps use sensors and algorithms to collect and analyze ECG signals in real time, providing patients with immediate feedback and promoting better management of heart disease. In addition to ML algorithms, the integration of IoT technology and sHealth applications has revolutionized the way we approach cardiovascular health. KardiaMobile is a compact, portable electrocardiogram (ECG) gadget that allows people to monitor their heart health and detect potential cardiac problems [3]. It works by recording a single-channel ECG through two electrodes on the device's back. It has been shown to be successful in clinical investigations for identifying atrial fibrillation (AF), with a sensitivity of 96.6% and a specificity of 94.1% [4]. An ECG check, like KardiaMobile, is a portable electrocardiogram (ECG) equipment that employs two or more electrodes to record the electrical activity of the heart [5]. The ECG check app transfers the recorded ECG data to a server for processing. Apple Watch is another device that can monitor the ECG signal and measure heart rate. The ECG feature, which is available on Apple Watch Series 4 and later, enables users to record an electrocardiogram, a test that examines the electrical activity of the heart. The watch can detect aberrant cardiac rhythms like AF and alert the user if one is identified [6]. Previously, our research team developed another smart health framework using body-worn flexible Inkjet-printed (IJP) sensors, commercial wearables such as smart wristbands, a scanner on a printed circuit board, and customized smartphone software [7]. The technology to collect and analyze ECG signals, providing patients with real-time feedback and enabling them to take control of their heart health using wearable devices and smart health apps has the potential to greatly improve the prevention, management, and treatment of heart diseases.

The use of multi-stage classification has shown significant potential in tackling the complexities of adjusting Artificial Intelligence (AI) models to novel sensor data or in the evolution of decision-making methodologies in smart systems [8]. Segmenting the AI model into various stages enhances its scalability and upgradability, offering a more flexible alternative compared to single-stage classification that could potentially struggle in adapting to changes. Multi-stage classification allows for independent modifications, enabling a more flexible and adaptable approach. This approach has shown great potential in the detection of cardiac diseases, demonstrating its scalability and upgradability for smart health systems. With the integration of AI into cardiac disease detection, multi-stage classification has emerged as a valuable tool in improving patient outcomes and promoting heart health. Its benefits include increased accuracy, efficiency, and the ability to adapt to changing data and circumstances. As such, multi-stage classification has the potential to revolutionize the field of smart health

and transform the way we approach cardiac disease detection and management [9–15].

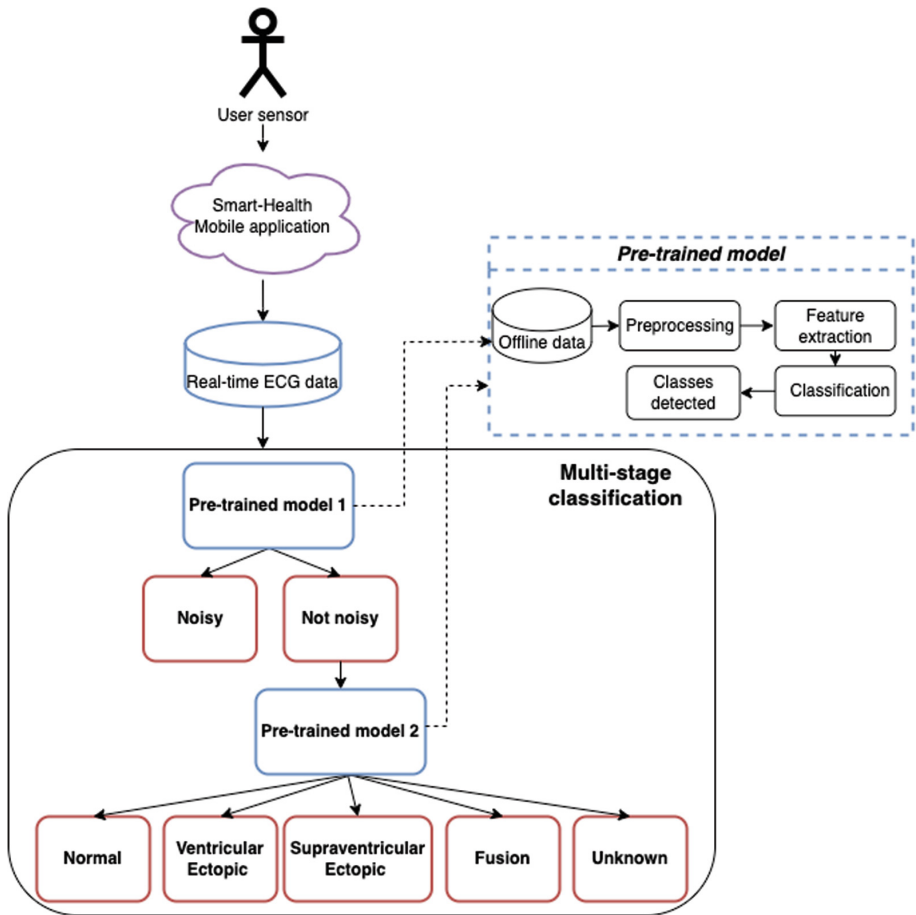


Fig. 1. The flowchart of the ECG signal processing, analysis, and classification.

A multistage algorithm for automatic ECG data classification combines different procedures for dimensionality reduction, consensus clustering, and fast supervised classification algorithms [9]. Two multilayer perceptron (MLP) and one self-organizing map (SOM) networks perform better than using raw data or individual features for classifying six common ECG waveforms with an average recognition rate of 0.883 within a short training and testing time [10]. A multistage deep learning classification model for automatic arrhythmia classification using ECG waveforms and Second Order Difference Plot (SODP) features in discriminating five types of heartbeats from the MIT-BIH Arrhythmia Database [11, 12].

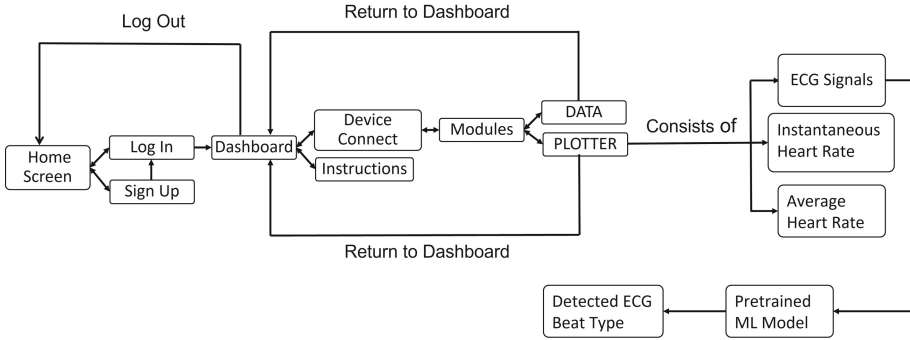


Fig. 2. Block Diagram of Smart-Health Application.

A multistage pruning technique to reduce the computational complexity of Convolutional neural network (CNN) models used in ECG classification for real-time detection of arrhythmias. [13]. The method presented entails a multi-tiered process to ensure precise classification of arrhythmia, leveraging 12-lead surface ECGs. This comprehensive procedure includes three distinct phases of noise reduction, an innovative feature extraction approach, and a finely optimized classification model. [14]. This technique employs features like dynamic amplitude range and autocorrelation maximum peak to identify and categorize various types of noise. [15].

This work presents a wearable ECG monitoring system that is cost-effective and capable of real-time monitoring through a smartphone application. The application provides real-time visualization of the ECG trace and heart rate detection, allowing for monitoring, assessment, and diagnosis. Also, we used Artificial Neural Network (ANN) as a pre-trained model for the application of disease detection. The main aspects of the proposed application include:

- It allows users to connect to an embedded system that collects ECG signals from wrists by electrodes via Bluetooth Low Energy (BLE) and monitor ECG signals on the screen of the application.
- The users can track their heart rate and rhythm over time, and share ECG recordings with healthcare providers for remote monitoring and diagnosis.
- The application can detect the Normal and Noisy signals from the ECG data.
- The application allows users to detect diseases from the incoming ECG signals.

2 Methodology

In this study, we utilized the MIT-BIH database, which follows the AAMI criteria to classify heartbeat types [16]. The MIT-BIH database consists of five categories of heartbeats, each containing multiple types of beats. Class N

includes normal heartbeats, class SV includes Supraventricular Ectopic heartbeats, class V includes Ventricular Ectopic beats, class F includes Fusion beats and class Q includes Unknown beats. To achieve our goal, we accessed the PhysioNet database, an open-source public data resource, and selected the MIT-BIH arrhythmia database (mitdb) [17]. We divided the records into training and testing datasets. The offline data is then used in a Machine Learning model a pre-trained model for training before using real-time data on the Smart-Health application. The ECG signal processing involves a machine learning algorithm for preprocessing, analysis, and classification. The flowchart of the ECG signal processing is shown in Fig. 1. Then, we passed the MIT-BIH test dataset to the application to validate the pre-trained model on the application.

2.1 ECG Data

ECG (electrocardiogram) data is a sort of medical data that captures the electrical activity of the heart. It is obtained by applying electrodes to the skin of the chest, arms, and legs and connecting them to a device that records and amplifies the electrical signals produced by the heart.

Signal processing and machine learning techniques are frequently used to analyze ECG data in order to extract diagnostic data and enhance clinical decision-making. Arrhythmia, myocardial infarction, and heart failure are just a few of the disorders that are frequently diagnosed and monitored using ECG data. Additionally, properties including heart rate variability, QT intervals, and P-wave morphology can be extracted from it. We collected ECG data using an AD8232 chip (Analog Devices, Wilmington, MA) implemented on our custom ECG data collection device [18]. Electrodes are attached to the wrists of the users and the other part of the electrodes are connected to Sparkfun nRF 52840 mini that is paired with the application via Bluetooth Low Energy (BLE) V5.3.

2.2 Application

The Smart-Health application is a mobile application that allows users to manage their health data. Figure 2 shows the block diagram of the Smart-Health Application.

Signal Preprocessing: Signal preprocessing is an important step in analyzing ECG data and removing any noise or artifacts that may interfere with the accurate signal analysis. In the Smart-Health application, the ECG signals collected from the embedded system are preprocessed to remove noise. The frequency range of interest for ECG signals is between 0.5 Hz and 150 Hz [19]. The lower cutoff frequency of 0.5 Hz is chosen to remove any DC offset or drift in the signal, while the upper cutoff frequency of 150 Hz is chosen to remove any high-frequency noise or artifacts in the signal. The ECG signal is filtered on the application using a bandpass filter to remove any noise outside the frequency range of interest.

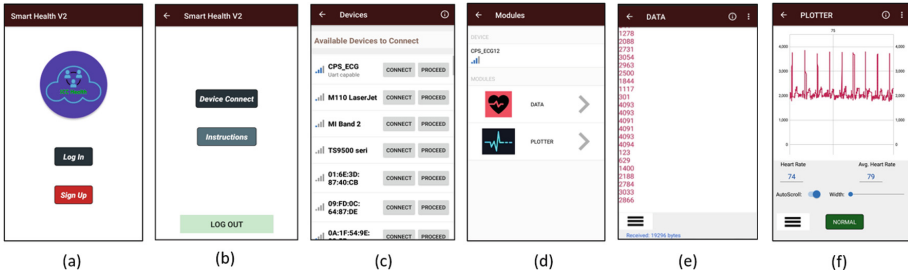


Fig. 3. Snapshots from the application (a) Home (b) Dashboard (c) Available Devices (d) Modules (e) Incoming Data (f) Real-time plot of ECG trace, Heart Rate and Signal type.

Peak Detection and Heart Rate Calculation: We used the Pan-Tompkins algorithm for peak detection of ECG signals which is a widely used method. First, we differentiated the filtered signal to emphasize the QRS complex’s high-frequency components. To accentuate the QRS complex and reduce the T and P waves, we squared the differentiated signal. Then, we passed the squared signal to a moving window to produce a smooth envelope where a threshold is applied to detect the R-peaks. After that, we determined the heart rates from R-Peaks. In addition, we displayed the Average Heart Rate on the application to provide a more complete picture of the user’s heart status. For that, we used the *Sliding Window* approach. We studied 30 heart rate measurements at the same time using a window size of 30. The same statistic was then computed for the next 30 data after adjusting the window by one heart rate value. As a result, users may simply monitor their heart rate and detect any abrupt changes.

Pre-trained Model on the Application: We used pre-trained machine learning models in the Smart-Health application to diagnose medical conditions in real-time. We trained the models on the MIT-BIH dataset and then integrated them into the application.

To use a pre-trained machine learning model in our Android Studio Java application, we followed these steps :

1. **Train and Save the model:** First, we trained the model on a suitable dataset and saved it in a format that can be loaded by TensorFlow Lite. Here, we saved the model as a *.h5* file using the `Keras model.save()` method.
2. **Convert the model to TensorFlow Lite format:** Next, we converted the machine learning model to TensorFlow Lite format using the TensorFlow Lite converter. This produced a *.tflite* file that we used in our Android application.
3. **Add the model in the android application:** We loaded the *.tflite* file in the *assets* folder of the Android Studio project. Alternatively, we could go to the *File* → *Other* → *Tensorflow Lite Model* and import the *.tflite* file. It will be added in the *ml* folder on the project.

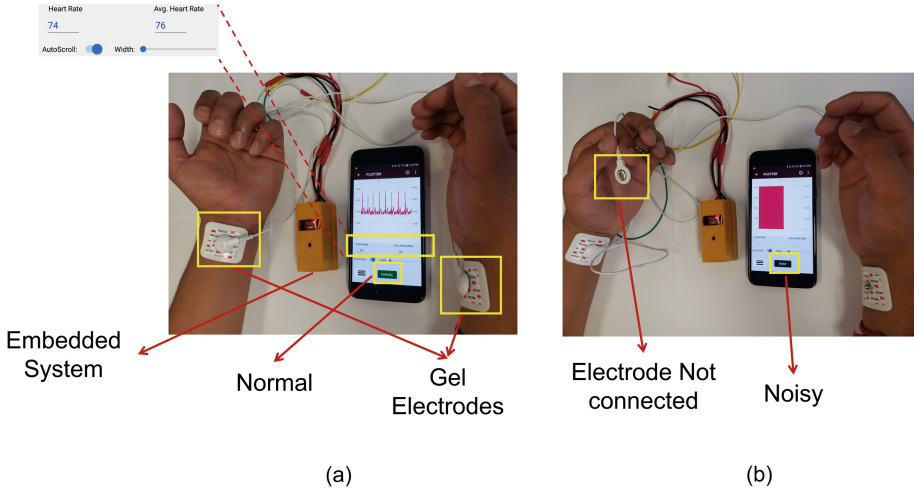


Fig. 4. Real-time data collection (a) Normal Signals (b) Noisy Signals.

4. **Add dependency:** After that, we added the TensorFlow Lite interpreter dependency to the project's build.gradle file: `implementation 'org.tensorflow:tensorflow-lite:2.6.0'`.
5. **Load the model:** Then, we loaded the model from the `assets/ml` folder and created a `ByteBuffer` object to hold the input data.
6. **Get the predicted class:** Finally, we passed the ECG signal data through the interpreter to get the predicted class.

We validated the pre-trained model on the application by passing the offline dataset to it. That means, instead of real-time data, we passed the offline MIT-BIH test dataset to the application to find the exact label. Also, it acted like real-time data on the application. With this validation, we can now go for a clinical trial.

Real-Time ECG Check: Fig. 3 shows some snapshots of the Smart-Health application. Initially, the user needs to register his or her details on the application. Then, s/he should *Log In* on the application. The various devices that can be connected through BLE connection are shown on the smartphone through the *Device Connect* button on the Dashboard screen. Then, the application establishes a BLE connection with the device and is ready to collect data. The user can then navigate to the *Modules* section, where they can view the *Data* or observe the ECG signals on the *Plotter*. Users can track their heart rate in *Plotter* section over an extended period of time in order to spot any potential problems. Additionally, the Smart-Health application provides users with an overview of their health through pre-trained model classification. In the *Plotter* section, there is a *Textbox* at the bottom where users can view their ECG signals' corresponding classification.

A depiction of the real-time data collection procedure is presented in Fig. 4. The experiment involved attaching electrodes to the wrists of users and utilizing an embedded system to collect data. This data was then transmitted to the Smart Health application via Bluetooth connectivity. The application processed the data and provided real-time information on heart rate and various heart rhythm patterns. The objective was to evaluate the application’s ability to accurately display and interpret users’ heart rates and identify different types of heartbeats. From Fig. 4(a), we can see the user’s ECG signal is *Normal* and the average Heart Rate is 76bpm. By detaching the electrodes, artificial noise can be created, which results in a noisy signal (Fig. 4(b)).

2.3 Pre-trained Model

Feature Extraction: In our research, utilizing the Time Series Feature Extraction Library (TSFEL) in Python, we were able to extract a comprehensive set of 175 features from the analyzed beats. To select the most informative features for classification, we applied analysis of variance (ANOVA) algorithms. ANOVA, a statistical technique, is utilized to examine variances in mean values across different groups. This method aids in pinpointing the features that significantly influence the accurate classification of ECG beats.

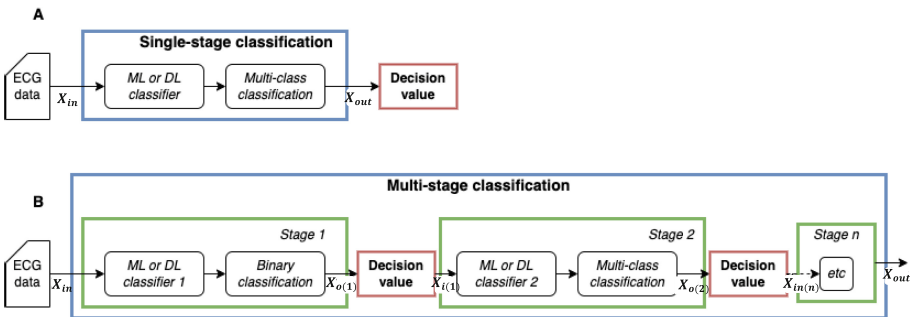


Fig. 5. The structure of single-stage and multi-stage classification is presented.

Classification: We performed heartbeat classification by assigning Normal heartbeats (N) as 0, Supraventricular Ectopic heartbeat (SV) as 1, Ventricular Ectopic beats (V) as 2, Fusion beats (F) as 3, and Unknown beats (Q) as 4. In our pursuit of precise classification, we harnessed a variety of machine learning methodologies, encompassing Decision Tree (DT), Artificial Neural Network (ANN), Support Vector Machine (SVM), Naive Bayes, and K-Nearest Neighbors (KNN), in addition to Bagged Tree. Furthermore, we utilized advanced Deep Learning models such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM). We tested both

single-stage classification and two-stage classification methods to optimize the accuracy of the results. Figure 5 shows the structure of single and multi-stage classification.

Single-Stage Classification: To perform single-stage classification, we first evaluated the effectiveness of various machine learning and deep learning models. We aimed to select the best-performing model by training and testing them with 10-fold cross-validation. This allowed us to assess the model’s ability to generalize to unseen data and avoided overfitting. We also adjusted the parameters of the models during the training process to optimize their performance. By doing so, we can determine the ideal combination of hyperparameters that results in the best performance for each model.

Multi-stage classification: In multi-stage classification, we first performed a binary classification to distinguish normal from abnormal noise. Then, a new classifier is built for the multi-class classification. We execute experimental trials using Decision Trees (DT) and Artificial Neural Networks (ANN), as these have proven to be the top-performing models among machine learning techniques. We also tweak parameters to gauge their performance. We also assessed power consumption, including memory usage, CPU usage, and running time to evaluate the efficiency of the classifiers.

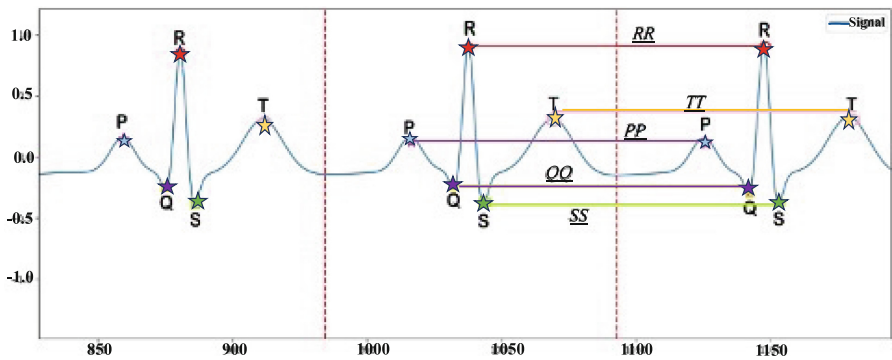


Fig. 6. The P, Q, R, S, and T peaks collectively create a heartbeat.

Performance: To assess the effectiveness of our model, we employed a range of statistical metrics such as accuracy, precision, and recall. Furthermore, we evaluated the power consumption, including memory usage, CPU usage, and running time, of the top-performing machine learning (ML) and deep learning (DL) algorithms. The Keras Model Profiler, a Tensorflow package, was utilized to

gather information on model parameters and memory requirements. To monitor system utilization, including CPU, memory, and network usage, we employed the Psutil package.

3 Results

3.1 Pre-trained Model

Accurately detecting R peaks is crucial in ECG heartbeat recognition. Figure 6 illustrates the P, Q, R, S, and T peaks, as well as a heartbeat. We utilized the Pan-Tompkins algorithm to detect the precise position of the R peak, which in turn affects the accurate positioning of the P, Q, S, T, and T' peaks. The algorithm identifies the R peak by employing a sliding window that spans two heartbeats and advances one beat at a time until completion. The peaks detected by the algorithm are then labeled as P, Q, R, S, and T, as demonstrated in Fig. 6, which showcases the interval from 850 to 1100 at the apex.

Table 1. Evaluate the performance of machine learning (ML) and deep learning (DL) methods on single-stage classification tasks by employing various classifier algorithms.

Single-stage Classification			
Classifier	<i>Accuracy %</i>	<i>Precision %</i>	<i>Recall %</i>
DT(10)	89	90	88
ANN(64)	94	92	93
SVM	78	78	79
Naive Bayes	81	83	82
KNN	75	75	74
Bagged tree	84	85	83
RNN	92	91	91
CNN	94	93	92
LSTM	90	89	87

In the machine learning domain, the Decision Tree (DT) classifier achieved an accuracy of 89%, with precision and recall rates of 90% and 88%, respectively. The Artificial Neural Network (ANN) exhibited an overall accuracy of 94%, accompanied by precision and recall scores of 92% and 93%. In the realm of deep learning, the Convolutional Neural Network (CNN) demonstrated superior performance, obtaining an accuracy of 94% and precision and recall values of 93% and 92%, respectively. Table 1 presents a comparative analysis of the performance of various machine learning (ML) and deep learning (DL) techniques in single-stage classification tasks.

Table 2 provides a summary of power consumption for single-stage classification using ANN and DT algorithms with varying parameters, such as the

number of layers and maximum depth. For the ANN models, we explore performance across a range of layer counts, from 1 to 256, with 64 layers as the standard configuration. Both the 128-layer and 256-layer ANN classifiers achieved 100% accuracy without significant changes in memory, CPU usage, or runtime. Beyond 128 layers, the accuracy and power consumption decreased, while the runtime became faster. For the DT models, we analyzed performance by varying the maximum depth parameter from 1 to 25, with 10 as the standard setting. The classifiers with a maximum depth of 25 and 24 achieved 100% accuracy, again without noticeable changes in memory, CPU usage, or runtime. Beyond a maximum depth of 24, the accuracy and memory usage decreased, while CPU usage and runtime remained consistent.

Table 2. Power consumption for single-stage classification using ANN and DT algorithms with varying parameters.

Single-stage Classification				
Parameter	Accuracy %	Memory usage (MiB)	CPU usage %	Run time(s)
ANN(Layers)				
256	100	431	4	20
128	100	431	4	18
64	94	429	3.5	12
32	82	427	3.5	9
16	67	424	3.0	9
8	52	423	3.0	7
4	49	420	2.0	7
2	33	417	2.0	5
1	23	417	2.0	3
DT(MaxDepth)				
25	100	375	1.0	1
24	100	375	1.0	1
23	99	375	1.0	1
20	98	375	1.0	1
15	94	375	1.0	1
10	89	360	1.0	1
6	80	352	1.0	1
2	79	348	1.0	1
1	75	347	1.0	1

Table 3 shows a summary of power consumption for multi-stage classification using ANN and DT algorithms with varying parameters and arrangements. For

multi-stage ANN&ANN classifiers, we achieved 100% accuracy with 256 to 64 layers without much change in memory usage, but accuracy decreased after 64 layers in the first stage. In multi-stage DT&DT classifiers, we achieved 100% accuracy with 25 to 20 max depths without much change in memory usage in the first stage, and accuracy dropped off after 24 max depths in the second stage. Overall CPU usage and run times remained the same when max depths were reduced.

Table 3. Overview of the power consumption for multi-stage classification using ANN and DT algorithms with different parameters and arrangements.

Multi-stage Classification				
Parameter	Accuracy%	Memory usage (MiB)	CPU usage %	Run time(s)
	<i>First stage; Second stage</i>	<i>First stage; Second stage</i>		
ANN(Layers)				
256	100 ; 100	342 ; 389	17	42
128	100 ; 99	340 ; 387	16.5	35
64	100 ; 94	336 ; 387	7.5	20
32	98 ; 84	329 ; 375	14	15
16	95 ; 74	325 ; 368	13.7	11
8	87 ; 67	321 ; 365	13	11
4	74 ; 53	321 ; 357	12	10
2	68 ; 39	318 ; 357	11.5	7
1	67 ; 27	315 ; 351	16	5
DT(MaxDepth)				
25	100 ; 100	384 ; 375	1.0	1
24	100 ; 100	383 ; 375	1.0	1
23	100 ; 99	383 ; 374	1.0	1
20	100 ; 94	382 ; 374	1.0	1
15	98 ; 88	383 ; 375	1.0	1
10	95 ; 79	382 ; 375	1.0	1
6	91 ; 53	383 ; 374	1.0	1
2	83 ; 42	383 ; 374	1.0	1
1	74 ; 36	382 ; 373	1.0	1

3.2 Application

The experiment involved validating the application’s functionality by simulating real-time data reception from the embedded system. Instead of using actual real-time data, we utilized an offline test dataset that contained labeled cardiac disease data. This allowed us to assess the application’s ability to accurately display the corresponding heart rhythms based on the provided labels.

As we don’t have any offline dataset containing the Noisy signal, we couldn’t validate our pre-trained model 1 which is a binary classifier and classifies ECG

signals into Noisy and Not Noisy signals (Fig. 1). But it can detect the Noisy and Not Noisy signals in our application perfectly. We validated our pre-trained model 2 using the MIT-BIH test dataset. It contains 21892 samples and 18118 of them are Normal beats, 556 beats are Supraventricular Ectopic(SV), 1448 beats are Ventricular Ectopic(V), 160 bears are Fusion(F) and the rest 1610 beats are Unknown.

Table 4. Evaluate the performance of Artificial Neural Network (ANN) algorithm on Smart-Health Application using the offline dataset.

MIT-BIH test Dataset	Number of Samples	Accuracy %
Set 1	5000	93.30
Set 2	5000	92.66
Set 3	5000	93.14
Set 4	5000	92.86
Set 5	1892	83.93
Total	21892	91.178

We have separated the MIT-BIH test dataset into five distinct sets. Each set was passed to the Smart-Health application, which contained a pre-trained model designed to classify heart diseases. We used Artificial Neural Network (ANN) as a pre-trained model 2 because it exhibited an overall accuracy of 94% which showed the best performance among ML models. The pre-trained model classified each set based on the available ECG signals. After classification, the labels obtained from the Smart-Health application were compared with the MIT-BIH offline dataset. The accuracy was calculated for each set by comparing the obtained labels from the Smart-Health application with the ground truth labels from the offline dataset. The average accuracy across all five sets was computed to evaluate the performance of the pre-trained model on the MIT-BIH test dataset.

Table 4 provides the accuracy of five distinct sets over the application. The overall accuracy of the pre-trained ANN model was 91.178%.

The Smart-Health application detected 16955 Normal beats, 486 SV beats, 1247 V beats, 127 F beats and 1371 Unknown beats correctly. Figure 7 demonstrates the beats of the MIT-BIH test dataset and the corresponding accurate beats obtained from the Smart-Health application.

Arrhythmia refers to any abnormality in the rhythm of the heart's electrical activity. SVs, Vs, and fusion beats are all types of arrhythmias that can occur in the heart. Figure 8 shows some snapshots of the Smart-Health application after the detection of Normal, Noisy, and Arrhythmic beats. In some cases, arrhythmias can be serious and lead to heart failure, stroke, or sudden cardiac death.

4 Future Work

It is incredibly challenging to collect data when walking, jogging, or engaging in any other action because of the complex system setup. There are some IoT gadgets in the package that could be improved in the future, at which point we could quickly attach those devices to the body and collect data. We are developing a custom wearable ECG data collection hardware, that can significantly simplify the data collection process. Also, the accuracy of the ANN model on the application can be improved. We will also try other algorithms which had a lower accuracy on offline datasets but can perform well with real-time data.

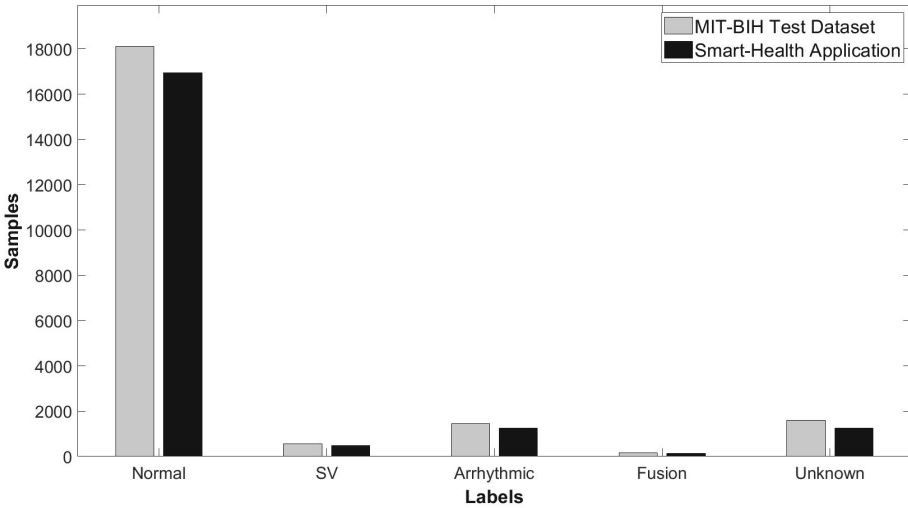


Fig. 7. Comparison of beat classification between the MIT-BIH test dataset and Smart-Health application.

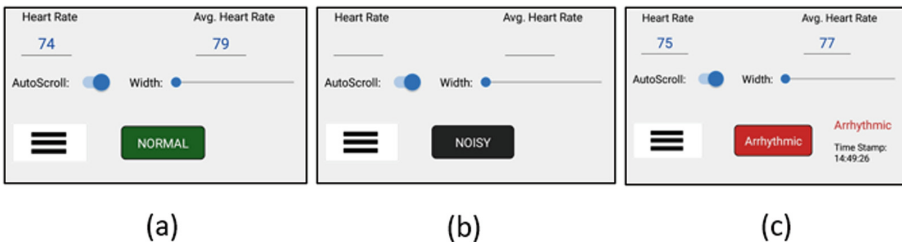


Fig. 8. Signal Detection (a) Normal (b) Noisy (c) Arrhythmic.

At present, the Smart-Health application can detect some types of cardiovascular diseases, including Arrhythmia, and show ECG signals and instantaneous

Heart Rate. As we have passed the MIT-BIH test dataset through the application to detect heart diseases and found quite a good accuracy, our next goals are:

- Plot study of the application with cardiac patients at a cardiac clinic.
- Improve our algorithms and find the best pre-trained model for the Smart-Health application to detect diseases.

5 Conclusion

In this study, we presented Smart-Health, a smartphone application that can continuously monitor ECG data, display Heart Rate and detect cardiac diseases using a pre-trained machine-learning model. The MIT-BIH test dataset was used to evaluate our model, and the findings suggest that our application can accurately detect various heart conditions. Patients can use this application to check their heart health in real time and take appropriate steps depending on the results. Overall, the Smart-Health application has the potential to be a valuable tool for the early detection and monitoring of cardiac problems. The Smart-Health application can aid in the prevention and control of cardiovascular illnesses, resulting in better health outcomes and a higher quality of life for patients by empowering them to actively participate in their own health management.

Acknowledgment. This material is based upon work supported by the National Science Foundation under Grant No. 2105766. The development of the ECG device was performed by Mahfuzur Rahman, Robert Hewitt, and Bashir I. Morshed.



References

1. Glovaci, D., Fan, W., Wong, N.D.: Epidemiology of diabetes mellitus and cardiovascular disease. *Curr. Cardiol. Rep.* **21**, 1–8 (2019)
2. Walker, A., Muhlestein, J.: Smartphone electrocardiogram monitoring: current perspectives. *Adv. Health Care Technol.* **8**(4), 15–24 (2018)
3. Koltowski, L., et al.: Kardia mobile applicability in clinical practice: a comparison of Kardia mobile and standard 12-lead electrocardiogram records in 100 consecutive patients of a tertiary cardiovascular care center. *Cardiol. J.* **28**(4), 543–8 (2021)
4. William, A.D., et al.: Assessing the accuracy of an automated atrial fibrillation detection algorithm using smartphone technology: the iREAD study. *Heart Rhythm* **15**(10), 1561–5 (2018)
5. Haverkamp, H.T., Fosse, S.O., Schuster, P.: Accuracy and usability of single-lead ECG from smartphones—a clinical study. *Indian Pacing Electrophysiol. J.* **19**(4), 145–9 (2019)
6. Raja, J.M., et al.: Apple watch, wearables, and heart rhythm: where do we stand? *Ann. Transl. Med.* **7**(17) (2019)
7. Rahman, M.J., Morshed, B.I., Harmon, B., Rahman, M.: A pilot study towards a smart-health framework to collect and analyze biomarkers with low-cost and flexible wearables. *Smart Health* **1**(23), 100249 (2022)

8. Senator, T.E.: Multi-stage classification. In: Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, TX, p. 8 (2005). <https://doi.org/10.1109/ICDM.2005.102>
9. Abawajy, J.H., Kelarev, A.V., Chowdhury, M.: Multistage approach for clustering and classification of ECG data. *Comput. Methods Programs Biomed.* **112**(3), 720–30 (2013)
10. Hosseini, H.G., Reynolds, K.J., Powers, D.: A multi-stage neural network classifier for ECG events. In: 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 2, pp. 1672–1675. IEEE (2001)
11. Kutlu, Y., Altan, G., Allahverdi, N.: Arrhythmia classification using waveform ECG signals. In: International Conference Advanced Technology & Sciences, Konya, Turkey (2016)
12. Altan, G., Kutlu, Y., Allahverdi, N.: A multistage deep belief networks application on arrhythmia classification. *Int. J. Intell. Syst. Appl. Eng.* **4**(Special Issue-1), 222–228 (2016)
13. Xiaolin, L., Panicker, R.C., Cardiff, B., John, D.: Multistage pruning of CNN based ECG classifiers for edge devices. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 1965–1968. IEEE (2021)
14. Zheng, J., et al.: Optimal multi-stage arrhythmia classification approach. *Sci. Rep.* **10**(1), 2898 (2020)
15. Satija, U., Ramkumar, B., Manikandan, M.S.: A simple method for detection and classification of ECG noises for wearable ECG monitoring devices. In: 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), pp. 164–169. IEEE (2015)
16. ANSI/AAMI. Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms. Association for the Advancement of Medical Instrumentation (1998)
17. Goldberger, A.L., et al.: PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* **101**(23), e215–e220 (2000). [Circulation Electronic Pages; <http://circ.ahajournals.org/content/101/23/e215.full>]
18. Rahman, M., Hewitt, R., Morshed, B.I.: Design and packaging of a custom single-lead electrocardiogram (ECG) sensor embedded with wireless transmission. In: IEEE Dallas Circuits and Systems (DCAS), pp. 14–16 (2023)
19. Golden, D.P., Wolthuis, R.A., Hoffer, G.W.: A spectral analysis of the normal resting electrocardiogram. *IEEE Trans. Biomed. Eng.* **5**, 366–72 (1973)



Reinforcement Learning Based Angle-of-Arrival Detection for Millimeter-Wave Software-Defined Radio Systems

Marc Jean^(✉)  and Murat Yuksel^(✉) 

University of Central Florida, Orlando, FL 32826, USA
marc1988@knights.ucf.edu, Murat.Yuksel@ucf.edu

Abstract. Millimeter-wave (mmWave) signals experience severe environmental path loss. To mitigate the path loss, beam-forming methods are used to realize directional mmWave beams that can travel longer. Yet, advanced algorithms are needed to track these directional beams by detecting angle-of-arrival (AoA) and aligning the transmit and receive antennas. To realize these advanced beam-forming algorithms in real world scenarios, Software-Defined Radio (SDR) platforms that allow both high-level programming capability and mmWave beam-forming are needed. Using a low-cost mmWave SDR platform, we design and prototype two reinforcement learning (RL) algorithms for AoA detection, i.e., Q- and Double Q-learning. We evaluate these algorithms and study the trade-offs involved in their design.

Keywords: Millimeter-Wave · Software-Defined Radio · Testbed · Beam-forming · Angle-of-Arrival · Reinforcement Learning · Q-learning · Double Q-learning

1 Introduction

Next generation 5G networks are being deployed at millimeter-wave (mmWave) bands, beyond 22 GHz [1]. Such high frequencies enable data rates in the order of gigabits per second due to the availability of large unlicensed bandwidth. This is especially beneficial to the future highly dense Internet-of-Things (IoT) networks, which demand large bandwidth. Further, mmWave systems have small form factor and are strong candidates for the emerging intelligent surfaces for IoT devices. Recent studies showed that mmWave antennas can be designed in a flexible and conformal manner [2–4], making them suitable for wearables.

Although mmWave bands allow for high data rate, the short wavelengths are heavily attenuated by the environment [5] mostly due to absorption. Thus, transmitted signals experience severe path loss. To combat the high path loss, mmWave antenna arrays with beam-forming features are being used for generating directional beams which attains longer communication range. However,

the directionality of the mmWave beams brings difficulty in mobile settings as they need constant alignment on both the mobile transmitter and receiver nodes [6]. mmWave channels can be quite complex as line-of-sight (LoS) and non-LoS (NLoS) signals can exist due to emphasized environmental effects. Characterizing mmWave channels, tracking mmWave beams, and Angle-of-arrival (AoA) detection have been challenging [7]. Handling this complexity requires the future mmWave systems to be highly integrated with software-defined radio (SDR) platforms, where advanced algorithmic methods can be practiced.

AoA detection [8] is a critical capability that can facilitate better alignment of the mmWave beams. It is an important directional wireless capability that is used to detect signals transmitted in the environment [9]. A good estimation of the AoA enables fine tuning of the beam alignment between the transmit and receive antennas, which leads to more accurate channel state information (CSI). As a result, the received signal strength (RSS) increases, which leads to a better overall signal-to-noise ratio (SNR) and link performance.

AoA detection has been studied extensively over the years. Numerous algorithms have been used to estimate AoA using synthetic data [9]. Deep learning has been the preferred machine learning (ML) choice for AoA detection, due to robustness to environmental noise. Other methods have been shown to perform poorly in estimating AoA in noisy environments [9]. However, the deep learning methods require extensive training which is not suitable for IoT devices operating in a highly dynamic environment with almost constantly changing channel behavior. More importantly, deep learning methods may require large memory which does not fit well with hardware-constrained IoT devices like wearables.

For mmWave AoA detection, we adapt unsupervised reinforcement learning (RL) algorithms to avoid the abovementioned complexities of deep learning. Our RL-based approach to detecting AoA is compatible with mmWave SDR systems as we show it by implementation. Our approach only considers the receiver side and can passively detect AoA without help from the transmitter or any other localization system. We utilize two RL methods, Q-learning and its variant double Q-learning, for AoA detection. Q-based learning algorithms are widely used for a wide variety of applications that require fast learning capability, such as in gaming, or fast detection capability, such as detecting a drone flying through an indoor environment [10].

Our RL algorithms follow a Markovian model, using actions to explore different states of a given environment [11, 12]. The actions can be based off a greedy policy. With this type of policy, the algorithms can use prior information learned to select the best actions to take [10]. Positive actions lead to positive rewards, while negative actions are punished with negative rewards. After a certain number of iterations, the algorithms learn which actions lead to the best states and converge to a solution. The algorithms use the Bellman equation, discussed later in Sect. 4. Beyond the observed reward for taken actions, the equation relies on several tunable input parameters: the learning rate α , the discount factor γ , and the exploration policy ϵ . It has been shown that tuning the learning rate α and exploration policy ϵ can lead to optimum solutions [12–14]. For our study we

measured the accuracy of detected AoA and convergence time, by tuning both the learning rate α and exploration policy ϵ . Further, the algorithms take a certain number of unknown iterations to converge. To tackle this problem, we use a threshold on the coefficient of variation (CoV) of RSS data samples as the criteria to detect convergence. We compare the performance of our algorithms using Horn antennas controlled by a Python-based SDR setup in connection with GNU radio [15].

Our main contributions are as follows:

- Adaptation of Q-learning and Double Q-learning methods for mmWave AoA detection.
- Tuning the hyper parameters (the learning rate α and exploration policy ϵ) of both Q- and Double Q-learning to detect AoA within 2°'s of accuracy.
- Design of a threshold-based convergence criteria for both Q- and Double Q-learning using CoV of RSS data samples.
- Implementation of a prototype of the algorithms in an affordable mmWave testbed platform using an off-the-shelf SDR platform.

The rest of the paper is organized as follows: Sect. 2 surveys the related literature on AoA detection and experimental mmWave SDR efforts. Section 3 presents our experimental platform and how the AoA detection algorithms are implemented in that platform. Next, Sect. 4 provides a detailed description of our Q-learning and Double Q-learning algorithms for AoA detection. Section 5 details experimental setup and discusses results from our experiments. Finally, Sect. 6 summarizes our work and outlines directions of future work.

2 Related Work

Angle (or direction) of arrival (AoA) detection/estimation has been an extensively studied problem within the context of wireless localization [16]. With the recent advent of directional beam-forming capabilities in super-6 GHz systems, AoA detection, in particular, has gained a renewed interest due to emerging applications using such systems [17].

Experimental demonstration and evaluation of AoA detection in super-6 GHz bands such as mmWave bands has been lacking. The main reason for this has been the limited availability of mmWave experimental testbeds due to the lack and high cost of mmWave hardware [18]. The U.S. National Science Foundation (NSF) is currently funding wireless communication testbed platforms to enable such experimentation. The COSMOS platform [19], for example, includes a 28 GHz phased array antenna, designed by IBM. The front end uses a custom software for steering the antenna beam with respect to azimuth and elevation angles. The AERPAW platform uses drones for 5G experimentation [20], which is the first of its kind. These platforms enable users to perform a variety of wireless communication experiments, such as, AoA detection. However, they are still being adapted by researchers. Unlike these high-end testbeds, we use a cheap

SDR platform and mmWave hardware to evaluate our AoA detection mechanisms. Further, the application programming interface (API) used by these testbed platforms can limit user experimentation. For example, the AERPAW API restricts users from running on the fly experiments. As a result, users aren't able to collect or train radio frequency (RF) data on the fly. This can restrict the types of algorithms users can use on the platform.

Researchers have relied on virtual environments and simulations to perform mmWave experiments. These virtual environments have gotten more sophisticated with the usage of 3D ray tracing. In [21], 3D ray tracing is used to simulate mmWave signals in virtual environments. Users can use the open source software to design large intelligent reflective surfaces and determine AoA using compressive sensing algorithms. Although using simulation-based approaches is cost effective, they do not render the physical world and fall short of precisely modeling complicated physical communication channel dynamics in mmWave or other super-6 GHz bands.

Recently, cheaper off-the-shelf SDRs have been used to setup testbed platforms for AoA detection. The testbed platform [9] uses a Kerberos radio with four whip antennas at the receiving end. At the transmitting end a long range (LoRa) radio is used to transmit a signal at 826 MHz. LoRa is beneficial for long range communication and uses low transmit power. The transmitter includes a GPS and compass unit used to label the direction of the transmitted signal. The labeled data set is the ground truth that is trained in the machine learning (ML) algorithm. The data is trained using a deep learning convolutional neural network (CNN) model [9].

Multiple Signal Classification (MUSIC) is a widely used AoA detection algorithm and assumes that the received signal is orthogonal to the noise signal [9]. MUSIC uses this assumption to decompose the noise from the received signal into separate sub-spaces. The power spectral density (PSD) of the signal is taken as a function of angle [22]. The angular value which results in the maximum power is estimated to be the AoA. The assumption that the received signal is orthogonal to the noise signal is not valid in real world scenarios. Therefore, MUSIC does poorly in environments that involve NLoS propagation. Since mmWave signals can experience severe environmental path loss and involve multiple NLoS signals, MUSIC may not be a good choice for mmWave AoA detection.

Support Vector Regression (SVR) has also been used to estimate AoA. SVR is a supervised ML algorithm. Regression does poorly in estimating AoA from impinging signals at multiple time steps [9]. The algorithm cannot be used to determine AoA since the number of impinging signals is unknown [23]. As a result, the algorithm can be used for detecting AoA for a single source at a time. This makes SVR less robust for AoA detection in environments with multiple signal sources. Therefore, SVR is not a good choice for mmWave AoA detection.

The CNN model used in [9] adapts a hybrid configuration. A classification method is used to determine the number of impinging receive signals and two regressive heads are used to determine the AoA. The study showed that CNN

outperformed the other classical ML methods, MUSIC and SVR. Further, the CNN model was able to estimate AoA within 2° 's of accuracy.

Our approach does not use a deep learning approach or supervised learning. These approaches are not the most suitable for many hardware-constrained IoT devices as the former requires large memory hardware to perform well and the latter requires availability of ground truth. Resources-constrained IoT devices like wearables do not have sufficient memory to keep trained models nor the extensive sensing or coordination capability to obtain the ground truth in AoA. To make it more relevant to IoT devices with high resource constraints, we design RL-based AoA detection methods that do not require the ground truth and determine the AoA based only on the RSS observations at the receiver.

3 mmWave Testbed

To perform a thorough evaluation of our reinforcement learning (RL)-based AoA detection methods, we use our mmWave testbed [24] that allows beam-steering capability from Python.

3.1 Hardware Setup

The architecture of our testbed can be seen in Fig. 1. The testbed uses a Universal Software Radio Peripheral (USRP) model N210. The USRP uses a Superheterodyne architecture for up-converting the transmit signal and down-converting the receive signal [25]. The architecture is built into the USRP's daughter-board to tune the signal within sub-6 GHz [25]. As a result, the USRP is only able to transmit and receive signals at a maximum frequency of 6 GHz.

To handle mmWave frequencies we connect the daughter-board to external RF mixers to further up/down convert the signal. We use Analog Devices up-converter ADMV 1013 [26] and down-converter ADMV 1014 [27]. The cost of each unit is reasonably priced at a few hundred dollars. This makes our mmWave testbed platform more affordable, compared to [18] and [19] that use RF front-ends that cost thousands of dollars. Two signal generators are used as local oscillators for mixing the signal. Two 26 GHz mmWave 15 dB gain horn antennas are used for transmission and reception. The receive horn antenna is mounted to a servo that can rotate from 0 to 180° . A pulse width modulated (PWM) signal is transmitted from the Arduino micro-controller to rotate the servo at a set angular value.

3.2 Software Setup

GNU radio software is used to program the USRP device. GNU radio is a Python-based graphical interface that is open source and readily available online. As seen in 2 [24], the source block is used to generate a cosine signal at a sampling frequency of 2.5 MHz. The samples are streamed in the USRP sink block that sets the frequency of the signal to 2 GHz. The signal is then transmitted from the

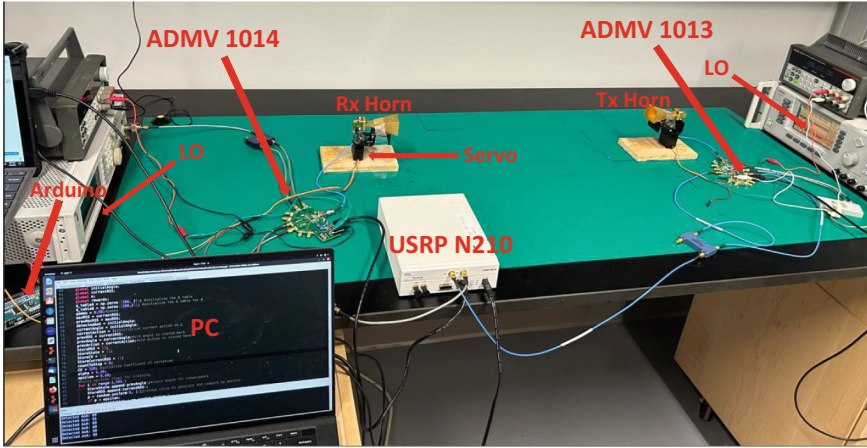


Fig. 1. Testbed Configuration

USRP daughter-board. The receive signal is mixed down to 2 GHz and injected into the daughter-board. The I and Q base-band data samples are streamed from the USRP source block. The samples are used to determine the RSS using the complex to mag-square block. The RSS samples are streamed into a socket using the ZMQ pub sink block. Python socket libraries are used to connect and receive the RSS data from the socket.

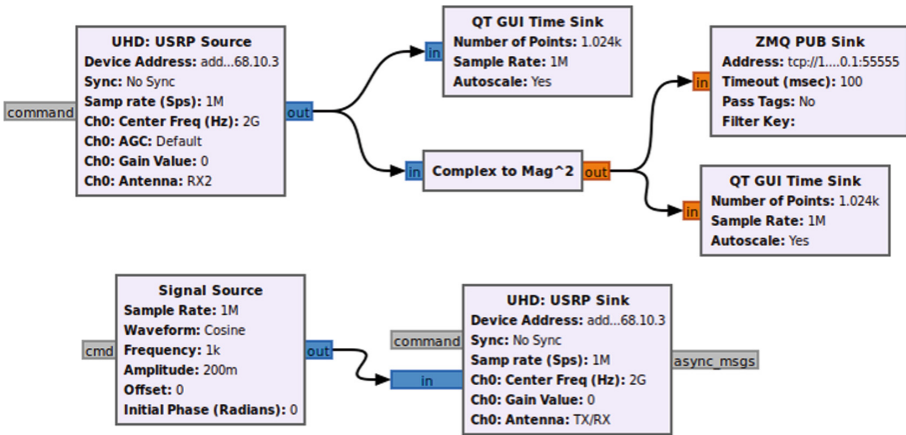


Fig. 2. GNU Radio Software Configuration

4 Q-Learning and Double Q-Learning

Q-learning and its variant Double Q-learning are model-free RL algorithms. Both algorithms are based on a Markovian approach that selects random actions [11]. The block diagram in Fig. 3 presents our RL model. The learning agent is located at the receiver horn antenna that is mounted onto a servo motor that can be steered. The agent can choose to take two possible actions: move left or right by one degree resolution. Since the servo can rotate up to a maximum angular value of 180° , the time-variant state values can be $s_t \in (0, 180)$. For our setup, a positive reward (i.e., when the action improves the RSS) is set as the difference of the current RSS_t and previous RSS_{t-1} , i.e., $RSS_t - RSS_{t-1}$. A negative reward (i.e., when the action reduces the RSS) is set to -5. This design incentivizes the agent to seek the angular position that maximizes the RSS, which is implied when the antenna is steered to the correct AoA.

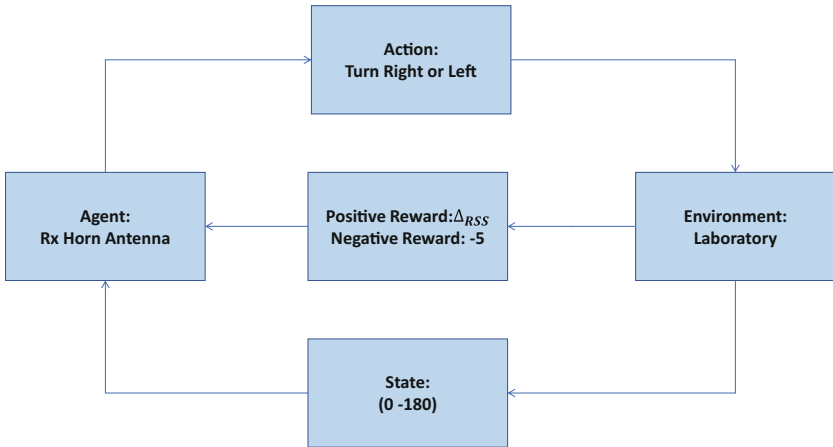


Fig. 3. RL Configuration

Q-learning uses the Bellman equation to populate the Q-table. The Bellman equation is

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha * (r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \quad (1)$$

which provides a mechanism to update the Q values as a function of state, action, and tunable parameters [10]. The variable s_t is our current state and a_t is our current action. When the agent performs an action the algorithm moves to the next state defined as s_{t+1} and a reward r_t is given. For our algorithm, the discount factor γ is set to 0.98 and the learning rate $\alpha \in (0, 1)$ was tuned to 0.1, 0.2, 0.3, and 0.4. Likewise, the greedy policy ϵ was tuned to the same values. Smaller values of ϵ cause the algorithm to exploit the Q table, by searching for

Algorithm 1. AoA Detection with Q-learning

```

1: Input:  $\alpha \in (0, 1)$ , the learning rate;  $\epsilon \in (0, 1)$ , the greediness policy;  $windowSize \in (5, 30)$ , the number RSS samples maintained for implementing convergence criterion;  $threshold \in (0.1, 0.4)$  threshold for convergence criterion.
2: Output: detectedAoA, AoA detected by the algorithm.
3:  $\gamma \leftarrow 0.98$  ▷ Initialize the discount factor
4:  $RSS[windowSize] \leftarrow []$  ▷ Initialize the array of RSS samples
5:  $CoV \leftarrow 1$  ▷ Initialize the coefficient of variation of RSS samples
6: Randomize currentAngle ▷ Initialize to a random angle respectively in [80,100] or [120,140] for the 90° and 130° scenarios
7:  $s_t \leftarrow currentAngle$  ▷ Initialize the current state to currentAngle
8:  $previousRSS \leftarrow$  Measure RSS at currentAngle
9:  $RSS.append(previousRSS)$  ▷ Store RSS sample in array
10:  $sampleCount \leftarrow 1$ 
11:  $Q_{table} \leftarrow 0$  ▷ Initialize the Q-table to 0
12: while  $threshold \leq CoV$  do
13:    $RSS.append(RSS_t)$  ▷ Store RSS data in array
14:   if  $Uniform(0,1) < \epsilon$  then
15:      $a_t \leftarrow Uniform[0,1]$  ▷ Randomly choose to turn left ( $a_t=0$ ) or right ( $a_t=1$ )
16:   else
17:      $a_t \leftarrow \max_a Q(s_t, a)$  ▷ Choose action based on maximum Q value
18:   end if
19:   if  $a_t == 0$  then
20:     Turn antenna beam left by 1 degree
21:      $currentAngle --$ ;
22:   else
23:     Turn antenna beam right by 1 degree
24:      $currentAngle ++$ ;
25:   end if
26:    $newRSS \leftarrow$  Measure RSS at currentAngle
27:    $RSS.append(newRSS)$  ▷ Store the new RSS sample and remove the oldest sample if needed
28:    $sampleCount ++$ ;
29:    $s_{t+1} \leftarrow currentAngle$ 
30:    $\Delta RSS \leftarrow newRSS - previousRSS$  ▷ Calculate the reward
31:   if  $0 < \Delta RSS$  then
32:      $r_t \leftarrow \Delta RSS$  ▷ Positive reward
33:   else
34:      $r_t \leftarrow -5$  ▷ Negative reward
35:   end if
36:    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha * (r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$  ▷ Bellman eqn.
37:    $s_t \leftarrow s_{t+1}$  ▷ Update current state with next state value
38:   if  $windowSize \leq sampleCount$  then
39:      $CoV \leftarrow RSS.std()/RSS.mean()$  ▷ Compute the CoV of RSS samples
40:   end if
41: end while
42: return  $s_t$ 

```

the action that results in the largest Q value. Increasing ϵ increases the likelihood that the algorithm will explore the environment, by selecting random actions.

Our software approach is presented in Algorithm 1. At the start of each run the hyper-parameters γ , α , and ϵ are initialized. The current state s_t is initialized to a random angular value, based off our experimental scenario. For the 90° experimental setup the random angle can be any value in [80,100] degrees and [120,140] degrees for the 130°. The Q_{table} used to store the Q values from Eq. 1 is initialized to zero. An outer while loop comparing the *threshold* value and *CoV* is used to decide the stopping condition for the algorithm until the threshold value is met. If the value of ϵ is greater than a random number in (0,1), the agent takes a random action, either turn the antenna using the servo left or right with one degree resolution. If the value of ϵ is greater than the random value, then the algorithm will exploit the Q_{table} by selecting the action that results in the largest Q value. A positive reward value is given if ΔRSS is greater than zero. If ΔRSS is less than zero, then the action resulted in a decrease in RSS. Therefore, the action yields a negative reward value of -5. The Bellman equation is updated with the reward at every iteration.

Q-learning based algorithms have to train for a certain number of iterations. Based on the number of iterations, the algorithm may or may not converge to the solution. This makes selecting the number of iterations trivial. To address this issue, we take CoV of a certain number of RSS samples (defined by *windowSize*) to detect convergence. CoV is a statistical measure of how dispersed data samples are from the mean of the sample space. It is the ratio of standard deviation and mean of a certain number of samples. In our algorithm, a window size is initialized to $windowSize \in (5, 30)$. The array *RSS* is used to store RSS samples as the algorithm is training on the fly. If the counter *sampleCount* is greater or equal to the *windowSize*, then enough samples have been collected to calculate the CoV. While the CoV is greater or equal to the selected threshold value, the algorithm will continue to train. When CoV is less then or equal to the threshold, the convergence criteria is met and the algorithm breaks out of the while loop. The smaller CoV means that the RSS samples have stabilized and it is safer to stop the algorithm and return the last steering angle as the detected AoA.

Q-learning uses a single estimator $\max_a Q(s_{t+1}, a)$, the maximum next state Q_{t+1} value for all possible actions. As shown in [12], this causes the algorithm to overestimate the desired solution, by causing a positive bias. As a result, standard Q-learning can perform poorly in certain stochastic environments. To improve the performance of standard Q-learning other variants, such as, Double Q-learning [12], Delayed Q-learning [28], Fitted Q-iteration [29], and Phased Q-learning [30] were developed to improve convergence time. To reduce over-estimation, the double Q-learning variant uses two Q functions $Q^A(s_t, a_t)$ and $Q^B(s_t, a_t)$ seen in Eqs. 2 and 3 below. $Q^A(s_t, a_t)$ is able to learn from

$$Q^A(s_t, a_t) = Q^A(s_t, a_t) + \alpha * (r_t + \gamma Q^B(s_{t+1}, \arg \max_a Q^A(s_{t+1}, a)) - Q^A(s_t, a_t)) \quad (2)$$

$$Q^B(s_t, a_t) = Q^B(s_t, a_t) + \alpha * (r_t + \gamma Q^A(s_{t+1}, \arg \max_a Q^B(s_{t+1}, a)) - Q^B(s_t, a_t)) \quad (3)$$

Algorithm 2. Double Q-learning

Same as lines 1-10 in Algorithm 1

```

11:  $Q_{table}^A \leftarrow 0$  ▷ Initialize the first Q-table to 0
12:  $Q_{table}^B \leftarrow 0$  ▷ Initialize the second Q-table to 0
13: while  $threshold \leq CoV$  do
14:    $RSS.append(previousRSS)$  ▷ Store RSS sample in array
15:   if  $Uniform(0,1) < \epsilon$  then
16:      $a_t \leftarrow Uniform[0,1]$  ▷ Randomly choose to turn left ( $a_t=0$ ) or right ( $a_t=1$ )
17:   else
18:      $Q_{table}^{AB} \leftarrow Q_{table}^A(s_t) + Q_{table}^B(s_t)$ 
19:      $a_t \leftarrow \max_a Q_{table}^{AB}$ 
20:   end if
21:   if  $a_t == 0$  then
22:     Turn antenna beam left by 1 degree
23:      $currentAngle --$ ;
24:   else
25:     Turn antenna beam right by 1 degree
26:      $currentAngle ++$ ;
27:   end if
28:    $newRSS \leftarrow$  Measure RSS at  $currentAngle$ 
29:    $RSS.append(newRSS)$  ▷ Store the new RSS sample and remove the oldest
sample if needed
30:    $sampleCount ++$ ;
31:    $s_{t+1} \leftarrow currentAngle$ 
32:    $\Delta RSS \leftarrow newRSS - previousRSS$  ▷ Calculate the reward
33:   if  $0 < \Delta RSS$  then
34:      $r_t \leftarrow \Delta RSS$  ▷ Positive reward
35:   else
36:      $r_t \leftarrow -5$  ▷ Negative reward
37:   end if
38:    $q \leftarrow Uniform(0,1)$ 
39:   if  $q < 0.5$  then
40:      $Q^A(s_t, a_t) = Q^A(s_t, a_t) + \alpha * (r_t + \gamma Q^B(s_{t+1}, \arg \max_a Q^A(s_{t+1}, a)) - Q^A(s_t, a_t))$ 
41:   else
42:      $Q^B(s_t, a_t) = Q^B(s_t, a_t) + \alpha * (r_t + \gamma Q^A(s_{t+1}, \arg \max_a Q^B(s_{t+1}, a)) - Q^B(s_t, a_t))$ 
43:   end if
44:    $s_t \leftarrow s_{t+1}$  ▷ Update current state with next state value
45:   if  $windowSize \leq sampleCount$  then
46:      $CoV \leftarrow RSS.std() / RSS.mean()$  ▷ Compute the CoV of RSS samples
47:   end if
48: end while

```

the experiences of $Q^B(s_t, a_t)$ and vice versa. This approach has been shown in [12] to cause the algorithm to underestimate, rather than overestimate the solution with a positive bias. In conditions where Q-learning performs poorly, double Q-learning has been shown to converge to the optimum solution [12].

Our AoA detection algorithm for Double Q-learning can be seen in Algorithm 2. Like Algorithm 1, the same parameters are initialized at the start of each run. Two Q tables Q^{Atable} and Q^{Btable} are initialized to zero. The same greedy ϵ and reward r_t values are also used in Algorithm 2. The variable q is set equal to a random uniform value $q \in (0, 1)$. If q is larger than the threshold value of 0.5 then $Q^A(s_t, a_t)$. Otherwise, if q is less than the threshold of 0.5, then $Q^B(s_t, a_t)$ is selected. The threshold value is set to 0.5 to give both $Q^A(s_t, a_t)$ and $Q^B(s_t, a_t)$ equal probability of being selected for training. The same convergence criteria is used in Algorithm 1 is used in 2.

5 Experimental Evaluation and Results

To understand the efficacy of our RL-based AoA detection algorithms, we measured the average AoA error and average convergence time for different (ϵ, α) combinations at each *Threshold* value. The *Threshold* value determines how strict the convergence criterion is. Hence, smaller *Threshold* causes the AoA detection algorithms to search the AoA for a longer period of time. This enables them to find a more accurate AoA. Hence, there is a trade-off between the error in AoA estimate and the convergence time of the algorithms. For a fixed *Threshold* value, we need to find the best (ϵ, α) combination by minimizing both the AoA error and convergence time. To do so, we search for the (ϵ, α) combination that minimizes the product of the two metrics, i.e.:

$$\min_{\epsilon, \alpha} \langle \text{AoA Error} \rangle * \langle \text{Convergence Time} \rangle . \quad (4)$$

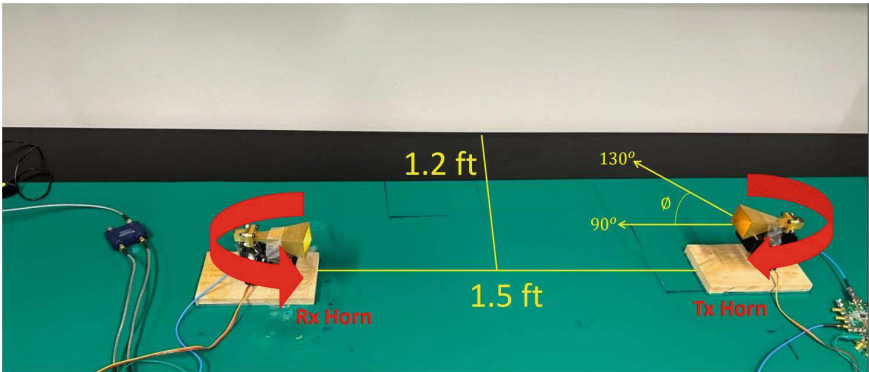


Fig. 4. Experimental Set-Up

We considered two scenarios for comparing the performance of the Algorithms 1 and 2. As seen in Fig. 4, the range between the transmit and receive horn antennas is 1.5 ft. The distance to the wall to the center is 1.2 ft. In the

first experiment scenario, the transmit antenna is fixed to 90° , pointing towards the receive antenna to compose an LoS path to the receiver. The receive horn antenna is initialized to a random angular state value between 80 and 100° . For the second scenario, the transmit horn antenna is rotated 130° to the right, composing a NLoS path to the receiver. The main lobe of the transmitted signal is reflected off the wall. The initial angle is set to a random value between 120 and 140° .

As previously mentioned, the hyper-parameters ϵ and α are tuned to 0.1 , 0.2 , 0.3 , and 0.4 . This was done to measure which combinations of (ϵ, α) resulted in the best performance with respect to both AoA detection and convergence time. The average AoA error and time of convergence was measured for thirteen threshold values from 0.1 to 0.4 in increments of 0.025 . This was done using Q and double Q-learning for both the 90° and 130° experimental scenarios.

5.1 Q-Learning Results

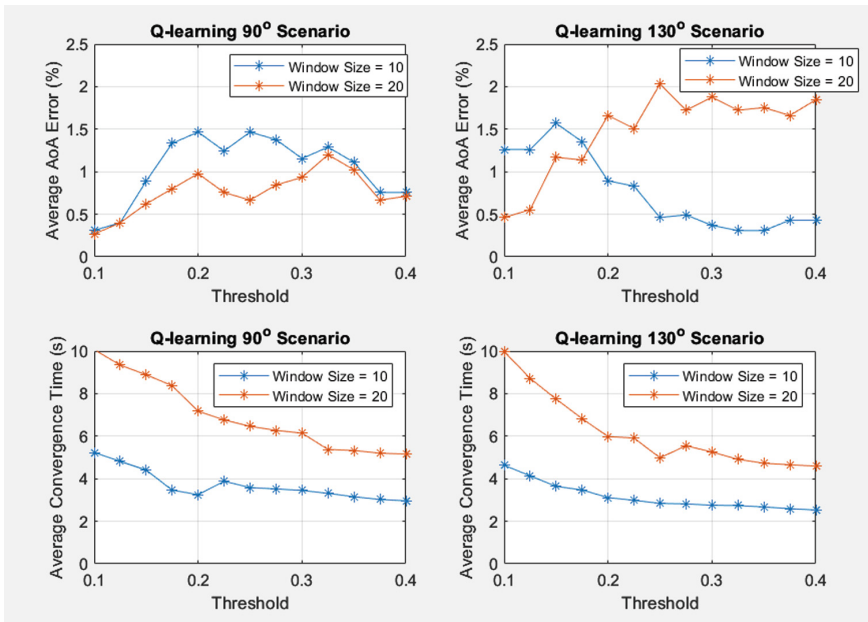


Fig. 5. Q-Learning for 90° and 130° scenarios

Figure 5 are graphs of *Threshold* vs. average AoA error and *Threshold* vs. average convergence time. For each combination of (ϵ, α) the average convergence time is reduced with respect to window size. A window size of 10 results in faster AoA detection compared to the window size of 20.

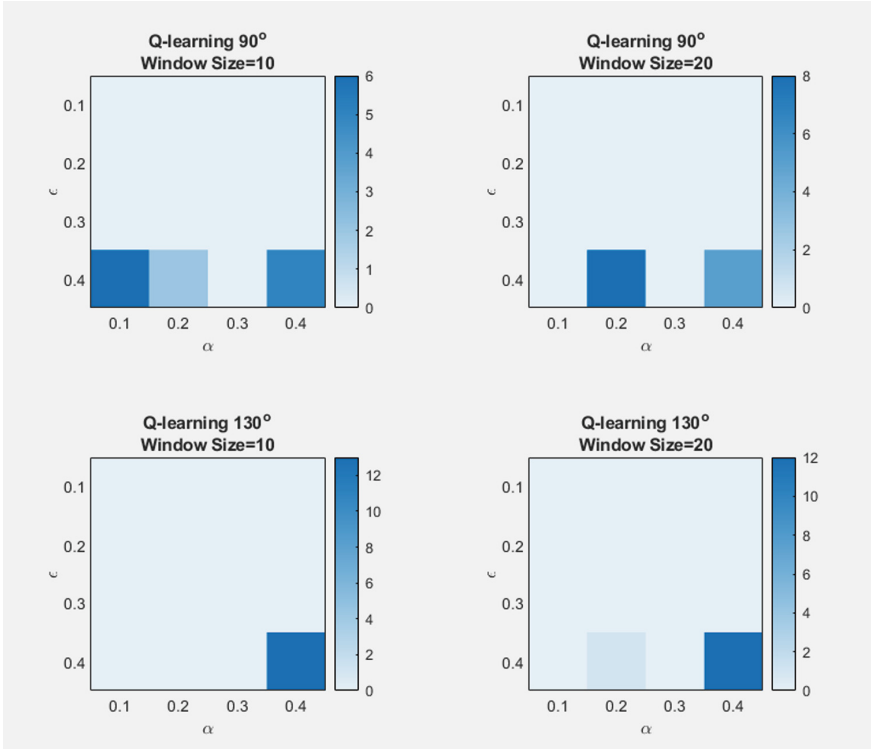


Fig. 6. Q-Learning Heat Map

Both the 90° and 130° scenarios resulted in average AoA error less than 1% for some combination of (ϵ, α) . This occurs for both window sizes of 10 and 20. With less than 1% AoA error, the detected AoA is within 1° of the correct AoA.

To understand which (ϵ, α) combinations give the best results, we look at the heat map of (α, ϵ) occurrences resulting in the minimum product in Eq. 4 as shown in Fig. 6. For the 90° case, ϵ of 0.4 with α of 0.1, 0.2, and 0.4 are the dominate cases. Likewise, ϵ of 0.4 is also the dominant case for the 130° scenario. The α values are also similar at 0.2 and 0.4 being the dominant cases. For Q-learning the combination of (ϵ, α) that are most effective are (0.4,0.1), (0.4,0.2) and (0.4,0.4).

5.2 Double Q-Learning Results

Figure 7 shows the average AoA error and average convergence time against *Threshold* for Double Q-learning. Like Q-learning, it is clear that an increase in window size results in larger convergence time. For Double Q-learning, both the average AoA error and convergence time are better than Q-learning. For nearly all (ϵ, α) combinations, the average AoA error is less than 0.15% for both the 90° and 130° scenarios.

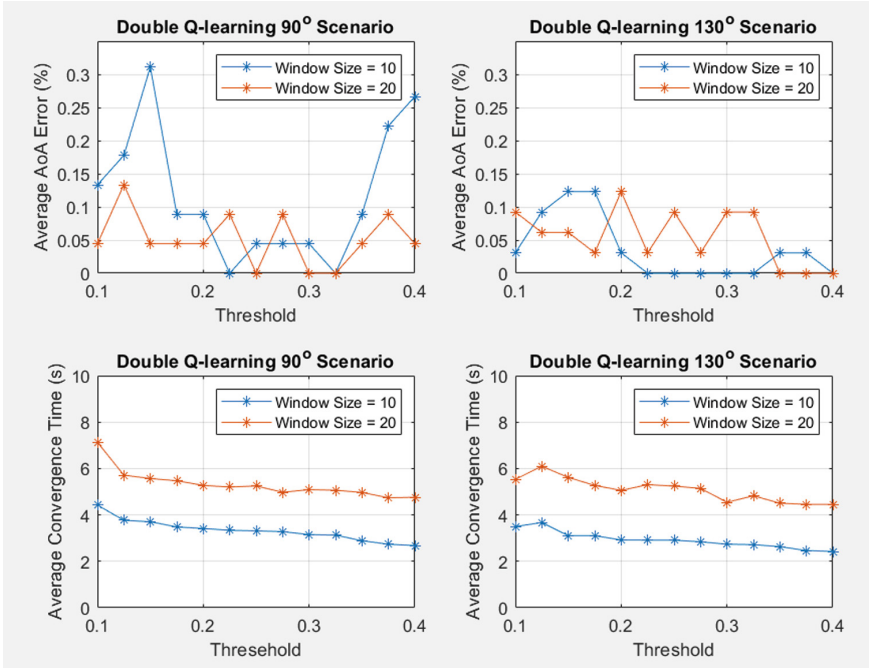


Fig. 7. Double Q-Learning for 90° and 130° scenarios

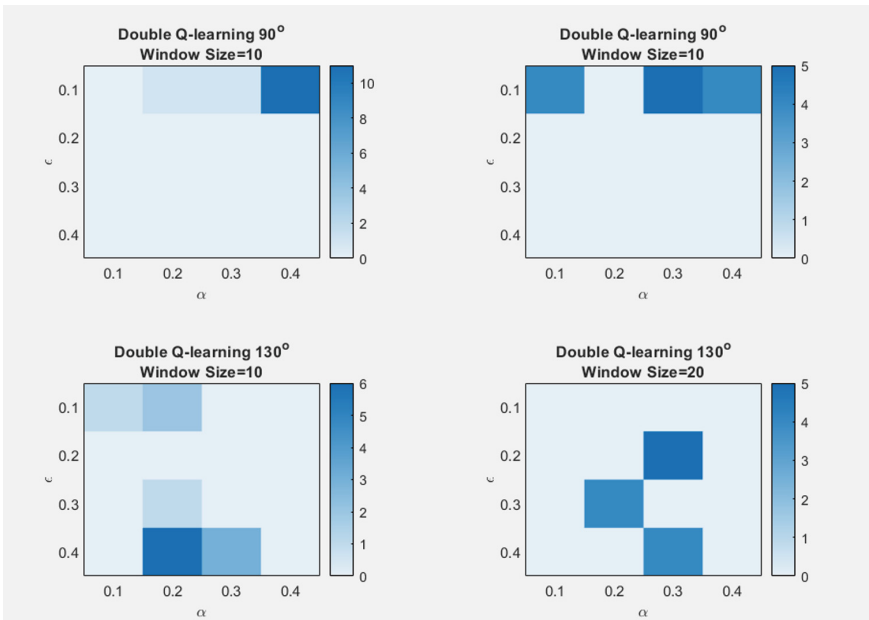


Fig. 8. Double Q-Learning Heat Map

Figure 8 shows the heat map of (α, ϵ) occurrences resulting in the minimum product in Eq. 4 for Double Q-learning. For the 90° scenario the (ϵ, α) combinations that occur the most often are $(0.1, 0.1)$, $(0.1, 0.3)$ and $(0.1, 0.4)$ for both window sizes. Like the 90° case, it can also be seen that an ϵ of 0.1 does occur for the 130° scenario for a window size of 10. However, we also see that ϵ values of 0.3 and 0.4 do occur for both window sizes. The dominant (ϵ, α) are $(0.1, 0.1)$, $(0.1, 0.2)$, $(0.2, 0.3)$, $(0.3, 0.2)$, $(0.4, 0.2)$, and $(0.4, 0.3)$.

6 Conclusion and Future Work

We presented RL-based AoA detection algorithms for mmWave systems that are operated by SDR. By adapting Q-learning and Double Q-learning to the AoA detection problem, we demonstrated the practicality of the approach and experimentally evaluated the methods in a mmWave SDR testbed. We achieved AoA detection within 2° of the correct AoA accuracy using the RL algorithms Q- and Double Q-learning. Compared to [9], our current setup uses unsupervised learning and does not rely on labeling data sets. The setup in [9] uses GPS to label transmitted signals. GPS does poorly in indoor environments, due to low power of reception. Further, for the RL algorithms Q- and Double Q-learning, our study investigated the best combinations of hyper-parameters that minimizes the AoA detection error and convergence time of the algorithms. We showed that double Q-learning outperforms Q-learning with respect to both AoA accuracy and convergence time. Compared to [18] and [19], our mmW platform is much cheaper and allows for a more user friendly interface.

Neural networks aren't used in our Q- and Double Q-learning algorithms. We plan on implementing and testing deep learning methods utilizing neural networks for AoA detection. With the usage of cheap servos, our current system set-up is limited. We plan to equip our testbed with phased array antennas [6], which can steer antenna beams in the order of microseconds. This will result in much faster convergence time and enable more flexible beamsteering enlarging the action space for the learning algorithms. We also plan to integrate field programmable gate array (FPGA) to our setup, in order to improve hardware and software run-time. An improvement in run-time will result in better overall convergence time.

Acknowledgment. This was supported in part by the U.S. National Science Foundation award 1836741.

References

1. Seth, S., Yuksel, M., Vosoughi, A.: Forming coalition sets from directional radios. In: MILCOM 2022–2022 IEEE Military Communications Conference (MILCOM), pp. 507–514 (2022)
2. Jilani, S.F., Munoz, M.O., Abbasi, Q.H., Alomainy, A.: Millimeter-wave liquid crystal polymer based conformal antenna array for 5G applications. *IEEE Antennas Wirel. Propag. Lett.* **18**(1), 84–88 (2019)

3. Dhruva, T., Rajesh, A., Abirami, K.: Design of conformal multi-band mmWave wearable antenna for 5G applications. In: Proceedings of IEEE International Conference on Smart Electronics and Communication (ICOSEC), pp. 573–577 (2022)
4. Hu, K., Zhou, Y., Sitaraman, S.K., Tentzeris, M.M.: Fully additively manufactured flexible dual-band slotted patch antenna for 5G/mmWave wearable applications. In: Proceedings of IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI), pp. 878–879 (2022)
5. Bogale, T., Wang, X., Le, L.: Chapter 9 - mmwave communication enabling techniques for 5g wireless systems: a link level perspective. In: Mumtaz, S., Rodriguez, J., Dai, L. (eds.) mmWave Massive MIMO, pp. 195–225. Academic Press (2017). <https://www.sciencedirect.com/science/article/pii/B9780128044186000091>
6. Jean, M., Velazquez, E., Gong, X., Yuksel, M.: A 30 ghz steerable patch array antenna for software-defined radio platforms. SoutheastCon **2023**, 856–860 (2023)
7. Fan, W., Xia, Y., Li, C., Huang, Y.: Channel tracking and aoa acquisition in quantized millimeter wave MIMO systems. IEEE Trans. Vehicular Technol., 1–15 (2023)
8. Yazdani, H., Seth, S., Vosoughi, A., Yuksel, M.: Throughput-optimal d2d mmwave communication: Joint coalition formation, power, and beam optimization. In: 2022 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1539–1544 (2022)
9. Dai, Z., He, Y., Vu, T., Trigoni, N., Markham, A.: Deepaoanet: learning angle of arrival from software defined radios with deep neural networks (2021)
10. Chowdhury, M.M.U., Erden, F., Guvenc, I.: Rss-based q-learning for indoor uav navigation. In: MILCOM 2019–2019 IEEE Military Communications Conference (MILCOM), pp. 121–126 (2019)
11. Watkins, C., Dayan, P.: Technical note: Q-learning. Mach. Learn. **8**, 279–292 (1992)
12. Van Hasselt, H.: Double q-learning, pp. 2613–2621 (Jan 2010)
13. Jaakkola, T., Jordan, M.I., Singh, S.P.: On the convergence of stochastic iterative dynamic programming algorithms. Neural Comput. **6**(6), 1185–1201 (1994)
14. Tsitsiklis, J.: Asynchronous stochastic approximation and q-learning. In: Proceedings of 32nd IEEE Conference on Decision and Control, vol. 1, pp. 395–400 (1993)
15. GNU Radio - the free and open source radio ecosystem, GNU Radio. <https://www.gnuradio.org>
16. Zafari, F., Gkelias, A., Leung, K.K.: A survey of indoor localization systems and technologies. IEEE Commun. Surv. Tutorials **21**(3), 2568–2599 (2019)
17. Wu, K., Ni, W., Su, T., Liu, R.P., Guo, Y.J.: Recent breakthroughs on angle-of-arrival estimation for millimeter-wave high-speed railway communication. IEEE Commun. Mag. **57**(9), 57–63 (2019)
18. Şahin, A., Sichertiu, M.L., Guvenc, İ.: A Millimeter-Wave Software-Defined Radio for Wireless Experimentation, [arXiv:2302.08444](https://arxiv.org/abs/2302.08444), (Feb. 2023)
19. Gu, X.: A multilayer organic package with 64 dual-polarized antennas for 28ghz 5g communication. In: 2017 IEEE MTT-S International Microwave Symposium (IMS), pp. 1899–1901 (2017)
20. Aerpaw: Aerial experimentation and research platform for advanced wireless. <https://aerpaw.org/>
21. Alkhateeb, A.: Deepmimo: a generic deep learning dataset for millimeter wave and massive MIMO applications. CoRR, vol. abs/ [arXiv: 1902.06435](https://arxiv.org/abs/1902.06435), (2019)
22. Kaveh, M., Barabell, A.: The statistical performance of the music and the minimum-norm algorithms in resolving plane waves in noise. IEEE Trans. Acoust. Speech Signal Process. **34**(2), 331–341 (1986)

23. Pastorino, M., Randazzo, A.: A smart antenna system for direction of arrival estimation based on a support vector regression. *IEEE Trans. Antennas Propag.* **53**, 2161–2168 (2005)
24. Jean, M., Yuksel, M., Gong, X.: Millimeter-wave software-defined radio testbed with programmable directionality. In: *Proceedings IEEE INFOCOM Workshop 2023* (in press)
25. UBX 10–6000 MHz Rx/TX (40 MHz, N series and X series), Ettus Research, a National Instruments Brand. <https://www.ettus.com/all-products/ubx40>
26. ADMV1013, Analog Devices. <https://www.analog.com/en/products/admv1013.html>
27. ADMV1014, Analog Devices. <https://www.analog.com/en/products/admv1014.html>
28. Kearns, M., Singh, S.: Finite-sample convergence rates for q-learning and indirect algorithms. In: *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pp. 996–1002. MIT Press, Cambridge, MA, USA (1999)
29. Ernst, D., Geurts, P., Wehenkel, L.: Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.* **6**, 503–556 (2005)
30. Szepesvári, C.: The asymptotic convergence-rate of q-learning. In: *Proceedings of the 10th International Conference on Neural Information Processing Systems*, ser. NIPS 1997, pp. 1064–1070. MIT Press, Cambridge, MA, USA (1997)



Empowering Resource-Constrained IoT Edge Devices: A Hybrid Approach for Edge Data Analysis

Rajeev Joshi^(✉), Raaga Sai Somesula, and Srinivas Katkoori

Department of Computer Science and Engineering, University of South Florida,
Tampa, FL 33620, USA

{rajeevjoshi, raagasai, katkoori}@usf.edu

Abstract. The efficient development of accurate machine learning (ML) models for Internet of Things (IoT) edge devices is crucial for enabling intelligent decision-making at the edge of the network. However, the limited computational resources of IoT edge devices, such as low processing power and constrained memory, pose significant challenges in implementing complex ML algorithms directly on these devices. This paper addresses these challenges by proposing a hybrid ML model that combines Principal Component Analysis (PCA), Decision Tree (DT), and Support Vector Machine (SVM) classifiers. By utilizing hardware-friendly techniques such as dimensionality reduction, optimized hyperparameters, and the combination of accurate and interpretable classifiers, the proposed hybrid model addresses the limitations of IoT edge devices. The proposed hybrid model enables intelligent decision-making at the edge while minimizing computational and energy costs. Experimental evaluations demonstrate the improved performance and resource utilization of the proposed model, providing insights into its effectiveness for IoT edge applications.

Keywords: Machine learning · Edge-AI · Internet-of-Things · ASICs

1 Introduction

In recent years, IoT has witnessed tremendous growth and widespread adoption, resulting in a substantial increase in data generation at the edge of the network. This data surge is primarily attributed to the diverse array of sensors, actuators, and embedded systems integrated into edge devices [12]. However, these devices are often constrained in terms of computational resources. This constraint presents a significant challenge when attempting to deploy resource-intensive ML algorithms directly on these devices. The limitations in processing power and memory capacity and the crucial requirement for energy efficiency have highlighted the pressing need for the development of hardware-friendly ML models [8, 9, 11] specifically designed to facilitate effective decision-making at the edge.

The advent of IoT has brought about a significant transformation in conventional computing approaches, emphasizing the transition from centralized

processing to distributed and decentralized systems. This shift in paradigm highlights the need for intelligent decision-making capabilities at the edge, where data is generated, to facilitate real-time analysis and response. However, the practical deployment of complex ML models directly on resource-constrained edge devices is frequently unfeasible owing to the inherent limitations of these devices. Consequently, there is a compelling requirement to design and implement hardware-friendly ML models that can operate effectively within the constraints of these devices. These models would facilitate effective and reliable decision-making at the edge, realizing the full capabilities of IoT applications [7, 10].

We propose a hybrid ML model that addresses the challenges posed by limited computational resources, memory constraints, energy efficiency requirements, and latency on IoT edge devices. We incorporate both supervised and unsupervised ML techniques. We integrate PCA, DT, and SVM classifiers that offer a holistic approach to improving hardware ML inference models for intelligent decision-making at the IoT edge. The motivation behind our hybrid model stems from the pressing requirement to effectively utilize limited computational resources while ensuring high accuracy and reliability on IoT edge devices. By incorporating PCA as a preprocessing step in our hybrid model, we aim to alleviate the burden of dimensionality in the input data. PCA allows for the extraction of the most informative features while reducing the computational and memory requirements associated with high-dimensional data. This dimensionality reduction technique results in concise and optimized feature vectors that are better suited for resource-constrained IoT edge devices. Similarly, the combination of DT and SVM classifiers in our hybrid model provides a comprehensive and robust approach for handling the diverse patterns and complexities often encountered in IoT data. DTs excel at capturing intricate relationships and generating interpretable predictions, making them ideal for understanding the underlying data structure. On the other hand, SVM classifiers are known for their ability to handle nonlinear data and achieve high classification accuracy. By leveraging the strengths of both models, our hybrid approach aims to enhance the overall performance of the model in terms of accuracy, interpretability, and generalization on IoT edge devices. In this preliminary research, we consider the crucial factor of energy efficiency, which holds paramount importance for IoT edge devices functioning under limited energy sources. The integration of PCA for dimensionality reduction, along with the utilization of optimized hyperparameters obtained through grid search, plays a pivotal role in minimizing unnecessary computations and energy consumption. Through meticulous resource management, our model guarantees energy-efficient operation while maintaining high levels of accuracy and performance. This hybrid model can be directly deployed at the IoT edge or further optimized using hardware optimization techniques to generate application-specific integrated circuits (ASICs) for IoT applications.

The rest of the paper is organized as follows: Sect. 2 presents background on contemporary machine learning models and an overview of the existing literature. Section 3 presents our proposed work. Section 4 reports experimental

design and results along with its discussion. Finally, we present the conclusion and potential future perspectives in Sect. 5.

2 Background and Related Work

In this section, we review the contemporary works on ML models, and related work to design efficient ML models for IoT edge applications.

2.1 Machine Learning

ML is an integral part of artificial intelligence (AI) that focuses on the creation of algorithms and models that empower computers to learn from data and make informed predictions or decisions without explicit programming. This rapidly advancing field has brought about revolutionary changes in diverse sectors, including healthcare, finance, transportation, etc. [14]. At the heart of ML are the ML models, which serve as mathematical representations or algorithms trained on labeled data to capture patterns, relationships, and valuable insights. These models play a pivotal role in making predictions, classifying new data, and uncovering hidden patterns within large datasets. ML models encompass various forms, such as DTs, neural networks (NNs), SVMs, etc., each possessing distinct strengths and applications. The development and implementation of accurate and efficient ML models hold paramount importance in facilitating intelligent decision-making, automation, and optimization across diverse industries and domains. This section provides a concise overview of some of the ML models employed in this work.

Principal Component Analysis. PCA is an unsupervised ML technique that is used to reduce the dimensionality of datasets. Its primary objective is to transform a high-dimensional dataset into a lower-dimensional representation while preserving as much of the original information as possible. This is achieved by identifying principal components, which are orthogonal vectors capturing the directions of maximum variance in the data [13]. These components form a new coordinate system, and the data is projected onto this reduced-dimensional space. The first principal component corresponds to the direction with the highest variance, and subsequent components follow in decreasing order of variance. By selecting a subset of the principal components that capture most of the variance, PCA enables efficient storage, visualization, and analysis of high-dimensional data. In addition, PCA finds applications in noise filtering, data compression, and feature extraction, making it a versatile tool in data preprocessing and analysis. PCA operates under the presumption that high-dimensional data frequently contains redundancy and that a lower-dimensional subspace can explain a large portion of the variation. By projecting the data onto a reduced-dimensional space, PCA can provide insights into the underlying structure and relationships within the dataset. However, it's important to note that PCA is

sensitive to the scale of the features, and data normalization is typically performed prior to applying PCA to ensure fair comparisons and accurate results. Overall, PCA serves as a valuable tool for dimensionality reduction and feature extraction to handle high-dimensional data efficiently and gain deeper insights from complex datasets.

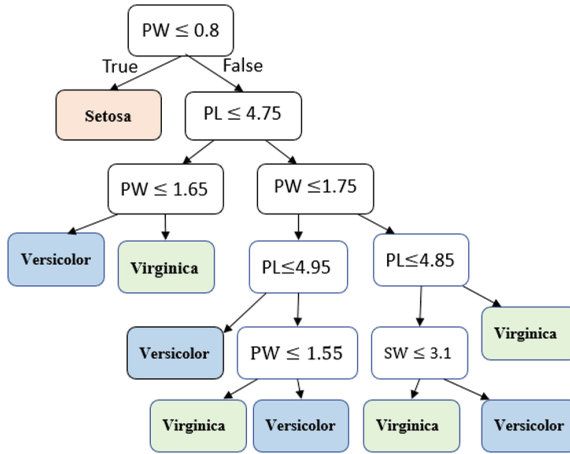


Fig. 1. Decision Tree model with Iris dataset classification

Decision Tree. DTs are ML techniques that are used for solving classification and regression problems. They represent decision-making processes through a hierarchical structure comprising nodes and branches. Figure 1 shows the classification of the Iris dataset using DT. At each node, the optimal split is determined using metrics like Gini impurity or entropy, aiding in the creation of distinct branches [17]. To address the issue of overfitting, pruning techniques can be employed to simplify the tree's structure. One of the notable advantages of decision trees is their ability to handle both categorical and numerical data, making them versatile for a range of applications. Furthermore, ensemble approaches such as random forest and gradient boosting can be employed to enhance the performance of decision trees by combining multiple trees. While decision trees offer interpretability, they can be sensitive to fluctuations in the data. Moreover, there exist extensions and modifications to decision trees, such as decision stumps (shallow trees) and advanced tree-based algorithms like XG-Boost, which further augment their capabilities.

Support Vector Machine. SVMs are supervised ML techniques extensively used for classification and regression tasks. The underlying principle of SVMs involves determining an optimal line or hyperplane that effectively separates different classes of data points while maximizing the margin between them. The

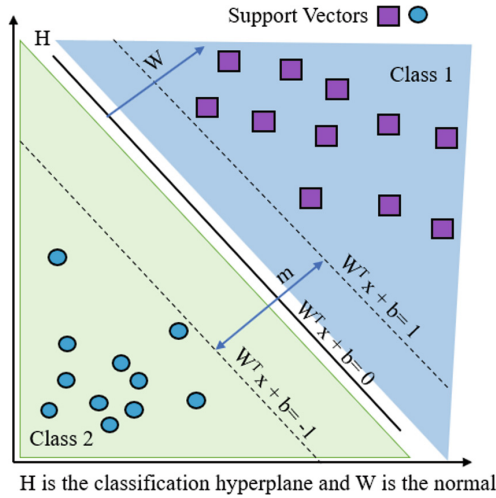


Fig. 2. Linear SVM model with two class classification

primary objective is to achieve a maximum margin, which refers to the distance between the hyperplane and the nearest data points in each class [6]. These nearest data points, known as support vectors, play a crucial role in defining the hyperplane and computing the margin. By focusing on support vectors, SVMs provide a computationally efficient approach to classification. Figure 2 shows the SVM model where it classifies two classes. To handle complex data patterns, SVMs leverage the kernel method, such as the linear kernel, polynomial kernel, Gaussian Radial Basis Function (RBF), sigmoid kernel, etc. This technique allows SVMs to implicitly map data into a higher-dimensional space, where it may become more separable. This mapping is performed without explicitly computing the coordinates in the higher-dimensional space. By identifying which side of the hyperplane a data point falls on, SVMs can effectively categorize new data, assigning it to the corresponding classes. SVMs offer several advantages, such as their ability to handle high-dimensional data, efficiently handle complex datasets, and exhibit resilience to overfitting. However, fine-tuning hyperparameters is necessary to ensure optimal performance.

2.2 Related Work

Ganaie et al. [5] proposed a method that utilizes twin-bounded support vector machines (TBSVM) to create oblique decision trees. These trees employ ensemble techniques and clustering hyperplanes to effectively separate data points into distinct groups. Ajani et al. [2], conducted a comprehensive analysis on embedded machine learning (EML), focusing on compute-intensive algorithms like k-nearest neighbors (k-NNs), support vector machines (SVMs), and deep neural networks (DNNs). They explored optimization techniques tailored for resource-

limited environments and discussed the implementation of EML on microcontrollers, mobile devices, and hardware accelerators. Struharik [18] introduced four hardware architectures aimed at accelerating axis-parallel, oblique, and non-linear decision tree ensemble classifiers. These architectures were optimized for FPGA and ASIC technology, highlighting their potential for embedded applications with limited system size. Shoaran et al. [15] introduced a hardware architecture that addresses the challenges of power and area constraints in applications like medical devices, specifically focusing on the implementation of gradient-boosted trees. Their architecture integrates asynchronous tree operation and sequential feature extraction techniques, resulting in notable energy and area efficiency. Yong et al. [21] presented a webshell detection system designed specifically for IoT networks, comprising both lightweight and heavyweight approaches. To enhance detection accuracy, ensemble methods utilizing traditional machine learning models such as DTs and SVMs were employed to improve the performance of the detection models. Hwang et al. [19] conducted an analysis of contemporary ML algorithms employed in edge computing to address security concerns, particularly in the context of IoT networks. Their assessment involved evaluating DTs, SVM, and logistic regression based on metrics such as computation complexity, memory footprint, storage requirements, and accuracy. This study also examined the applicability of these algorithms to various cybersecurity problems and explored their potential utilization in different use cases.

3 Proposed Work

In this section, we provide a comprehensive overview of our proposed approach. Our proposed work entails the development of a hybrid ML model that aims to design resource-efficient inference models. These inference models can be efficiently deployed using microcontrollers or used to design custom hardware inference models for IoT edge applications. With a primary focus on resource efficiency, we integrate contemporary ML techniques to create optimized inference models capable of optimizing resource utilization. By utilizing the synergies between ML algorithms, the proposed hybrid model optimizes resource usage, providing effective and efficient processing capabilities for IoT edge devices.

3.1 Hybrid Model Architecture

The proposed approach combines a dimensionality reduction technique with contemporary classification ML algorithms to form a hybrid model framework. We present a hybrid model that incorporates PCA, DT, and SVM. The proposed hybrid model leverages PCA, DT, and SVM to effectively address the challenging scenarios where the input dataset has a large number of features, and the DT or SVM alone may suffer from the curse of dimensionality. Through the process of reducing the dataset's dimensionality prior to the application of ML algorithms, this approach enhances classifier performance while concurrently decreasing the computational resources necessary for training and deploy-

ing models on resource-constrained IoT edge devices. By combining these techniques, the model aims to optimize the accuracy and resource utilization before the efficient inference model hardware implementations in IoT edge devices. In this approach, PCA is initially employed to reduce the number of features in the dataset by extracting the most informative principal components. This reduction in dimensionality helps alleviate the computational complexity and memory requirements associated with processing high-dimensional data without sacrificing much accuracy of the model. The DT and SVM are then finetuned utilizing the output of PCA to create a hybrid model. Finally, the predictions from these models are combined using a voting mechanism and averaging, resulting in an inference model that is both efficient and effective. This resource-efficient implementation ensures the feasibility of deploying the hybrid model on IoT edge devices with limited resources, enabling real-time classification tasks in a low-power and energy-efficient manner. The hybrid model architecture is shown in Fig. 3.

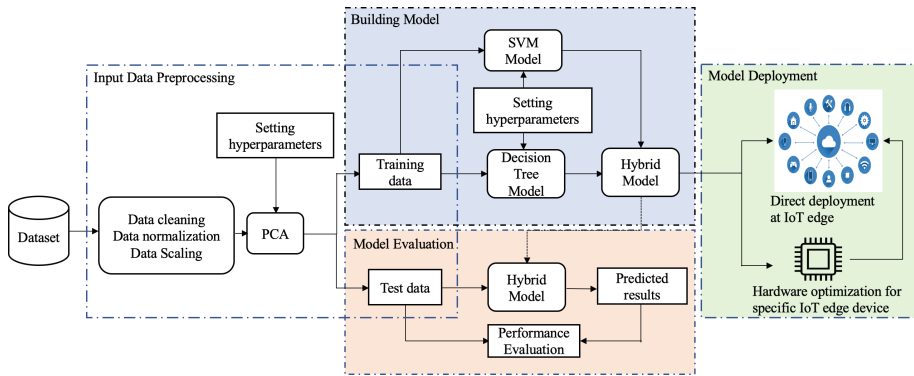


Fig. 3. Hybrid model architecture framework with workflow.

3.2 A Hybrid Dimensionality Reduction and Classification Workflow

The workflow for this hybrid approach incorporating PCA, DT, and SVM for dimensionality reduction and classification for a resource-efficient inference model hardware implementation for IoT edge devices consists of several key steps:

- **Input Dataset Preprocessing:** To ensure the accuracy and consistency of data, as well as to improve the overall accuracy of the model, initially, we perform necessary data cleaning, normalization, and scaling on the input dataset.

- **Optimizing the Hyperparameters of PCA:** We split the preprocessed data into training and validation sets. And then, we employed grid search for hyperparameter tuning, allowing us to explore various combinations of PCA hyperparameters. Our objective is to determine the optimal hyperparameters for PCA, including the number of principal components. Once we identified the ideal hyperparameters, we proceeded to fit PCA on the training set using these selected values. Finally, we applied the fitted PCA model to transform both the training and validation sets, ensuring that they were properly adjusted according to the chosen hyperparameters.
- **Hyperparameter Tuning and Training of Decision Tree Classifier:** We utilize the transformed training and validation sets obtained from the PCA step. This allowed us to capture the most informative features and reduce the dimensionality of the data. Subsequently, we employed grid search to systematically explore a range of decision tree hyperparameters, aiming to identify the most suitable combinations. The decision tree classifier was then trained using the training set, and its performance was evaluated on the validation set. If the obtained performance did not meet our expectations, we repeated the grid search process with different hyperparameter values. This iterative approach allowed us to fine-tune the decision tree model until optimal performance was achieved, ensuring that our classifier effectively captured the underlying patterns in the data.
- **Hyperparameter Tuning and Training of SVM Classifier:** We begin by utilizing the transformed training and validation sets obtained from the PCA step, again. To enhance the performance of our SVM classifier, we employed grid search to explore various combinations of hyperparameters. This iterative process allowed us to identify the optimal hyperparameter values that would maximize the SVM's performance. Subsequently, we trained the SVM classifier using the training set and evaluated its performance on the validation set. If the achieved performance did not meet our desired expectations, we iteratively conduct a grid search by exploring different hyperparameter values in order to enhance the results. This iterative refinement process ensures that we achieve the highest possible performance for our SVM classifier, enabling accurate and reliable classification of the feature vectors.
- **Evaluation of the Hybrid Model:** In this step, we combine the optimized decision tree classifier and SVM classifier into a unified framework. By integrating the decision tree and SVM outputs, we aimed to leverage the complementary strengths of these two models. We combine the outputs of the decision tree and the SVM using a voting mechanism to make the final prediction. Then, we conduct a comparative analysis of the accuracy of the hybrid model with those of the individual decision tree classifier and SVM classifier. If the performance did not meet our desired criteria, we revisit the hyperparameter tuning process for the PCA, decision tree, and SVM classifier. To achieve optimal results, we iteratively repeat the previous steps, adjusting the hyperparameters for the PCA, DT, and SVM classifiers using grid search, until we attained satisfactory performance. This iterative refinement process ensured

that our hybrid model is both accurate and resource-efficient, making it a viable solution for classification tasks in resource-constrained environments.

- Hybrid Model Deployment: After evaluating the hybrid model and confirming its satisfactory performance, deployment on the IoT edge becomes feasible. Direct deployment utilizing microcontrollers is a viable option for deploying the hybrid model. Additionally, the hybrid model can be further optimized for specific IoT edge applications through hardware optimization techniques, such as quantization, to develop custom hardware IoT devices.

4 Experimental Results

This section outlines the experimental design and results of our proposed method. In order to gauge the efficacy of the proposed methodology, a comprehensive evaluation is conducted utilizing five well-known classification dataset benchmarks, including Heart Disease [4], Breast Cancer Wisconsin (Diagnostic) [20], Lung Cancer [1], Fetal Health [3], and Pima Indian Diabetes datasets [16]. The Heart Disease dataset comprises 1,025 instances and 10 input features. The Breast Cancer Wisconsin (Diagnostic) dataset consists of 30 input features and 569 instances. The dataset pertaining to Lung Cancer comprises 309 instances and 15 input features. Similarly, the dataset on Fetal Health comprises 21 input features and 2,126 instances, and the Pima Indian Diabetes dataset consists of 11 input features and 767 instances. The experimental flow involves training the classification datasets using our proposed hybrid workflow. The training and testing data are randomly partitioned in an 80:20 ratio to generate the hybrid model. Subsequently, the performance of the hybrid model is evaluated through performance analysis in terms of accuracy.

Table 1. Optimized hybrid model input configurations

Dataset	Original no. of input features	No. of features in hybrid model	(%) of features reduced
Heart Disease	10	3	70%
Breast Cancer	30	12	60%
Lung Cancer	15	7	53%
Fetal Health	21	10	52%
Pima Indian Diabetes	11	5	55%

Extensive experiments are conducted to evaluate the efficacy of the proposed methodology. In the context of experimental design, the initial step involves the implementation of input data preprocessing procedures. These procedures include data cleaning, data normalization, and data scaling, which are intended to enhance the precision and uniformity of the input data. Following the preprocessing of the input dataset, a grid search technique is employed to optimize the

hyperparameters of a hybrid model consisting of PCA, DT, and SVM. The objective is to identify the optimal combination of hyperparameters that produces the best performance for each of these algorithms. Specifically, for PCA, we focus on fine-tuning hyperparameters such as the selection of the optimal number of orthogonal components that effectively capture the most significant variance in the input features. This allows us to significantly reduce the dimensions of the input features by condensing as much information as possible from the input features into a smaller subset of transformed features, which are then utilized as input for DT and SVM models. This reduction in dimensionality also leads to more efficient memory utilization and computational resources, making it suitable for resource-constrained IoT edge devices with limited processing power and storage capacity. Moreover, it results in lower energy consumption during the implementation of the hybrid inference model on IoT edge devices as well as helps to minimize the computational workload and data movement, enabling real-time and low-latency predictions, which are crucial for time-sensitive IoT applications. Table 1 presents the optimized hybrid model input configurations obtained after applying PCA. For the Heart Disease dataset, we observe a reduction in the number of features from the original 10 to 3 transformed features, resulting in a significant 70% reduction in the dimensionality of the input features. Similarly, for the Breast Cancer dataset, there is a 60% reduction in dimensionality as the number of input features decreases from 30 to 12 transformed features. The Lung Cancer, Fetal Health, and Pima Indian Diabetes datasets also demonstrate substantial reductions in dimensionality, with percentage reductions of 53%, 52%, and 55% respectively, in their respective input feature sets.

Subsequently, we proceed with fine-tuning the hyperparameters of DT and SVM in our hybrid model. Given our objective of creating a robust hybrid model specifically designed for IoT applications with hardware efficiency in mind, we meticulously approach the process of selecting hyperparameters for DT and SVM. In the case of DT, we pre-prune the DT prior to training by determining the optimal hyperparameters as follows: The criterion for DT is set as Gini impurity, considering its computational efficiency compared to entropy and its tendency to produce shorter and more cohesive branches. The maximum depth of the decision tree is set to 10, limiting the depth to control overfitting and model complexity. The minimum sample split is set to 2, ensuring that a split at a node requires at least 2 samples, preventing further divisions with insufficient data. The splitter strategy is selected as “best,” indicating that the best-split strategy is chosen at each node during the tree construction process. Other hyperparameters are left as default values. Similarly, we carefully determine the hyperparameters for SVM as follows: To optimize hardware implementation and efficiency, we select the sigmoid kernel for SVM, which offers simpler mathematical operations compared to more complex kernels like Gaussian (RBF). This choice facilitates easier implementation and optimization in hardware architectures, resulting in reduced computational complexity and memory requirements. Consequently, our approach reduces hardware resource utilization, energy efficiency, and execution time. Specifically, we set the regularization parameter (C)

Table 2. Evaluation of hybrid model training performance

Dataset	Accuracy (%)	Accuracy (%)
	Hard Voting	Soft Voting
Heart Disease	96.24	97.5
Breast Cancer	97.52	99.26
Lung Cancer	97.77	98.51
Fetal Health	99.26	99.65
Pima Indian Diabetes	99.75	99.87

Table 3. Evaluation of hybrid model testing performance

Dataset	Accuracy (%)	Accuracy (%)
	Hard Voting	Soft Voting
Heart Disease	95.12	96.67
Breast Cancer	95.39	98.04
Lung Cancer	93.54	95.23
Fetal Health	96.58	98.53
Pima Indian Diabetes	96.09	99.02

Table 4. Performance evaluation of DT and SVM models for training and testing

Dataset	DT		SVM	
	Training Accuracy (%)	Testing Accuracy (%)	Training Accuracy (%)	Testing Accuracy (%)
Heart Disease	99.14	98.50	98.00	96.08
Breast Cancer	99.99	98.79	99.58	94.26
Lung Cancer	99.71	97.74	98.00	96.45
Fetal Health	99.99	97.00	99.99	97.00
Pima Indian Diabetes	99.95	99.00	96.00	93.53

to 100 and the kernel coefficient (Gamma) to 10 based on experimentation. We maintain the remaining hyperparameters at their default values. By fine-tuning these hyperparameters, we aim to optimize the performance and efficacy of our hybrid model.

After determining the optimal hyperparameters for our hybrid model, we conducted comprehensive training and testing on the transformed datasets obtained through PCA using both DT and SVM. This enabled us to generate an optimized hybrid inference model specifically designed for resource-constrained IoT edge devices. The evaluation of our hybrid model's performance, as demonstrated in Tables 2 and 3, involved the utilization of two different techniques for output prediction: the hard and soft voting techniques.

Upon carefully analyzing the results presented in Table 2 and Table 3, we observed that the soft voting technique consistently yielded superior output predictions for both the training and testing of our hybrid model. Additionally, we compared the performance of our hybrid model with that of individual DT and SVM models, specifically for training and testing, as shown in Table 4. Our hybrid model achieved output predictions on par with these individual models, despite the latter not being optimized and exhibiting higher computational complexity, memory requirements, power consumption, and latency. The evaluation further revealed that the average loss in accuracy of our hybrid inference model remained below 4%, signifying its robustness and effectiveness.

In our review of the existing literature, we encountered limited related work for direct comparison with our proposed approach. We found only one relevant study [5] that utilized the Breast Cancer Wisconsin (Diagnostic) dataset for comparison purposes. Our proposed work demonstrates superior performance compared to the models presented in [5], yielding a modest improvement in results. Our approach achieves this improvement while utilizing simpler mathematical operations, demonstrating the effectiveness of our proposed method. Moreover, this hybrid model is suitable for direct deployment on IoT edge devices, and it also holds the potential for further optimization through hardware optimization techniques. By leveraging ASICs, we can achieve even more efficient implementations, thereby enhancing the overall performance and efficiency of the model on IoT edge devices.

In this preliminary work, our research findings reveal the promising capabilities of our optimized hybrid model in delivering accurate predictions while addressing the computational and resource constraints commonly encountered in IoT edge devices. The hybrid model exhibits improved accuracy and highlights reliable performance when tested on multiple datasets. These experimental results collectively show the effectiveness of our proposed method and its potential to enhance classification tasks across diverse domains.

5 Conclusions

In this work, we present a hybrid approach that integrates PCA, DT, and SVM for deployment in IoT edge devices. The experimental findings demonstrate the effectiveness of this approach in enhancing the performance of ML inference models for IoT edge applications. By integrating these contemporary ML techniques, the proposed approach achieves improved accuracy while effectively addressing computational and memory limitations of resource-constrained IoT edge devices. In the future, we are interested to focus on further optimizing the hybrid model for specific IoT edge device architectures and exploring additional feature selection and classification techniques to improve performance and efficiency.

References

1. Ahmad, A.S., Mayya, A.M.: A new tool to predict lung cancer based on risk factors. *Heliyon* **6**(2), e03402 (2020)
2. Ajani, T.S., Imoize, A.L., Atayero, A.A.: An overview of machine learning within embedded and mobile devices-optimizations and applications. *Sensors* **21**(13), 4412 (2021)
3. Ayres-de Campos, D., Bernardes, J., Garrido, A., Marques-de Sa, J., Pereira-Leite, L.: SisPorto 2.0: a program for automated analysis of cardiocograms. *J. Maternal-Fetal Med.* **9**(5), 311–318 (2000)
4. Detrano, R.: UCI Machine Learning Repository: Heart Disease Data Set (2019)
5. Ganaie, M., Tanveer, M., Suganthan, P.N.: Oblique decision tree ensemble via twin bounded SVM. *Expert Syst. Appl.* **143**, 113072 (2020)
6. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. *IEEE Intell. Syst. Appl.* **13**(4), 18–28 (1998)
7. Joshi, R., Zaman, M.A., Katkoori, S.: Novel bit-sliced near-memory computing based VLSI architecture for fast Sobel edge detection in IoT edge devices. In: 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), pp. 291–296. IEEE (2020)
8. Joshi, R., Kalyanam, L.K., Katkoori, S.: Simulated annealing based integerization of hidden weights for area-efficient IoT edge intelligence. In: 2022 IEEE International Symposium on Smart Electronic Systems (iSES), pp. 427–432 (2022). <https://doi.org/10.1109/iSES54909.2022.00093>
9. Joshi, R., Kalyanam, L.K., Katkoori, S.: Area efficient VLSI ASIC implementation of multilayer perceptrons. In: 2023 International VLSI Symposium on Technology, Systems and Applications (VLSI-TSA/VLSI-DAT), pp. 1–4. IEEE (2023)
10. Joshi, R., Zaman, M.A., Katkoori, S.: Fast Sobel edge detection for IoT edge devices. *SN Comput. Sci.* **3**(4), 302 (2022)
11. Kalyanam, L.K., Joshi, R., Katkoori, S.: Range based hardware optimization of multilayer perceptrons with RELUs. In: 2022 IEEE International Symposium on Smart Electronic Systems (iSES), pp. 421–426 (2022). <https://doi.org/10.1109/iSES54909.2022.00092>
12. Laghari, A.A., Wu, K., Laghari, R.A., Ali, M., Khan, A.A.: A review and state of art of internet of things (IoT). *Arch. Comput. Methods Eng.* 1–19 (2021)
13. Lee, L.C., Jemain, A.A.: On overview of PCA application strategy in processing high dimensionality forensic data. *Microchem. J.* **169**, 106608 (2021)
14. Shanthamallu, U.S., Spanias, A., Tepedelenlioglu, C., Stanley, M.: A brief survey of machine learning methods and their sensor and IoT applications. In: 2017 8th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1–8. IEEE (2017)
15. Shoaran, M., Haghi, B.A., Taghavi, M., Farivar, M., Emami-Neyestanak, A.: Energy-efficient classification for resource-constrained biomedical applications. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **8**(4), 693–707 (2018)
16. Smith, J.W., Everhart, J.E., Dickson, W., Knowler, W.C., Johannes, R.S.: Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In: Proceedings of the Annual Symposium on Computer Application in Medical Care, pp. 261. American Medical Informatics Association (1988)
17. Somesula, R.S., Joshi, R., Katkoori, S.: On feasibility of decision trees for edge intelligence in highly constrained internet-of-things (IoT). In: Proceedings of the Great Lakes Symposium on VLSI 2023, pp. 217–218 (2023)

18. Struharik, R.: Decision tree ensemble hardware accelerators for embedded applications. In: 2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY), pp. 101–106. IEEE (2015)
19. Wang, H., Barriga, L., Vahidi, A., Raza, S.: Machine learning for security at the IoT edge—a feasibility study. In: 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), pp. 7–12. IEEE (2019)
20. Wolberg, W., Street, W., Mangasarian, O.: Breast cancer wisconsin (diagnostic). UCI Machine Learning Repository (1995)
21. Yong, B., et al.: Ensemble machine learning approaches for Webshell detection in internet of things environments. *Trans. Emerg. Telecommun. Technol.* **33**(6), e4085 (2022)



Energy-Efficient Access Point Deployment for Industrial IoT Systems

Xiaowen Qi¹, Jing Geng², Mohamed Kashef²,
Shuvra S. Bhattacharyya¹, and Richard Candell²

¹ University of Maryland, College Park, MD 20740, USA
{xqi12,ssb}@umd.edu

² National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
{jing.geng,mohamed.kashef,richard.candell}@nist.gov

Abstract. Internet of Things (IoT) technologies have impacted many fields by opening up much deeper and more extensive integration of communications connectivity, sensing, and embedded processing. The industrial sector is among the areas that have been impacted greatly — for example, IoT has the potential to provide novel capabilities for more effective tracking, control and optimization of industrial processes. To maintain reliable embedded processing and connectivity in industrial IoT (IIoT) systems, including systems that involve intensive use of smart wearable technologies, energy consumption is often a critical consideration. With this motivation, this paper develops an energy-efficient deployment strategy for access points in IIoT systems. The developed strategy is based on a novel genetic algorithm called the Access Point Placement Genetic Algorithm (AP2GA). Simulation results with our proposed deployment strategy demonstrate the effectiveness of AP2GA in optimizing energy consumption for IIoT systems.

Keywords: Green Communication · Wireless Industrial IoT · Genetic Algorithm

1 Introduction

Wireless communications technologies are of increasing interest in industrial environments because of their important potential benefits compared to full reliance on wired communications [3]. As such, Industrial Internet of Things (IIoT) is playing a huge role in industry due to the connectivity capabilities provided by wireless technology, revolutionizing the sector. For example, various sensors can be deployed to monitor temperature, humidity, and vibrations of machines to create safer production environments and to report early warnings of possible malfunctions. By seamlessly connecting various devices and sensors, IIoT enables more efficient data collection, analysis, and process control, bringing productivity into higher levels.

However, with ever-increasing system complexity, the increasing amounts of energy consumed by wireless communication devices has attracted significant

attention from both academia and industry. The large amount of energy consumption also poses challenges to the environment, as renewable green energy is typically not used as a power source for wireless networks [2].

The energy consumption attributable to Information and Communication Technology (ICT) has exhibited large increases with the advent of new technologies, such as Fifth Generation (5G) and Multiple-Input and Multiple-Output (MIMO), as such technologies require more power consumption to increase response speed and accommodate more users. Therefore, innovation in green communications technologies is in urgent need.

To help address this need, we propose and demonstrate a systematic Access Point (AP) deployment strategy for energy-efficient IIoT systems. The remainder of this paper is organized as follows. Section 2 discusses background and related literature about AP deployment strategies. Section 3 explains the proposed energy-efficient AP deployment strategy. Section 4 reviews the factory system flow model used in our simulation experiments. Section 5 presents the results of the proposed strategy, which are obtained from the aforementioned simulation experiments. Finally, the paper is concluded in Sect. 6.

2 Background and Related Work

In wireless networks, an AP plays a crucial role by providing wireless connectivity and forwarding communication between devices or even networks as a relay node. To effectively support these functionalities and meet users' requirements, proper deployment of APs is essential. In this paper, by deployment of APs, we specifically mean the physical placement of the APs for operation in a given site.

A range of approaches regarding AP deployment has been proposed in the literature, with a goal of seeking optimal positions based on various objectives, such as reducing the number of APs used [8] or improving network performance [6, 10, 12–14].

In [8], the authors apply a continuous optimization technique known as A new Global OPTimization algorithm (AGOP) to minimize the number of APs used to cover a service area containing obstacles. The authors of [6] employ a multiobjective Tabu algorithm to search the set of candidate locations. The algorithm jointly considers coverage, interference, and Quality of Service (QoS). The final selection is made based on the most important factor to the end user. In [10], AP deployment and channel allocation are optimized together with a computationally-efficient local search algorithm to maximize system throughput and achieve fair resource sharing. In order to reflect the dynamic movement of users in an indoor wireless local area network (WLAN) system, the authors of [12] first use statistical theory to model the location and probability of the user distribution, and then model and solve the corresponding AP deployment problem with the fuzzy C-clustering algorithm.

In addition to the above algorithms, Genetic Algorithms (GAs) have also been widely used in identifying efficient AP deployment locations, especially under relevant multi-objective constraints. GAs are heuristic optimization methods inspired by Darwin's Theory of Evolution. They form an important sub-class

of evolutionary algorithms [1]. A GA iteratively evolves to a solution of the given problem by using principles of natural selection. GAs have been shown to perform well on complex optimization problems where it is infeasible to derive optimal solutions with manageable time-complexity [1].

In [13], the authors take non-uniform user distribution into account, using a GA to cooperatively optimize the coverage, number of APs, and interference. In [14], an optimized placement of APs is selected with a GA such that the transmit power and overlap rate are minimized under the constraint of full coverage. The average transmit power is substantially optimized; it is reduced by about 61%. However, this average value may not be very useful in real-life situations, since it is possible that the device in the system with the smallest transmit power only processes a limited amount of traffic. The communication energy consumption would be a better metric to consider to more accurately inform system analysis and optimization.

Motivated by the above observations, we propose a novel energy-efficient AP placement method. The method mutually considers non-uniform user distribution and unbalanced communication activity on the premise of complete coverage. Our method considers total communication energy consumption as a key metric to guide the optimization process. This is a complex optimization problem, and a GA is designed to derive an efficient deployment setup for a given deployment scenario.

3 Proposed Methods

In the problem formulation that is addressed in this work, the energy cost to be optimized refers to the energy consumed by communication activities that occur during normal operation of the IIoT system. Specifically, the problem definition targeted in this work is the optimization of communication energy given a placement of networked devices, which may be unevenly distributed, and a characterization of the traffic demand for each device.

The communication energy considered in this paper refers to the transmission energy. Energy associated with communication reception for the devices is not taken into account in the methods developed in the paper, as it is common in related analysis contexts to focus on transmission power, and consideration of the transmission energy provides an approximation of the overall energy consumption due to communication. Incorporation of models for reception energy into the developments of this paper is an interesting direction for future work.

To save energy, we consider optimal placement of the APs so that they can deliver packets to all stations (STAs) in an area with an appropriate transmit power according to their activity rates. The fitness function can be mathematically expressed as follows:

$$\begin{aligned}
& \min_{x_i, y_i, \alpha_{i,j}} \sum_{i=1}^n \sum_{j=1}^s \mathbf{1}_{i,j} \alpha_{i,j} t_{i,j} \beta_{i,j} & (1) \\
s.t. \quad & C1 : (x_i, y_i) \in \mathbb{Q} \\
& C2 : \sum_{i=1}^n \mathbf{1}_{i,j} = 1, \forall j \in [1, s] \\
& C3 : \alpha_{min} \leq \alpha_{i,j} \leq \alpha_{max}, \forall i \in [1, n], \forall j \in [1, s] \\
& C4 : \alpha_{i,j} + G_{i,j} - L_{i,j} \geq \alpha_0, \exists i \in [1, n], \forall j \in [1, s]
\end{aligned}$$

Here, n and s denote the number of used APs and STAs respectively, (x_i, y_i) is the position of AP_i , $\mathbf{1}_{i,j}$ indicates whether STA_j is associated to AP_i , $\alpha_{i,j}$, $t_{i,j}$, $\beta_{i,j}$ are the used transmit power, transmission time, and total communication activity rate (including both downlink and uplink activity) occurring in the link between AP_i and STA_j respectively, \mathbb{Q} constrains the service area, α_{min} and α_{max} set the lower and upper bound for the transmit power, $G_{i,j}$ and $L_{i,j}$ are the antenna gains and losses of the communication link between AP_i and STA_j , and α_0 refers to the receiver sensitivity of signal detection. For simplicity we assume a single transmit power setting for both directions of a link; the framework can readily be extended to handle differing transmit power values.

The antenna gains add both the transmitter antenna gain and receiver antenna gain. Similarly, the loss $L_{i,j}$ of each link contains three components: cable and connector losses on both sides, path loss, and miscellaneous losses such as fading margin. The propagation loss is estimated using the log-distance path loss model:

$$L = L_0 + 10\gamma \log_{10}\left(\frac{d}{d_0}\right), \quad (2)$$

where L_0 is the path loss at the reference distance d_0 , γ is the decay component, and d is the distance between transmitter and receiver.

Figure 1 illustrates the communication activities in a simple network consisting of two STAs and one AP. Different colors (i.e., blue and black) are used to distinguish different directions of transmission. Dashed lines represent expected/imagined communication paths, while solid lines represent the corresponding actual occurring communication paths. Suppose STA_1 needs to send 3 messages to STA_2 . After receiving and analyzing the messages, STA_2 sends a message back to STA_1 . The intermediate AP AP_1 acts as a relay node to perform the above operations. In this case, $\beta_{1,1} = 3(\text{uplink}) + 1(\text{downlink}) = 4$ and $\beta_{1,2} = 1(\text{uplink}) + 3(\text{downlink}) = 4$. Note that they are equal because there are only two links existing in this scenario. $\alpha_{1,1}$ is the transmit power used by STA_1 and AP_1 , and $t_{1,1}$ is the transmission time of packets in the link between AP_1 and STA_1 . $\alpha_{1,2}$ and $t_{1,2}$ have similar meaning but between AP_1 and STA_2 .

If the constraints $C1$, $C3$, and $C4$ are jointly satisfied, then STA_j is efficiently covered by AP_i in the given environment. Depending on the settings, it

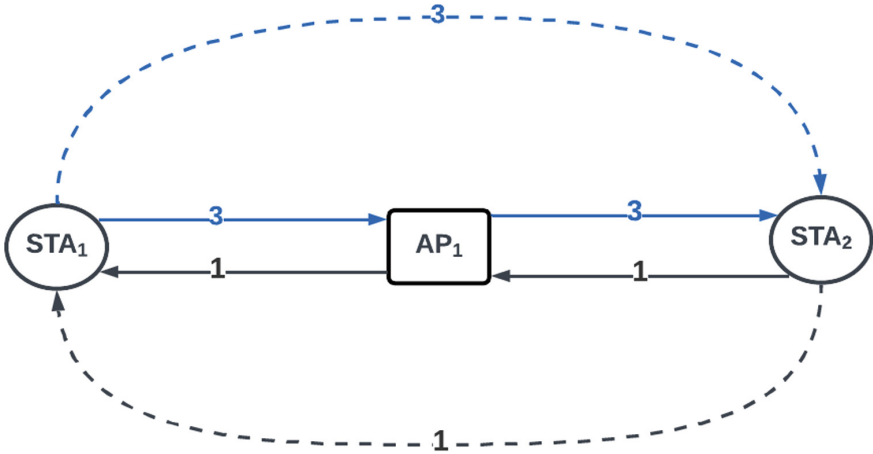


Fig. 1. An illustration of communication activity between two STAs.

is possible that an STA is covered by multiple APs. The constraint $C2$ takes this situation into account and restricts an STA to only communicate with the AP offering the best cost.

We have developed a GA to solve the multivariate optimization problem formulated above, and we refer to our GA-based AP placement approach as the AP Placement GA (AP2GA). We have developed a prototype implementation of AP2GA using the DEAP Framework for GA implementation [4]. AP2GA iterates through a series of genetic operations to evolve the population (current set of candidate solutions). After a pre-determined number of iterations, AP2GA produces its final population, and from the final population, a solution with maximum fitness (see Eq. 1) is selected as the final solution to the optimization process. Figure 2 illustrates this operation of AP2GA in a flowchart.

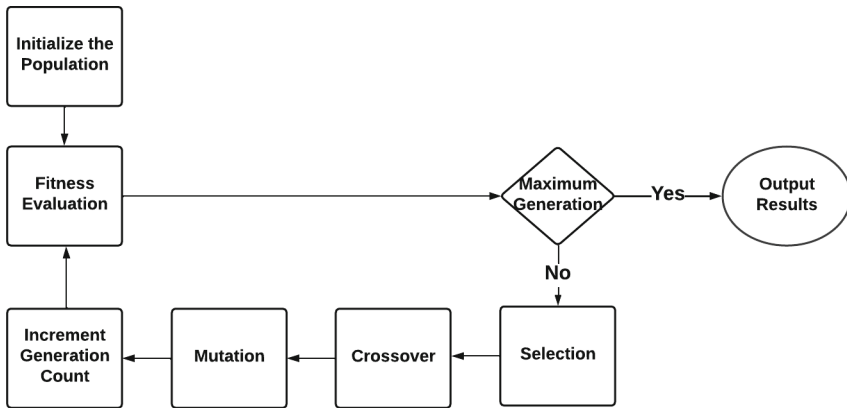


Fig. 2. A flowchart of the proposed deployment strategy.

AP2GA starts by randomly initializing a set of candidate solutions, which will form the initial population for the optimization process. Each candidate solution in a GA population is referred to, in its encoded form, as a chromosome. A candidate solution in a GA population is also referred to as an individual. Each chromosome in the population consists of a set of genes (bits) that encode x_i , y_i , and $\alpha_{i,j}$ ($i = 1, \dots, n, j = 1, \dots, s$) in binary format. The crossover and mutation operations, which are used to evolve the population, operate directly on the bits of the chromosome. A gene bit-string is initialized under the constraints $C1$ and $C3$, and its length depends on a user-specified precision value.

After that, the fitness function is called iteratively for each individual. Based on the obtained fitness score, a tournament selection process is used to select parents to breed offspring. There is a feasibility check to see if each individual violates any constraint. If so, a large penalty value is added to the fitness score of the individual. The subsequent tournament selection process selects the parents to breed offspring based on the fitness score, so invalid individuals with large fitness scores are less likely to be selected for survival. In our formulation, higher fitness scores correspond to lower-quality solutions, so more “fit” individuals (higher levels of fitness) in the GA population correspond to lower fitness scores.

A two-point crossover follows to exchange information between the selected parents. As the name suggests, two crossover points are randomly chosen and the genes in-between are swapped to reproduce different offspring (derived candidate solutions) with different bit patterns as chromosomes. Subsequently, mutation is applied on the chromosomes. Mutation refers to the unpredictable change in certain genes during the genetic process, which is not guaranteed to have a positive or negative influence on fitness, but will enhance genetic diversity in the population [7]. The above process of evaluation and genetic manipulation (selection, crossover, and mutation) is repeated until a pre-determined number of generations has been reached.

4 Factory System Modeling

We extend our previously proposed factory process-flow model to evaluate our new deployment strategy [5]. An illustration is shown in Fig. 3.

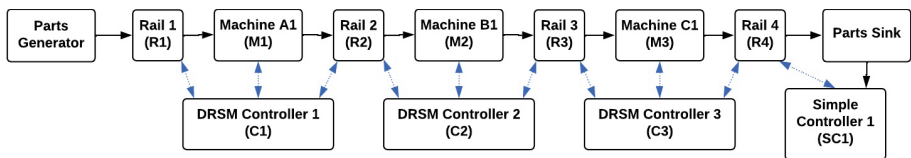


Fig. 3. The factory process flow model that we use in our experiments.

Five types of functional units (actors) are used in the system: part generator, rail, machine, controller, and part sink, where they cooperatively model a basic

work cell in a production environment. Each actor effectively encapsulates a finite state machine, where each state (mode) of the actor corresponds to a specific sub-function executed by the actor. The raw components produced by the part generator undergo processing by machines that add features, and are ultimately stored in the part sink once all processing is completed. There are two types of edges shown in the figure: the one-way black edges represent physical links, including both physical entity transport and any associated information flow, while the two-way blue edges represent the transfer of data across wireless communication links.

Rails, machines, and controllers in the environment are equipped with communication devices. Rails and machines report their status to the controller whenever mode transitions happen. After receiving state information, the responsible dual-rail-single-machine (DRSM) controller performs some computation and sends instructions back to the actors to which it is connected. Additionally, there is a special controller, called a simple controller, as shown in the lower right side of Fig. 3. The simple controller records the capacity information of the part sink and controls the release of the last rail. Thus, the modeled workflow can run smoothly with continuous information exchange.

Communication capability is enabled by a specific type of actor called a communication interface actor. Communication tasks are divided into sending and receiving sub-tasks, which are undertaken by the send interface actor (SIA) and receive interface actor (RIA), respectively. For more details on the factory system modeling approach that we build upon in this paper, including the modeling of communication functionality, we refer the reader to [5,9].

In [5,9], it is assumed that the machines used in the factory floor are homogeneous. However, in general the operation of a factory may involve the cooperative work of different types of machines, which are specialized for diverse tasks. This diversity generally results in varying processing times and varying levels of communication traffic. Therefore, unbalanced processing and communication activities need to be considered for more general system modeling scenarios.

5 Experiments

In this section, we present simulation results that demonstrate the effectiveness of the proposed AP2GA approach. Our simulations are carried out using ns-3 [11].

The service area for our simulated systems is 20 m x 20 m. There is a total of 13 actors involved in the factory system model of which 11 of the actors involve data communication to other actors (the part generator and part sink do not involve data communication in this model). The communication relationships between the actors are illustrated in Fig. 3.

Each actor is characterized by an activity rate, which characterizes both the outgoing and incoming traffic for the actor. Except for the first rail, each rail has a fixed activity rate of 5 messages/cycle. Here, by a “cycle”, we meant the entire processing of a single part through the entire factory pipeline, from the

part source to the part sink. Three types of machines are used, and they have activity rates of 8, 6, and 3 messages/cycle, representing high activity, medium activity, and low activity, respectively. The activity rate of a DRSM controller D is the sum of the activity rates of the two rails and one machine that are connected to D , while the simple controller is only responsible for the last rail.

Considering the different volumes of machines and different lengths of conveyors that are typically found in practice, the spacing between actors is nonuniform in our experiments.

We apply the same channel configuration across the entire system model. Unless otherwise stated, the activity rate and placed location of each actor is as listed in Table 1, and other aspects of the simulation setup are as listed in Table 2.

Regarding the threshold for signal detection, two similar values are used in related literature: -65 dBm [13], and -70 dBm [14]. We used the value of -65 dBm to account for the severe multipath fading typical in industrial environments.

Table 1. Communication activity rate and position for each actor. The units for the activity rate are messages/cycle.

Actor	R_1	M_1	R_2	M_2	R_3	M_3
Activity	3	8	5	6	5	3
Position	(0, 0)	(0, 2)	(0, 5)	(0, 7)	(0, 9)	(0, 15)
Actor	R_4	C_1	C_2	C_3	SC_1	
Activity	5	14	11	8	2	
Position	(0, 18)	(1, 2)	(1, 6)	(1, 14)	(1, 17)	

Table 2. Other simulation parameters.

Parameter	Value
Number of GA generations	1000
Population size	300
Crossover rate	0.5
Mutation rate	0.2
Tournament selection size	10
Number of bits in each variable	6
Maximum transmit power of AP (α_{max})	17 dBm
Minimum transmit power of AP (α_{min})	0 dBm
Path loss exponent (γ)	3
Reference distance (d)	1 m
Threshold for signal detection (α_0)	-65 dBm

Two scenarios are intensively considered in our experiments: (1) all devices are cable-connected to their power supplies, and (2) all devices are powered by batteries.

5.1 Devices with Cable-Connected Power Supplies

In this simulation, the goal of our deployment strategy is to minimize the communication energy consumption of the network. The position of each actor is listed in Table 1. The actors are non-uniformly distributed in this layout.

First, we do not take into account the non-uniform distribution of actors, nor do we take into account unbalanced communication activity, and we place the AP at the center position (0.50, 10.00) of the pipeline. This center position is a simple and intuitive choice if we do not take into account non-uniform distribution and unbalanced communication, as described above. We execute the simulator for 1000 cycles and record the obtained energy consumption.

Next, we execute AP2GA to take into account the non-uniform actor distribution and unbalanced communication activity, and derive an optimized position for a single AP. The resulting AP position is (0.70, 7.90). We move the AP to this position in our simulation model, and again execute 1000 simulation cycles. We compare the energy consumption brought by (1) deploying the AP in the center position (“middle”), and (2) deploying the AP based on the result produced by AP2GA. The results are shown in Fig. 4.

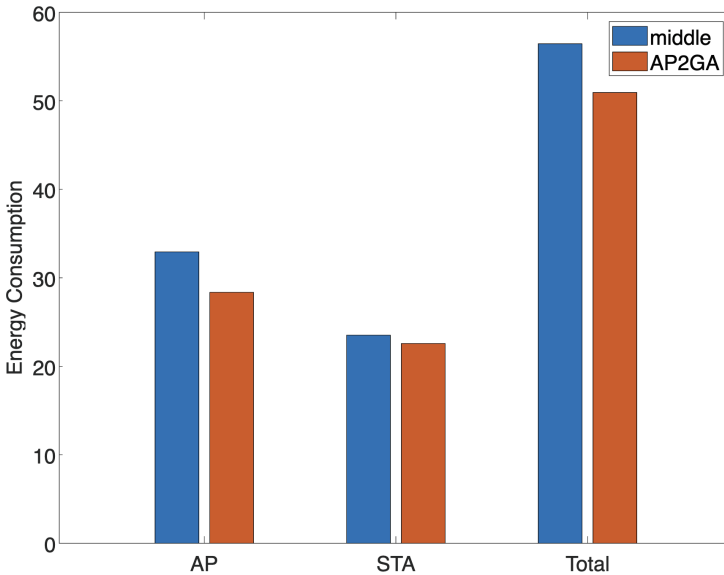


Fig. 4. Energy consumption in different locations.

In Fig. 4, the total energy consumption is the summation of the energy consumed by both the AP and the STAs. The AP column represents the overall downlink energy consumption, while the STAs column represents the overall uplink energy consumption. Since the maximum/minimum allowed transmit power and available power levels are all set to be the same for every communication node in the simulation model, the differences in the energy consumption between the downlink and uplink come from the uneven inflow and outflow the actors.

It can be clearly seen from Fig. 4 that even for this relatively simple and small-scale example, AP2GA results in a significant reduction in total energy consumption compared to the simple/intuitive strategy of placing the AP at the center position. The relative energy savings provided by AP2GA is about 10%, which will amount to significant absolute energy savings over long-term operation. Since the uneven outgoing and incoming traffic makes the downlink bear more long-distance workload, there is more significant reduction in the energy consumed by the AP. For example, when *Rail 2* sends packets to *DRSM Controller 1*, the downlink transmission distance (from the AP to the controller) is longer than the uplink transmission distance (from the rail to the AP).

5.2 Battery-Powered Devices

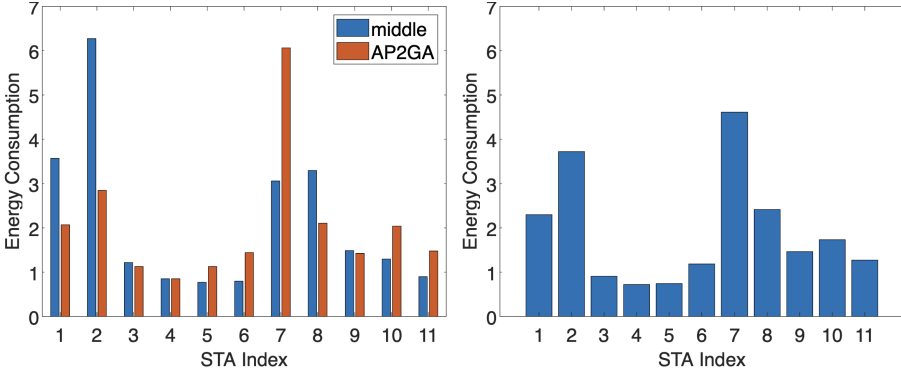
In Sect. 5.1, we optimized the total energy consumption supposing that all of the STAs and the AP are cable connected to power supplies. However, due to their low-price and easy installation, an increasing proportion of communication devices in industrial environments are powered by batteries.

When battery-powered communication devices are employed, it is important to consider the network durability when designing and configuring the network. Intuitively, by network durability, we mean the length of time that the network remains operational as batteries in the communication devices are drained. There are various ways to measure the network durability depending on the particular kinds of operational scenarios that are of interest. Since our scenario requires the mutual work of all devices in the network, we use a measure of network lifetime to assess durability, and we regard the time until one STA's battery is drained as the network lifetime. That is, the network lifetime is the time from the beginning of operation until the time when the first STA stops operating due to insufficient energy availability.

Assuming that all devices have the same battery capacity, maintaining a long network lifetime requires that all devices consume energy at approximately the same average rate — that is, the variance of energy consumption across the battery-powered devices should be low. To assess energy consumption variance, we plotted the energy consumption of each STA under both the center-position and AP2GA-based AP deployment obtained from case 1 in Fig. 5a, and tabulated their corresponding standard deviations (“Std.”) values in Table 3.

Table 3. Standard deviation of STA energy consumption.

Label	center pos.	AP2GA pos. 1 with cables	AP2GA pos. 2 with battery
AP Position	(0.50, 10.00)	(0.70, 7.90)	(-0.16, 8.57)
Std.	1.74	1.38	1.26



(a) In cable-connected configuration and (b) In AP2GA position 2 with constraint AP2GA position 1. C5 (Equation 3).

Fig. 5. Energy consumption levels of the different STAs under different AP deployment configurations.

From Fig. 5a, we can see that the energy consumption of the STAs is unbalanced in both deployment scenarios — center-position and AP2GA-based. Peaks appear on different devices depending on the combination of communication distance and activity rate. However, the distribution of STA energy consumption under AP2GA-based deployment from case 1 has better performance in terms of standard deviation.

To prolong network lifetime and ameliorate the imbalance described above, the dispersion of STA energy consumption can be taken into account in AP2GA. For this purpose, a maximum value for the standard deviation std_{max} can be imposed as another constraint:

$$C5 : \sqrt{\frac{\sum_{j=1}^s (e_j - \mu)^2}{s}} \leq std_{max}, \text{ where } e_j = \alpha_{i,j} t_{i,j} \beta_j, \mu = \frac{\sum_{j=1}^s e_j}{s}. \quad (3)$$

Moreover, when optimizing deployment for battery-powered devices, the activity rates used in the AP2GA are changed to only include the outgoing traffic for each device (i.e. $\beta_{i,j} \rightarrow \beta_j$). In our formulation, the updated fitness function measures the total transmission energy consumed by all STAs in the network, rather than the combination for all the STAs together with the AP, which was assumed in Sect. 5.1.

Through simulation experiments, we empirically determined that for our deployment case study, an effective maximum standard deviation value — for use in Eq. 3 — is $std_{max} = 1.3$. We executed AP2GA to find optimized deployment positions for this value of the maximum standard deviation. Then for the resulting deployment, we ran a simulation for 1000 cycles and plotted the energy consumption, as shown in Fig. 5b. In comparison with Fig. 5a, we can see that the results in Fig. 5b are more concentrated and the peak value has decreased. The standard deviation of 1.26, which results from imposing $std_{max} = 1.3$, represents a significant improvement compared to 1.74, which is the standard deviation measured from center-position deployment.

AP2GA can be applied in or extended for a wide variety of design space exploration scenarios to incorporate different combinations of decisions that are involved in deploying communication devices. For example, in our experiments, we assumed that the STAs have identical battery capacities. This condition can be relaxed to explore design spaces where batteries of different types are considered — ranging from small and less costly low-capacity batteries to large and more costly high-capacity batteries. The AP2GA fitness function may be extended in such a case to consider the cost of the deployment as well as the energy consumption, while taking into account the different available battery types. A candidate network configuration C would then include an assignment of battery types to the STAs. Various candidate configurations C_1, C_2, \dots, C_n can be optimized using AP2GA and evaluated through simulation to determine a single configuration to select among those that are evaluated. Such extension of AP2GA to assist with more general or comprehensive design space exploration is an interesting direction for future work.

5.3 Summary

In summary, from the study and experimental results presented in this paper, two main findings and implications emerge. First, in environments where users are unevenly distributed and their communication traffic varies, proper deployment of APs can significantly reduce the transmission energy consumption of the entire network. Second, the original formulation of AP2GA can be readily extended to other energy-related scenarios by manipulating selected parameters and introducing additional constraints. Averaging the transmission energy of battery-powered devices is an example, and there are many additional possibilities for performing other types of design space exploration.

6 Conclusion

In this paper, we have introduced an energy-efficient AP deployment strategy for industrial Internet of things (IIoT) environments. The developed strategy, which is based on a novel genetic algorithm called the Access Point Placement Genetic Algorithm (AP2GA), optimizes energy consumption in an environment with uneven distribution of communication stations that can have varying levels

of communication traffic. Simulation results involving two factory process flow scenarios demonstrate the effectiveness of the AP2GA approach in improving the energy efficiency of AP deployments. For environments in which stations have cable-connected power supplies, we demonstrate the use of AP2GA in optimizing total energy consumption, while in environments that involve battery power, we demonstrate the use of AP2GA in maximizing the operational network lifetime. A current limitation of AP2GA is that the algorithm assumes a single communication channel configuration, which is used uniformly in the modeled industrial environment. Interesting directions for future work include incorporating diverse channel configurations, and also extending the approach to consider additional metrics, such as communication throughput and deployment cost.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process. During the preparation of this work the authors used ChatGPT in order to correct possible grammatical errors and improve the readability of the paper. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Disclaimer. Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

1. Back, T., Hammel, U., Schwefel, H.: Evolutionary computation: comments on the history and current state. *IEEE Trans. Evol. Comput.* **1**(1), 3–17 (1997)
2. Björnson, E., Larsson, E.G.: How energy-efficient can a wireless communication system become?. In: 2018 52nd Asilomar Conference on Signals, Systems, and Computers (2018)
3. Candell, R., Kashef, M., Liu, Y., Lee, K.B., Fofou, S.: Industrial wireless systems guidelines: practical considerations and deployment life cycle. *IEEE Ind. Electron. Mag.* **12**(4), 6–17 (2018)
4. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagn'e, C.: DEAP: evolutionary algorithms made easy. *J. Mach. Learn. Res.* **13**, 2171–2175 (2012)
5. Geng, J., et al.: Model-based cosimulation for industrial wireless networks. In: Proceedings of the IEEE International Workshop on Factory Communication Systems, Imperia, Italy, pp. 1–10 (June 2018)
6. Jaffres-Runser, K., Gorce, J., Ubeda, S.: QoS constrained wireless LAN optimization within a multiobjective framework. *IEEE Wirel. Commun.* **13**(6), 26–33 (2006)
7. Katoch, S., Chauhan, S.S., Kumar, V.: A review on genetic algorithm: Past, present, and future. *Multimed Tools Appl.* **80**(5), 8091–8126 (2021)
8. Kouhbor, S., Ugon, J., Rubinov, A., Kruger, A., Mammadov, M.: Coverage in WLAN with minimum number of access points. In: 2006 IEEE 63rd Vehicular Technology Conference (2006)

9. Li, H., Geng, J., Liu, Y., Kashef, M., Candell, R., Bhattacharyya, S.: Design space exploration for wireless-integrated factory automation systems. In: Proceedings of the IEEE International Workshop on Factory Communication Systems, Sundsvall, Sweden (May 2019), 8 pages in online proceedings
10. Ling, X., Yeung, K.L.: Joint access point placement and channel assignment for 802.11 wireless LANs. *IEEE Trans. Wireless Commun.* **5**(10), 2705–2711 (2006)
11. Project, n.: The network simulator ns-3 (2016). <https://www.nsnam.org/> (Accessed 20 October 2016)
12. Tang, S., Ma, L., Xu, Y.: A novel AP placement algorithm based on user distribution for indoor WLAN system. *China Commun.* **13**(10), 108–118 (2016)
13. Zhang, Z., Di, X., Tian, J., Zhu, Z.: A multi-objective WLAN planning method. In: 2017 International Conference on Information Networking (ICOIN) (2017)
14. Zhi, Z., Wu, J., Meng, X., Yao, M., Hu, Q., Tang, Z.: AP deployment optimization in non-uniform service areas: a genetic algorithm approach. In: 2019 IEEE 90th Vehicular Technology Conference (VTC 2019-Fall) (2019)

Hardware/Software Solutions for IoT and CPS (HSS)



FAMID: False Alarms Mitigation in IoMT Devices

Shakil Mahmud^(✉), Myles Keller, Samir Ahmed, and Robert Karam

University of South Florida, Tampa, FL 33620, USA
shakilmahmud@usf.edu

Abstract. Wearable and Implantable Medical Devices (WIMDs) and Physiological Closed-loop Control Systems (PCLCS) are crucial elements in the advancing field of the Internet of Medical Things (IoMT). Enhancing the safety and reliability of these devices is of utmost importance as they play a significant role in improving the lives of millions of people every year. Medical devices typically have an alert system that can safeguard patients, facilitate rapid emergency response, and be customized to individual patient needs. However, false alarms are a significant challenge to the alert mechanism system, resulting in adverse outcomes such as alarm fatigue, patient distress, treatment disruptions, and increased healthcare costs. Therefore, reducing false alarms in medical devices is crucial to promoting improved patient care. In this study, we investigate the security vulnerabilities posed by WIMDs and PCLCS and the problem of false alarms in closed-loop medical control systems. We propose an implementation-level redundancy technique that can mitigate false alarms in real-time. Our approach, *FAMID*, utilizes a cloud-based control algorithm implementation capable of accurately detecting and mitigating false alarms. We validate the effectiveness of our proposed approach by conducting experiments on a blood glucose dataset. With our proposed technique, *all* the false alarms were detected and mitigated so that the device didn't trigger any false alarms.

Keywords: alert system · false alarm · internet of medical things (IoMT) · physiological closed-loop control systems (PCLCS)

1 Introduction

The Internet of Medical Things (IoMT) refers to a group of medical devices and software programs interconnected via computer networks that supports healthcare information technology systems to collect and exchange data. By 2030, the global IoMT market is projected to reach USD 861.3 billion, expecting a compound annual growth rate of 16.8% from 2023 to 2030 [41]. Physiological closed-loop control systems (PCLCS), smart medical devices, wearable and implantable medical devices (WIMDs), remote patient monitoring systems, and telemedicine platforms are some examples of IoMT devices. These devices can gather health-related information, including vital signs, medication dosage, blood glucose levels, drug concentration, etc. Real-time transmission of this data to healthcare

doctors and other systems makes it possible to monitor patients from a distance, make more precise diagnoses, and develop better treatment strategies [47]. IoMT devices have a wide range of security issues, such as vulnerabilities in software and firmware, physical attacks, weak encryption/authentication, and a lack of security patches or updates [35]. Additionally, system *reliability* is paramount so that healthcare practitioners can not only access the data when required but also trust that data is accurate, so patients can receive safe and effective care. The addition of autonomous, closed-loop treatment adds additional complexity to IoMT devices. Any vulnerabilities or potential reliability issues become even more serious in this context. A fault in any system component can cause the device to malfunction. A faulty device could result in inaccurate diagnoses, improper treatment, or even patient injury if the device does not perform as intended. Besides, IoMT devices must be able to communicate with other healthcare systems to provide continuity of treatment. The delivery of patient care may be delayed or fail altogether if the devices are unstable or produce inaccurate data. Additionally, regular maintenance, including updates and patching of a defective IoMT device can be expensive. Therefore, a PCLCS must be resistant to failure in order to ensure long-term in-field safety and reliability. The overall IoMT architecture is illustrated in Fig. 1, which shows the various scenarios involving WIMDs and PCLCS. For example, a person can have WIMDs that are IoMT-enabled but not part of a PCLCS. Alternatively, a WIMD may be utilized in a PCLCS that is not IoMT-enabled. Additionally, a WIMD can be part of both a PCLCS and an IoMT.

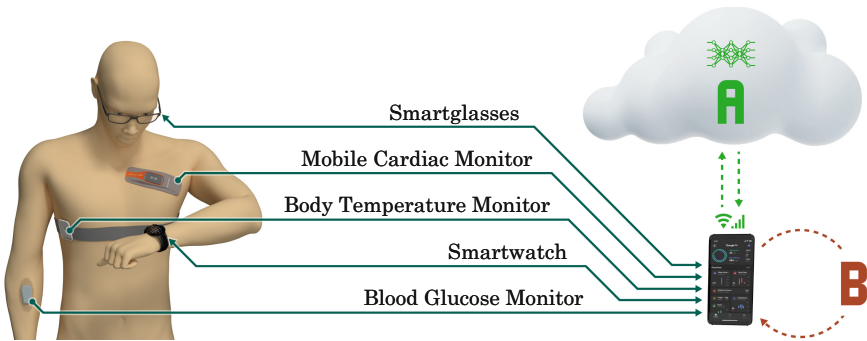


Fig. 1. Overall IoMT architecture, illustrating different WIMDs and cloud-based (A) vs. local processing (B) for closed-loop systems.

This paper focuses on IoMT devices that specifically include a PCLCS. In particular, we seek to leverage the connected nature of the device to help enhance the safety and reliability of the PCLCS through false alarm mitigation. The main components of a PCLCS are (1) a biosensor to measure physiologic variables from the patient, (2) a controller/control algorithm to determine the automatic actions, and (3) an actuator to perform the intended action, like delivering a

drug or therapy [13]. System safety features are crucial for any PCLCS design to enhance the safety and reliability of the device. For example, an alert system can help the patient or the healthcare provider in a variety of ways, such as indicating over- or under-medication, low battery, or any potential issue with the device that needs to be addressed. While alarms can help to take prompt action, false alarms can make it difficult for patients to distinguish between genuine and fake incidents. In addition, false alarms impact the battery life and the overall functionality of the device by unnecessarily consuming additional power. *Detecting and mitigating false alarms in real-time is essential for ensuring the reliability of the device and improving the overall effectiveness of the system.*

Several studies in the literature have explored various methods for identifying and addressing false alarms in different fields, such as cyber-physical systems [26] and medical systems or units [19, 53]. However, there has been a lack of research on detecting and correcting false alarms in closed-loop medical control systems. This paper aims to address that research gap by exploring the following **research questions**:

- RQ-1:** How to effectively identify false alarms in PCLCS that may occur due to natural failures or malicious modifications?
- RQ-2:** How can we efficiently mitigate false alarms in PCLCS to ensure the safety and reliability of the device?

In this paper, we assume the controller or the control algorithm to be defective, with the defects either being natural faults due to manufacturing errors or intentional malicious modifications, i.e., hardware Trojans [7]. A defective controller output signal can generate false alarms, and we propose a technique that can effectively detect and mitigate those alarms in real-time. We utilize the concept of implementation-level redundancy, where the redundant nodes are cloud-based PCLCS controllers. In particular, we implement a WebSocket API that facilitates communication between the local and cloud-based controllers. We then compare the local and cloud-based controller outputs to detect and mitigate false alarms. To test our approach, we model an artificial pancreas system (APS) and utilize a CGM dataset [4], which contains information about meal-times, insulin use, blood glucose measurements, and other factors for 30 patients, aged 18 to 66, with type-1 diabetes. A modified local control algorithm simulates a faulty device that triggers false alarms in order to evaluate the effectiveness of our technique. In summary, we make the following novel contributions:

1. We model different types of faults or attacks that can potentially disrupt the alert mechanism system of a PCLCS and cause it to generate false alarms.
2. We propose an implementation-level redundancy approach for the PCLCS controller that utilizes cloud-based parallel controllers to detect false alarms in real-time.
3. We implement an efficient alarm mitigation technique, *FAMID*, that effectively reduces false alarms arising from erroneous controller outputs to decrease the number of false alarms.

The rest of the paper is organized as follows: In Sect. 2, we explore the security and reliability concerns that arise from the use of IoMT devices with PCLCS components, describe the alert mechanism system and provide an overview of the specific (APS) control algorithms. In Sect. 3, we discuss related works, emphasizing the various false alarm mitigation techniques found in the literature and how they relate to IoMT devices. In Sect. 4, we delve into the details of the threat model. We then present our methodology in detail in Sect. 5. Section 6 presents and discusses the findings. Finally, we conclude in Sect. 7.

2 Background

In this section, we provide an in-depth background on the safety and security concerns associated with IoMT devices, specifically WIMDs, and PCLCS. We also elaborate on the alert mechanism system and control algorithms used in the PCLCS.

2.1 Safety and Reliability of IoMT Devices

The complexity of the software and hardware components of WIMDs is increasing as the healthcare domain transitions in response to technological and therapeutic advances. WIMDs have seen widespread use in recent years, making it critical to address the security and reliability concerns associated with these devices. Vulnerabilities in WIMDs can be exploited by attackers, granting them access to sensitive patient information and the ability to manipulate the device's normal operation [32]. WIMDs are vulnerable to various types of attacks, including hardware attacks like Hardware Trojans, software attacks such as malware and counterfeit firmware, communication channel attacks like denial-of-service and man-in-the-middle, and side-channel attacks like power analysis and electromagnetic interference. Table 1 illustrates a summary of these attacks, which can lead to data modification, device malfunctions, and other serious consequences [28]. Therefore, to ensure the protection of patient data and the proper functioning of WIMDs, it is imperative to prioritize the security and reliability of these devices. The increased use of embedded and customized software in WIMDs grows the attack surface, and various software and hardware defects have been discovered that can lead to different types of attacks [43]. For example, a WIMD with malware can malfunction intermittently, operate slowly, or even become unusable. Another significant security risk is updating WIMDs with counterfeit firmware, as attackers can access the system and change the programs. For example, the authors in [15] analyzed an automated external defibrillator (AED) and identified a set of vulnerabilities, including buffer overflow, password protection defects, and flaws in the software update mechanism, which make the device vulnerable to counterfeit firmware. An unverified firmware allows for a man-in-the-middle attack, and an example of updating an unverified firmware of a home monitoring device connected to an ICD is presented in [42]. The security vulnerabilities of wearable devices loaded with sensors, including firmware reverse engineering or

Table 1. Survey: Attacks on WIMDs

Category	Types	Components	Security	Examples
Comm. Channel	DoS	Network	A	AED [36], any IMD [18]
	Eavesdropping	Network	I, C	BCI [25], Insulin pump [6]
	Man-in-the-middle	Network	I, C	ICD [30]
	Ransomware	Data, HCP	I, C	Healthcare facilities [46]
	Replay	Network	I, C	ICD [14,30]
	Unauthorized Access	Network	I, C	Insulin pump [17], AED [36]
Hardware	EMI	Sensor	A	ICEDs [24]
	Hardware Trojan	Sensor	I	Medical microchips [49]
Software/Firmware	Battery Depletion	Device	A	ICEDs [39]
	Counterfeit Firmware	Device, Data	I, A	AED [15], ICD [42]
	Malware	Device, Data, HCP	I, A	WIMDs [43]
	Sensor Spoofing	Sensor	A	Insulin pump [17,30]

Notes: Security concerns → Availability (A), Confidentiality (C), Integrity (I); EMI → Electromagnetic Interference; HCP → Health care provider; AED → Automated external defibrillator; BCI → Brain computer interface; ICD → Implantable cardioverter defibrillator; ICED → Implantable cardiac electrical devices

compromising software gateways to extract and manipulate private user information, have also been studied [11,23]. Extensive research has been conducted in the literature to investigate the security concerns associated with WIMDs. The findings suggest that attackers can exploit wireless communication vulnerabilities to compromise these devices, both actively and passively. Consequently, critical security aspects such as authentication, confidentiality, integrity, availability, and authorization may be compromised. While some efforts have been made to enhance the security of the communication channel, such as implementing external proxies [50], biometric access control [16], and proximity-based security [40], the challenge of preserving device reliability in the presence of defective components in a closed-loop system still needs to be addressed.

The security and reliability aspects of PCLCS have been well-studied in the literature. For example, the authors in [2] investigated the impact of replay attacks to analyze the behavior of the APS under two well-known control algorithms: Proportional-Derivative (PD) and Multi-Basal (MB) control using simulation and model checking for security analysis. The same authors also investigated the impact of closed-loop anesthesia control under temporal sensor faults and reported only how the performance of the controllers was impacted because of the faulty sensor [3]. Furthermore, the authors in [38] observed that patients might be at serious risk if attackers get unauthorized access to a deep brain stimulation device. Blind and targeted attacks were on their list of potential ways to harm the patients. Examples of blind attacks included stopping stimulation, draining battery power, causing tissue injury, and collecting data. In contrast, targeted attacks consisted of interfering with motor control, disrupting impulse control, manipulating emotions, imposing pain, and manipulating the reward system. All of these studies have highlighted the impact of different attacks on PCLCS. However, none of them presented any solutions or strategies to ensure the reliability of these systems.

2.2 Alert Mechanism System

Alert mechanism systems can be used in various domains to ensure the safety of individuals. For example, the authors in [37] propose an SMS-based alert system for detecting car accidents and notifying emergency services. The SMS-based alert system in this approach takes advantage of the widespread use of mobile devices to provide timely notifications to first responders. When the system detects an accident, it uses GPS technology to pinpoint the location and sends a text message to rescue services, providing them with the information they need to dispatch assistance to the scene of the accident. The app-based alert system in Cerberus leverages the widespread use of mobile devices to provide timely notifications to users [12]. The mobile app can be easily installed on the user's device and communicate with the cloud server to receive and process alerts. The app provides information about the alert's location, type, and severity, allowing users to take appropriate action to stay safe. For example, if there is a flood alert, the app will notify the user of the unsafe water level at a particular location and provide guidance on avoiding the area. Similarly, if there is a fire alert, the app will provide information on the location and severity of the fire and recommend evacuation routes. A device equipped with both visual and audio alert mechanisms was created to ensure that miners are alerted in real-time of hazardous conditions [33].

In medical devices, alert mechanism systems are used to alert healthcare providers of potential issues or risks related to a patient's medical condition or the functionality of a medical device [22]. For example, an alert mechanism system in a patient monitoring system may alert healthcare providers if a patient's vital signs indicate a potential medical emergency. In addition, medical device manufacturers must implement alert mechanism systems to comply with regulatory requirements and ensure patient safety [21]. Alert mechanisms are essential tools in various industries to enhance safety measures for workers and individuals. These mechanisms can come in various forms, such as mobile apps, wearable devices, or SMS-based systems, and can provide timely and crucial notifications about potential hazards or emergencies. With the increasing importance of safety measures, alert mechanism systems are becoming more widespread and are now an integral part of regulatory requirements in many industries, including healthcare. Furthermore, the use of alert mechanisms is expected to grow as new technologies and innovations emerge, making workplaces and public spaces safer for all.

2.3 Control Algorithms

The control algorithm determines the automatic operations of the PCLCS. The primary purpose of the control algorithm is to apply modifications to the given medicine or therapy to ensure that the PCLCS satisfies clinically related performance requirements [13]. Characterizing the physiologic variable's response and the interaction of any elements that could impact these processes should be the basis for the control algorithms employed in a PCLCS. For example, in

an automated blood glucose system, the blood glucose measurement and the meals taken need to be considered by the controller to maintain the patient's insulin level safely. To meet Food and Drug Administration (FDA) recommendations, control algorithms should be designed to operate under potential risks and environmental interference.

In this paper, as a case study, we used an APS as a closed-loop system, which includes a continuous glucose monitor (CGM). Computer simulation has accelerated the development of AP control algorithms. For example, the authors in [29] developed the UVA/PADOVA Type 1 diabetes simulator, which can mimic meal challenges in virtual subjects, represented by a model parameter vector that was randomly selected from a suitable joint parameter distribution. The control algorithm implemented consists of a I_{bo} (bolus insulin) calculator as described by Eqn 1,

$$I_{bo} = \frac{CHO}{CR} + \frac{(G_p - G_t)}{CF} \quad (1)$$

with parameters G_p (amount of glucose in plasma), G_t (patient target glucose), CF (patient correction factor), CR (carbohydrate ratio), and CHO (ingested carbohydrate). The controller reads a patient's glucose level and carbohydrate amount from the dataset [4] stored in a .csv file for 10 patients with 200 samples each. The algorithm examines the patient's glucose readings and identifies any values below 70 or above 180, known as threshold alert values, as true alarms [5]. Furthermore, the algorithm computes the bolus amount by considering the patient's target glucose value and other relevant parameters.

3 Related Work

Detecting false alarms can help lower emotional or mental anxiety, reduce unnecessary computation power of the system, and ensure safety and reliability. Many false alarms can increase the risk of poor responses from the system to actual emergencies, which can harm people or cause financial loss. The primary goal of false alarm detection is to ensure that appropriate actions are taken for authentic situations instead of false ones. Researchers from various disciplines have investigated several methods to detect and mitigate false alarms. For example, several scientific research has been conducted to detect false alarms in a variety of application domains like the internet of connected vehicles [1], marine environments [10], wind turbine [31], and medical cyber-physical systems [26]. However, as the WIMDs are evolving rapidly with sophisticated components, it is required to develop new techniques that can effectively detect and mitigate false alarms to enhance patient experiences.

The authors in [44] proposed an approach to identify sensor anomalies by analyzing the historical data collected from various biosensors to detect and reduce false alarms. Their methodology consists of four algorithms: the correlation coefficient for physiological data preprocessing, random forest for sensor value prediction, the dynamic threshold for error calculation, and the majority

voting for alarm trigger. They used a large real-time healthcare dataset to evaluate their methodology and found a high false alarm detection rate and a low false positive rate. Furthermore, in [53], the authors proposed a robust methodology to detect seizures for wearable devices and tested that with the CHB-MIT electroencephalogram (EEG) dataset. To note, EEG acquisition is a time-consuming and error-prone process, and many seizure detection techniques are associated with unacceptably high false-alarm rates. Compared to the other studies on the same issue, their approach resulted in a 34.70% reduction in false alarms and demonstrated that their technique could extend the battery life of a cutting-edge wearable device by reducing the frequency of false alarms. Decreasing the number of false alarms is essential in intensive care Unit (ICU) to improve patient safety. In [52], the authors proposed a game-theoretic feature selection technique that uses a genetic algorithm to find the most useful biomarkers from signals obtained from various monitoring equipment. To reduce the false alarms in the ICU, the authors presented this low-computational complexity approach to estimate Shapley values for physiological signal characteristics. They reported that their proposed technique captured the actual alarms with better sensitivity and equivalent specificity compared to other feature selection methods and reduced the false alarms considerably.

In recent years, neural networks (NN), such as deep neural networks (DNNs), convolutional neural networks (CNNs), etc., have been used by researchers to detect false alarms by identifying patterns in the dataset. For example, the authors in [19] utilized the evolutionary and swarm algorithm to improve the DNN performance in detecting false alarms in ICU. They reported reduced suppressed true alarms by improving the accuracy compared to the benchmark Physionet challenge by utilizing dispersive flies optimization (DFO). In their study, 5-fold cross-validation was done using two models with different architectures. The results showed that compared to other results, including the benchmark, the DFO-modified version for both models performed better. Furthermore, the authors in [54] used CNNs to learn the feature representations of physiological waveforms to differentiate between authentic and false arrhythmia alarms. Their method utilizes contrastive learning to reduce binary cross-entropy classification and similarity loss. They tested their approach with a benchmark, the 2015 PhysioNet Computing in Cardiology Challenge, and reported that their proposed deep-learning framework performed better than the benchmark challenge.

Although various methods have been developed to detect and reduce false alarms in multiple fields, there is a clear research gap in utilizing false alarm detection and reduction techniques in closed-loop medical control systems. This paper aims to fill this gap by introducing an innovative and effective false alarm mitigation technique for PCLCS. A fault-tolerant system is required to ensure reliability and effective treatment. One potential technique for fault tolerance in control systems is hardware redundancy, where identical hardware modules perform the same functions. In case of a fault in one module, the other modules continue the process, maintaining the system's functionality. The triple modular redundancy (TMR) approach is frequently used to improve hardware fault

tolerance. Our proposed approach benefits from the TMR method, particularly N-version programming, where the same basic requirements are used to build independent versions of many functionally identical programs. IoMT is then leveraged to support cloud-based redundancy by offloading multiple versions or implementations of the control algorithm in a trusted cloud environment. Diversifying the alert mechanism in this way can help to improve reliability against both natural faults and intentional attacks.

4 Threat Model

Recent healthcare advancements have renewed interest in clinical automation and encouraged researchers to explore new techniques for physiological closed-loop control systems (PCLCS). In addition to delivering reliable and effective care while reducing the possibility of human error, PCLCS have the potential to improve medical support, particularly in emergency or overload situations. Consequently, it is crucial to consider patient safety when assessing the potential benefits of PCLCS. The authors in [48] explored the security threats and attacks and the various challenges associated with medical cyber-physical systems and closed-loop control systems. An example of a PCLCS is an APS, also known as automated closed-loop insulin delivery [20]. This system integrates a continuous glucose sensor, an insulin pump, and a control algorithm to regulate insulin delivery based on real-time measurements of blood glucose levels. A well-functioning APS can provide numerous benefits to patients. However, if the APS is defective, it may result in an underdose or overdose of insulin, which could pose a danger to the patient. In [8], the authors investigated the safety and design requirements of the APS, focusing on individual components or the system as a whole. While assessing the potential gains of PCLCS, the assurance of patient safety must be considered. Thus, it is a core requirement to ensure the reliable and effective operation of PCLCS [13].

The closed-loop medical control systems rely on a feedback mechanism to execute the intended operation automatically. Any external or internal disturbances to any system components can disrupt the regular operation of the device. For example, a faulty control algorithm of a PCLCS can lead to unintended alarm generation from the alert system, which can confuse the patient in distinguishing between authentic and false events and eventually increase their stress level. In addition to potentially affecting the patient's behavior, false alarms increase the device's power consumption, a critical problem for resource-constrained edge medical devices [34]. The design, implementation, and clinical translation of PCLCS are significantly impacted by crucial considerations such as area, power, and reliability [27]. Researchers have investigated security concerns related to IoMT devices, indicating how crucial it is to protect them from internal and external disruptions [9,51]. The threats with the PCLCS components can be both natural and intentional. The natural or unintended faults in the components can arise from device aging, high temperature, biosensor drift, accidental human error, etc. On the other hand, malicious threats could involve physically

tampering with the PCLCS or changing any system component. For example, the intentional malicious modification to any circuitry of an integrated circuit is known as a hardware Trojan, and if that is undetected during the manufacturing process, it can result in device malfunction [28]. From the perspective of IoMT devices, protecting against such intentional attacks is crucial as software security is sometimes inadequate [45].

PCLCS can malfunction as a result of faults in any of the system components. This paper focuses on a PCLCS with a faulty controller that occasionally generates incorrect actuating values leading to false alarms. We are assuming that: 1) the controller is defective and it can be a result of a natural fault or intentional malicious attack, and 2) the other components (biosensor, transducer, signal processing unit, analog to digital hardware, alert system, etc.) are error-free. Our goal is to design and implement an effective reliability mechanism that can reduce false alarms and maintain the safety and dependability of the PCLCS, as any incorrect reading from the defective controller can cause the device to malfunction and generate false alarms.

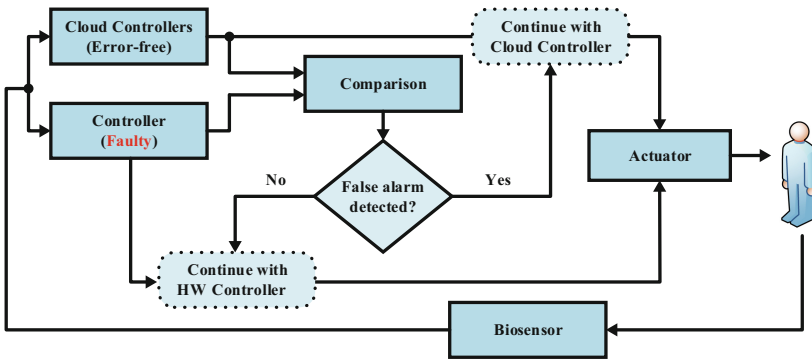


Fig. 2. An overview of our proposed approach illustrating how cloud controllers can help detect and mitigate false alarms in a PCLCS.

5 Methodology

This section presents a detailed description of the methodology and the experimental setup. We divide our approach into four main parts: 1) implementing cloud-based controllers, 2) implementing WebSocket API for two-way interactive communication, 3) generating false alarms to simulate accidental or intentional faults with the controller, and 4) detecting and mitigating the false alarms in real-time. An overview of our proposed approach is illustrated in Fig. 2.

5.1 Cloud-Based Controller

Our proposed approach is built upon the TMR method, a fault-tolerant technique commonly used in safety-critical systems. Specifically, we utilize the N-version programming aspect of TMR, where multiple independent versions of the same program are created based on the same requirements. Each version is designed to be functionally identical but implemented using different algorithms and/or programming languages. We implemented three versions of the same control algorithm, written in different programming languages, which include C++, Rust, and Python. These languages were chosen based on their memory efficiency and speed, with Python being used for the local physical device (Raspberry Pi) and C++ and Rust for their speed and memory efficiency. It is important to note that our approach to implementing cloud-based controllers is not limited to specific programming languages and is not limited to implementation diversity.

We analyzed the compiled code of the Rust and C++ algorithms and found significant differences in the number of instructions and clock cycles. Specifically, the Rust compiled algorithm had nearly 48% more instructions than the C++ version. The version of the Rust compiler used was `rustc v.1.69`, while the version of the C++ compiler used was `gcc v.17`. We wrote a script to convert the Python-interpreted code into C to find the differences between all three programs. We observed the execution time to be not significantly different among the three languages. However, the number of main memory access in the Rust version was the highest, followed by C++ and C. Therefore, it is important to consider the programming language and compiler version when implementing control algorithms, as these factors can significantly impact the resulting program's performance and efficiency.

5.2 WebSocket API

The system architecture of our WebSocket-based communication system consisted of a Raspberry Pi acting as a WebSocket client, a PC acting as a WebSocket server, and multiple other clients running in virtual machines. The Wi-Fi of the Raspberry Pi was configured to function in Ad-hoc mode, negating the need for a dedicated access point. The PC was running Windows 10 and hosted the WebSocket server which was written using Python. The virtual machines were running on the same PC and each had a unique IP address. The Raspberry Pi and all virtual machines were running the same client-side Python script to communicate with the WebSocket server. The WebSocket server was implemented using the `WebSocketServer` class, which allowed us to handle WebSocket connections and messages easily. The server was configured to listen on a specific port (e.g., 8000) and accept connections from any IP address. Once a connection was established, the server would maintain a persistent connection with all clients and handle incoming messages. The WebSocket client was implemented on the Raspberry Pi using the `WebSocket` client library. The client was configured to connect to the WebSocket server running on the PC using the IP address

of the PC and the port number specified during server configuration. Once a connection was established, the client awaited a predefined symbol to arrive in a WebSocket message to indicate that all clients on the network should run an iteration of the simulation. We also tested the system using multiple virtual machine clients running on the same PC as the WebSocket server. Each virtual machine was configured with a unique IP address and ran a simple Python script that connected to the WebSocket server as a client and printed incoming messages to the console. This allowed us to simulate multiple clients and test the scalability and resiliency of the system. We collected data by storing the returned result values received by the WebSocket server in a .csv file for later analysis. Figure 3 shows our experimental communication setup between cloud and local devices.

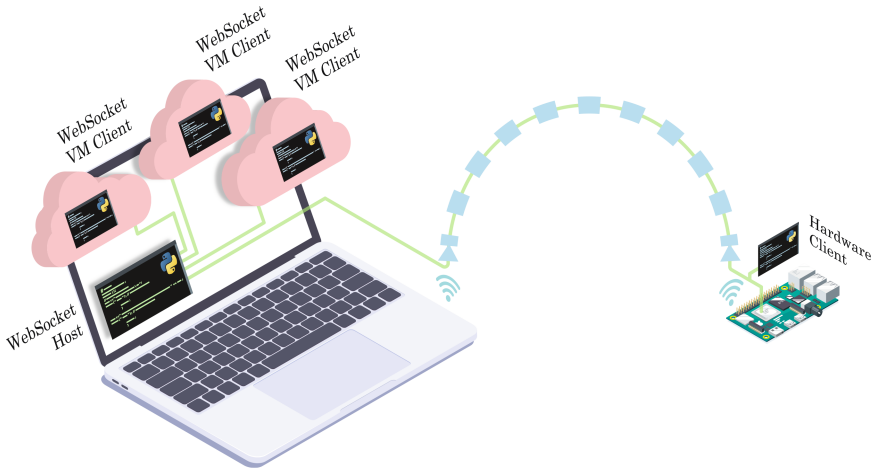


Fig. 3. Diagram of methods and technologies used in our experimental setup to communicate between cloud and local devices.

5.3 Generation of False Alarms

To validate our proposed approach, we conducted experiments by generating false alarms based on potential issues that could affect the controller in the PCLCS. We considered two scenarios: 1) controller malfunction and 2) intentional controller modification. The first scenario, a malfunctioning controller, could occur accidentally during or after the manufacturing process or due to communication issues. To simulate this scenario, we modeled the controller to randomly send a zero bolus value to the actuator in a non-sequential manner. In the second scenario, intentional controller modification, we introduced a single-bit error (SBE) into the calculated bolus measurements by flipping a random bit with a fault probability of 1%, which adds additional 10–15% false alarms to the total number of alarms. This type of attack could be aimed at disrupting the PCLCS. These scenarios and parameters were chosen to demonstrate that

our proposed approach is effective in detecting and mitigating false alarms generated by a faulty controller. It is important to note that the chosen scenarios and parameters are worse than what would typically occur in the real world. However, they are still comparable to real-world scenarios but with a higher probability of faults due to the limited dataset.

5.4 False Alarms Detection and Mitigation

To detect false alarms in the PCLCS, the output of the local controller is compared with the outputs of the cloud controllers, which are expected to produce identical values. If there is a mismatch, an alarm is triggered based on the alarm-triggering condition. In such a scenario, our technique can detect the false alarm and prevent it from triggering. Figure 2 illustrates how our approach compares

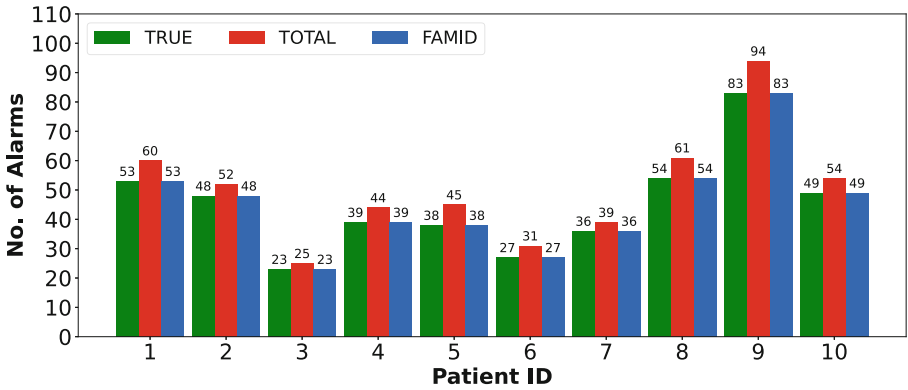


Fig. 4. Bar charts illustrating the true alarms, total alarms including false alarms, and the number of alarms after applying FAMID technique for random nonsequential dropped values (RNDV).

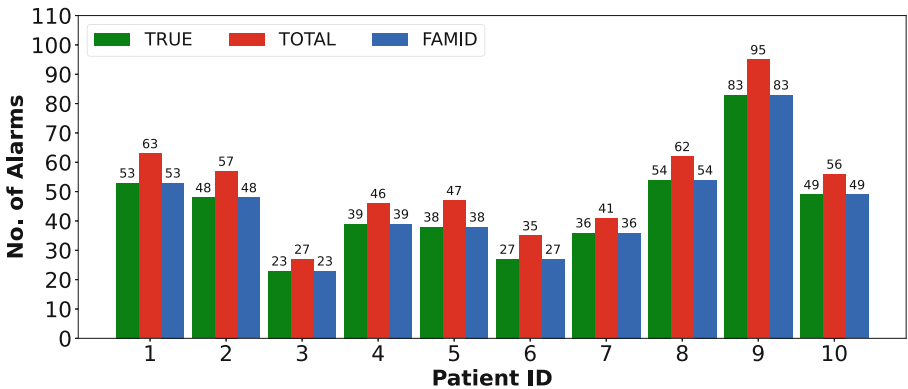


Fig. 5. Bar charts illustrating the true alarms, total alarms including false alarms, and the number of alarms after applying FAMID technique for single-bit error (SBE).

the output signals of the local and cloud controllers to detect false alarms and then chooses to continue operation with either the local or cloud controller. By detecting and mitigating false alarms generated by the local controller, the system's reliability is improved.

6 Results and Discussion

We present the results and discussions in this section divided into two main parts: 1) the accuracy of cloud-based controllers and 2) false alarm detection and mitigation using FAMID. We recorded the number of true alarms triggered by the alarm-triggering condition to observe the number of occurrences in the local and cloud controllers. The alarm-triggering condition for true alarms was set when the blood glucose value was greater than 180 or less than 70. For all samples in the test dataset, we confirmed that the cloud-based algorithms generated the same alarms as the unmodified local physical controller.

Our approach for detecting and mitigating false alarms addresses two critical issues with the local controller: single-bit errors (SBE) and random non-sequential dropped values (RNDV). To validate the effectiveness of our proposed approach, we conducted evaluations using data from ten different patients. In Fig. 4, we showcase the results of our technique in reducing the number of additional false alarms caused by RNDV in the controller's output signal. For instance, for patient 1, there were initially 53 true alarms and 7 false alarms, totaling 60 alarms before implementing our technique. However, after applying our proposed approach, the number of alarms was significantly reduced to 53, matching the number of true alarms. This significant reduction demonstrates our approach's high accuracy and efficiency in mitigating false alarms. Similarly, Fig. 5 presents compelling results when an SBE was introduced into the local controller's output values. Our proposed technique also successfully mitigated false alarms in this scenario, showcasing its versatility and robustness. To provide a comprehensive overview of our results, Table 2 summarizes each patient's true, false, and total alarms before and after implementing our FAMID technique. The data further confirms the effectiveness of our approach across various patient cases.

Our proposed technique has proved its effectiveness in substantially reducing false alarms caused by both RNDV and SBE, presenting a promising solution to enhance alarm accuracy and ensure patient safety in medical systems. The evaluation results clearly illustrate the potential advantages of implementing our technique in real-world applications, significantly contributing to creating a more reliable and efficient healthcare environment. By addressing false alarms, our approach can minimize unnecessary alerts and enable medical professionals to focus on critical cases promptly, improving patient outcomes and a more streamlined healthcare system. The positive results from this study highlight the importance of further integrating our approach in various medical devices and settings to optimize alarm management and overall patient care.

Table 2. Comparing Total Alarms (A), True Alarms, and False Alarms for RNDV and SBE using FAMID

Sub	True Alarms	False Alarms		Alarms		FAMID	
		SBE	RNDV	A1	A2	A1	A2
S-1	53	10	7	63	60	53	53
S-2	48	9	4	57	52	48	48
S-3	23	4	2	27	25	23	23
S-4	39	7	5	46	44	39	39
S-5	38	9	7	47	45	38	38
S-6	27	8	4	35	31	27	27
S-7	36	5	3	41	39	36	36
S-8	54	8	7	62	61	54	54
S-9	83	12	11	95	94	83	83
S-10	49	7	5	56	54	49	49
Total	450	79	55	529	505	450	450

7 Conclusion

In this paper, we have presented a comprehensive overview of the security and reliability challenges associated with WIMDs and PCLCS in the IoMT. These devices are globally significant in improving individual well-being, but their growing complexity, driven by hardware and software advancements, introduces crucial concerns regarding their safety and effectiveness. Securing WIMDs and PCLCS is essential to ensure patient safety and reliability. Several security issues have been identified in these devices, and false alarms in PCLCS are particularly concerning, as they can lead to alarm desensitization, alarm fatigue, and distress for the patient. The technique proposed in this work aims to mitigate false alarms in real-time using implementation-level redundancy by implementing cloud-based controllers to improve the reliability of the PCLCS. Our approach demonstrated a complete success rate in detecting and mitigating false alarms, thereby ensuring the reliability of the PCLCS. Future research on this work will focus on enhancing the robustness of our approach by integrating algorithmic-level diversity with implementation-level redundancy and testing the proposed technique on additional datasets to evaluate its effectiveness in different scenarios.

References

1. Al Zamil, M.G., et al.: False-alarm detection in the fog-based internet of connected vehicles. *IEEE Trans. Veh. Technol.* **68**(7), 7035–7044 (2019)
2. Alshalalfah, A.L., Hamad, G.B., Mohamed, O.A.: Towards system level security analysis of artificial pancreas via uppaal-smc. In: 2019 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5. IEEE (2019)

3. Alshalalfah, A.L., Hamad, G.B., Mohamed, O.A.: System-level analysis of closed-loop anesthesia control under temporal faults via uppaal-smc. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 2508–2511 (2020)
4. Anderson, S.M., et al.: Multinational home use of closed-loop control is safe and effective. *Diabetes Care* **39**(7), 1143–1150 (2016)
5. Atlas, E., Nimri, R., Miller, S., Grunberg, E.A., Phillip, M.: Md-logic artificial pancreas system: a pilot study in adults with type 1 diabetes. *Diabetes Care* **33**(5), 1072–1076 (2010)
6. Beardsley, T.: R7–2016-07: Multiple vulnerabilities in animas onetouch ping insulin pump. Rapid7 blog (2016)
7. Bhunia, S., Hsiao, M.S., Banga, M., Narasimhan, S.: Hardware trojan attacks: threat analysis and countermeasures. *Proc. IEEE* **102**(8), 1229–1247 (2014)
8. Blauw, H., Keith-Hynes, P., Koops, R., DeVries, J.H.: A review of safety and design requirements of the artificial pancreas. *Ann. Biomed. Eng.* **44**(11), 3158–3172 (2016)
9. Chacko, A., Hayajneh, T.: Security and privacy issues with iot in healthcare. *EAI Endorsed Trans. Pervasive Health Technol.* **4**(14) (2018)
10. Chen, X., Su, N., Huang, Y., Guan, J.: False-alarm-controllable radar detection for marine target based on multi features fusion via CNNs. *IEEE Sens. J.* **21**(7), 9099–9111 (2021)
11. Classen, J., Wegemer, D., Patras, P., Spink, T., Hollick, M.: Anatomy of a vulnerable fitness tracking system: dissecting the fitbit cloud, app, and firmware. *Proc. ACM Interact. Mobile Wearable Ubiquit. Technol.* **2**(1), 1–24 (2018)
12. Dasari, S.: Cerberus: a novel alerting system for flood, fire, and air quality. In: 2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), pp. 98–101 (2020)
13. FDA-Guidance: Technical considerations for medical devices with physiologic closed-loop control technology (2021). <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/technical-considerations-medical-devices-physiologic-closed-loop-control-technology>
14. Halperin, D., et al.: Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In: 2008 IEEE Symposium on Security and Privacy (sp 2008), pp. 129–142. IEEE (2008)
15. Hanna, S., Rolles, R., Molina-Markham, A., Poosankam, P., Fu, K., Song, D.: Take two software updates and see me in the morning: The case for software security evaluations of medical devices
16. Hei, X., Du, X.: Biometric-based two-level secure access control for implantable medical devices during emergencies. In: 2011 Proceedings IEEE INFOCOM, pp. 346–350. IEEE (2011)
17. Hei, X., Du, X., Lin, S., Lee, I., Sokolsky, O.: Patient infusion pattern based access control schemes for wireless insulin pump system. *IEEE Trans. Parallel Distrib. Syst.* **26**(11), 3108–3121 (2014)
18. Hei, X., Du, X., Wu, J., Hu, F.: Defending resource depletion attacks on implantable medical devices. In: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, pp. 1–5. IEEE (2010)
19. Hooman, O.M., Al-Rifaie, M.M., Nicolaou, M.A.: Deep neuroevolution: training deep neural networks for false alarm detection in intensive care units. In: 2018 26th European Signal Processing Conference (EUSIPCO), pp. 1157–1161. IEEE (2018)

20. Hovorka, R.: Closed-loop insulin delivery: from bench to clinical practice. *Nat. Rev. Endocrinol.* **7**(7), 385–395 (2011)
21. International Organization for Standardization: ISO 13485:2016 - Medical devices - Quality management systems - Requirements for regulatory purposes (2018)
22. Joshi, S., Joshi, S.: A sensor based secured health monitoring and alert technique using iomt. In: 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), pp. 152–156 (2019)
23. Kim, D., Park, S., Choi, K., Kim, Y.: BurnFit: analyzing and exploiting wearable devices. In: Kim, H., Choi, D. (eds.) WISA 2015. LNCS, vol. 9503, pp. 227–239. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31875-2_19
24. Kune, D.F., et al.: Ghost talk: mitigating emi signal injection attacks against analog sensors. In: 2013 IEEE Symposium on Security and Privacy, pp. 145–159. IEEE (2013)
25. Li, Q., Ding, D., Conti, M.: Brain-computer interface applications: security and privacy challenges. In: 2015 IEEE conference on communications and network security (CNS), 663–666. IEEE (2015)
26. Li, W., Meng, W., Su, C., Kwok, L.F.: Towards false alarm reduction using fuzzy if-then rules for medical cyber physical systems. *IEEE Access* **6**, 6530–6539 (2018)
27. Mahmud, S., Majerus, S.J., Damaser, M.S., Karam, R.: Design tradeoffs in bioimplantable devices: a case study with bladder pressure monitoring. In: 2018 IEEE 24th International Symposium on On-Line Testing and Robust System Design (IOLTS), pp. 69–72. IEEE (2018)
28. Mahmud, S., Zareen, F., Olney, B., Karam, R., et al.: Trojan resilience in implantable and wearable medical devices with virtual biosensing. In: 2022 IEEE 40th International Conference on Computer Design (ICCD), pp. 577–584. IEEE (2022)
29. Man, C.D., Micheletto, F., Lv, D., Breton, M., Kovatchev, B., Cobelli, C.: The uva/padova type 1 diabetes simulator: new features. *J. Diabetes Sci. Technol.* **8**(1), 26–34 (2014)
30. Marin, E., Singelée, D., Garcia, F.D., Chothia, T., Willems, R., Preneel, B.: On the (in) security of the latest generation implantable cardiac defibrillators and how to secure them. In: Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 226–236 (2016)
31. Marugán, A.P., Chacón, A.M.P., Márquez, F.P.G.: Reliability analysis of detecting false alarms that employ neural networks: a real case study on wind turbines. *Reliability Eng. Syst. Safety* **191**, 106574 (2019)
32. Newaz, A.I., Sikder, A.K., Rahman, M.A., Uluagac, A.S.: A survey on security and privacy issues in modern healthcare systems: attacks and defenses. *ACM Trans. Comput. Healthcare* **2**(3), 1–44 (2021)
33. Noorin, M., Suma, K.: Iot based wearable device using wsn technology for miners. In: 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 992–996 (2018)
34. Olney, B., Mahmud, S., Karam, R.: Evaluating edge processing requirements in next generation iot network architectures. In: Casaca, A., Katkooi, S., Ray, S., Strous, L. (eds.) IFIPIoT 2019. IAICT, vol. 574, pp. 252–269. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43605-6_15
35. Papaioannou, M., et al.: A survey on security threats and countermeasures in internet of medical things (iomt). *Trans. Emerging Telecommun. Technol.* **33**(6), e4049 (2022)

36. Papp, D., Ma, Z., Buttyan, L.: Embedded systems security: threats, vulnerabilities, and attack taxonomy. In: 2015 13th Annual Conference on Privacy, Security and Trust (PST), pp. 145–152. IEEE (2015)
37. Parveen, N., Ali, A., Ali, A.: Iot based automatic vehicle accident alert system. In: 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), pp. 330–333. (2020)
38. Pycroft, L., et al.: Brainjacking: implant security issues in invasive neuromodulation. *World Neurosurgery* **92**, 454–462 (2016)
39. Ransford, B., et al.: Cybersecurity and medical devices: a practical guide for cardiac electrophysiologists. *Pacing Clin. Electrophysiol.* **40**(8), 913–917 (2017)
40. Rasmussen, K.B., Castelluccia, C., Heydt-Benjamin, T.S., Capkun, S.: Proximity-based access control for implantable medical devices. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 410–419 (2009)
41. Research, G.V.: Internet of things in healthcare market worth \$861.3 billion by 2030 (March, 2023). <https://www.grandviewresearch.com/press-release/global-iot-in-healthcare-market>
42. Rios, B., Butts, J.: Security evaluation of the implantable cardiac device ecosystem architecture and implementation interdependencies. WhiteScope, sl (2017)
43. Ronquillo, J.G., Zuckerman, D.M.: Software-related recalls of health information technology and other medical devices: Implications for fda regulation of digital health. *Milbank Q.* **95**(3), 535–553 (2017)
44. Saraswathi, S., Suresh, G., Katiravan, J.: False alarm detection using dynamic threshold in medical wireless sensor networks. *Wireless Netw.* **27**, 925–937 (2021)
45. Sidhu, S., Mohd, B.J., Hayajneh, T.: Hardware security in IOT devices with emphasis on hardware trojans. *J. Sens. Actuator Netw.* **8**(3), 42 (2019)
46. Spence, N., Niharika Bhardwaj, M., Paul III, D.P.: Ransomware in healthcare facilities: a harbinger of the future? *Perspectives in Health Information Management*, pp. 1–22 (2018)
47. Sundaravadivel, P., Lee, I., Mohanty, S., Koungianos, E., Rachakonda, L.: Rm-iot: an iot based rapid medical response plan for smart cities. In: 2019 IEEE International symposium on smart electronic systems (iSES)(Formerly iNiS), 241–246. IEEE (2019)
48. Tyagi, A.K., Sreenath, N.: Cyber physical systems: analyses, challenges and possible solutions. *Internet of Things Cyber-Phys. Syst.* **1**, 22–33 (2021)
49. Wehbe, T., Mooney, V.J., Javaid, A.Q., Inan, O.T.: A novel physiological features-assisted architecture for rapidly distinguishing health problems from hardware trojan attacks and errors in medical devices. In: 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 106–109. IEEE (2017)
50. Xu, F., Qin, Z., Tan, C.C., Wang, B., Li, Q.: Imdguard: securing implantable medical devices with the external wearable guardian. In: 2011 Proceedings IEEE INFOCOM. IEEE, pp. 1862–1870 (2011)
51. Yang, Y., Wu, L., Yin, G., Li, L., Zhao, H.: A survey on security and privacy issues in internet-of-things. *IEEE Internet Of Things J.* **4**(5), 1250–1258 (2017)
52. Zaeri-Amirani, M., Afghah, F., Mousavi, S.: A feature selection method based on shapley value to false alarm reduction in icus a genetic-algorithm approach. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 319–323. IEEE (2018)

53. Zanetti, R., Aminifar, A., Atienza, D.: Robust epileptic seizure detection on wearable systems with reduced false-alarm rate. In: 2020 42nd annual international conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 4248–4251. IEEE (2020)
54. Zhou, Y., et al.: A contrastive learning approach for icu false arrhythmia alarm reduction. *Sci. Rep.* **12**(1), 4689 (2022)



Dynamic Task Allocation and Scheduling for Energy Saving in Edge Nodes for IoT Applications

Shubhangi K. Gawali^(✉) , Lucy J. Gudino , and Neena Goveas 

BITS Pilani Goa Campus, Vasco da Gama, India
{shubhangi,neena}@goa.bits-pilani.ac.in,
lucy.gudino@pilani.bits-pilani.ac.in
<https://www.bits-pilani.ac.in>

Abstract. Internet of Things with an Edge layer is a trending approach in areas such as healthcare, home, industry, and transportation. While scheduling the tasks of such applications, if the edge node utilizes its energy in computing latency-insensitive tasks then it might fail in executing the future latency-sensitive task due to low energy. Thus conserving the energy of the edge node is a key aspect to be considered while designing task allocation and scheduling policies. This can be done by exploiting the inactive state of the edge nodes which is due to less execution time taken than the predicted worst-case time. As this inactive node consumes energy, the best way is to utilize this energy by executing the other node's task or by transiting to the zero energy state like shutdown. Managing the inactive interval in such a way also reduces the number of idle intervals in the schedule and the overall idle duration of the edge server which effectively reduces energy. In a homogeneous multi-edge (HME) system, techniques like Dynamic Procrastination (DP) combined with migration can help the edge node qualify for the shutdown. Other nodes can be slowed down to execute the tasks with later deadlines using the dynamic voltage/frequency scaling (DVFS) technique to further save energy. Migration combined with DP and DVFS effectively results in improved system utilization and reduced overall energy without affecting performance. This introduces challenges like dynamic allocation of tasks to edge nodes and meeting deadlines. In this work, we propose a dynamic task allocation and scheduling approach for an HME system that can decide on slowing down or shutting down the edge node. We observe that by decreasing the number of idle intervals and increasing the duration of the inactive state, our approach gives improved results for energy consumption over state-of-the-art energy reduction techniques.

Keywords: IoT · task allocation · task scheduling · Multi-Edge · Dynamic Procrastination · Dynamic Voltage/Frequency Scaling · Job Migration · Energy efficient scheduling

1 Introduction

In IoT systems, along with timeliness, response time, and waiting delay, energy consumption is also a significant design consideration. The usage of an edge layer with multiple servers is now been proposed to help in satisfying these timing constraints. An Edge server can be in an active or inactive state consuming static and/or dynamic energy. Static energy is due to leakage current and dynamic energy is due to switching current. The inactive state duration and the energy consumed by the edge server during this period can be further reduced if it can be transited from an inactive state to a state that consumes an infinitesimally small amount of static energy. This can be done by shutting down the unutilized processing elements. To do this, the benefits achieved by transiting to a lower-energy state have to be more than the energy consumed in this decision-making, shutting down and waking up the processing elements, leading to a threshold time. Thus if the edge server remains inactive for a longer duration than the threshold, static energy consumption can be reduced, thus reducing the overall energy consumption. The inactive duration can be increased by postponing the less critical task as much as possible i.e. completing it just before its deadline. If the edge server remains idle for less than the threshold duration, it is better to share the workload from other edge servers and transit to an active state. By doing this, there is a possibility for other servers to remain in a lower-energy state for a long time.

In a multi-Edge (ME) system, there is a fragmentation of task occupancy of any Edge Server. One of the reasons is the variations in actual execution time (AET) which is usually less than or the same as the worst-case execution time (WCET). Due to this, the schedule has small inactive intervals. Merging the active intervals leads to a reduction in the number of active and inactive intervals which effectively produces longer inactive intervals. One of the techniques used is the Dynamic Procrastination technique, which achieves this merging without missing any deadlines [1,2]. For homogeneous multi-core systems, job migration and other techniques have been used to improve the performance [3]. The energy saving is proportional shutdown period achieved [4,5].

In this work, we propose an energy-saving technique for IoT applications with hard real-time periodic workload generation on a homogeneous Multi-Edge (HME) system connected with dedicated links having multi-mode energy levels. The scheduler produces a valid schedule with optimal energy per productive work by minimizing inactive intervals of all edge servers with the help of DP, DVFS, and controlled job migration.

The rest of the paper is organized as follows: Sect. 2 discusses the work done in the areas related to DP, DVFS, and migration techniques for energy saving. In Sect. 3 the proposed technique is explained. Section 4 details the experimental evaluation and results. Section 5 concludes the paper with future directions.

2 Related Work

Recent literature on hard real-time schedulers using various slowdown and shutdown techniques is analyzed in this section. We also explore these techniques when combined with job migration. Researchers address leakage current energy management i.e. saving the static energy while scheduling by considering temperature-dependent leakage on a processor with the help of accumulated execution slack while some modulate the core between active and sleep states by combining DPS and Real-Time Calculus for computing the core idle intervals [6–8]. To combine idle durations ES-RHS uses harmonization for uncore and MC systems [9]. In [10], the optimization goal is to minimize time delay, but power consumption is not considered. [11] explains Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices. [12] studied the tradeoff between time delay and energy consumption of mobile devices. Other approaches include DVSLK which merges the scattered idle intervals [13], Single Frequency Approximation (SFA) with procrastination [14], Critical Speed DVS with Procrastination (CS-DVS-P) [15], systems without constraints on maximum processor speed [16]. Here SFA and CS-DVS-P show good performance in all jobs execution with WCET. [17], studied the potential of dynamic procrastination with AET.

Job migrations have been used for load balancing, but can also exploit for improving response time and energy saving. In an ME system, in addition, to which Edge-Server to migrate the jobs other aspects to be considered are when and which jobs are to be considered for migration. In the ME system, the inactive/shutdown intervals being spread across multiple Edge servers makes optimal energy efficiency difficult. The push and pull migration strategies used by Linux can help in merging these intervals [18]. A polynomial-time scheduling algorithm for minimizing energy consumption with job migration was proposed by [16]. Energy savings in a multi-core system using the push-procrastinate-pull (Pcube) technique have been studied in [19]. Here we extend their Pcube technique further with Dynamic Voltage/Frequency scaling (DVFS) technique along with migration for optimal energy consumption. [20] used DVFS technique to develop a hierarchical scheduling algorithm called Slack Aware Frequency Level Allocator (SAFLA) for saving energy while scheduling multiple real-time periodic task graphs on heterogeneous systems.

Researchers working on energy-saving techniques in edge computing platforms try to balance the other parameters like system throughput, latency, network delay, edge server utilization, and task success rate along with energy parameters. We explored the literature that uses various strategies for energy savings in edge computing. One of the reasons for task failure in edge computing is less energy with the edge nodes to which the task was allocated. Thus task allocation is significant in edge computing. [21] proposed a hybrid MAC protocol-based adaptable sleep mode and demonstrated the effectiveness that improves the network throughput and enhances energy conservation. [22] proposed a framework called EASE for job scheduling and migration within the edge hosts with distributed renewable energy resources. EASE uses distributed con-

sensus step, to reach the migration agreement. [23] proposed online upload scheduler, named FReshness-aware Energy efficient ScHeduler (FRESH), to minimize the update energy consumption subject to information freshness constraints. [24] provide insights on the optimal look-ahead time for energy prediction. They formulated an energy-aware scheduler for battery-less IOT devices using the Mixed Integer Linear Program.

In addition to slowdown and shutdown techniques we look at optimal decisions on migration to reduce energy consumption. Our focus is not only on reducing the number of idle intervals but also on reducing the idle duration. This is achieved by migrating the upcoming jobs as their arrival times are known due to their periodic nature. In push migration, We call the edge server from where the job is pushed as a source node and to which it is pushed as a target node. Similarly, in pull migration, vice-versa nomenclature is used. Although push migration increases the active duration in the target edge node, it helps in increasing the idle duration in the source edge node. This can potentially extend to a duration suitable for the shutdown. Similarly, pull migration increases the utilization of the source edge node and increases inactive duration on the target edge node. Migration can thus help in increasing the inactive duration for both pull and push mechanisms. Whenever such migrations do not help in achieving the shutdown threshold, instead of remaining idle, the edge server executes the tasks at low frequencies thus saving dynamic idle energy. Our proposed technique results in the creation of a schedule combining the DP and DVFS with migration. We show that our proposed method reduces overall energy consumption. By optimally making use of affinity features of the ME system with multiple energy-state support, our proposed algorithm schedules the tasks on edge servers with optimal energy consumption.

3 Problem Statement

The optimization problem aims to minimize the overall energy consumption of the multi-edge system having multi-mode energy levels that supports migration. For the source edge server from where the jobs are migrated, maximize the shutdown duration and for the target edge server having insufficient duration for shutdown to which the jobs are migrated, minimize the idle duration. The idle duration is produced due to variations in AET. The optimization problem is not only to reduce the idle duration but also the number of idle intervals. The overall energy consumption is the sum of energy consumed at active, idle, and shutdown state energy of all the edge servers. This includes the overhead energy caused by edge server state transitions from active to shutdown and wakeup which is to be considered while deciding upon the shutdown threshold. Due to scaled-down frequency, the active duration increases but the energy in this duration remains low due to low voltage. Thus overall energy consumption reduces.

4 Proposed Technique

The optimization problem aims to minimize the overall energy consumption of the homogeneous multi-edge system supporting multi-mode energy levels. For the source edge server from where the jobs are migrated, maximize the shutdown duration and for the target edge server having insufficient duration for shutdown to which the jobs are migrated, minimize the idle duration. The idle duration is produced due to variations in AET. The optimization problem is not only to reduce the idle duration but also the number of idle intervals. The overall energy consumption is the sum of energy consumed at active, idle, and shutdown state energy of all the edge servers. This includes the overhead energy caused by edge server state transitions from active to shutdown and wakeup which is to be considered while deciding upon the shutdown threshold. Due to scaled-down frequency, the active duration increases but the energy in this duration remains low due to low voltage. Thus overall energy consumption reduces.

Our proposed scheduler follows slowdown and shutdown techniques along with migration for less energy consumption. Thus named Slowdown or Shutdown with Migration (SoSM). Whenever it is possible for the edge server to be transited from an idle state to a shutdown state in the future, it executes the jobs at full voltage and frequency. To qualify for shutdown, it postpones the upcoming job execution using the Dynamic Procrastination (DP) technique and migrates the jobs to other Edge Servers. When it is not possible to push the jobs to other Edge Servers, the unproductive inactive time is utilized by pulling the jobs from other Edge Servers to aid in shutting down the other Edge Servers. If the server still continues to have an idle state, our scheduler SoSM executes the non-critical jobs at the lowest possible voltage and frequency to save energy. This technique is called Dynamic Voltage/Frequency technique (DVFS). Migration with DP and DVFS helps in increasing the overall Multi-Edge system utilization and shutdown duration resulting in reducing energy consumption.

At the beginning, the scheduler considers the worst-case execution time (WCET) of jobs to find the active and idle durations. If the idle duration is not large enough for shutdown, it finds jobs that can be pushed to other Edge Servers - **Push**. The non-migratable jobs are then procrastinated to increase the idle duration -**Procrastinate**. If after the actions it does not have a duration large enough to qualify for shutdown, instead of remaining idle, it pulls jobs from other edge servers -**Pull**. If there are not enough jobs to pull, it finds the appropriate voltage and frequency for job executions and scales down the voltage/frequency accordingly.

When the Edge node becomes idle, SoSM uses **Push - Procrastinate - Pull** policy [19] to decide whether to keep the edge node idle or shut it down. This effectively combines idle/shutdown intervals to longer shutdown duration using aggressive procrastination and migration. When the edge node wakes up, the scheduler computes the appropriate voltage and frequency for task execution. Algorithm 1 shows our scheduler SoSM.

5 Experimental Evaluation

We designed a framework for finding the resultant schedule of a task set along with measuring various energy parameters like inactive, static, dynamic, and total energy consumption for the state-of-the-art Dynamic Procrastination (DPS) and DVFS technique along with the proposed scheduler Shutdown or Slowdown with Migration (SoSM). Each edge node is allocated a set of randomly generated tasks using a first-fit bin packing approximation algorithm based on CPU utilization. The utilization of each edge node is varied between 20% to 90%. The simulations for utilization percentages ranging from 225% to 355% in a set of 3 and 4 edge nodes are performed. The following ranges for mean, $\mu = (\text{WCET} + \text{AET})/2$, and standard deviation $= \int (\text{WCET} - \text{AET})/\text{number of tasks}$ is given where WCET is the worst-case execution time and AET is the actual execution time. The AET of the task is varied between 5% and 100% of its WCET in steps of 5%. For the analysis purpose, the schedules are generated for one hyper period because the situation of all the periodic tasks is ready at the same time repeats at the least common multiple of the periods which is called a hyper period. The total energy consumption (E_{tot}) during execution is measured by considering energy components like Static energy (E_{stat}), Dynamic energy (E_{dyn}), Scheduler decision making energy (E_{dm}), processor shutdown and wakeup energy (E_{psd}).

$$E_{tot} = E_{stat} + E_{dyn} + E_{dm} + E_{psd} \quad (1)$$

Jejurikar et al. [15] have given a set of energy parameters which we use in our simulations as shown in Table 1 [19]. We take the maximum frequency at 1 V to be 3.1 GHz with 0.43 nF of capacitance [15]. For procrastination with migration decisions, we take $2 \mu\text{J}$. For the idle duration to qualify for shutdown, the energy saved must be more than the energy spent shutting down and waking up the edge server i.e. the static and dynamic energy consumed during idle duration must be more than the overhead energy of state transition. If T is the shutdown threshold, $(22 \text{ nJ} + 11 \text{ nJ}) T > 483 \mu\text{J}$. Thus we have considered T as the time equivalent to 15000 CPU cycles.

Table 1. Energy parameters

Symbol	Energy per cycle
E_{stat}	22nJ
E_{dyn}	11nJ: idle, 44nJ: active
E_{dm}	40 μJ for dispatcher
E_{psd}	483 μJ

Algorithm 1. SHUTDOWN OR SLOWDOWN WITH MIGRATION SCHEDULER (SoSM)

On Event: At time t at the beginning of the active periodInput: ready queue of all edge servers at time t .

Output: state active/idle/shutdown at the end of the active period and scaled voltage (SV) for the active period.

Step 1: Compute the idle duration at the end of the active period.

Step 2: If (idle duration > shutdown threshold)

return state = shutdown and SV = Max voltage

Else Find the list of pushable jobs.

Find the procrastinated idle duration (PID)

by considering the pushable jobs and

procrastination of non-pushable jobs using DP.

If (PID > shutdown threshold)

return state = shutdown and SV = Max voltage

Else Find the pushable jobs from the end of the active period.

This will create extra idle duration (EID).

If (EID + PID > shutdown threshold)

return state = shutdown and SV = Max voltage

Else Find the list of pullable jobs from other edge servers.

Find the appropriate scaled voltage/frequency using DVFS.

If the list of pullable jobs is non-empty

return state = active and SV = scaled voltage

Else return state = idle and SV = scaled voltage

On Event: Empty ready queue i.e when the edge node is idleInput: ready queue of other edge servers at time t .

Output: state idle/shutdown and scaled voltage (SV)

Step 1: Find the PID by considering the pushable jobs and procrastination of non-pushable jobs using DP.

Step 2: If (PID > shutdown threshold)

return state = shutdown and SV = zero

Else Find the list of pullable jobs from other edge servers.

Find the appropriate scaled voltage/frequency using DVFS.

If the list of pullable jobs is non-empty

return state = active and SV = scaled voltage

Else return state = idle and SV = minimum voltage

6 Experimental Results

We compare the results of our proposed technique SoSM with state-of-the-art energy-saving techniques Dynamic Procrastination (DPS) and DVFS schedulers, combined DPS and DVFS (SoS) i.e. without migration in Figs. 1 and 2.

We find that the utilization of migration in our proposed technique results in increased shutdown duration. A possible reduction in the chances of core shutdown occurs due to the scaled active duration. Figure 1 shows that on average SoSM reduces shutdown duration by 2.76% over DPS and increases by 0.74% over SoS schedules.

On average, SoSM reduces static energy by 46% over DVFS. Since static energy is inversely proportional to the shutdown duration, this implies that on average, our proposed technique increases the static energy by 2.88% over DPS and reduces it by 0.81% over the SoS algorithm.

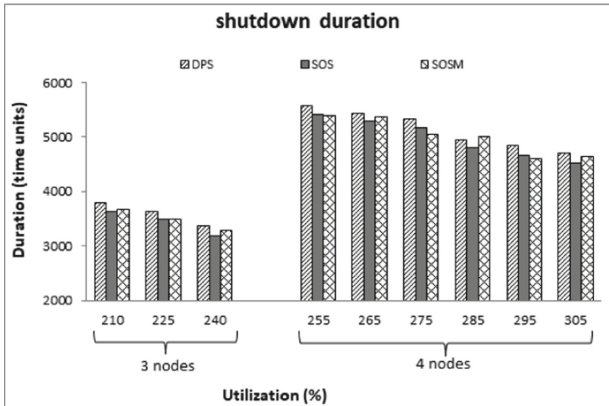


Fig. 1. Shutdown duration for different utilizations

On average the dynamic energy consumption of SoSM is more than the DVFS by 4.5% over and less than by 5% and 1.34% over DPS and SoS algorithms respectively.

We find that on average, SoSM reduces the total energy consumption by 18.6%, 2.3%, and 1.2% over DVFS, DPS, and SoS algorithms respectively Fig. 2.

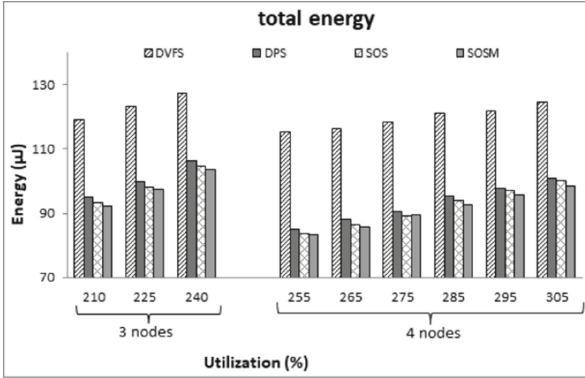


Fig. 2. Total energy consumption per unit for different utilizations

7 Conclusion

For IoT applications using Edge layer computations having Edge servers connected with dedicated links, energy efficiency is an important criterion for a successful deployment. In the future, the conservation of energy and the long life of deployed systems is likely to be deciding factor for large-scale adoption. A vision of the future world has all household and industrial devices being part of some IoT network. Satisfying the requirements of collecting, analyzing, and transmitting data requires optimized and energy-efficient Edge servers.

In this work, we have proposed a dynamic task allocation and scheduling technique for periodic tasks on multilayer IoT systems, having Edge Servers that can conserve energy by staying in a very low-energy state like shutdown. Along with the primary constraint of timeliness on real-time tasks, our proposed scheduler achieves a reduction in overall energy consumption. It achieves enhancement in overall shutdown duration with the help of migration, dynamic procrastination of tasks, and dynamic voltage/frequency scaling of task execution. Migration helps to utilize any unused idle duration of Edge servers by executing the jobs from other Edge servers and increasing the possibility of a shutdown in Edge servers. This results in reduced static and dynamic energy consumption of the system. In the future, with the deployment of Embedded devices such as Edge Servers, the use of an effective task-scheduling mechanism will result in conserving energy and an increase in the lifetime of IoT deployments.

References

1. Liu, B., Foroozannejad, M.H., Ghiasi, S., Baas, B.M.: Optimizing power of many-core systems by exploiting dynamic voltage, frequency and core scaling. In: IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1–4 (2015)

2. Chen, Y.L., Chang, M.F., Liang, W.Y., Lee, C.H.: Performance and energy efficient dynamic voltage and frequency scaling scheme for multicore embedded system. In: IEEE International Conference on Consumer Electronics (ICCE), pp. 58–59 (2016)
3. Keng-Mao, C., Chun-Wei, T., Yi-Shiuan, C., Chu-Sing, Y.: A High Performance Load Balance Strategy for Real-Time Multicore Systems. Hindawi Publishing Corporation, The Scientific World Journal (2014)
4. Devdas, V., Aydin, H.: On the interplay of voltage/frequency scaling and device power management for frame-based real-time embedded applications. In: IEEE Transactions on Computers, pp. 31–44 (2012)
5. Legout, V., Jan, M., Pautet, L.: A scheduling algorithm to reduce the static energy consumption of multiprocessor real-time systems. In: 21st ACM International conference on Real-Time Networks and Systems (RTNS), pp. 99–108 (2013)
6. Yang, C.Y., Chen, J.J., Thiele, L., Kuo, T.W.: Energy-efficient real-time task scheduling with temperature-dependent leakage. In: Design, Automation Test in Europe Conference Exhibition (DATE), pp. 9–14 (2010)
7. Awan, M., Petters, S.: Enhanced race-to-halt: a leakage-aware energy management approach for dynamic priority systems. In: 23rd Euromicro Conference on Real-Time Systems (ECRTS), pp. 92–101 (2011)
8. Huang, K., Santinelli, L., Chen, J., Thiele, L., Buttazzo, G.: Periodic power management schemes for real-time event streams. In: Proceedings of the 48th IEEE Conference on Decision and Control (CDC) with 28th Chinese Control Conference, pp. 6224–6231 (2009)
9. Rowe, A., Lakshmanan, K., Zhu, H., Rajkumar, R.: Rate-harmonized scheduling and its applicability to energy management. In: IEEE Transactions on Industrial Informatics, pp. 265–275 (2010)
10. Liu, J., Mao, Y., Zhang, J., Letaief, K.B.: Delay-optimal computation task scheduling for mobile-edge computing systems. In: IEEE International Symposium on Information Theory (ISIT), pp. 1451–1455 (2016)
11. Mao, Y., Zhang, J., Letaief, K.B.: Dynamic computation offloading for mobile-edge computing with energy harvesting devices. In: IEEE Journal on Selected Areas in Communications, pp. 3590–3605 (2016)
12. Mao, Y., Zhang, J., Letaief, K.B.: Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2017)
13. Han, L., Canon, L.-C., Liu, J., Robert, Y., Vivien, F.: Improved energy-aware strategies for periodic real-time tasks under reliability constraints. In: IEEE Real-Time Systems Symposium (RTSS), pp. 17–29 (2019)
14. Pagani, S., Chen, J.J.: IEEE 19th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Energy efficiency analysis for the Single Frequency Approximation (SFA) Scheme, pp. 82–91 (2013)
15. Jejurikar, R., Pereira, C., Gupta, R.: Leakage aware dynamic voltage scaling for real time embedded systems. In: 41st IEEE Design Automation Conference, pp. 275–280 (2004)
16. Chen, J.J., Heng-Ruey, H., Kai-Hsiang, C., Chia-Lin, Y., Ai-Chun, P., Tei-Wei, K.: Multiprocessor energy-efficient scheduling with task migration considerations. In: Proceedings of 16th Euromicro Conference on Real-Time Systems (ECRTS), pp. 101–108 (2004)
17. Gawali, S., Raveendran, B.: DPS: a dynamic procrastination scheduler for multi-core/multi-processor hard real time systems. In: IEEE International Conference on Control, Decision and Information Technologies (CoDIT), pp. 286–291 (2016)

18. Yu, K., Yang, Y., Xiao, H., Chen, J.: An improved DVFS algorithm for energy-efficient real-time task scheduling. In: IEEE 22nd International Conference on High Performance Computing and Communications, IEEE 18th International Conference on Smart City, IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 263–272 (2020)
19. Gawali, S.K., Goveas, N.: P3: A task migration policy for optimal resource utilization and energy consumption. In: 2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS), pp. 1–6 (2022)
20. Roy, S.K., Devaraj, R., Sarkar, A.: SAFLA: scheduling multiple real-time periodic task graphs on heterogeneous systems. In: IEEE Transactions on Computers, pp. 1067–1080 (2023)
21. Al-Janabi, T.A., Al-Raweshidy, H.S.: An energy efficient hybrid MAC protocol with dynamic sleep-based scheduling for high-density IoT networks. In: IEEE Internet of Things Journal, pp. 2273–2287 (2019)
22. Perin, G., Meneghello, F., Carli, R., Schenato, L., Rossi, M.: EASE: energy-aware job scheduling for vehicular edge networks with renewable energy resources. In: IEEE Transactions on Green Communications and Networking, pp. 339–353 (2023)
23. Zhang, L., Yan, L., Pang, Y., Fang, Y.: FRESH: freshness-aware energy-efficient Scheduler for cellular IoT systems. In: IEEE International Conference on Communications (ICC), pp. 1–6 (2019)
24. Delgado, C., Famaey, J.: Optimal energy-aware task scheduling for batteryless IoT devices. In: IEEE Transactions on Emerging Topics in Computing, pp. 1374–1387 (2022)



Deep Learning Based Framework for Forecasting Solar Panel Output Power

Prajnyajit Mohanty^(✉), Umesh Chandra Pati^(ID),
and Kamalakanta Mahapatra

National Institute of Technology Rourkela, Rourkela, India
prajnyajitmohanty@gmail.com, {ucpati, kkm}@nitrkl.ac.in

Abstract. Energy Harvesting from diverse renewable energy sources has experienced rapid growth due to the adverse environmental impacts of using fossil fuels. Solar energy is a significant energy source frequently used for power generation in various applications. Due to the variable nature of solar irradiation, temperature, and other metrological parameters, Photovoltaic (PV) power generation is highly fluctuating. This unstable nature of output power has been evolved as a considerable issue in various applications of solar energy prediction system. In this work, a hybrid Deep Learning (DL) model based on Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Attention mechanism to forecast solar cell output power has been proposed. The proposed model is implemented, and its performance is compared with other DL models, including CNN, LSTM, and LSTM with an attention mechanism. The proposed model has been trained and evaluated with a publicly available dataset which contains 20 parameters on which solar panel output power is relatively dependent. The model yields maximum coefficient of determination (R^2) up to 84.5%. A lightweight model has also been developed using the pruning technique to implement the DL model into a low-end hardware.

Keywords: Energy Prediction · Energy Harvesting · Solar Energy · Deep Learning · Internet of Things

1 Introduction

The dependency transition from non-renewable energy sources such as oil, natural gas, coal, and nuclear energy to renewable energy sources such as solar, mechanical vibration, kinetic, thermal, and wind has been hastened in various sectors for electricity generation in response to the growing need of restriction on non-renewable energy uses in power production. Solar energy is one of the most promising and prominent renewable energy source due to its high energy density, ubiquitous nature, and cost-effectiveness [1]. Due to recent advancements in Photovoltaic (PV) technology, the use of solar panels and solar cells is not only restricted to power grids but also has captured immense popularity in

powering Internet of Things (IoT) nodes and consumer electronics products to feature them with sustainability. Energy forecasting framework has been widely adopted in power grids for grid stabilization, while in IoT applications, it is used for task scheduling. However, power generation from solar panels or solar cells is fraught with high degree of uncertainty due to its unavoidable dependence on variable environmental parameters or characteristics. Therefore, a reliable forecasting framework that can effectively predict solar panel output power must be developed, which can help balancing the energy consumption at the load side as per the energy generation at the source side.

Several number of models have been proposed to forecast the output power of solar panel in last decade. These models can be classified into three categories such as physical model, statistical model, and Machine Learning (ML) model based on their underlying methodology and assumptions. The physical model is a mathematically established model following the principle of solar panel power generation. It uses parameters such as temperature, solar radiation, humidity, air pressure, wind speed, cloud volume, solar panel installation angle etc. The physical prediction model depends on precise station geography information, reliable meteorological data, and comprehensive PV battery information instead of historical data. The statistical and ML models are primarily data-driven and use various weather parameters to build the forecasting model. The objective of the statistical model is to predict the future output power of the solar panel by establishing the correlation mapping between the input-output data through curve fitting and parameter estimation. The ML models can extract non-linear, high-dimensional features and directly map them to output. ML models have emerged as one of the most popular techniques for forecasting time series [2]. Traditional neural networks can only increase the number of hidden layers and input layer nodes to recognize more complex relationships between dimensional input and output because the cognitive ability of the traditional neural network is limited in the new situation of dealing with more input variables. However, Deep Neural Networks (DNN) can extract more features than traditional neural networks and reduce the vanishing gradient problem. Convolutional Neural Networks (CNN) [3], Long Short Term Memory (LSTM) [4], and Deep Belief Networks (DBN) [5] are the most prevailing DNN networks used to forecast output power of solar panel. Recent research in this domain proved that prediction accuracy is higher for DL models than ML models.

A hybrid DL model has been proposed in this manuscript for forecasting output power of solar panel based on historical data. The contributions of this manuscript are as follows.

- A hybrid DL model based on CNN, LSTM, and the attention mechanism has been proposed.
- Attempts have been made to implement various DL models such as LSTM, CNN, Auto-LSTM, LSTM-attention, Gated Recurrent Units (GRU), CNN-LSTM with the same dataset for forecasting the output power of solar panel to have a comparative analysis of the proposed model.
- A lightweight model of the proposed DL model has been developed using the pruning technique which can be implemented on low-end microprocessors.

Further, the related prior work has been discussed in Sect. 2. The methodology and proposed model are described in Sect. 3 and 4, respectively. Results and discussions have been presented in Sect. 5. The manuscript has been concluded in Sect. 6.

2 Related Prior Work

In [6], various Deep Neural Network (DNN) architectures such as LSTM, auto-LSTM, Deep Belief Networks (DBN), and ANN architectures such as Multilayer Perceptron (MLP) have been implemented for energy forecasting. It has been observed that DNN models outperform ANN models. In [7], Auto-GRU is proposed and compared with other models such as LSTM, Auto-LSTM, GRU, and theta model. This work employs a ML and Statistical Hybrid model with two diversity techniques: structural and data diversity. In [8], a hybrid CNN and LSTM model is proposed. The results show that the hybrid CNN-LSTM model outperforms others, such as CNN and LSTM. It has also been observed that accuracy is improved by increasing the input sequence length. In [9], a hybrid DL model based on CNN and LSTM forecasts solar power and is compared using different benchmark models, including Persistence, BPNN, and RBFNN. In [10], a hybrid DL model based on wavelet decomposition and LSTM has been proposed. The proposed model is compared with LSTM, Recurrent Neural Network (RNN), GRU, and Multi-Layer Perceptron (MLP) model.

In [11], three DL models, LSTM, GRU, and RNN, are studied and applied to identify the best-performing model for solar panel output power forecasting. It is observed that LSTM and RNN outperform GRU. In [12], a simplified LSTM model built on ML methodology is introduced. It is observed that the LSTM model outperforms MLP. In [13], a DL model based on Long Short-Term Memory Neural Network (LSTMNN), has been proposed for forecasting solar power output. The proposed model has been compared with various ANN models such as feed-forward Extreme Learning Machine (ELM) and the shallow learning Elman Neural Network (ENN). In [14], three DL models, LSTM, Temporal Convolutional Network (TCN), and GRU, have been implemented to forecast solar panel output power using different time intervals. In this work, it has been reported that LSTM outperforms all other models. In [15], various ML and DL models are studied and compared using RMSE metrics. The results show that DL models such as LSTM, GRU, and RNN outperform ML models such as Baseline regression, Linear regression, Lasso regression, Elastic net regression, Ridge regression, Random forest regression, Gradient boost regression, Extra trees regression, Light Gradient Boosting Machine (LGBM) regression, and K-Nearest Neighbors (KNN) regression.

3 Methodology

Figure 1 represents the functional methodology of the proposed solar panel output power forecasting framework. The dataset for this work has been taken from

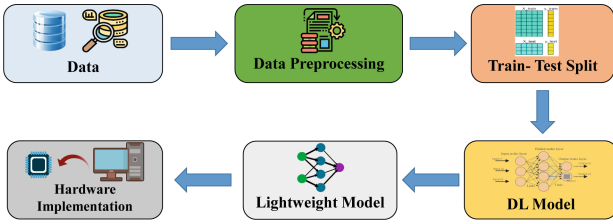


Fig. 1. Basic block diagram of DL based energy forecasting framework

Kaggle’s website [16]. The raw data need preprocessing, which includes data cleaning, which typically deals with finding the null values and handling the missing data and data transformations. The preprocessed data is split into training and testing data in the ratio of 80:20. The DL model itself can extract the significant features from the dataset. The model iteratively learns during training process and tunes the hyperparameters to reduce the error while increasing the forecasting accuracy. Subsequently, the testing data has been used to evaluate the performance of the model. The proposed DL model can not run in a low-end microprocessor/microcontroller. Thus, a lightweight model can be developed in order to implement it in the field to forecast the solar panel output power. The lightweight model has lesser trainable parameters compared to the original model thus, it is supposed to consume lesser computation power compared to the original model.

4 Proposed Work

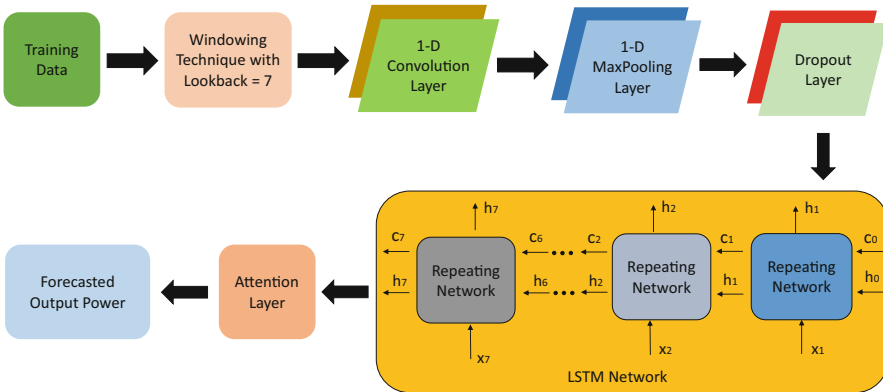


Fig. 2. Block diagram of the proposed forecasting model

Figure 2 depicts the proposed hybrid CNN and LSTM with an attention mechanism. The employed CNN extracts meaningful insights about the spa-

tial representations or patterns related to solar power forecasting, such as cloud cover, solar radiation backward, and other weather parameters from the input data. The CNN component comprises the 1-D convolutional, 1-D max pooling, and dropout layer. A set of learnable filters often called the kernel, is convolved with input data in the convolutional layer. Each kernel is a small matrix that slides across the receptive fields and performs convolution with the input values. Each filter in the convolution layer captures the visual patterns from the input data. The convolutional layer can recognize local structures and task-related visual information by learning these filters. The critical advantage of a convolutional layer is parameter sharing. The parameter sharing reduces the number of learnable parameters, lowering the training time and enabling efficient generalization. The MaxPooling layer follows this convolutional layer. The pooling layer will reduce the dimensionality of the feature map, which helps lower the computational complexity. Then dropout layer is introduced, which prevents overfitting and improves generalization capability. LSTM layers will take the output of the CNN and process it sequentially, considering the historical patterns in the data. LSTM layers capture the temporal dependencies of the historical data by learning the patterns from the data. The attention layer assigns weights to different time steps of the LSTM output, which helps to focus on significant time steps or dependencies while forecasting future values.

5 Results and Discussion

The publicly available dataset, which has been used as the dataset in this work has a total of 21 parameters and 4203 samples of data [16]. The correlation of various parameters with output power of the solar panel has been shown in Fig. 3. It can be noted that correlation coefficients closer to 1 indicate higher correlation, while correlation coefficients far from 1 indicate lower correlation. Among all the features, five highly correlated features are the angle of incidence, shortwave radiation, total cloud cover, humidity, and temperature in descending order. The correlation between output power and these most significant parameters are shown in Fig. 4. The dataset has been preprocessed using the MinMax normalization technique. The preprocessing stage ensures that no missing values exist in the dataset. Then the data is split into training and testing data in the ratio 80:20, respectively. The proposed model can extract the significant input features from the dataset. The model has been trained using training data, and the hyperparameters are tuned using validation data to minimize the error. Figure 5(a) represents the plot between training loss and validation loss. The model is trained for 100 epochs, and it has been observed that the training loss is getting saturated. The gap between training loss and validation loss is minimum, indicating no traces of model underfitting or overfitting. Figure 5(b) typically depicts the justification for choosing the number of layers of LSTM and CNN considering R^2 as the performance metric. It can be observed that the model achieves maximum R^2 of 0.845 when the hybrid model uses two convolutional layers and one LSTM layer with attention mechanism indicating low

error values, low trainable parameters and thus low computational complexity. The performance of the model is evaluated using four different error metrics, such as Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), coefficient of determination and (R^2). The mathematical formulae for calculating MAE, MSE, RMSE, and R^2 are mentioned below. Figure 6 illustrates the graph between the actual output power of the solar panel against the predicted output power of the solar panel. It has been plotted only for 250 samples in the testing data to visualize the difference between predicted and original values.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - p_i)^2 \quad (1)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - p_i)^2} \quad (2)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - p_i| \quad (3)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - p_i)^2}{\sum_{i=0}^N (y_i - p_{mean})^2} \quad (4)$$

A lightweight model has been developed using pruning technique that reduces the trainable parameters. It can be noted that a large number of weights is near zero in a trained fully connected neural network. These insignificant weights whose values are near zero are removed using pruning technique. The number of trainable parameters is decreased as a result without decreasing accuracy. This lightweight model can be deployed into a low-end microprocessor such as raspberry pi to incorporate the proposed framework in field. The proposed model has been compared with various DL models which are majorly implemented to predict output power of solar panel. Table 1 summarises the comparative analysis of the proposed model with similar models. The error metrics is graphically shown in Fig. 7. Thus, the proposed light weight model can be directly implemented on off-the-shelf microprocessors and deployed on field to forecast the harvestable solar energy in short term. Along with the microprocessor, a low-power Data Acquisition (DAQ) system can be designed and deployed to measure the necessary environmental parameters, which task is to provide real-time data to the proposed model allowing it to be scaled up for use on a real-world test bench. The proposed model can receive the data from the DAQ and process them to forecast the energy that can be harvested in future.

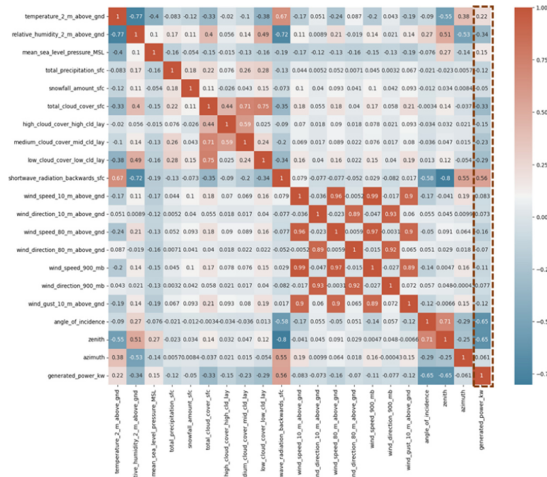


Fig. 3. Heating map

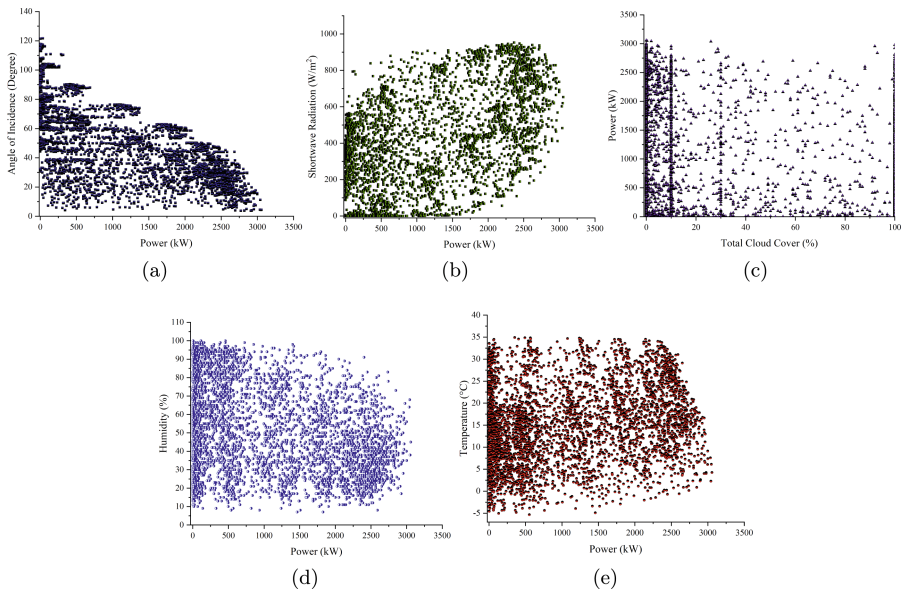


Fig. 4. Correlation between output power of solar panel and various parameters (a) Output power vs. angle of incidence of the solar panel (b) Output power vs. shortwave radiation (c) Total cloud cover vs. output power (d) Output power vs. humidity (e) Output power vs. temperature

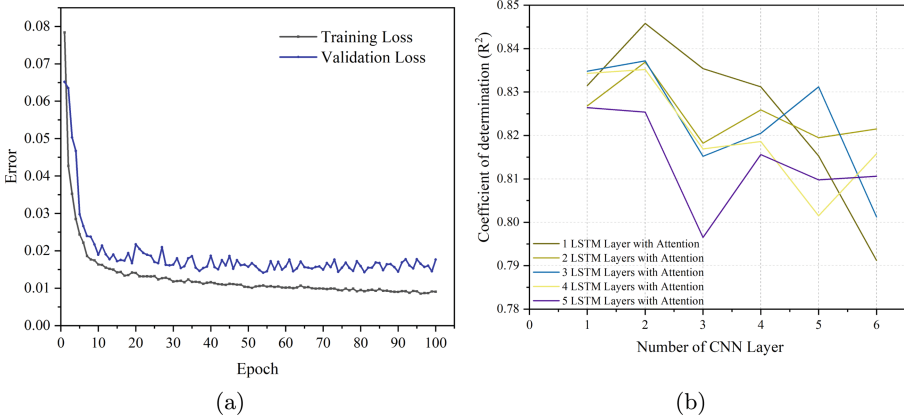


Fig. 5. Correlation between output power of solar panel and various parameters (a) Loss curve (b) Layers analysis of the proposed model

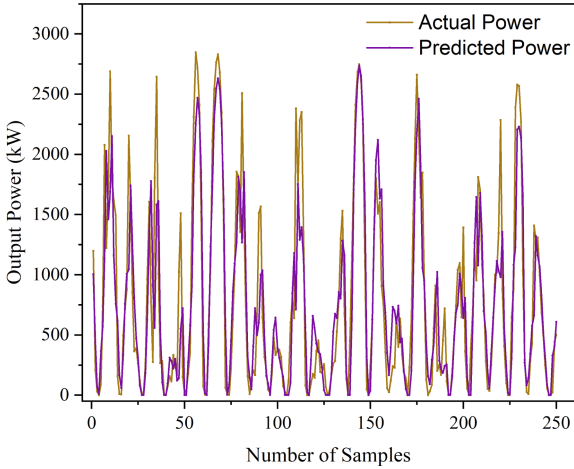


Fig. 6. Predicted output power vs original power output

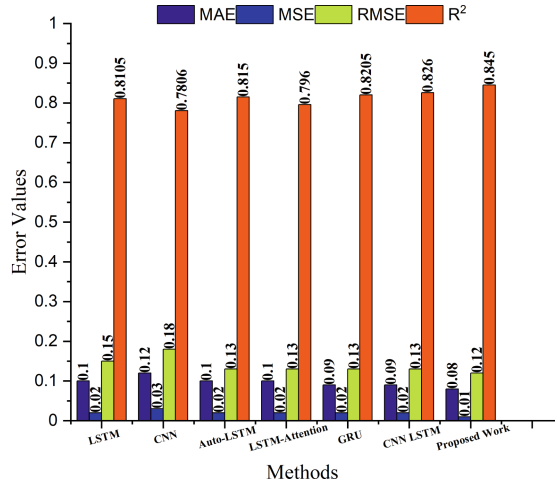


Fig. 7. Error metrics for prediction output power of solar panel

Table 1. Performance comparison of proposed model with various DL models

Model	MAE	MSE	RMSE	R ²
LSTM	0.10	0.02	0.15	0.8105
CNN	0.12	0.03	0.18	0.7806
Auto-LSTM	0.10	0.02	0.13	0.815
LSTM-attention	0.10	0.02	0.13	0.796
GRU	0.09	0.02	0.13	0.8205
CNN-LSTM	0.09	0.02	0.13	0.826
Proposed Work	0.08	0.01	0.12	0.845

6 Conclusion

In this manuscript, a hybrid DL model for forecasting the output power of solar panel has been proposed. The proposed model uses two layers of CNN hybridized with one layer of LSTM followed by an attention layer. CNN is used for feature extraction, and LSTM with an attention layer predicts future output power. The model has been analyzed using four error metrics such as MAE, MSE, RMSE, and R². The proposed model is compared with various popular DL models used to forecast solar panels in most of the literature. It outperforms other models by achieving maximum R² of 0.845. A lightweight model has also been developed for the proposed model using the pruning technique, which has less number of trainable parameters and possesses almost the same accuracy. The lightweight model can be used to implement the framework on low-end microprocessors to forecast the output power of solar panels in field.

In future, the proposed model can be implemented on hardware to forecast the output power of smaller solar cells. Thus, this forecasting framework can be used to incorporate task scheduling mechanism in energy autonomous IoT devices.

References

1. Mohanty, P., Pati, U., Mahapatra, K.: Self-powered intelligent street light management system for smart city. In: IEEE 18th India Council International Conference (INDICON), Guwahati, India, pp. 1–6 (2021). <https://doi.org/10.1109/INDICON52576.2021.9691575>
2. Yagil, G., Yang, D., Srinivasan, D.: Automatic hourly solar forecasting using machine learning models. *Renew. Sustain. Energy Rev.* **105**, 487–498 (2019). <https://doi.org/10.1016/j.rser.2019.02.006>
3. Yao, G., Lei, T., Zhong, J.: A review of convolutional-neural-network-based action recognition. *Pattern Recogn. Lett.* **118**, 14–22 (2019). <https://doi.org/10.1016/j.patrec.2018.05.018>
4. Rao, G., Huang, W., Feng, Z., Cong, Q.: LSTM with sentence representations for document-level sentiment classification. *Neurocomputing* **308**, 49–57 (2018). <https://doi.org/10.1016/j.neucom.2018.04.045>
5. Tao, Y., Chen, H., Qiu, C.: Wind power prediction and pattern feature based on deep learning method. In: IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), pp. 1–4. (2014). <https://doi.org/10.1109/APPEEC.2014.7066166>
6. Gensler, A., Henze, J., Sick, B., Raabe, N.: Deep Learning for solar power forecasting - an approach using AutoEncoder and LSTM Neural Networks. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, pp. 002858–002865 (2016). <https://doi.org/10.1109/SMC.2016.7844673>
7. AlKandari, M., Ahmad, I.: Solar power generation forecasting using ensemble approach based on deep learning and statistical methods. *Appl. Comput. Inform.* (2019). <https://doi.org/10.1016/j.aci.2019.11.002>
8. Wang, K., Qi, X., Liu, H.: A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network. *Appl. Energy* **251**, 1–14 (2019). <https://doi.org/10.1016/j.apenergy.2019.113315>
9. Li, G., Xie, S., Wang, B., Xin, J., Du, S.: Photovoltaic power forecasting with a hybrid deep learning approach. *IEEE Access* **8**, 175871–175880 (2020). <https://doi.org/10.1109/ACCESS.2020.3025860>
10. Li, P., Zhou, K., Lu, X., Yang, S.: A hybrid deep learning model for short-term PV power forecasting. *Appl. Energy* **259**, 1–11 (2020). <https://doi.org/10.1016/j.apenergy.2019.114216>
11. Jebli, I., Belouadha, F., Kabbaj, M., Tilioua, A.: Deep learning based models for solar energy prediction. *Adv. Sci. Technol. Eng. Syst. J.* **6**(1), 349–355 (2021). <https://doi.org/10.25046/aj060140>
12. Liu, C., Gu, J., Yang, M.: A simplified LSTM neural networks for one day-ahead solar power forecasting. *IEEE Access* **9**, 17174–17195 (2021). <https://doi.org/10.1109/ACCESS.2021.3053638>
13. Montoya, A., Mandal, P.: Day-ahead and week-ahead solar PV power forecasting using deep learning neural networks. In: North American Power Symposium (NAPS), Salt Lake City, UT, USA, pp. 1–6 (2022). <https://doi.org/10.1109/NAPS56150.2022.10012199>

14. Chen, M., Chiang, H., Chang, C.: Solar photovoltaic power generation prediction based on deep learning methods. In: IET International Conference on Engineering Technologies and Applications (IET-ICETA), Changhua, Taiwan, pp. 1–2 (2022). <https://doi.org/10.1109/IET-ICETA56553.2022.9971676>
15. Machina, S., Koduru, S., Madichetty, S.: Solar energy forecasting using deep learning techniques. In: 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, pp. 1–6 (2022). <https://doi.org/10.1109/PARC52418.2022.9726605>
16. Solar energy power generation dataset. <https://www.kaggle.com/datasets/stucom/solar-energy-power-generation-dataset>, (Accessed 1 May 2023)

AI and Big Data for Next-G Internet of Medical Things (IoMT)



EHR Security and Privacy Aspects: A Systematic Review

Sourav Banerjee¹ , Sudip Barik¹ , Debashis Das² , and Uttam Ghosh³ 

¹ Computer Science and Engineering, Kalyani Government Engineering College,
Kalyani, Nadia, India

mr.sourav.banerjee@ieee.org

² Department of Computer Science and Engineering, Narula Institute of Technology,
Kolkata, West Bengal, India

³ Department of CS and DS, Meharry Medical College, Nashville, TN, USA

Abstract. Electronic Health Records (EHRs) have become increasingly popular in recent years, providing a convenient way to store, manage and share relevant information among healthcare providers. However, as EHRs contain sensitive personal information, ensuring their security and privacy is most important. This paper reviews the key aspects of EHR security and privacy, including authentication, access control, data encryption, auditing, and risk management. Additionally, the paper discusses the legal and ethical issues surrounding EHRs, such as patient consent, data ownership, and breaches of confidentiality. Effective implementation of security and privacy measures in EHR systems requires a multi-disciplinary approach involving healthcare providers, IT specialists, and regulatory bodies. Ultimately, the goal is to come upon a balance between protecting patient privacy and ensuring timely access to critical medical information for feature healthcare delivery.

Keywords: Electronic Healthcare · Record (EHR) · Deep Federated learning (DFL) · Deep Learning · Artificial Intelligence · Machine Learning

1 Introduction

Electronic Health Records (EHRs) have become an integral part of modern healthcare, providing a convenient way to store, manage and share patient information among healthcare providers. With the increasing adoption of EHRs, ensuring their security and privacy has become an essential characteristic of healthcare delivery. This paper provides an overview of the key aspects of EHR privacy and security, including authentication, access control, data encryption, auditing, and risk management. Additionally, the paper discusses the legal and ethical issues surrounding EHRs, such as consent, data ownership, and breaches of confidentiality [33].

1.1 Importance of EHR in Modern Healthcare

EHRs contain sensitive personal information, including medical history, diagnoses, and treatments, which cybercriminals can exploit for identity theft or insurance fraud. In addition, unauthorized access to EHRs leads to serious breaches of patient confidentiality, resulting in reputational damage to healthcare providers and legal repercussions. Moreover, patients have the right to control their health information, and healthcare providers are responsible for protecting that information. To protect the confidentiality, integrity, and availability of Electronic Health Records (EHRs), it is imperative to establish vigorous security and privacy protocols [39].

1.2 Key Aspects of EHR Security and Privacy

- **Access Control:** Access control is the process of dealing with the situation who can access EHRs and what actions they can perform. It includes authentication, authorization, and accountability. Authentication verifies the identity of the user, authorization determines what resources the user can access, and accountability ensures that the actions of the user are recorded for auditing purposes [7].
- **Data Encryption:** Data encryption is the process of converting plaintext data into ciphertext to prevent unauthorized access. Encryption ensures that the data is secure during transmission and storage, making it unreadable to unauthorized users [30].
- **Auditing:** Auditing is the process of recording and monitoring EHR access and use. Auditing helps to detect and investigate any unauthorized access, modification or disclosure of EHRs, ensuring compliance with regulatory requirements and standards [24].
- **Risk Management:** Risk management refers to the series of activities aimed at identifying, evaluating, and minimizing risks related to the security and privacy of electronic health records (EHRs). This involves the creation of policies and procedures to manage EHRs, training employees on how to maintain EHR security and privacy, and implementing various technical controls such as firewalls, intrusion detection systems, and prevention mechanisms. Through this process, organizations can effectively safeguard the confidentiality, integrity, and availability of their EHRs, while complying with relevant laws and regulations governing the handling of medical information [31].

1.3 Legal and Ethical Issues

The protection of Electronic Health Records (EHRs) poses significant legal and ethical challenges that demand thoughtful contemplation. Patients are entitled to manage their health information, which encompasses the ability to regulate who can retrieve it and for what reason. Conversely, healthcare providers have a responsibility to secure patients' information from being accessed or revealed

without authorization. Furthermore, healthcare providers must adhere to various regulations such as the Health Insurance Portability and Accountability Act (HIPAA) that establishes nationwide criteria for preserving the privacy and security of patient health information [17,21,31].

1.4 Motivation

EHR security and privacy are critical aspects of modern healthcare, requiring a multi-disciplinary approach that involves healthcare providers, IT specialists, and regulatory bodies. Effective implementation of security and privacy measures in EHR systems is crucial for protecting patient privacy and ensuring timely access to critical medical information for quality healthcare delivery. In conclusion, protecting EHRs requires constant attention and vigilance, and healthcare providers must remain up-to-date with the latest security and privacy measures to protect their patients' sensitive personal information.

This work is structured into several sections for clarity and organization. In Sect. 2, we will explore the role of digitalization in the healthcare sector. Section 3 will delve into the importance of ensuring the security and privacy of Electronic Health Records (EHRs) in the context of federated learning. Major challenges in this area will be analyzed in Sect. 4, followed by an examination of the current state of the art in Sect. 5. Section 6 will analyze the limitations of some existing approaches. In Sect. 7, we will propose federated learning-based solutions for ensuring security and privacy in the healthcare sector. Finally, in Sect. 8, we will discuss possible directions for future work.

2 Digital Advancements in Healthcare

Digital technology has become more and more important in healthcare innovation and has introduced several tools and methods for improving healthcare services. These measures consist of maintaining secure storage of patient information in a centralized location and implementing software that enhances the availability of health-related data for patients. However, the digitization of healthcare is still in its early stages, and several multidimensional problems need to be addressed.

Healthcare organizations are adopting digital technologies to improve performance and efficiency, save costs, and increase efficacy. This trend is fueled by the availability of cost-effective and energy-efficient equipment and software, as well as the success of high-profile projects in many countries. Digital health systems can be especially beneficial in low-income countries, helping organizations achieve cost savings and improve healthcare delivery, which is critical in the time-reactive nature of healthcare.

A periodic survey is carried out by the World Health Organization (WHO) to gather information on the scope and structure of healthcare digitization across different countries. However, a recent report on digital healthcare innovation

in France indicates that the integration of innovation is still lacking, which is preventing the expansion of healthcare digitization in the country.

Expanding the scope of National Health Service (NHS) mobile health services is of paramount importance, given that a significant number of such services are currently restricted to limited pilot studies and have yet to achieve widespread adoption. The integration of technology in healthcare has both advantages and disadvantages. On one hand, it promotes innovation in health services and administrative processes, leading to reduced healthcare costs and improved efficiency in both internal and inter-hospital services [23]. On the other hand, there are multidimensional challenges that must be addressed, such as cybersecurity risks, inadequate integration of innovation, and infrastructure issues. Some factors influencing Digital Health are shown below in Fig. 1.

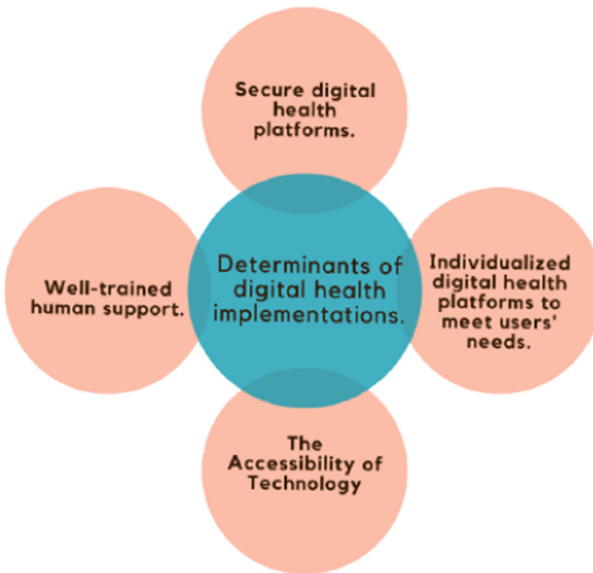


Fig. 1. The Key Factors Influencing Digital Health.

According to researchers, the integration of technologies in healthcare has both positive and negative sides. On the positive side, it promotes novel health services and streamlined administrative processes, leading to decreased healthcare costs and increased efficiency of both internal and inter-hospital services [3]. However, some challenges must be addressed, such as understanding the social barriers that may arise, including conflicts with hospital strategies and medical staff's behaviour. In addition, there is a significant of technical risk associated with information security.

Table 1 outlines the security goals that are paramount in the healthcare sector, which primarily revolve around protecting patient data, guaranteeing the

privacy and confidentiality of sensitive information, and upholding the availability and integrity of healthcare systems [16].

Table 1. Some solutions to enhance privacy and security in the healthcare sector.

Security objective	Description	Techniques
Availability	Authorized users can always access healthcare systems, even in situations where failures or attacks occur	For modern computing architecture, distributed storage, virtualization, and data backup/recovery are essential
Confidentiality	Only authorised healthcare personnel have access to patient information	Virtual private networks and encryption
Privacy	Ensure that only authorized individuals have access and safeguard against any breaches of personal data	The processes of rendering data anonymous, using pseudonyms, and encrypting it
Integrity	Ensuring that patient information is not altered without authorization	Digital signatures, hash functions, data checksums, version control, audit trails
Authentication	The authentication of users and their access to healthcare is crucial matter in preventing unauthorized entry to patients' confidential information	Passwords, two-factor authentication, biometrics, smart cards, tokens, certificates, and PKI are examples of popular authentication techniques
Authorization	Regulating the availability of patient data by considering a user's position and duties within the healthcare institution	Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Access Control Lists (ACL), OAuth
Nonrepudiation	The prohibition of denial of participation in acts such as changing patient information and accessing private data supports healthcare accountability	Blockchain and Digital signatures

3 The Need for EHR Security and Privacy in Federated Learning

Federated Learning (FL) is a technique for machine learning that enables multiple organizations to collaborate on a model without compromising the privacy of their sensitive data. This approach is decentralized and allows the participants to train the model locally while sharing only the necessary information with the central server. However, Electronic Health Records (EHRs) contain sensitive personal information that can be exploited by malicious actors if not adequately protected. Therefore, ensuring the security and privacy of EHRs in FL is critical to protect patient privacy and maintaining public trust.

EHRs typically contain sensitive information, such as patient names, addresses, social security numbers, medical histories, and other personal health information. This information is highly valuable and can be used by malicious

actors for identity theft, insurance fraud, and other criminal activities. Moreover, the unauthorized disclosure or misuse of EHRs can harm patients' reputations, cause emotional distress, and lead to physical harm.

In FL, multiple institutions collaborate on a machine learning model without sharing their sensitive data. This collaborative approach can provide significant benefits, such as improved accuracy, reduced bias, and faster model training. However, it also introduces new security and privacy risks, such as data breaches, data poisoning attacks, and model inversion attacks.

Therefore, it is critical to ensure the security and privacy of EHRs in FL to protect patients' sensitive personal information. This requires implementing robust security and privacy measures, such as access control, data encryption, auditing, and risk management. Additionally, participants in FL must comply with various regulatory requirements, such as the Health Insurance Portability and Accountability Act (HIPAA), which sets national standards for protecting the privacy and security of patient's health information.

In summary, protecting EHRs in FL is essential to maintain public trust, protect patient privacy, and comply with regulatory requirements. Healthcare providers, data scientists, and regulatory bodies must work together to implement effective security and privacy measures to protect patient's sensitive personal information [8, 35].

3.1 Federated Learning

Federated learning provides a secure and privacy-focused approach for distributed machine learning models among different devices in the context of the Internet of Medical Things (IoMT). To leverage the benefits of federated learning in the IoMT, it is essential to establish a connection between the devices with sensors and other data-generating components and a central server. After collecting data from the devices, the server uses it to train a machine-learning model. This model is then sent back to each device for further use. The local storage of data ensures its safety against potential data breaches, while predictions can be made using the trained model. The general architecture of FL is illustrated in Fig. 2. This allows medical professionals to benefit from IoMT insights without compromising patient privacy. The utilization of federated learning to combine data from various devices can have a substantial impact on enhancing prediction accuracy and yielding better outcomes in the healthcare domain.

The healthcare industry has been slow to adapt to the digital advancements seen in other fields, but various digital developments are now causing significant changes. Figure 3 demonstrates the practical application of a Federated Learning architecture within a healthcare context. The trend towards digitization and real value in healthcare is being propelled by various factors, including the proliferation of digital firms, the cost management initiatives of payers, and the growing demand for improved care among elderly patients. By adopting digital transformation, healthcare providers can improve their services and reduce costs, leading to macroeconomic disruption and improved business models. Furthermore, established companies can team up with newer firms to minimize investment

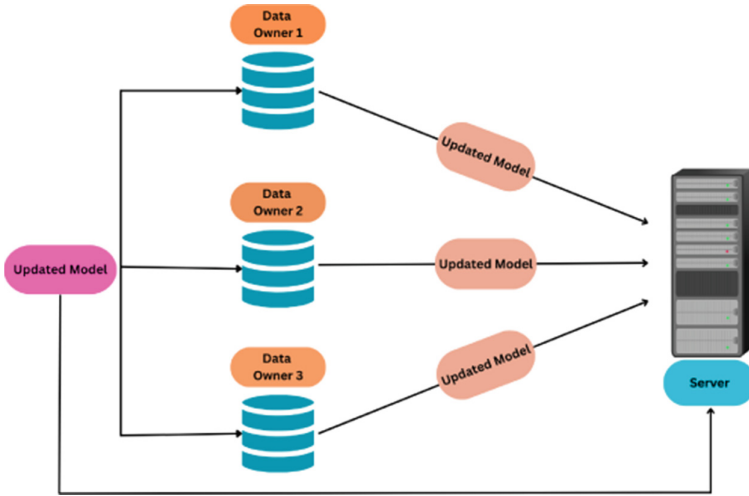


Fig. 2. General Architecture of Federated Learning

expenses. According to the authors referenced in [5], established organizations' expertise, when combined with proper regulations, can assist startups in digitally disrupting the healthcare industry.

Digitalization has the potential to enhance healthcare outcomes while also reducing costs. The capacity to process large amounts of varied data quickly and with flexibility is a key advantage of digital technologies. However, to fully leverage these benefits, data warehouses, and cloud-based data management technologies must be employed. Although data warehouses are still prevalent in health IT, they may not be sufficient for utilizing big data. To utilize big data efficiently, it is essential to have appropriate IT infrastructure, visualization techniques, workflows, user interfaces, and tools. Moreover, big data must be employed in a manner that balances societal benefits with patient privacy, in order to create value for healthcare. In order to make optimal use of big data in healthcare, institutions need to be ready to elaborate modifications in their database utilization, accessibility, sharing, privacy measures, sustainability practices, and compliance requirements [26,37].

4 Major Challenges

Healthcare organizations worldwide are embracing Electronic Health Records (EHRs), which are digital versions of a patient's medical history. However, EHRs also pose significant security and privacy challenges, some of which are:

Unauthorized access: EHRs contain sensitive patient information, such as medical history, social security number, and insurance details. Unauthorized access to this information by individuals can result in identity theft or fraud.

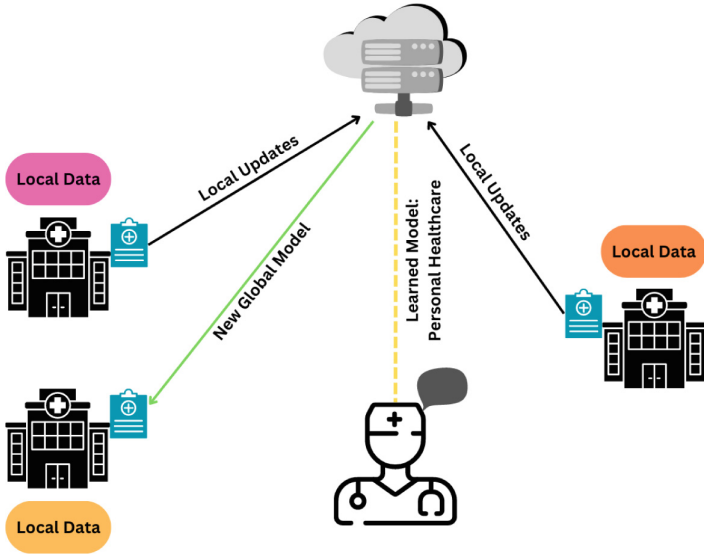


Fig. 3. Federated Learning used in healthcare industry

Cybersecurity threats: Healthcare institutions are a prime target for cyber-criminals as they contain sensitive and valuable patient information in electronic health records (EHRs). Cyber threats such as malware, phishing attacks, and ransomware can jeopardize the security of EHRs, potentially resulting in data breaches. Such data breaches can lead to significant harm to patients, including identity theft, fraud, and medical identity theft. Therefore, healthcare organizations must prioritize cybersecurity measures to safeguard patient data and ensure the confidentiality, integrity, and availability of their EHRs.

Human error: EHRs can be misused, intentionally or unintentionally, by authorized users, resulting in the exposure of sensitive information. For example, a healthcare worker may accidentally upload a patient’s medical record to a public-facing website.

Interoperability: EHRs may need to be shared between different healthcare providers to facilitate patient care. However, sharing EHRs between organizations can increase the risk of unauthorized access, data breaches, and privacy violations.

Legal compliance: EHRs must comply with numerous regulations and laws, including HIPAA, GDPR, and others. Healthcare organizations must ensure that they are compliant with all applicable regulations, which can be challenging, given the complexity of the regulations and the frequency of updates.

Patient consent and control: Patients must be given control over their health information, including the right to access, modify, and delete their data. Ensuring that patients are fully informed and have given their informed consent to the

use and sharing of their data can be challenging, especially given the complexity of the healthcare system and the varied interests of different stakeholders.

5 State of the Art

Electronic Health Record (EHR) security and privacy are critical aspects that need to be addressed to protect patient health information. Here are some of the state-of-the-art measures that are being implemented to ensure EHR security and privacy:

Access control: EHR systems use access control mechanisms to ensure that only authorized personnel can access patient health information. These mechanisms include password-protected logins, two-factor authentication, and role-based access controls [7].

Encryption: Encryption is used to protect data stored in EHR systems, making it unreadable to unauthorized users. Encryption can be applied to data both in transit and at rest.

Audit trails: EHR systems maintain an audit trail of all access and modifications to patient health information. This allows organizations to track who has accessed the information and when, and to detect any unauthorized access or modifications.

Data backup and recovery: Electronic health record (EHR) systems are crucial in maintaining patient health information, and backups of this data are regularly created to safeguard against system failure or cyberattacks. These backups serve as a failsafe mechanism and are frequently tested to guarantee their efficiency in restoring data. In case of a disaster, EHR backups can be relied upon to restore vital information, ensuring continuity of care and patient safety.

Data minimization: EHR systems implement data minimization principles, meaning that they collect only the minimum amount of information necessary to provide patient care. This helps reduce the risk of data breaches and protects patient privacy.

Regular vulnerability assessments: EHR systems undergo regular vulnerability assessments to identify and address potential security weaknesses. This helps prevent security breaches and ensures that the EHR system remains secure over time.

Employee training and awareness: EHR systems implement training and awareness programs to ensure that employees are aware of the security and privacy policies and procedures. This helps prevent accidental breaches of patient health information.

Compliance with regulations: EHR systems comply with relevant regulations and standards, such as HIPAA and GDPR. This ensures that patient health information is protected and that organizations are not subject to legal or financial penalties.

So EHR security and privacy are essential components of healthcare IT systems, and these state-of-the-art measures are crucial to ensuring that patient health information is protected from unauthorized access, use, and disclosure.

6 Limitations of Some Existing Work

While there has been significant research on EHR security and privacy, there are several limitations to existing research, including:

Limited scope: Much of the existing research has focused on specific aspects of EHR security and privacy, such as access control or data encryption. However, EHR security and privacy are complex issues that require a holistic approach.

Lack of real-world data: Many studies rely on simulated data or hypothetical scenarios, which may not reflect real-world threats and vulnerabilities.

Small sample sizes: Some studies have small sample sizes, making it difficult to generalize findings to larger populations.

Limited diversity: Many studies have focused on healthcare organizations in developed countries, which may not reflect the challenges faced by organizations in developing countries or underserved communities.

Outdated technology: Some research may be based on outdated EHR systems or security protocols, which may not reflect the current state-of-the-art in EHR security and privacy.

Limited longitudinal data: There is a lack of long-term studies on the effectiveness of EHR security and privacy measures. It is essential to evaluate the long-term effectiveness of these measures to ensure that they continue to provide adequate protection against evolving threats.

Lack of standardized evaluation methods: There is a lack of standardized methods for evaluating EHR security and privacy. This makes it difficult to compare findings across studies and to establish best practices for EHR security and privacy.

So, while existing research has provided valuable insights into EHR security and privacy, there is a need for more comprehensive, real-world studies that can help healthcare organizations better understand and address the challenges they face in protecting all patient health information. A few works have been carried out on this EHR security and privacy issues which are mentioned in Table 2.

Table 2. Some solutions to enhance privacy and security in the healthcare sector.

Cite	Network Model	Method	Security Models	Advantage	Limitations
[15] 2020	IoMT	Blockchain-based solution	anonymous and untraceable	Health records are safely kept on a tamper-proof blockchain that is managed by cloud servers	Advanced encryption and decryption techniques are employed as part of the protocol
[25] 2020	Internet of Health Things	Federated Learning	Data privacy	Federated learning and differential privacy address privacy and security concerns	A full decentralisation of FL is impossible due to the federated nodes' limited training capacity
[28] 2022	The network architecture comprises patients, telecare servers, and the registration centre	Elliptic Curve Cryptography	Patient Anonymity	Insiders, privileged individuals, and thieves using stolen equipment cannot attack the protocol	Compared to other protocols, the protocol's cryptography techniques use more energy
[36] 2022	Medical monitoring system based on RFID	The encryption process utilizes a combination of cyclic shifting and XOR operations	RFID security authentication	Medical monitoring systems with RFID technology guarantee the privacy and confidentiality of patient records	Designing an efficient and effective authentication protocol is challenging due to the resource constraint imposed by RFID tags/readers
[18] 2022	IoMT-based cloud-healthcare infrastructure	Elliptic curve cryptography	Patient anonymity	According to the comparative analysis, RAPCHI has shown better effectiveness than other protocols	The absence of practical application
[38] 2022	Wireless Medical Sensor Network	Blockchain-based solution	Anonymity and Untraceability	The utilization of smart contracts and PUF in the suggested approach offers both decentralization and security	There is no indication in the paper regarding the practicality of implementing the proposed approach in an actual real-life situation
[6] 2022	Internet of Health Things	An Authentication Protocol with Minimal Overhead	Mutual authentication	The process includes biometric measures for user anonymity, authentication, key negotiation, privacy, and access control	The procedure requires intricate techniques for encrypting and decrypting data

(continued)

Table 2. (*continued*)

Cite	Network Model	Method	Security Models	Advantage	Limitations
[2] 2022	IoT-based healthcare	Homomorphic Encryption	Privacy-preserving	By utilizing data aggregation, the EPPADA scheme aims to decrease energy usage by eliminating unnecessary data	The plan entails utilizing intricate techniques for encrypting and decrypting data
[1] 2022	Utilizing an IoT network for remote patient monitoring	A solution based on Elliptic Curve Cryptography	Privacy-preserving	The suggested RPM system provides secure authentication via RFID, ensures secure communication, and protects privacy	Challenges with dependability, restricted availability, and expensive communication
[34] 2023	Smart healthcare systems	Federated Learning	Privacy-preserving	FRESH uses certificate ring signatures as a source inference attack (SIA) defence	The system being considered is susceptible to attacks through adversarial machine learning techniques
[13] 2023	Smart healthcare utilizing the Internet of Things technology	Cryptographic primitives designed for low computational and memory requirements are commonly referred to as lightweight	Privacy-preserving	The effectiveness of the proposed authentication technique is evaluated through security and performance analysis in comparison to established and widely-used schemes	Challenges of dependability, restricted availability, and expensive communication.
[4] 2023	Internet of Things (IoT) network for healthcare	Data aggregation	Privacy-preserving	Compared to traditional methods, it lowers both the expense of communication and computation	The absence of practical application in actual situations
[29] 2023	A financial system for smart healthcare utilizing the Internet of Things	Blockchain-based solution	Data privacy	The suggested solution protects user data privacy and enables information sharing across devices using blockchain and zero-knowledge evidence	The inherent properties of blockchain technology may impose limitations on the system's ability to scale

7 Federated Learning Based Security and Privacy Solutions for the Healthcare Sector

In their research, Rahman et al. [25] suggested the use of the Internet of Health Things (IoHT) for managing health, while emphasizing the importance of protecting privacy through secure data management. They identified a lack of training capabilities and trust management as key challenges to IoHT adoption and proposed a hybrid federated learning framework that incorporates blockchain smart contracts to manage trust and authentication among federated nodes.

The framework ensures encryption and anonymity of IoHT data using differential privacy (DP) and was evaluated for COVID-19 patient data using deep learning applications, showing potential for widespread adoption.

Wang et al. [34] proposed a smart healthcare framework, known as FRESH, that aims to facilitate the sharing of physiological data while ensuring data privacy. This framework leverages Federated Learning (FL) and ring signature defence to protect against source inference attacks (SIAs).

The data collection process for FRESH involves gathering data from wearable devices and processing it using edge computing devices for local training of machine learning models. The model parameters are subsequently uploaded to the central server for joint training. The authors utilized ring signature technology to hide the source of parameter updates, which significantly reduces the success rate of SIAs. They also introduced a batch verification algorithm to improve the efficiency of signature verification.

According to the authors, FRESH is highly suitable for large-scale smart healthcare systems that cater to multiple users. This framework represents a major milestone in the quest to enhance data privacy and security in the healthcare industry.

8 Future Works

8.1 Advancing Privacy and Security in the Healthcare Industry: The Need for Further Study

The healthcare industry regards the privacy and security of patient data as essential issues, and continuous investigation is imperative to enhance current protocols and confront evolving obstacles. A potential research direction could be to investigate the effectiveness of current privacy and security regulations, such as HIPAA and GDPR, in protecting patient information. This research could examine the gaps and limitations in the existing regulatory framework and propose recommendations for improvements.

Another potential avenue of research is to investigate how emerging technologies like artificial intelligence and blockchain can improve privacy and security within the healthcare industry. For example, blockchain technology offers a decentralized and tamper-proof platform for storing and sharing patient information, which could reduce the risk of data breaches and ensure the accuracy of health records. Similarly, artificial intelligence can be used to detect and prevent potential security breaches and unauthorized access to patient information.

Furthermore, research could be conducted on the impact of privacy and security breaches on patient trust and healthcare outcomes. A breach of patient information can lead to a loss of trust between patients and healthcare providers, which can have long-lasting effects on patient health and well-being. Thus, understanding the effects of privacy and security breaches and developing strategies to restore patient trust could be a valuable research direction.

8.2 Study of the Impacts of Digitization on Health Outcomes

In addition to the benefits mentioned, digitization has also improved patient safety. For example, electronic prescribing (e-prescribing) has reduced medication errors by eliminating the need for handwritten prescriptions, which can be misread or contain errors. EHRs also can flag potential drug interactions or allergies, alerting healthcare providers to potential issues before they occur. The utilization of barcode scanning technology has enhanced medication safety by verifying that the correct medication is administered to the accurate patient at the appropriate time.

Digitization has also enabled better coordination of care among healthcare providers. With EHRs, providers can share patient information more easily and efficiently, ensuring that all members of the care team have access to the same information. This can help to reduce the risk of errors or duplicative testing, leading to improved patient outcomes.

However, despite the many benefits of digitization in healthcare, there are also challenges and potential drawbacks to consider. For example, there may be concerns about the security and privacy of patient information, as well as issues related to data ownership and access. Additionally, there may be concerns about the potential for technology to replace human interaction and the importance of maintaining the human touch in healthcare.

Future research in this area could focus on exploring the benefits and challenges of digitization in healthcare, as well as identifying ways to optimize the use of technology to improve patient outcomes and quality of care. This could include examining the role of patient engagement and education in promoting the adoption and effective use of digital technologies in healthcare, as well as the potential for technology to improve patient-centred care and promote better health results.

8.3 An Evaluation of Artificial Intelligence's Role in Healthcare

AI is transforming healthcare by leveraging natural language processing, virtual assistants, and AI-powered chatbots as well as AI-powered imaging analysis and diagnostic tools, among other technologies, to enhance patient outcomes. AI-powered healthcare solutions have the potential to significantly reduce healthcare costs, increase efficiency, and improve patient outcomes by providing faster and more accurate diagnoses, personalized treatment recommendations, and improved patient communication and engagement. However, it's important to note that while AI has the potential to revolutionize healthcare, it's important to ensure that these systems are developed ethically and that patient privacy is protected.

8.4 The Significance of Patient Engagement in Ensuring Security and Privacy

In the healthcare industry, ensuring privacy and security heavily relies on patient engagement as a crucial factor. When patients are engaged in their healthcare,

they are more likely to be aware of the risks associated with the use of personal health information and are more likely to take steps to protect it. Healthcare providers can encourage patient engagement by providing clear and concise information about privacy and security policies, as well as by offering patient education resources, such as online portals, educational videos, and other materials. Healthcare providers can promote patient privacy and security, as well as enable informed decision-making regarding healthcare, by actively engaging patients in safeguarding their personal information [20, 22, 32].

8.5 Exploring the Potential of Blockchain Technology in Healthcare

The healthcare sector has the prospect of a significant transformation through the adoption of blockchain technology, as it offers secure and transparent means for storing and exchanging patient information. Blockchain's distributed ledger technology can ensure that patient data is protected from unauthorized access, while also allowing for the efficient sharing of that data among healthcare providers. The use of blockchain technology for electronic medical records can also reduce errors, streamline workflows, and increase the accuracy of medical data [14, 19].

By utilizing blockchain technology, smart contracts can automate and simplify the procedure of insurance claims and reimbursements for both patients and insurance companies. By reducing the time and costs associated with traditional payment processing systems, blockchain-based smart contracts can help to reduce healthcare costs and improve patient outcomes [12, 27].

Blockchain technology can leverage the healthcare industry to facilitate supply chain management via vehicular communication [9] securely, enabling the verification and traceability of medical devices and drugs to ensure their genuineness. This can improve patient safety by reducing the risk of counterfeit products, and can also help to improve supply chain efficiency and transparency [10, 11].

9 Future Scope and Advancements

Digitalization, including IoMT (Internet of Medical Things) and blockchain technology, offers significant opportunities and advantages to the healthcare sector. However, successful implementation requires careful consideration of social, organizational, and collaborative aspects. Fostering a positive attitude and providing necessary support enable healthcare organizations to adopt digital technologies, improving patient care and cost savings while addressing potential difficulties and constraints. Future work aims to enhance Electronic Health Record (EHR) security through novel approaches, leveraging IoMT and blockchain technology for concise and secure record-keeping in electronic mode.

10 Conclusion

In conclusion, digitalization offers significant opportunities and advantages to the healthcare sector, but its implementation requires careful consideration of the social, organizational, and collaborative aspects of the workplace. Even if digitization can enhance healthcare performance and accomplish strategic goals, it is crucial to be aware of potential difficulties and constraints. By focusing on a positive attitude and providing necessary support, healthcare organizations can successfully adopt digital technologies and streamline their procedures, resulting in improved patient care and cost savings.

Acknowledgement. This work was supported by the National Science Foundation, under award number 2219741.

References



1. Ahmed, M.I., Kannan, G.: Secure and lightweight privacy preserving internet of things integration for remote patient monitoring. *J. King Saud Univ.-Comput. Inform. Sci.* **34**(9), 6895–6908 (2022)
2. Alam, M.A., Al Riyami, K.: Shear strengthening of reinforced concrete beam using natural fibre reinforced polymer laminates. *Constr. Build. Mater.* **162**, 683–696 (2018)
3. Alloghani, M., Al-Jumeily, D., Hussain, A., Aljaaf, A.J., Mustafina, J., Petrov, E.: Healthcare services innovations based on the state of the art technology trend industry 4.0. In: 2018 11th International Conference on Developments in eSystems Engineering (DeSE), pp. 64–70. IEEE (2018)
4. Bhowmik, T., Banerjee, I.: Eeppda-edge-enabled efficient privacy-preserving data aggregation in smart healthcare internet of things network. *Inter. J. Network Manag.* e2216 (2023)
5. Chae, B.: Mapping the evolution of digital business research: a bibliometric review. *Sustainability* **14**(12), 6990 (2022)
6. Chen, C.M., Chen, Z., Kumari, S., Lin, M.C.: Lap-ioht: a lightweight authentication protocol for the internet of health things. *Sensors* **22**(14), 5401 (2022)
7. Dagher, G.G., Mohler, J., Milojkovic, M., Marella, P.B.: Ancile: privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustain. Urban Areas* **39**, 283–297 (2018)
8. Dang, T.K., Lan, X., Weng, J., Feng, M.: Federated learning for electronic health records. *ACM Trans. Intell. Syst. Technol. (TIST)* **13**(5), 1–17 (2022)
9. Das, D., Banerjee, S., Chatterjee, P., Ghosh, U., Biswas, U.: A secure blockchain enabled v2v communication system using smart contracts. *IEEE Trans. Intell. Trans. Syst.* (2022)
10. Das, D., Banerjee, S., Chatterjee, P., Ghosh, U., Mansoor, W., Biswas, U.: Design of a blockchain enabled secure vehicle-to-vehicle communication system. In: 2021 4th International Conference on Signal Processing and Information Security (ICSPIS), pp. 29–32. IEEE (2021)
11. Das, D., Banerjee, S., Chatterjee, P., Ghosh, U., Mansoor, W., Biswas, U.: Design of an automated blockchain-enabled vehicle data management system. In: 2022 5th International Conference on Signal Processing and Information Security (ICSPIS), pp. 22–25. IEEE (2022)

12. Das, D., Banerjee, S., Dasgupta, K., Chatterjee, P., Ghosh, U., Biswas, U.: Blockchain enabled sdn framework for security management in 5g applications. In: 24th International Conference on Distributed Computing and Networking, pp. 414–419 (2023)
13. Das, S., Namasudra, S.: Lightweight and efficient scpprivacy-preserving/scp mutual authentication scheme to secure scpinternet of things/scp-based smart healthcare. *Trans. Emerging Telecommun. Technol.* (2023)
14. Dutta, K., Guin, R.B., Chakrabarti, S., Banerjee, S., Biswas, U.: A smart job scheduling system for cloud computing service providers and users: modeling and simulation. In: 2012 1st international conference on recent advances in information technology (rait), pp. 346–351. *IEEE* (2012)
15. Garg, N., Wazid, M., Das, A.K., Singh, D.P., Rodrigues, J.J., Park, Y.: Bakmpiomt: design of blockchain enabled authenticated key management protocol for internet of medical things deployment. *IEEE Access* **8**, 95956–95977 (2020)
16. Herrmann, M., Boehme, P., Mondritzki, T., Ehlers, J.P., Kavadias, S., Truebel, H.: Digital transformation and disruption of the health care sector: Internet-based observational study. *J. Med. Internet Res.* **20**(3), e104 (2018)
17. Kigera, J., Kipkorir, V.: Electronic health records-the ethical and legal issues. *Annals African Surgery* **20**(1), 1–2 (2023)
18. Kumar, V., Mahmoud, M.S., Alkhayat, A., Srinivas, J., Ahmad, M., Kumari, A.: Rapchi: robust authentication protocol for iomt-based cloud-healthcare infrastructure. *J. Supercomput.* **78**(14), 16167–16196 (2022)
19. Lahiri, P.K., Das, D., Mansoor, W., Banerjee, S., Chatterjee, P.: A trustworthy blockchain based framework for impregnable iov in edge computing. In: 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 26–31. *IEEE* (2020)
20. Lahiri, P.K., Mandal, R., Banerjee, S., Biswas, U.: An approach towards developments of smart covid-19 patient’s management and triaging using blockchain framework (2020)
21. Li, H., et al.: Review on security of federated learning and its application in healthcare. *Futur. Gener. Comput. Syst.* **144**, 271–290 (2023)
22. Mandal, R., Banerjee, S., Islam, M.B., Chatterjee, P., Biswas, U.: Qos and energy efficiency using green cloud computing. In: *Intelligent Internet of Things for Healthcare and Industry*, pp. 287–305. Springer (2022). https://doi.org/10.1007/978-3-030-81473-1_14
23. Manogaran, G., Thota, C., Lopez, D., Sundarasekar, R.: Big data security intelligence for healthcare industry 4.0. *Cybersecurity for Industry 4.0: Analysis for Design and Manufacturing*, pp. 103–126 (2017)
24. Parker, M.: Managing threats to health data and information: toward security. In: *Health Information Exchange*, pp. 149–196. Elsevier (2023)
25. Rahman, M.A., Hossain, M.S., Islam, M.S., Alrajeh, N.A., Muhammad, G.: Secure and provenance enhanced internet of health things framework: a blockchain managed federated learning approach. *IEEE Access* **8**, 205071–205087 (2020)
26. Roski, J., Bo-Linn, G.W., Andrews, T.A.: Creating value in health care through big data: opportunities and policy implications. *Health Aff.* **33**(7), 1115–1122 (2014)
27. Roy, R., Haldar, P., Das, D., Banerjee, S., Biswas, U.: A blockchain enabled trusted public distribution management system using smart contract. In: *International Conference on Electronic Governance with Emerging Technologies*, pp. 25–35. Springer (2022). https://doi.org/10.1007/978-3-031-22950-3_3
28. Ryu, J., et al.: Secure ecc-based three-factor mutual authentication protocol for telecare medical information system. *IEEE Access* **10**, 11511–11526 (2022)

29. Singh, R., Dwivedi, A.D., Srivastava, G., Chatterjee, P., Lin, J.C.W.: A privacy preserving internet of things smart healthcare financial system. *IEEE Internet of Things J.* (2023)
30. Sonkamble, R.G., Bongale, A.M., Phansalkar, S., Sharma, A., Rajput, S.: Secure data transmission of electronic health records using blockchain technology. *Electronics* **12**(4), 1015 (2023)
31. Tertulino, R., Antunes, N., Morais, H.: Privacy in electronic health records: a systematic mapping study. *J. Public Health*, 1–20 (2023)
32. Tiwari, S., et al.: Applications of machine learning approaches to combat covid-19: a survey. In: *Lessons from COVID-19*, pp. 263–287 (2022)
33. Wang, S., Kirillova, K., Lehto, X.: Travelers' food experience sharing on social network sites. *J. Travel Tourism Market.* **34**(5), 680–693 (2017)
34. Wang, W., Li, X., Qiu, X., Zhang, X., Zhao, J., Brusica, V.: A privacy preserving framework for federated learning in smart healthcare systems. *Inform. Proc. Manag.* **60**(1), 103167 (2023)
35. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. *J. Healthcare Inform. Res.* **5**, 1–19 (2021)
36. Yang, C., Everitt, J.H., Murden, D.: Evaluating high resolution spot 5 satellite imagery for crop identification. *Comput. Electron. Agric.* **75**(2), 347–354 (2011)
37. Yin, X., Zhu, Y., Hu, J.: A comprehensive survey of privacy-preserving federated learning: a taxonomy, review, and future directions. *ACM Comput. Surv. (CSUR)* **54**(6), 1–36 (2021)
38. Yu, S., Park, Y.: A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions. *IEEE Internet Things J.* **9**(20), 20214–20228 (2022). <https://doi.org/10.1109/JIOT.2022.3171791>
39. Zhao, Y., et al.: Growth traits and sperm proteomics analyses of myostatin gene-edited Chinese yellow cattle. *Life* **12**(5), 627 (2022)



SNN Based Neuromorphic Computing Towards Healthcare Applications

Prasenjit Maji¹ , Ramapati Patra², Kunal Dhibar³,
and Hemanta Kumar Mondal² 

¹ Department of CSD, BCREC Durgapur, Durgapur, India
maji.katm@gmail.com

² Department of ECE, NIT Durgapur, Durgapur, India
rp.19ec1103@phd.nitdgp.ac.in, hemanta.mondal@ece.nitdgp.ac.in

³ Department of CSE, BCET Durgapur, Durgapur, India

Abstract. The diagnosis, treatment, and prevention of diseases may be revolutionized by integrating neuromorphic computing, artificial intelligence (AI), and machine learning (ML) into medical services. A novel method of processing complex data that more effectively and quickly mimics how the human brain works is called neuromorphic computing. This paper provides an overview of neuromorphic computing and its uses in AI and ML-based healthcare. We talk about the advantages and disadvantages of using these technologies as well as how it helps to accelerate the entire diagnostic procedure. We also provide case studies of how neuromorphic applications have been successfully used in the medical field to diagnose and predict diseases. Additionally, we provide the medical and healthcare industries with enhanced Spiking Neural network application results with up to 98.5% accuracy.

Keywords: Artificial Intelligence · Machine Learning · Neuromorphic Computing · Neural Networks · Spiking Neural Network

1 Introduction

Medical healthcare is an area that can significantly benefit from the integration of neuromorphic computing, AI, and ML. Neuromorphic computing is a revolutionary computer paradigm that mimics the operation of the human brain. It uses spiking neural networks (SNNs), Deep Neural Networks (DNNs), and in some cases, Recurrent Neural Networks (RNNs) to process and analyze data more efficiently and faster than traditional computing systems. AI and ML are also becoming increasingly important in healthcare, allowing for analyzing large amounts of data and developing predictive models. In the proposed work, we explore the applications of these technologies in medical healthcare and their potential to transform the field. Artificial Neural Networks (ANNs) are widely used in a variety of applications such as image recognition, audio identification, and processing of natural languages. Traditional ANNs, on the other hand, are unsuitable

for low-power and real-time applications because to their high computational complexity and energy consumption. Because of their excellent energy economy and processing capacity, SNNs have emerged as a potential replacement to standard ANNs. In this paper, we comprehensively study the theory and applications of SNNs [1].

1.1 Neuromorphic Computing

Neuromorphic computing is a form of computing inspired by the human brain's structure and function. It employs SNNs, which are connections of artificial neurons that interact with one another via electrical impulses. These networks can process and analyze data more efficiently and faster than traditional computing systems. Neuromorphic computing is particularly useful for complex data processing applications, such as medical healthcare [2].

1.2 Spiking Neural Networks (SNNs)

SNNs are a class of artificial neural networks that simulate the spiking behavior of biological neurons. SNNs have gained importance in recent years due to their potential to achieve high energy efficiency and computational power. This work provides a widespread review of the theory and applications of SNNs. We first introduce the basic principles of SNNs, including the spiking neuron model and the synaptic plasticity rule. Following that, we look at current improvements in SNN learning algorithms, such as supervised, unsupervised, and reinforcement learning approaches. Next, we discuss the hardware implementations of SNNs, including neuromorphic chips and Field-Programmable Gate Arrays (FPGAs) [3].

SNN and neuromorphic computing are related; however not the same thing. So, while SNNs can be used in neuromorphic systems, not all neuromorphic systems use SNNs. Neuromorphic computing is a larger field that includes a variety of ways to creating electronic circuits and devices that imitate the behavior of biological neurons and synapses, such as analogue and digital circuits, memristive devices, and others. SNNs are one type of neural network that can use in neuromorphic systems. However, different kinds of networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), can also be used.

While there are several opportunities to use neuromorphic computing, AI, and ML in medical healthcare, there are also challenges to solve. Security and confidentiality of information are two of the most challenging concerns to address. Medical information is particularly sensitive and must be protected against unauthorized access. Ethical concerns, such as the use of AI and ML throughout decision-making processes, must additionally be addressed. The general concept of the proposed work, as well as how it will be implanted in edge devices via cloud and IoT platforms, is depicted in Fig. 1.

Another issue is regulation. The application of AI and machine learning in medical healthcare is still in its early stages, and there is a need for clear laws and standards to guarantee that these technologies are utilised safely and efficiently [4]. Spiking Neural Networks (SNNs) have several potential applications in medical healthcare due to their ability to simulate the spiking behavior of biological neurons. Here are some examples of how SNNs are used in medical healthcare.

Brain-Computer Interfaces (BCIs): SNNs can be used to develop BCIs, which enable patients to communicate with computers or other devices using of their brain signals. SNNs can be used to decode the electroencephalogram (EEG) signals generated by the patient's brain and translate them into specific commands.

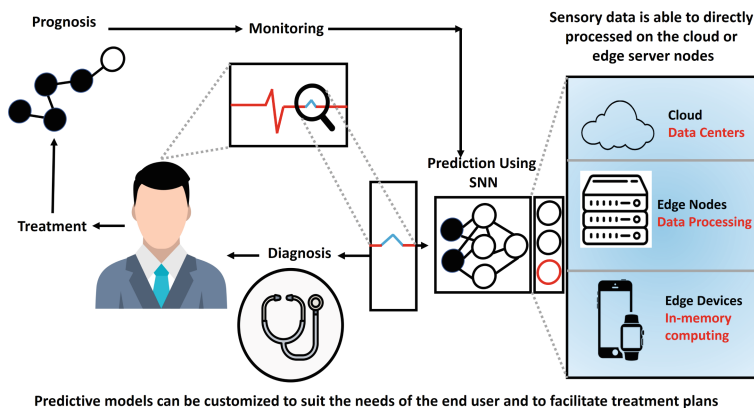


Fig. 1. Working model of SNN with the advanced application

Epilepsy Prediction: SNNs can be used to predict epileptic seizures in patients with epilepsy. SNNs can learn to recognize the patterns in EEG signals that indicate an imminent seizure and issue an alert to the patient or caregiver.

Parkinson's Disease Diagnosis: SNNs can be used to diagnose Parkinson's disease by analyzing the patient's gait. SNNs can learn to recognize the characteristic changes in gait associated with Parkinson's disease and issue a diagnosis.

Drug Discovery: SNNs can be used to accelerate the drug discovery process by predicting the efficacy and toxicity of candidate drugs. SNNs can be trained on large datasets of molecular structures and their corresponding biological activities and used to predict the activity of new compounds.

Medical Image Analysis: SNNs can be used to analyze medical images, such as MRI and CT scans. SNNs is used to detect and classify abnormalities, such as tumors, and aid in the diagnosis of diseases.

Medical Robotics: SNNs can be used to control medical robots, such as surgical robots. SNNs can learn to recognize the desired surgical actions and movements and issue commands to the robot.

Overall, SNNs have the potential to revolutionize medical healthcare by enabling faster and more accurate diagnosis, prediction, and treatment of diseases [5]. The salient contribution of the proposed work is as follows:

1. Spiking Neural Network's (SNN) biological spiking behavior is used to generate desired results.

2. The proposed work validates a Neuromorphic approach for various disease detection.
3. The proposed Neuromorphic framework considerably speeds up the process of evaluating required outputs accurately.

The remainder of the work is organized as follows: Sect. 2 contains a literature review, Sect. 3 contains a proposed method and data for the application of neuromorphic systems, and Sect. 4 contains a Dataflow Diagram. Part 4 discusses the findings, and Sect. 5 concludes with a conclusion.

2 Literature Review

There are already several examples of neuromorphic applications in medical healthcare. Artificial Intelligence (AI), Spiking Neural Networks (SNNs), Machine Learning (ML), and Neuromorphic Computing (NC) are rapidly gaining traction in medical healthcare due to their ability to process vast amounts of data and make accurate predictions. In this literature review, we will explore the various ways in which these technologies are being used in medical healthcare.

AI and ML have been used in medical healthcare for decades, primarily for tasks such as image analysis and diagnosis. However, recent advancements in SNN and NC have opened up new possibilities for medical applications. One of the primary advantages of SNN and NC is their ability to mimic the behavior of biological neurons, enabling them to process information in a way that is similar to the human brain [6].

One of the primary applications of SNN and NC in medical healthcare is in the development of Brain-Computer Interfaces (BCIs). BCIs use SNNs to translate the electrical signals generated by the brain into specific commands, enabling patients to control devices such as prosthetic limbs or communication devices. This technology has the capability to vastly enhance the quality of life for people who are paralysed or have other impairments. [7].

SNN and NC have also been used in the diagnosis and prediction of neurological disorders, such as epilepsy and Parkinson's disease. By analyzing the patterns in brain activity, SNNs can predict the onset of seizures or diagnose Parkinson's disease based on changes in gait. This technology has the potential to enable earlier diagnosis and more effective treatment of these conditions [8].

In addition to neurological disorders, SNN and NC have also been used in the analysis and treatment of other diseases, such as cancer. For example, SNNs can analyze medical images to detect and classify tumors, enabling earlier detection and more effective treatment. Another area where SNN and NC have shown promise is in the development of new drugs. SNNs can analyze large datasets of molecular structures and predict the efficacy and toxicity of new compounds. This technology has the potential to speed up the drug unearthing process and enable the development of more effective and safer drugs [17].

Overall, SNN and NC have the potential to revolutionize medical healthcare by enabling faster and more accurate diagnosis, prediction, and treatment of diseases. While there are still many challenges to overcome, such as the need for more data and the development of more efficient hardware, the potential benefits of these technologies are

Table 1. Comparative table for different neuromorphic approaches in the medical field with resultant.

Ref	Neuromorphic System	Medical Application	Results/Output
Qiao [9]	SpiNNaker	Brain-Computer Interfaces (BCIs)	Achieved state-of-the-art BCI performance with accuracy of 90.6% for 4-class classification
Livi [10]	TrueNorth	Neuromorphic Vision	High-accuracy facial recognition with 99.45% accuracy on the FERET dataset
Yamakawa [11]	BrainScaleS	Epileptic Seizure Prediction	Outperformed traditional algorithms with a sensitivity of 88.4% and a specificity of 90.2%
Pereira [12]	IBM Neurosynaptic	Brain Tumor Segmentation	Increased accuracy over traditional methods with a Dice similarity coefficient of 0.83
Michaelis [13]	Loihi	Predictive Maintenance for Medical Equipment	Identified equipment faults with high accuracy, achieving a mean-area under the ROC curve of 0.947
Hatem [14]	SpiNNaker	Motor Rehabilitation of Stroke Patients	Improved rehabilitation outcomes compared to traditional therapy, with patients showing a significant improvement in Fugl-Meyer Assessment scores
Klietz [15]	BrainScaleS	Parkinson's Disease Diagnosis	Achieved high accuracy in diagnosis, with a sensitivity of 90.0% and a specificity of 94.0%
Andreou [16]	TrueNorth	Neuromorphic Auditory Processing	Improved speech recognition compared to traditional methods, achieving a word error rate of 16.5% on the TIMIT dataset

immense. Table 1. clearly reflect the comparisons among the work done in the field of neuromorphic systems and application area.

3 Proposed Method and Data

The proposed work defines and trains a spiking neural network to perform binary classification on the Pima Indian's diabetes and Wisconsin Cancer datasets. The US National Institute of Diabetes and Digestive and Kidney Diseases first gathered the data, then made it available to the general public for research. It is a dataset that is frequently used in the study of machine learning and predictive modeling. The dataset has 768 instances, each with 8 attributes. Dr. William H. Wolberg originally gathered the breast cancer data at the University of Wisconsin Hospital in Madison, Wisconsin, and made it accessible to the general public for research. There are 569 instances in the collection, and each instance has 32 attributes.

The spiking neural network is defined using the Nengo framework. It consists of three layers: **a dense layer with 16 units**, a layer of leaky integrate-and-fire (LIF) neurons, and a dense output layer with a single sigmoid activation unit. The input and output nodes are defined as well.

The network is then compiled using the NengoDL simulator and trained using the fit method with the mean squared error loss and a stochastic gradient descent optimizer. The training is performed for the number of epochs, and we attain the best in 10 epochs.

Finally, the trained model is used to predict the training data and compute the classification accuracy. The predictions are threshold at 0.5 to obtain binary labels, and the accuracy is computed as the mean of the correct predictions.

3.1 Proposed SNN Implementation Algorithm

In this proposed work, we explain with the diabetic dataset only; however, we have tested the proposed model with both the said dataset and also explain in the result section. We first load the Pima Indians Diabetes dataset from the UCI repository and split it into input features (X) and output labels (y). We then scale the input features to the range $\{0, 1\}$ to ensure that all the features are on a similar scale.

We define a spiking neural network using the Nengo and NengoDL frameworks. The network has two dense layers and a spiking LIF layer. We then compile the model using NengoDL's Simulator object and train the model on the dataset.

Once the SNN framework is installed and the data is preprocessed, we begin to build the SNN. The steps involved in building an SNN model for diabetes prediction are as follows.

Here's the step-wise detailed working principle of the algorithm:

-
- Step 1.** Load the Pima Indians Diabetes dataset from a URL and store the column Names in a list.
 - Step 2.** Load the dataset into a Pandas DataFrame.
 - Step 3.** Split the dataset into input features (X) and output labels (y).
 - Step 4.** Scale the input features to the range [0, 1].
 - Step 5.** Define a spiking neural network using the Nengo library
 - 5.1.** Create a Nengo network.
 - 5.2.** Define the input and output nodes.
 - 5.3.** Create the first dense layer with 16 units using the NengoDL Layer.
 - 5.4.** Create the second dense layer with 16 units using the Nengo LIF Neuron model.
 - 5.5.** Create the final output layer with a sigmoid activation Function using the NengoDL layer.
 - 5.6.** Create probes to record the output of the final layer.
 - Step 6.** Configure the NengoDL settings to make the network non-trainable.
 - Step 7.** Compile the model using the MSE loss function and With a learning rate of 0.001 along with Stochastic Gradient descent optimizer
 - Step 8.** Train the model using the training data for 10 epochs.
 - Step 9.** Predict new data using the trained model.
-

The spiking neural network has two dense layers, each with 16 units, and a final output layer with a sigmoid activation function. The LIF neuron model is used in the second layer. The model is trained using the MSE loss function and SGD optimizer with a learning rate of 0.001 for 10 epochs. The final classification accuracy of the model is printed to the console, which is more than 90% for the diabetes dataset and 98.5% in cancer dataset.

4 Result and Discussion

The investigations occurred on the Google Colab platform and also in Jupyter Notebook with Python 3.7 and Keras. Machine learning techniques have been applied using the “SKLearn” library. Moreover, we monitor the binary cross-entropy with Adam Optimizer and MSE metric function. For SNN, we employ the Nengo library. The capacity of Nengo to simulate massive spiking neural networks is one of its key advantages. For that purpose, a network of interconnected neurons is defined, and each neuron is represented using LIF neurons (Fig. 2).

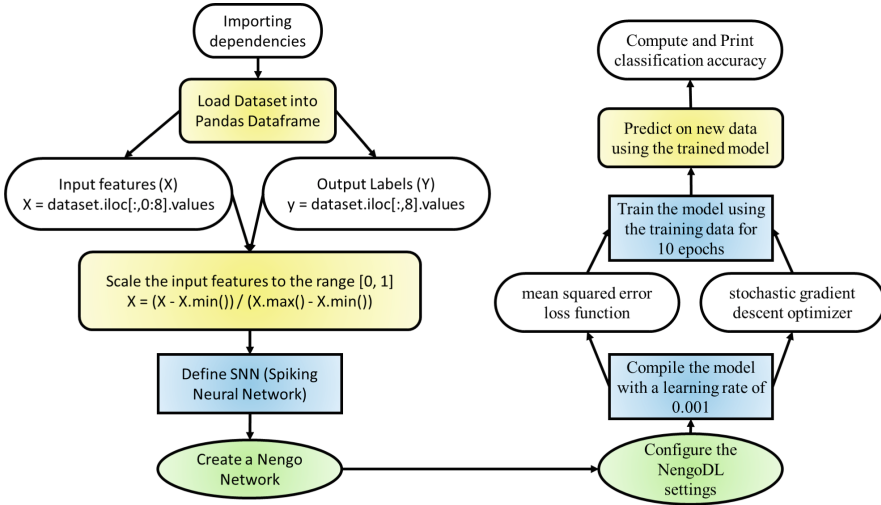


Fig. 2. Dataflow Diagram of the proposed method

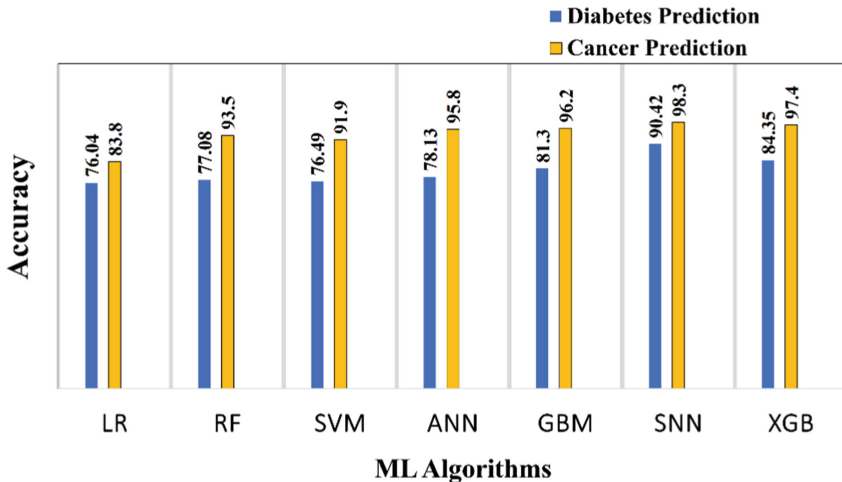


Fig. 3. Comparison of performance in disease prediction using ML and Neuromorphic approach

A number of backend, such as the CPU, GPU or specialized neuromorphic hardware, can be used to run the network. The output of the implementation is the accuracy of classification on trained spiking neural network model on the Pima Indian Diabetes dataset. The exact accuracy may vary each time the code is executed due to the stochastic nature of the spiking neural network and the randomness introduced by the training process. The hyper parameters used for the applications are as follows:

Hyper parameters:

tau_rc: Time constant of the RC circuit in the LIF neurons.

tau_ref: Refractory period of the LIF neurons.

Units: Number of neurons in the dense layers.

Activation: Activation function used in the dense layers.

Loss: Loss function used to train the model.

Optimizer: Optimization algorithm used to train the model.

Trainable: Boolean flag to specify whether to train the layers or not.

Synapse: Synapse model used for filtering the output probe.

In this dataset, the spiking neural network model achieves significant classification accuracy, correctly predicting the presence or absence of diabetes in 90.42% of the samples. For the breast cancer dataset, the accuracy is 98.5%. The overall performance of different ML algorithms and SNN for two other datasets are shown in Fig. 3.

In SNN, only those neurons that cross a certain threshold are activated, which aids in energy efficiency. Unlike other neural networks (such as CNN, ANN, etc.), all neurons are active and functioning. The fact that SNN does not use back propagation like some other systems does make it unique and more challenging to deploy. Back propagation in SNN must be explicitly implemented when the neuron grid is utilized after the feature extraction block rather than ReLU or softmax function, typically employed after feature extraction for classification. Figure 4 makes this understandable.

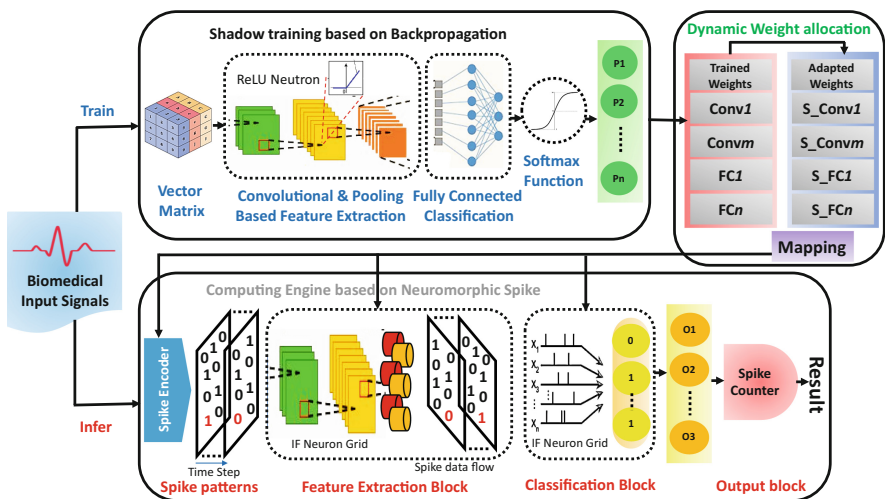


Fig. 4. Framework for generic neuromorphic edge computing in medical applications

When assessing the effectiveness of regressors, we also utilize metrics like mean squared error (MSE) or mean absolute error (MAE) in place of accuracy. The performance of the model improves as these metrics' values decrease. When predicting a continuous parameter (such as blood sugar level) in regression assignments like the one in this code, precision is typically not an essential factor. The ROC curve, used to assess the effectiveness of binary classification models, cannot be calculated because the code provided is for a regression problem. The ROC curve is used to calculate the trade-off between true and false favourable rates at various categorization levels. Braindrop is also used, which is an open-source mixed-signal neuromorphic architecture designed

to be programmed at a high degree of abstraction. Neuromorphic techniques can offer superior results when used in conjunction with hardware, particularly FPGAs. We intend to develop this project further in the next, especially by implementing hardware.

5 Conclusion

Classification of neuron modelling approaches, comparison of the investigated chips, and identification of present trends and limits impeding the development of neuromorphic technologies for medical applications. The application of Neuromorphic systems in biological healthcare and signal processing holds great potential for medical professionals, practitioners and their patients. Neural Networks specially SNNs can be utilized to improve the quality of life for chronically sick patients by allowing ambient observations for anomalies, hence reducing the strain on medical resources. In the proposed work, we can implement SNN for healthcare data with up to 98.5% accuracy compared to other ML algorithms. Proper application can result in decreased workloads for medical practitioners, allowing them to focus on time-critical jobs that demand a quality higher than what neural networks can attain now.

When combined with Neuromorphic hardware, an SNN-based application performs better than conventional ML methods. Our observation is that neuromorphic computing have the potential to significantly enhance medical applications and the optimization profession. Each of these applications is likely to benefit from the massively parallel, event-driven, and/or stochastic processes of neuromorphic computers when combined with neural networks.

References

1. Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., Maass, W.: A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat. Commun.* **11**(1), 1–13 (2020)
2. Nunes, J.D., Carvalho, M., Carneiro, D., Cardoso, J.: Spiking neural networks: a survey. *IEEE Access.* **10** (2022). <https://doi.org/10.1109/ACCESS.2022.3179968>
3. Liu, J., Wu, T., Ma, X., Zhang, Y., Hu, J.: A survey on deep learning-based neuromorphic computing. *Front. Neurosci.* **15**, 655935 (2021)
4. Koo, M., Srinivasan, G., Shim, Y., Roy, K.: sBSNN: stochastic-bits enabled binary spiking neural network with on-chip learning for energy efficient neuromorphic computing at the Edge. *IEEE Trans. Circ. Syst. I: Regular Papers*, 1–10 (2020). <https://doi.org/10.1109/TCSI.2020.2979826>
5. Li, Z., Tang, W., Zhang, B., Yang, R., Miao, X.: Emerging memristive neurons for neuromorphic computing and sensing. *Sci. Technol. Adv. Mater.* **24**(1), 2188878 (2023). <https://doi.org/10.1080/14686996.2023.2188878>. PMID:37090846;PMCID:PMC10120469
6. Schuman, C., et al.: A Survey of Neuromorphic Computing and Neural Networks in Hardware (2017)
7. Esser, S.K., Merolla, P.A., Arthur, J.V., et al.: Convolutional networks for fast, energy-efficient neuromorphic computing. *PNAS* **113**(41), 11441–11446 (2016)
8. Tavanaei, A., Ghodrati, M., Kheradpisheh, S.R., Masquelier, T., Maida, A.: Deep learning in spiking neural networks. *Neural Netw.* **111**, 47–63 (2019). <https://doi.org/10.1016/j.neunet.2018.12.002>, ISSN 0893–6080

9. Qiao, N., et al.: Reconfigurable online learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Front. Neurosci.* **9**, 141 (2015)
10. Livi, P., Indiveri, G.: A current-mode conductance-based silicon neuron for address-event neuromorphic systems. In: *IEEE International Symposium on Circuits and Systems*, Taipei, Taiwan 2009, pp. 2898–2901 (2009). <https://doi.org/10.1109/ISCAS.2009.5118408>
11. Yamakawa, T., et al.: Wearable epileptic seizure prediction system with machine-learning-based anomaly detection of heart rate variability. *Sensors* **20**, 3987 (2020). <https://doi.org/10.3390/s20143987>
12. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging* **35**(5), 1240–1251 (2016). <https://doi.org/10.1109/TMI.2016.2538465>
13. Michaelis, C., Lehr, A.B., Oed, W., Tetzlaff, C.: Brian2Loihi: an emulator for the neuromorphic chip Loihi using the spiking neural network simulator brian. *Front. Neuroinform.* **9**(16), 1015624 (2022). <https://doi.org/10.3389/fninf.2022.1015624>
14. Hatem, S.M., et al.: Rehabilitation of motor function after stroke: a multiple systematic review focused on techniques to stimulate upper extremity recovery. *Front. Hum. Neurosci.* **13**(10), 442 (2016). <https://doi.org/10.3389/fnhum.2016.00442>. PMID:27679565;PMCID:PMC5020059
15. Klietz, M., Bronzlik, P., Nösel, P., et al.: Altered neurometabolic profile in early parkinson’s disease: a study with short echo-time whole brain MR spectroscopic imaging. *Front. Neurol.* **17**(10), 777 (2019). <https://doi.org/10.3389/fneur.2019.00777>
16. Andreou, A.G.: Real-time sensory information processing using the TrueNorth Neurosynaptic System. In: *IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, QC, Canada 2016, pp. 2911 (2016). <https://doi.org/10.1109/ISCAS.2016.7539214>
17. Navamani, T.M.: *Deep Learning and Parallel Computing Environment for Bioengineering Systems* (2019)
18. Wei, O., Shitao, X., Chengyu, Z., Wenbao, H., Qionglu, Z.: An overview of brain-like computing: Architecture, applications, and future trends. *Front. Neuro.* **16** (2022). <https://doi.org/10.3389/fnbot.2022.1041108>, ISSN=1662–5218



Crossfire Attack Detection in 6G Networks with the Internet of Things (IoT)

Nicholas Perry[✉] and Suman Bhunia[✉]

Department of Computer Science and Software Engineering, Miami University,
Oxford, OH 45056, USA
{perryna4,bhunias@miamioh.edu}

Abstract. As the internet plays an increasingly vital role in our daily lives, the threat of denial of service (DoS) attacks continues to loom, posing significant challenges to network security. With the proliferation of internet-of-things (IoT) devices, including those in the healthcare sector (IoMT), the need to secure these networks becomes even more critical. The emergence of Mobile Edge Computing (MEC) servers has shifted the focus toward processing data near the network edge to alleviate network congestion. However, a new form of DoS attack, known as the crossfire attack, presents a complex challenge as it is difficult to detect and can have devastating effects on networks. While Software Defined Networks (SDNs) offer promise in mitigating DoS attacks, they also introduce vulnerabilities of their own. This paper explores the current landscape of IoT, IoMT, DoS attacks, and crossfire attacks. It discusses existing defense strategies and proposes a defense mechanism that leverages packet header inspection to differentiate between adversarial and benign packets. The paper concludes with the execution of a crossfire attack in a Mininet environment with the RYU SDN controller, highlighting the need for multiple approaches to protect critical servers in the face of persistent DDoS attacks.

Keywords: IoT · Cross Fire attack · Neural Network · Mininet

1 Introduction

As the internet continues to evolve and become more essential to daily lives than ever, there is a growing population looking to destroy it. Denial of service (DoS) attacks have been a threat to the internet for years and continue to cause issues even in today's networks. The continued improvement and introduction of internet-of-things (IoT) devices have pushed mobile carriers to update their existing infrastructure to support the increased number of devices. With the use of Internet of Medical Things (IoMT) servers, securing the network has become even more critical. If an adversary were to disrupt the operations of IoMT devices, entire hospitals and healthcare networks may be affected, leading

to disastrous, if not deadly consequences. Due to the increased load on the network, a focus on Mobile Edge Computing (MEC) has increased. These MEC servers keep data from going across the center of the network and instead process the data near the edge of the network. A newer type of DoS attack, known as the crossfire attack has plagued the internet. This type of attack is extremely difficult to detect and can be lethal to networks if executed correctly. Software defined networks (SDNs) have been discussed as a promising mitigation technology that could detect DoS attacks. While SDNs are helpful, they are not perfect and open up a different set of vulnerabilities to exploit. The end of DDoS attacks is not in sight, and therefore many different approaches must be taken to protect critical servers.

About 25 billion devices are currently interconnected and by 2025, 60 billion devices are expected to be connected [3]. As the Internet continues to develop, traditional devices are becoming “smart”, meaning that they are connected to the Internet. The term “Internet of Things” (IoT) was coined in 1999 by Kevin Ashton which describes a global network of interconnected devices [3]. The motivation for IoT devices is to create large “smart” systems [8]. Technological advancements are the reason for the increased motivation to link devices together [3]. IoT devices take many forms and almost any traditional device can be converted to a smart device. Some examples of IoT devices include smart plugs, smart washing machines, smart lights, smart refrigerators, etc. The Internet of Medical Things (IoMT) is an extension of the IoT with a focus on medical devices. These IoMT devices are medical things that have the capability to transfer information across a network without requiring human-to-human or human-to-computer interaction. These devices enable physicians to diagnose illnesses more easily by connecting various vital parameters using IoMT monitors [16].

The crossfire attack is a type of DoS attack that is more difficult to detect. This attack uses many devices across large geographic regions to send low-intensity requests across the network to various servers on the other side of the network. This is especially problematic with the advent of IoT and IoMT, because even though these devices often have extremely limited processing power, these devices can be compromised and used since the attack only requires low-intensity attacks to be sent from any given device.

Previous works seek to defend against these crossfire attacks using various methodologies. Routing around congestion (RAC) attempts to mitigate the crossfire attack by changing routing decisions based on the congestion of a given link. This solution, though also slows down legitimate traffic as all traffic is routed around that congestion. If an adversary was able to force routing decisions to consistently change, packets may be dropped as the network tries to continually determine the best route but is unable to do so. Another defense strategy, moving target defense (MTD) seeks to make the scanning phase of the attack more difficult by randomly updating routes so that an adversary would not be able to identify a consistently shared link between nodes in the network. This approach

also suffers from the fact that oftentimes these routes are non-ideal, meaning legitimate traffic is degraded at the expense of security.

The summary of contributions are:

- The paper proposes a statistical detection model for crossfire attacks using Analysis of Variance (ANOVA) and neural networks.
- The proposed mechanism only uses packet headers and not packet content to determine if a packet is adversarial that achieves an accuracy of 95.3% in detecting these packets
- We evaluated the proposed defense mechanism on a real-world network topology from the ATT North America backbone topology using the Mininet simulation environment.

The rest of the article is organized as follows: Sect. 2 discusses some background information about IoT, IoMT, DoS attacks, and crossfire attacks. Section 3 details the threat model of the attack. Section 4 discusses defense strategies against the crossfire attack. Section 5 Shows the execution of the crossfire attack.

2 Background

This section discusses the essential knowledge required to successfully execute a crossfire attack, a critical aspect of modern network security. It explores key networking concepts, including the revolutionary 6G technology, the Internet of Things (IoT) and Internet of Medical Things (IoMT), Mobile Edge Computing, and the pervasive security concerns surrounding these advancements. Moreover, it provides a comprehensive examination of the crossfire attack itself, shedding light on its intricacies and implications for network defenses. By thoroughly examining these interconnected topics, this section aims to contribute to the understanding and mitigation of cyber threats in contemporary network environments

2.1 6G

5G/6G Architecture. Until recently, mobile communication was handled by fourth-generation (4G) and Long Term Evolution (LTE) systems. Recently with the rise of Internet of Things (IoT) devices and a larger focus on edge computing, a new standard had to be created in order to support the rapidly growing Internet. Fifth-generation (5G) is the next standard of mobile communication that will be able to support such a wide variety of devices simultaneously. 5G services are attempting to meet 3 main constraints as it develops: ubiquitous connectivity, zero latency, and high-speed gigabit connections [11].

Network Architecture. As the number of mobile devices exponentially increases, there is a need for an architecture redesign from the previous generation. Differences in the waves used for 5G that permit increased speed require careful consideration due to differences in propagation.

5G cellular network architecture is distinct from previous generations but retains many features of those generations. A renewed focus on interior architecture is necessary. With 4G, an outdoor base station had the ability to allow both inside and outside users to communicate. Due to the constraints and changes in architecture, the shorter waves of 5G cannot penetrate walls as easily [5]. Therefore, 5G architecture must consider distinct interior architecture to overcome the issue of penetration loss [17]. The use of multiple-input, multiple-output (MIMO) technology can help reduce the burden of penetration loss.

2.2 IoT and IoMT

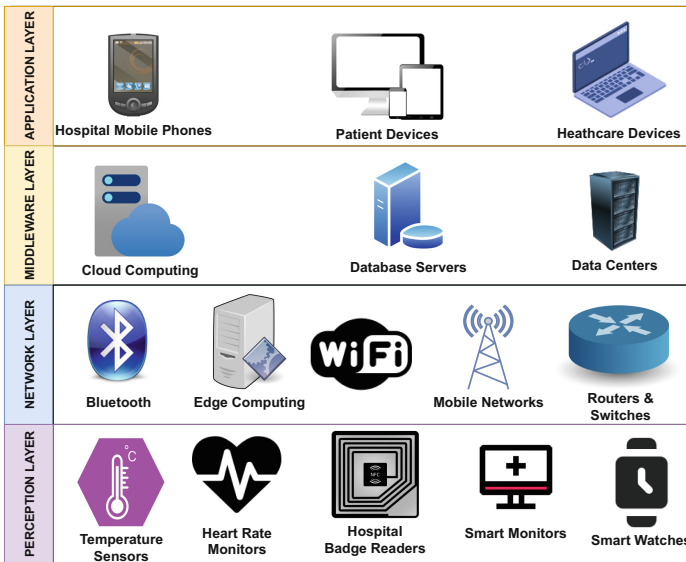


Fig. 1. The 4 Layer IoT Architecture

The Internet of Things has its own distinct architecture that works with the Internet. Typically IoT is categorized into 4 layers. Figure 1 details the types of devices that are present on each layer. The perception layer is the lowest level of the IoT architecture. The perception layer contains the sensor-enabled physical objects which act as endpoints to the IoT ecosystem. The next layer, the network layer, consists of various communication protocols, edge computing, and network connectivity. This layer transfers information securely from IoT end-points to a

processing device. The middleware layer receives data from the network layer and stores it in a database. Cloud computing servers and database management systems are typically middleware devices that allow applications and sensors to connect. Finally, the top layer of the four-layer IoT architecture is the application layer. This layer IoT exists as a result of many technologies. These technologies work together to create a holistic system that is able to communicate across the internet. Radio frequency identification (RFID) was a large precursor to IoT as it allowed machines to record metadata, recognize objects, and control devices through radio waves [8].

2.3 Mobile Edge Computing

Mobile edge computing (MEC) is an architecture where cloud computing services are placed on the edge of the network using mobile base stations [1]. With the ever-increasing need for cloud computing services while using mobile devices, placing computing servers within the radio access network (RAN) and in close proximity to these devices allows mobile traffic to connect to the nearest cloud service edge network. By placing MEC services within the RAN, bottlenecks associated with traveling through the core of the internet can be reduced [1]. The European Telecommunications Standards Institute characterizes MEC by the following criteria: [12]

1. On-Premises - The edge services should be located at the edge of the network, meaning it should be able to run isolated from the core of the network
2. Proximity - By being close to the source of the data/information, MEC is useful for analytics and data collection
3. Lower Latency - By being closer to the edge devices, latency is considerably reduced. This can be used to reduce latency or improve user experience.
4. Location Awareness - When connected to WiFi or cellular, services can use low-level signaling to determine the location of connected devices.
5. Network Context Information - Real-time network statistics can be used by applications to provide context-specific services that can be monetized and change the experience of mobile communication.

Mobile edge computing can be used in many sectors to offload core services. Augmented reality (AR) systems typically require high computational power. Many users use (AR) on their mobile devices, so computations have to be offloaded to servers. Edge computing would allow these high-demand, low-latency tasks to remain at the edge of the network [1]. Edge computing also will play a key role with respect to web performance and caching HTML content. By deploying content delivery servers at the edge, HTTP requests would travel through these servers that would handle many of these requests, reducing traffic across the core network [1]. MEC services allow 5G to continue to work towards the core goal of “zero-latency” as reducing congestion in the core allows more traffic to be routed. This in turn improves the experience for users of 5G technology.

2.4 Security Concerns

IoT and IoMT devices may provide useful services, however, they currently present a large security problem in the world of networking. The first concern that arises with the introduction of IoT is that creating additional devices that are addressable can allow attackers to intrude [8]. Security measures are only as good as the weakest link, and the introduction of new devices opens the door to additional vulnerabilities that could be exploited by an adversary. Due to the low cost of IoT devices, corners may be cut in terms of manufacturing. Oftentimes, IoT devices may use default or anonymous logins which an adversary can use to intrude on a network [13]. These concerns are magnified in healthcare settings. If sensors are compromised, they may report false data, or no data at all leading to misdiagnoses, or in the worst case, a patient unable to call for medical staff in a time of emergency. Therefore, it is necessary that additional security and safety measures are in place to prevent these critical devices from failing or becoming compromised.

2.5 Crossfire Attack

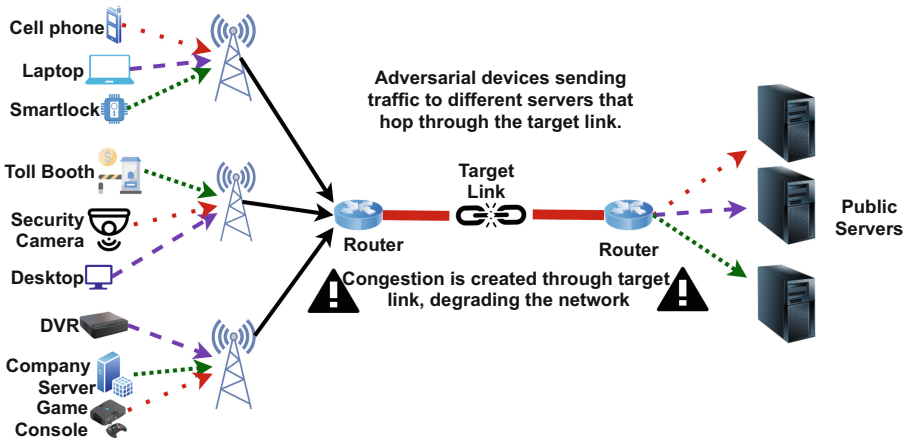


Fig. 2. Execution of the crossfire attack

The crossfire attack is a type of Link Flooding attack that attempts to degrade or disable connections to a specific geographical region of the internet. This attack is perpetuated by directing low-intensity requests to various public servers that share a common link to flood that shared link. The attack uses multiple attacking nodes, each sending low-intensity traffic to different destination nodes. This type of attack does not affect one specific destination, instead, it targets a large geographical region, served by the target link node. This type of attack is devastating to a specific geographical region, as both upstream and

downstream traffic is affected [7]. The crossfire attack is more difficult to detect, as crossfire is an *indirect* attack, contrary to most other attacks. Since the attack spreads low-intensity traffic to various destinations, this allows the attack traffic to blend in with legitimate traffic and is virtually undetectable in standard DoS detection and mitigation protocols, at least until after substantial damage has been done [7].

To execute a crossfire attack, first, a potential adversary would select a list of public servers within the targeted areas and a set of decoy servers away from the target area. Since these servers are publicly accessible, they can be easily found. Next, the adversary would have to perform reconnaissance and generates a link map. This link map would be a map of layer 3 links that connect their decoy servers to public servers. The map enables the adversary to select a set of “target servers” that when flooded, would be able to cut off the target area from the open internet. The adversary then coordinates the decoy servers to flood the target link, effectively blocking most flows to the target area. Each individual server sends low-intensity traffic so as to not arouse suspicion of anomaly-based detection system. The individual low-intensity flows are indistinguishable from legitimate traffic. Finally, the adversary begins to flood each target link one at a time to not trigger automatic packet route mutation [7].

2.6 Related Works

MTD. One proposed crossfire defense solution is the moving target defense (MTD). Traditional networks are static in nature, allowing attacks to spend as much time as needed to gather information and find vulnerabilities [4]. MTD has historically been a warfare strategy but recently has been adopted into the IT world. A moving target defense can first delay network mapping and reconnaissance. ICMP and UDP scans can be disrupted by imposing fake hosts and services on random ports that do not exist. The fake listeners can increase the time and workload an attacker would need in order to launch an attack [6]. Additionally, a mutable network can be created that changes IP addresses and ports of network hosts dynamically. This means that machines will not be able to be located at the same address at any given time [4]. Oftentimes, MTD takes the form of random route and address mutations. Randomization has been a common strategy in moving target defenses. This strategy is based on the idea that if addresses of targets within the network constantly change or access policies between the attacker and target change, then the attack success rate will drastically reduce. An attacker’s ability to effectively do reconnaissance is sharply diminished as well due to the ever-changing network landscape [6]. Obfuscation of links via SDN has also been proposed to confuse and thwart attackers [2]. By obfuscating links between the attacker and target, an adversary would not be able to identify common links, a key step in performing a crossfire attack.

Rerouting-Based Defenses. Moving target defense can also be used to create virtual “routable IPs”. While the real IPs of hosts remain unchanged and static,

the virtual IPs assigned by the network consistently changed frequently and synchronously. However, the higher the rate of mutation, the more overhead is required to run the network [4]. This type of approach is often used by load-balancers to send traffic to different destinations depending on network load.

Another proposed way to mitigate large-scale DDoS attacks is by using a routing around congestion (RAC) defense. The RAC defense works by routing traffic between a service deployer and a critical autonomous system around degraded links. RAC defense asserts that attack traffic is irrelevant and does not need to be filtered when using this defense [14]. The RAC defense offers path isolation by dynamically creating detour routes for critical flow [15].

Infeasibility of Current Mitigation Techniques. While the proposed defense solutions may work in theory, they are infeasible in nature. Rerouting-based defenses like RAC are not feasible in production servers. RAC defense uses border gateway protocol poisoning to avoid specific autonomous systems. The current border gateway protocol is incompatible and may not be able to be updated in such a way as to make this defense feasible. Even so, if this defense were made possible, it could be misused with malicious intent to attack autonomous systems [15]. Rerouting-based defense may also be able to be hijacked to force rerouting constantly. This in practice may cause packets to get dropped or delayed. Additionally, the aforementioned overhead required to implement a moving target defense may not be practical on large-scale networks.

Current Detection Efforts. Current defense mechanisms treat both legitimate and attack traffic the same, degrading the performance of legitimate users. Current attack traffic detection methods point to detecting DoS and DDoS attacks, not link-flooding attacks. These detection efforts often rely on traffic being directed to a singular end-point. Therefore, models that detect standard DoS and DDoS attacks may not be able to accurately detect crossfire attack traffic.

One study, Narayanadoss et al. [10] proposes a deep-learning model to detect crossfire attacks in intelligent transport systems. This study provides a machine-learning-based model to detect vehicles in a network that are involved in the attack. The created models reflected a detection rate of 80% in a network of 35 nodes [10]. This model includes data irrelevant to traditional networks (vehicle speed) that may impact the model's accuracy in networks that are not intelligent transport systems. Additionally, as the network grew, detection accuracy decreased. Only a maximum of 35 nodes were implemented. As noted by the author, as the number of nodes increased, “[m]any legitimate flows could be detected as part of attacking traffic as they may have a temporal correlation with other attacking flows” [10]. This model may prove infeasible in larger networks, such as networks in large cities. Beyond this singular model, significant work has not been done to detect crossfire attacks and classify traffic based on characteristics.

3 Threat Model

To successfully execute a crossfire attack, adversaries must be able to mask malicious traffic behind legitimate traffic to avoid detection by the system. The aforementioned mitigation techniques focus on adversaries using the same attack nodes repeatedly. If an adversary were to have a botnet sufficiently large, they would be able to slowly introduce attack nodes into the attack, and remove attack nodes that have become ineffective. By staying within the confines of thresholds, an attack could be executed for longer without being detected.

During the reconnaissance phase of the attack, an adversary would use multiple attack nodes to execute *trace route* commands. These commands would be done at low intensity, and low rate to avoid route mutation from the SDN. By spreading out these *trace route* commands, common links can be discovered and mapped with minimal error due to route mutation. This would allow the adversary to create a route map, and understand where secondary nodes are used when the primary link is being flooded. For maximum efficiency, the adversary would choose a time during routine peak demand, as SDNs would anticipate this stress on the system, and additional strain may be attributed to regular demand fluctuations.

Once launching the attack, the nodes would monitor the route of the low-intensity traffic to destination servers, to ensure that the traffic is being routed through the link node. If a node determines that it has been rerouted, it shall continue directing traffic to the target node, and wait before disconnecting and changing targets. This can prevent the “common denominator” defense. The moving target defense can be leveraged in itself to disrupt legitimate traffic. By forcing the SDN to continually change routes, legitimate traffic can be slowed down beyond usability.

The adversary would not launch all attack nodes at once, as this may cause a spike in demand, which the system would detect. Instead, the adversary would gradually increase attack nodes in order to mask the increase in demand as organic demand, thereby potentially circumventing anomaly-based detection [7].

As the attack propagates on the network, constant monitoring of network routing would be required. As the system responds to the attack, we would monitor the change in performance during route mutation, and when the attack is taking place undetected. This would allow for the practicality of leveraging route mutation-based mitigation to be measured.

4 Defense Mechanism

Since crossfire attacks are so lethal, it is important to detect when they are occurring as soon as possible. Therefore, using a software defined network (SDN) is ideal so that a holistic view of the network can be obtained. The use of SDN is proposed as an ideal approach to obtain a holistic view of the network. In an SDN, the control plane is decoupled from the data plane, allowing centralized control and management of the network. The SDN architecture consists of

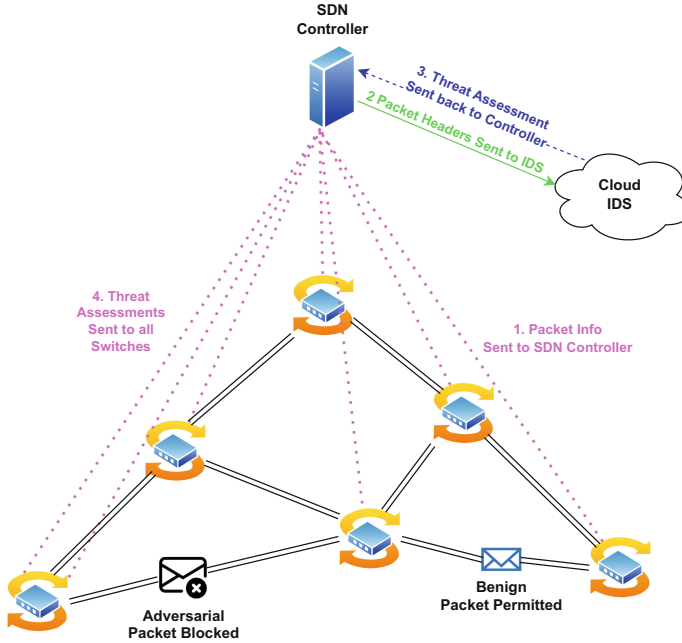


Fig. 3. Defense Mechanism

OpenFlow switches that forward network traffic based on instructions received from the SDN controller. Each OpenFlow switch in the network reports the packet headers of incoming packets to the SDN controller. Packet headers contain important information such as source and destination IP addresses, transport protocol, port numbers, etc. By inspecting these headers, the SDN controller can gain visibility into the network and analyze the characteristics of the packets flowing through it. To defend against crossfire attacks, a traffic classification model is proposed to determine whether a packet is adversarial (part of the attack) or benign. This model is implemented on the SDN controller. It leverages machine learning or rule-based techniques to analyze the packet headers and make an informed decision about the nature of the packet. The SDN controller sends the packet headers to a cloud-based IDS for further analysis. The IDS hosts the proposed traffic classification model, which evaluates the received packet headers and determines if they correspond to an adversarial or benign packet. The IDS is equipped with computational resources and advanced analysis techniques to perform this task effectively. Once the IDS determines that a packet is adversarial, the SDN controller instructs the respective OpenFlow switch to drop the offending packet(s) from the processing pipeline. By discarding the malicious packets, congestion on the network can be reduced, preventing the crossfire attack from spreading further. In summary, this defense mechanism combines the capabilities of SDN, packet header inspection, a traffic classifica-

tion model, and a cloud-based IDS to detect and mitigate crossfire attacks. By inspecting packet headers, identifying adversarial packets, and dropping them in real time, the mechanism helps protect the network from the detrimental effects of crossfire attacks, minimizing potential damage and maintaining network performance. Figure 3 details the mechanism.

5 Experiment Setup

The execution of a crossfire attack, in theory, appears straightforward. By flooding a specific link, the attacker aims to overwhelm it with traffic. This process involves identifying potential routes that utilize the targeted link and generating low-intensity requests across those routes to flood the link effectively. However, in practice, executing a crossfire attack can be challenging due to the limited availability of information regarding the specific routes taken by packets. The lack of public access to this crucial routing data presents a significant hurdle for attackers attempting to orchestrate such attacks. In this section, we delve into the intricacies of executing a crossfire attack, exploring the methodologies used to overcome these obstacles and the implications of this type of attack on network performance and security.

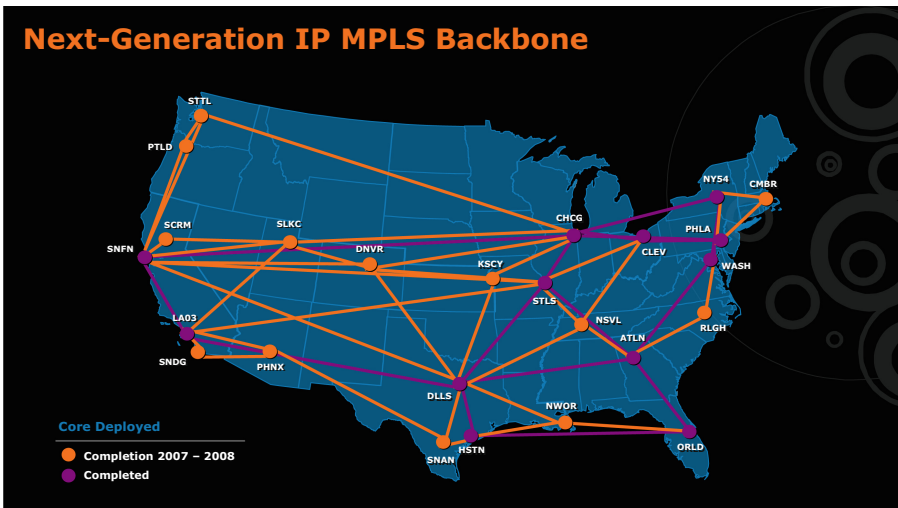


Fig. 4. Test Network Diagram [9]

Our crossfire attack was executed on a network simulated by the MiniNet network simulator and the RYU SDN controller. These tools allowed for the creation and management of a virtual network environment for experimentation and analysis. To set up the network, a python script was employed, which utilized the ATT North America Backbone network from The Internet Topology

Zoo [9] as the basis for the network configuration. The network topology, as depicted in Fig. 4, provides a visual representation of the structure and inter-connections of the various network components. It showcases the arrangement of links within the simulated network. To introduce additional functionality and explore specific scenarios, a Mobile Edge Computing (MEC) server was strategically placed between the ORLD and CLEV nodes. The MEC server served as a centralized computing platform that brought computing resources closer to the network edge, enabling efficient processing and analysis of data generated within the network. In this particular setup, the MEC server had a direct connection between the ORLD and CLEV nodes, facilitating seamless communication and data exchange between them. The choice of the ATT North America Backbone network from The Internet Topology Zoo [9] was driven by its complexity and size, which allowed for a more realistic simulation of network traffic. By utilizing a network with a sufficient number of components and diverse connections, researchers and analysts could better understand and evaluate the performance, scalability, and security aspects of network systems under various conditions.

5.1 Executing the Attack

Once the network setup is complete, the testing scenario involves a sequence of events. First, a ping request is sent from the NY54 node, which represents an external connection, to the MEC (Multi-Access Edge Computing) server. This initial interaction confirms the connectivity between these nodes.

Following the establishment of the network, servers within the network begin initiating HTTP connections randomly across the infrastructure. Approximately 80% of the servers are engaged in requesting HTTP resources at any given time. This random traffic generation simulates unpredictable and legitimate network activity, replicating real-world usage patterns.

After a period of 30s dedicated to legitimate traffic flows, the attack commences. Multiple zombie servers, compromised devices controlled by the attacker, start streaming video traffic over TCP (Transmission Control Protocol) connections to each other. TCP is deliberately chosen for this attack to obscure the nature of the traffic being transmitted. To further obfuscate the content, the videos are streamed over HTTPS (Hypertext Transfer Protocol Secure), making it difficult to distinguish the packets as video traffic based on their packet types.

Each zombie server strategically selects its destinations, ensuring that the attacking video packets pass through, but do not end at, either the ORLD or CLEV nodes. This strategic routing aims to block external connections to the MEC server. Consequently, the switches connecting these networks experience congestion due to the significant volume of data flowing through each link.

The attack is executed in three phases, with a third of the zombie servers initiating the attack during each phase. This staged approach helps distribute the attack traffic and potentially evade detection or mitigation measures.

Throughout the entire process, packet headers are captured using Wireshark, a widely used network protocol analyzer. Despite the use of HTTPS, which encrypts the content of the packets, the packet headers remain visible. Therefore,

in this scenario, the network would only have access to information contained in the packet headers to analyze and identify the attack.

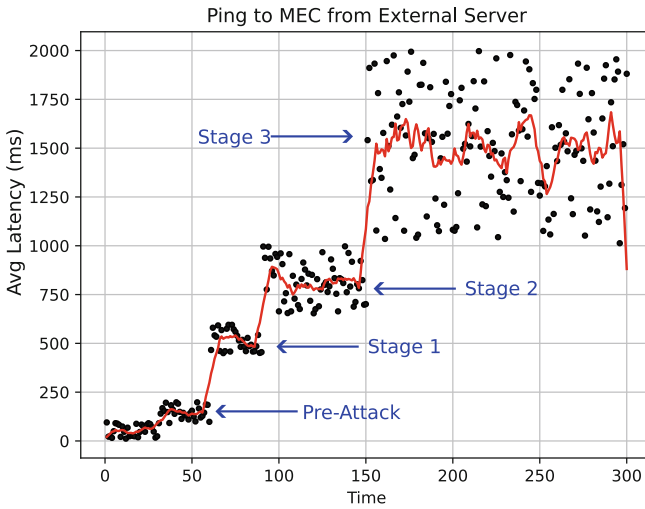


Fig. 5. Average Latency Over Time

5.2 Attack Impact

After running the network for 5 min, the ping round trip times were collected and plotted in Fig. 5. The moving average was plotted as a line. As the attack continues, the ping increases on average. During the first stage, the average ping remained around 50–100 ms. Once the zombie servers began attacking, the average increased to about 500 ms. After the second phase, the ping increases to about 800 ms. Finally, during the third phase the ping increases and hovers around 1500 ms.

6 Crossfire Detection

Detecting a crossfire attack directly can be difficult. Since HTTPS encrypts the packets, the contents of the packets cannot be inspected. Only the headers of each packet are able to be inspected. Therefore, each model created only analyzes the headers of the packets and makes decisions based on those headers.

After running the experiment described in Sect. 5, packet headers were collected for every packet sent on the network. About 30,000 packets were collected. After the packets were collected, the data was aggregated. First, a standard Analysis of Variance (ANOVA) was conducted. After conducting the initial ANOVA.

Table 1. Initial ANOVA Full Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	176.18978	12	352.3796	<.0001
Full	408.10276			
Reduced	584.29254			

Table 2. Initial ANOVA Parameter Estimates

Term		Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	Unstable	-6.18762050	89752.103	0.00	0.9999
tcp.window_size		0.00057399	8.7147e-5	43.38	<.0001
tcp.len		-0.00051130	0.0003489	2.150	0.1428
tcp.stream		-0.08826740	0.0343179	6.620	0.0101
tcp.flags[0x00000010]	Unstable	-15.9381260	89752.102	0.000	0.9999
tcp.flags[0x00000011]	Unstable	9.78867015	116551.53	0.000	0.9999
tcp.flags[0x00000012]	Unstable	12.0348821	117261.45	0.000	0.9999
tcp.flags[0x00000018]	Unstable	-15.0194920	89752.102	0.000	0.9999
tcp.analysis.ack_rtt		-63.1695000	7.2849454	75.19	<.0001
frame.time_relative		0.11828939	0.0393738	9.030	0.0027
frame.time_delta		1.88407238	1.1200274	2.830	0.0925
tcp.time_relative		25.6221427	3.1284271	67.08	<.0001
tcp.time_delta		3.24388117	4.7044249	0.480	0.4905

6.1 Analysis of Variance (ANOVA)

An analysis of variance was first performed with all the data. The initial ANOVA gave the results depicted in Table 1 and Table 2. As shown, the largest predictor of adversarial packets were the window size, the ack_rtt, and the time_relative.

After running the original ANOVA, we removed any non-significant factors to achieve the following ANOVA shown in Table 3 and Table 4. The test as a whole is able to determine whether or not a packet is adversarial based on the window size, ack_rtt, frame time, and tcp time. This model may not be practical given a node that consistently has a significant delay. If a node has significant delay, all packets may be marked as adversarial. Additionally, during an attack, the delay of packets through the congested links may present a problem where the model detects all packets as adversarial and blocks essentially all connections, worsening the effects of the attack.

Table 3. Revised ANOVA Full Model Test

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	105.98066	4	211.9613	<.0001
Full	478.31188			
Reduced	584.29254			

Table 4. Revised ANOVA Parameter Estimates

Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	-13.9094560	3.0268445	21.120	<.0001
tcp.window_size	0.000396320	6.8154e-5	33.810	<.0001
tcp.analysis.ack_rtt	-38.0040650	3.7611744	102.10	<.0001
frame.time_relative	0.01735100	0.0023833	53.000	<.0001
tcp.time_relative	15.8395049	2.3819797	44.220	<.0001

6.2 Neural Network

A neural network was also created based on all the criteria. The diagram for the neural network is drawn in Fig. 6. The confusion matrices for training and validation data are pictured in Table 5 and Table 6. The neural network correctly predicted 99.82% of legitimate packets and 95.3% of attack packets correctly in the training data. In the validation data, the model correctly identified 99.81% of legitimate packets and 95.88% of attack packets in the validation data. An additional neural network was created with 25 nodes by 2 layers, which yielded negligibly better results. The second model only yields an extremely limited increase. Tables 7 and 8 Since these devices are IoT devices and have limited processing power, keeping the neural network model as minimal as possible is ideal. Therefore, using a smaller model for IoMT devices is the better approach.

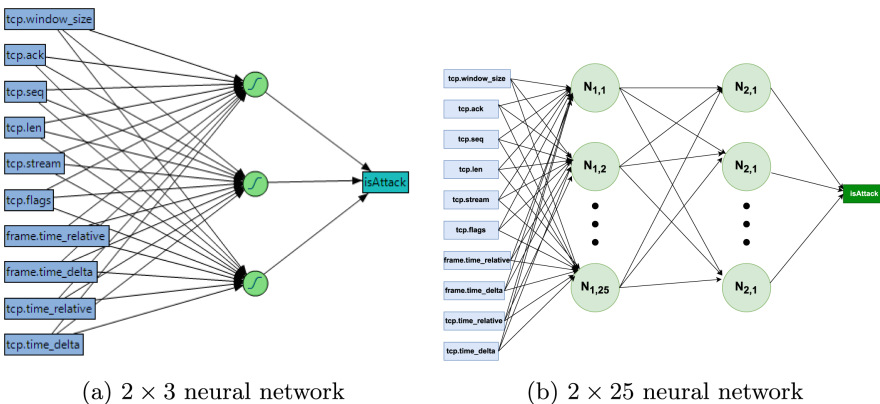


Fig. 6. Neural Network Diagrams Using Hyperbolic Tangent Nodes

Table 5. Confusion Matrix for Training Data for 1×3 neural network

		Predicted	
		Adversarial	Benign
Actual	Adversarial	19206	33
	Benign	242	4928

Table 6. Confusion Matrix for Validation Data for 1×3 neural network

		Predicted	
		Adversarial	Benign
Actual	Adversarial	6373	12
	Benign	72	1679

Table 7. Confusion Matrix for Validation Data with 2×25 neural network

		Predicted	
		Adversarial	Benign
Actual	Adversarial	19308	28
	Benign	140	4933

Table 8. Confusion Matrix for Validation Data with 2×25 neural network

		Predicted	
		Adversarial	Benign
Actual	Adversarial	6375	10
	Benign	31	1720

7 Conclusion

As the internet continues to evolve and become increasingly essential to daily lives, the threat of cyber attacks, particularly Denial of Service (DoS) attacks, looms large. The rise of Internet of Things (IoT) devices, including Internet of Medical Things (IoMT) devices, has further exacerbated the need for robust security measures to protect critical networks, such as those in hospitals and healthcare systems. The advent of Mobile Edge Computing (MEC) servers has addressed some of the challenges posed by the growing number of IoT devices by processing data near the edge of the network, reducing the strain on the central network infrastructure. However, this progress has also introduced new vulnerabilities that can be exploited by attackers. One of the emerging threats is the crossfire attack, a sophisticated and difficult-to-detect type of DoS attack. The crossfire attack targets a specific geographical region by flooding low-intensity traffic from multiple devices, causing congestion and disrupting network operations. Traditional DoS detection and mitigation protocols struggle to identify and counter this type of attack effectively.

While Software Defined Networks (SDNs) have been proposed as a promising mitigation technology for DoS attacks, they are not without their vulnerabilities and limitations. Therefore, it is crucial to explore multiple approaches and strategies to protect critical servers from these evolving threats. The security concerns surrounding IoT and IoMT devices must be addressed to prevent potential intrusions and compromises. The low cost and default login credentials of many IoT devices make them attractive targets for attackers. Robust security measures and safety protocols should be implemented to ensure the integrity and reliability of these critical devices. Moving forward, the adoption of moving target defense (MTD) strategies, such as route and address mutation, and the use of obfuscation techniques can enhance network security and make it more challenging for attackers to carry out crossfire attacks. Rerouting-based defenses

and the deployment of intrusion detection systems that inspect packet headers can also contribute to the detection and prevention of adversarial packets. As the number of interconnected devices continues to grow, with an estimated 60 billion devices expected to be connected by 2025, the importance of securing critical servers and networks cannot be overstated. Ongoing research and collaboration among cybersecurity experts, network administrators, and device manufacturers are essential to developing effective defense mechanisms and ensuring the uninterrupted operation of vital services, particularly in healthcare settings. Overall, protecting critical servers from DoS attacks, including the evolving crossfire attack, requires a multi-faceted approach that combines advanced technologies, robust security protocols, and proactive defense strategies. By addressing these challenges and investing in cybersecurity measures, we can safeguard the integrity and reliability of the Internet and its essential services for the benefit of all.

References



1. Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T.: Mobile edge computing: a survey. *IEEE Internet Things J.* **5**(1), 450–465 (2017)
2. Aydeger, A., Saputro, N., Akkaya, K., Rahman, M.: Mitigating crossfire attacks using SDN-based moving target defense. In: 2016 IEEE 41st Conference on Local Computer Networks (LCN), pp. 627–630 (2016). <https://doi.org/10.1109/LCN.2016.108>
3. Balaji, S., Nathani, K., Santhakumar, R.: IoT technology, applications and challenges: a contemporary survey. *Wirel. Pers. Commun.* **108**(1), 363–388 (2019). <https://doi.org/10.1007/s11277-019-06407-w>
4. Gudla, C., Sung, A.H.: Moving target defense application and analysis in software-defined networking. In: 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 641–646 (2020). <https://doi.org/10.1109/IEMCON51383.2020.9284847>
5. Gupta, A., Jha, R.K.: A survey of 5G network: architecture and emerging technologies. *IEEE Access* **3**, 1206–1232 (2015). <https://doi.org/10.1109/ACCESS.2015.2461602>
6. Kampanakis, P., Perros, H., Beyene, T.: SDN-based solutions for moving target defense network protection. In: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, pp. 1–6 (2014). <https://doi.org/10.1109/WoWMoM.2014.6918979>
7. Kang, M.S., Lee, S.B., Gligor, V.D.: The crossfire attack. In: 2013 IEEE Symposium on Security and Privacy, pp. 127–141 (2013). <https://doi.org/10.1109/SP.2013.19>
8. Kavre, M., Gadekar, A., Gadhade, Y.: Internet of Things (IoT): a survey. In: 2019 IEEE Pune Section International Conference (PuneCon), pp. 1–6 (2019). <https://doi.org/10.1109/PuneCon46936.2019.9105831>
9. Knight, S., Nguyen, H., Falkner, N., Bowden, R., Roughan, M.: The internet topology zoo. *IEEE J. Select. Areas Commun.* **29**(9), 1765–1775 (2011). <https://doi.org/10.1109/JSAC.2011.111002>
10. Narayanadoss, A.R., Truong-Huu, T., Mohan, P.M., Gurusamy, M.: Crossfire attack detection using deep learning in software defined its networks. In: 2019

- IEEE 89th Vehicular Technology Conference (VTC2019-Spring), pp. 1–6 (2019). <https://doi.org/10.1109/VTCSpring.2019.8746594>
11. Panwar, N., Sharma, S., Singh, A.K.: A survey on 5G: the next generation of mobile communication. *Phys. Commun.* **18**, 64–84 (2016)
 12. Patel, M., et al.: Mobile-Edge Computing - Introductory Technical White Paper (2019). <https://www.scirp.org/reference/ReferencesPapers.aspx?ReferenceID=2580032>
 13. Patton, M., Gross, E., Chinn, R., Forbis, S., Walker, L., Chen, H.: Uninvited connections: a study of vulnerable devices on the Internet of Things (IoT). In: 2014 IEEE Joint Intelligence and Security Informatics Conference, pp. 232–235. IEEE (2014)
 14. Smith, J.M., Schuchard, M.: Routing around congestion: defeating DDoS attacks and adverse network conditions via reactive BGP routing. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 599–617 (2018). <https://doi.org/10.1109/SP.2018.00032>
 15. Tran, M., Kang, M.S., Hsiao, H.C., Chiang, W.H., Tung, S.P., Wang, Y.S.: On the feasibility of rerouting-based DDoS defenses. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 1169–1184 (2019). <https://doi.org/10.1109/SP.2019.00055>
 16. Vishnu, S., Ramson, S.J., Jegan, R.: Internet of medical things (IoMT)-an overview. In: 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), pp. 101–104. IEEE (2020)
 17. Wang, C.X., et al.: Cellular architecture and key technologies for 5G wireless communication networks. *IEEE Commun. Mag.* **52**(2), 122–130 (2014). <https://doi.org/10.1109/MCOM.2014.6736752>

IoT for Wearables and Smart Devices (IWS)



Prediction of Tomato Leaf Disease Plying Transfer Learning Models

B. S. Vidhyasagar¹ , Koganti Harshagnan¹ , M. Diviya¹  ,
and Sivakumar Kalimuthu² 

¹ Department of Computer Science and Engineering,
Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India
{bs_vidhyasagar, m_diviya}@ch.amrita.edu,
ch.en.u4cse19013@ch.students.amrita.edu

² Faculty of Computer Science and Information Technology (FOCSIT),
University Putra Malaysia, Seri Kembangan, Malaysia

Abstract. The tomato has a high market value and is one of the vegetables grown in the most significant quantity globally. Tomato plants are susceptible to diseases, which can negatively impact the fruit's yield and quality. Detecting these illnesses at an early stage and their accurate identification is necessary for successfully managing diseases and reducing losses. In recent years, deep learning methods such as convolutional neural networks (CNNs) have demonstrated significant promise in identifying plant diseases from images. This research suggested a CNN-based strategy for detecting tomato leaf diseases using transfer learning. Transfer learning enables us to enhance the performance of our disease detection model using a smaller dataset by leveraging pre-trained CNN models that have been trained on large datasets. The proposed transfer learning model through Resnet50 and Inception V3 is effective by applying it to a dataset of tomato leaf images. As a result, a high level of accuracy is achieved and could be indulged for practical applications in agriculture.

Keywords: Resnet50 · Inception V3 · Transfer Learning · CNN · Tomato leaf disease · deep learning

1 Introduction

Tomato plants are vulnerable to various diseases that can severely harm the plant's leaves, fruit, and overall health. These diseases may result from multiple causes, including bacterial, fungal, and viral infections, nutrient deficiencies, and extreme weather conditions. Some common tomato leaf diseases include early blight, late blight, Septoria leaf spot, bacterial spot, and tomato yellow leaf curl virus, which can cause symptoms like yellowing and wilting of leaves, brown spots on leaves, stem cankers, and stunted growth [1]. Farmers can take preventative measures such as proper plant spacing, adequate watering, and regular sanitation practices to minimize the risk of tomato leaf disease. Additionally, early detection and timely treatment with suitable fungicides or bactericides can help control the spread of these diseases and ensure a healthy tomato crop.

© IFIP International Federation for Information Processing 2023

Published by Springer Nature Switzerland AG 2023

D. Puthal et al. (Eds.): IFIPIoT 2023, IFIP AICT 683, pp. 293–305, 2023.

https://doi.org/10.1007/978-3-031-45878-1_20

A significant agricultural advancement is automated systems for detecting tomato diseases by analyzing tomato leaves. Detecting and treating tomato diseases promptly and efficiently can positively impact crop productivity and quality. However, identifying diseases in tomatoes can be challenging for experienced agriculturists and pathologists may struggle to detect illnesses by observing diseased leaves due to the vast range of crops grown [2]. In countrified areas of blossoming countries, visual inspection is still the primary method of disease detection, and farmers may have to make time-consuming and expensive trips to experts' offices. Therefore, deep learning techniques, such as convolutional neural networks (CNNs), have become famous for image classification tasks, including detecting plant leaf diseases. They offer a more efficient and cost-effective solution to identifying and treating tomato leaf diseases.

Convolutional neural network (CNN) offers the most promising approach for disease detection to learn, decide and discriminate features from data automatically. The model consists of multiple convolutional layers that realize various elements from the input data. These models can be applied to detect plant diseases with high accuracy. However, deep learning has limitations as it requires significant data to train the network effectively. Performance may suffer if the dataset is insufficient in size and does not contain enough images for the model to learn from. As such, we can use transfer learning which offers numerous advantages, one of which is that it doesn't demand a large volume of data to train the network effectively. Transfer learning enhances the learning process by leveraging the knowledge gained from a previously learned task, allowing for knowledge transfer to the current task [4]. Neural networks also used in many research studies have utilized Transfer learning in disease detection strategies diagnose the human diseases as well, approving to be a beneficial technique [5]. Some of the advantages of adopting Transfer learning for disease detection include the following:

- **Reduced training time:** Reusing pre-trained models can significantly reduce the time necessary to train a new model. It is one of the primary benefits of transfer learning that comes in particularly handy in situations where a limited quantity of labeled data can be used for training.
- **Improved generalization:** Model's generalization performance by capitalizing on the information gained from a pre-trained model. It can improve performance on new data that has not been seen before.
- **Reduced computational cost:** The cost of training a new model from inception by beginning with a model that has already been trained and using that model as a starting point for the learning process. It may be beneficial in settings with limited resource access, such as mobile devices or integrated systems.
- **Robustness:** Because pre-trained models have already been trained on various large datasets, they are more resistant to overfitting and have greater robustness overall. Learning through Transfer enables this robustness to be transferred to the new model, which ultimately results in the new model being more dependable and accurate.
- **Adaptability:** For various applications, such as picture classification, object detection, natural language processing, etc. Because of their adaptability, machine algorithms can be applied in multiple contexts.
- **Improved accuracy:** Improve the accuracy of disease detection models by enabling the model to leverage the knowledge acquired from pre-trained models. This method

allows the model to utilize the information available better. Transfer learning can contribute to more accurate predictions, thereby reducing the number of false positives and negatives.

In this study, we identify tomato illnesses using two deep-learning models.

1.1 Inception v3

Google researchers in 2015 brought into the limelight Inception v3, a convolutional neural network designed to classify images and features in a deep neural network with over 23 layers. The architecture of the model is given in Fig. 1.

Inception v3 uses a unique module called the Inception module, which consists of multiple convolutional filters with different sizes applied to the same input, allowing for efficient feature extraction at different scales [6]. It also incorporates batch normalization and regularization techniques to enhance the accuracy of the architecture.

Inception v3 has accomplished benchmark performance on several image recognition tasks. Additionally, it has a relatively small model size compared to other deep neural network architectures, making it efficient to train and deploy.

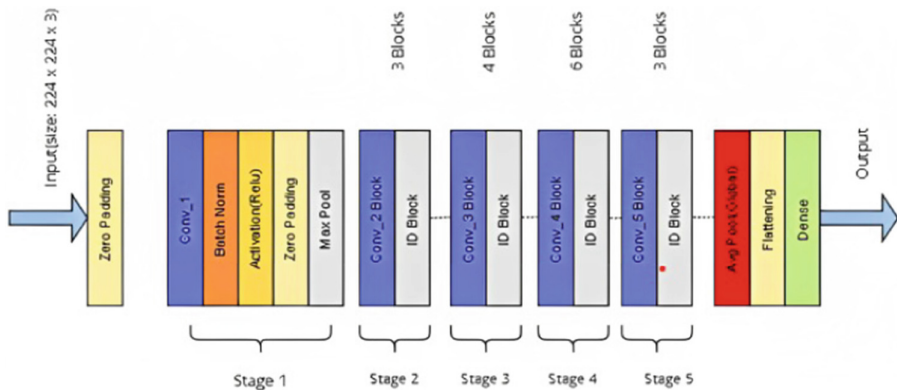


Fig. 1. Inception v3 architecture

1.2 Resnet 50

Researchers from Microsoft devised the ResNet50 convolutional neural network architecture first used in the company's research in 2015. It is a subset of the ResNet (Residual Network) family of deep neural networks and was developed to enhance the training of very deep neural networks with more than 50 levels [6]. ResNet50 uses skip connections, allowing information to flow from earlier layers to the last layers without passing through any non-linear transformations. This makes it much simpler to train very deep networks. It has attained state-of-the-art performance on various image recognition tasks, such as recognizing objects and parsing scenes, among other image recognition tasks. In addition to this, it has been demonstrated to be effective in transfer learning, which is the

process whereby a model that has been pre-trained can be fine-tuned on a new task using relatively small quantities of data. The architecture of Resnet50 is described in Fig. 2.

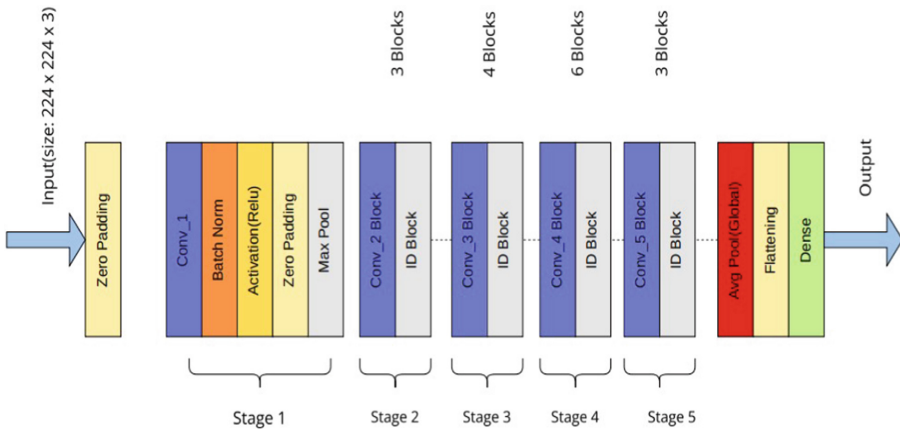


Fig. 2. Resnet50 Architecture

Using these two pre-trained models, train them on the tomato leaf dataset once it has been preprocessed, and then assess how well the trained models perform on the image assessment set. The effective measures, such as accuracy, precision, recall, and F1-score, will be computed to determine how successful the model is. Then, compare these two models to the remaining four models to choose which models have the most outstanding performance out of the six.

2 Related Works

The author primarily emphasizes creating a CNN model that can differentiate between five distinct diseases that can affect tomato leaves. They utilized a dataset containing 1,583 images of healthy tomato leaves and images of leaves that were affected by these diseases according to the research findings, a CNN model that included data augmentation obtained an accuracy of 98.9%, outperforming a control group that used a model that did not have data augmentation, which received an accuracy of 91.2%. The author offers insightful recommendations on the application of CNNs to diagnose diseases affecting tomato leaves and emphasizes the significance of data supplementation techniques for enhancing the model's overall performance [1].

Kurup et al. investigate how capsule networks can be used to classify plant varieties and detect plant diseases. They introduced a novel method that improves the accuracy of plant disease classification and plant species recognition by combining capsule networks and transfer learning. The authors describe the architecture of their suggested model, which is based on the well-known Convolutional Neural Network (CNN) architecture, ResNet-50. The authors have utilized Transfer learning by seeding the model with weights that have already been pre-trained using the ImageNet dataset. The algorithm is then fine-tuned on two different plant datasets, namely PlantVillage and Flavia,

to classify plant diseases and recognize plant species. The capsule network-based model outperformed other state-of-the-art techniques by achieving an accuracy of 98.95% for plant disease classification on the PlantVillage dataset. The suggested model achieved an accuracy of 96.27% when it came to recognizing plant species using the Flavia dataset [2]. A pre-trained CNN model, the VGG16 model, has been demonstrated to perform admirably when given the image classification task. They describe how they got rid of the model's last fully connected layer and used the layer's output before it as input to the SVM classifier has fewer layers. An overall accuracy of 95.8% was achieved on the authors' dataset, which the authors describe as promising results[3]. Followed by various researchers, the Inception-v3 model is used in image categorization tasks. They describe the training and testing process and the metrics used to evaluate the performance of their method and fine-tuned model with the tomato leaf dataset [4].

To identify tomato diseases, the authors applied various state-of-the-art convolutional neural networks (CNNs) classification network architectures, such as ResNet18, MobileNet, DenseNet201, and InceptionV3, to a total of 18,162 plain tomato leaf images. They describe the process of training and testing, as well as the evaluation metrics that were used to evaluate the performance of their method [6]. A deep learning-based technique with Inception V3 architecture is employed for detecting potato leaf diseases. The accuracy of the suggested approach for detecting five different types of potato leaf diseases was 95.12%. The authors propose that this method can be used for early detection and monitoring of potato leaf diseases, which can help reduce crop losses and increase agricultural productivity [8].

Ahmad et al. suggest a method for detecting diseases that affect tomato leaves using pre-trained convolutional neural networks. Four distinct models of CNNs VGG16, ResNet50, InceptionV3, and Xception are used in addition to data augmentation strategies to expand the scope of their information. The authors analyzed the performance of their model in terms of accuracy, precision, recall, and F1 score, and they compared it to the performance of other techniques that are considered to be state-of-the-art. They discovered that the Xception model achieved the highest level of precision, which was 99.6% [9].

Agarwal et al. suggest a method for detecting tomato leaf diseases called ToLeD that makes use of a convolutional neural network to hypothesize that their technique can be adapted to identify a wider variety of plant diseases. Because the dataset that was used in the research is on the smaller side, the generalizability and scalability of the technique may be compromised as a result [10].

Traditional deep learning models have been trained over various plant disease evaluations. When Transfer learning in conjunction with deep neural networks is involved, it has shown a par improvement in diagnosing leaf diseases in grape and mango plants. They used two pre-trained models, namely Inception V3 and ResNet50, and then fine-tuned them using a dataset consisting of grape and mango leaf images. Their approach had an accuracy of 97.4% when detecting three different diseases that can affect grape leaves and 94.6% when detecting two other diseases that can affect mango leaves [11].

A technique for detecting tomato plant diseases using Transfer learning with synthetic images generated by a conditional generative adversarial network is proposed by Abbas et al. The authors suggest that their method can be used for the early detection

and monitoring of tomato plant diseases, which can help increase crop yields and reduce losses. They also suggest expanding their approach to identify other plant diseases utilizing synthetic images generated by C-GAN. It is possible for the synthetic images included in the dataset to introduce noise or biases, both of which can harm the model’s performance. The dataset may be flawed if the synthetic images are not authentic or have artifacts. Training is necessary for both the C-GAN and the classifier models, which are parts of the technique. It may cause the method to become more complicated and call for additional computational resources and training time [12].

The paper proposes a novel method called aGROdet, designed to detect plant diseases and estimate leaf damage severity within an Agriculture Cyber-Physical System implemented at the edge platform of IoT systems. A convolutional neural network-based model trained on large publicly available datasets achieves over 97% accuracy in initial experiments. The study also addresses damage estimation challenges, such as leaf shadows and surrounding areas [17].

The article emphasizes the significance of early detection and disease severity estimation for effective disease management and prevention. The proposed solution suggests a fully automated approach utilizing deep neural networks, specifically the Mask R-CNN network, for disease detection and localization in leaf images. The method employs image augmentation and transfer learning to enhance precision and save time. The approach is applicable to various imaging devices, such as smartphone cameras or low-altitude unmanned aerial vehicle (UAV) cameras. It has been demonstrated using apple leaves as a case study [18].

Table 1. Types of diseases in tomato plant

S.No	Type	Plant	Diseases	Number Of Images (Train + Valid)
1	Healthy	Tomato	Healthy	1926 + 481
2	Diseased		Bacterial Spot	1702 + 425
3			Early Blight	1920 + 480
4			Late Blight	1851 + 463
5			Leaf Mold	1882 + 470
6			Septoria Leaf Spot	1745 + 436
7			Two Spotted Spider Mites	1741 + 435
8			Target Spot	1827 + 457
9			Yellow Curl Virus	1961 + 490
10			Mosaic Virus	1790 + 448

3 Dataset

Each image in the Plant Village dataset is associated with a label indicating the type of plant and the type of disease or pest. The dataset covers over 20 crops, including tomato, potato, apple, grape, and soybean, and includes more than 80 different plant diseases and pests. The dataset also contains additional metadata, such as geographic location, disease severity, and time of the year the image was captured. We separated the dataset into ten directories with ten different tomato leaf diseases with 22,998 images. The ten directories of tomato leaf images are taken for the project and divided into training and validation datasets. The training dataset has 18403 images, and the validation dataset has 4595 images belonging to 10 classes. The details is depicted in Table 1.

4 Methodology

4.1 Resnet50

The dataset was obtained from Kaggle and is named plant village. Before training the model, Images are preprocessed to ensure they were in a standard format. Firstly, resized the images to 224x224 pixels to reduce the computational complexity of the model. Then converted the images to RGB format to ensure they were in the same color space as the pre-trained ResNet 50 model. Lastly, normalization of the pixel values is done to ensure they were between 0 and 1, which helped the model converge faster during training. To further improve the model's performance and prevent overfitting, we augmented the dataset using various data augmentation techniques such as random rotation, flip, and zoom. This helped increase the dataset's size and improve the model's ability to generalize to new data.

Transfer learning is employed to fine-tune the pre-trained ResNet 50 model on the tomato disease dataset. Frozing the initial layers of the model and retraining the last few layers to adapt to the new dataset. This approach allowed to leverage the pre-trained weights of the ResNet 50 model, which were learned from millions of images from the ImageNet dataset, and apply them to our tomato disease detection task. During training, experimentation with different hyperparameters, such as batch size and optimizer, has been done to find the optimal combination with the best performance. Various 32 batch sizes and Stochastic gradient descent (SGD) optimizers are used for the proposed model's best hyperparameters. Finally, the model's performance is evaluated on a hold-out validation set by calculating accuracy, precision, recall, and F1-score metrics. This helped to assess the model's ability to detect tomato diseases accurately and efficiently. By evaluating the model's performance on a separate validation set, an evaluation on how well the model would perform on new, unseen data could be justified. The proposed Transfer learning architecture for Resnet is shown in Fig. 3.

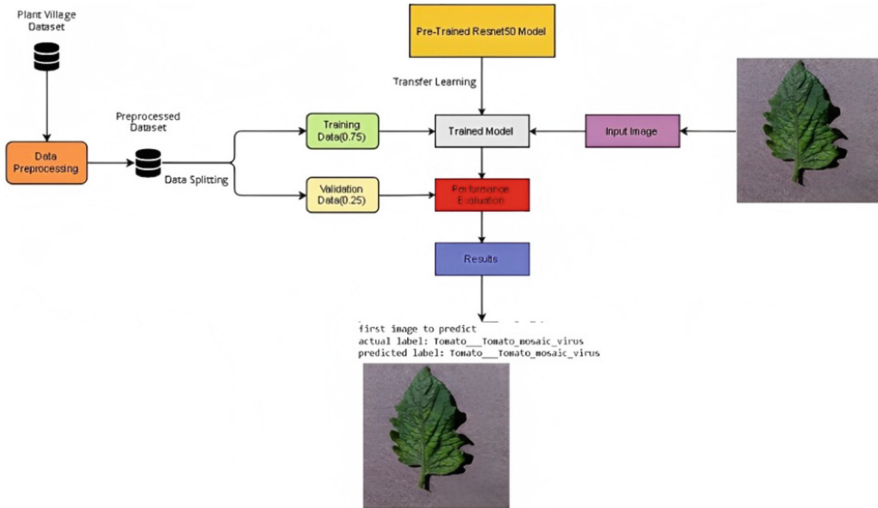


Fig. 3. Proposed Resnet50 model with Transfer Learning

4.2 Inceptionv3

To build a tomato disease detection model using Inceptionv3, the first step is to collect a large dataset of tomato images [13–16]. This dataset should include images of healthy tomatoes and tomatoes affected by various diseases, such as bacterial spots, late blight, and early blight. The dataset should contain various tomato images captured in different lighting conditions and angles. The images can be obtained from multiple sources, such as online image repositories or by taking pictures of actual tomatoes. Once the dataset is done, the images should be preprocessed to ensure consistency and reduce noise in the data. The images should be resized to a standard size, such as 224x224, to ensure that all images have the exact dimensions. The pixel values should be normalized to provide the input data has a similar range of values. Image enhancement strategies, including random rotations, flips, and other transformations, may be utilized to expand the dataset’s size and enhance the model’s performance. Separating the dataset into a training group and a validation set is necessary.

It has been demonstrated that the Inceptionv3 architecture is a robust convolutional neural network (CNN) that is capable of achieving state-of-the-art outcomes when it comes to image classification tasks. Convolutional layers, pooling layers, and ultimately connected layers are some of the types of layers that can be found in the deep neural network known as Inceptionv3. Because the architecture was created to process images of various sizes and resolutions, it is particularly well-suited for detecting tomato diseases. To categorize the input picture into a variety of tomato diseases, the final layer of Inceptionv3 should be removed and replaced with a global average pooling layer, which should then be followed by a fully connected layer that uses softmax activation. Figure 4 Represents the Inception v3 model over transfer learning.

The optimization technique Root Mean Squared Propagation (RMSProp) with a batch size of 32 can be used when training the model. Categorical cross-entropy loss is the objective function that the algorithm is trying to achieve. It is recommended that the model be trained for a total of ten epochs, and early stopping can be utilized to avoid overfitting.

To determine how well the trained model can effectively identify tomato diseases, it should be validated using a dataset that is kept separate from the training dataset. The effectiveness of the model can be evaluated based on several different metrics, including accuracy, precision, recall, and the F1 score.

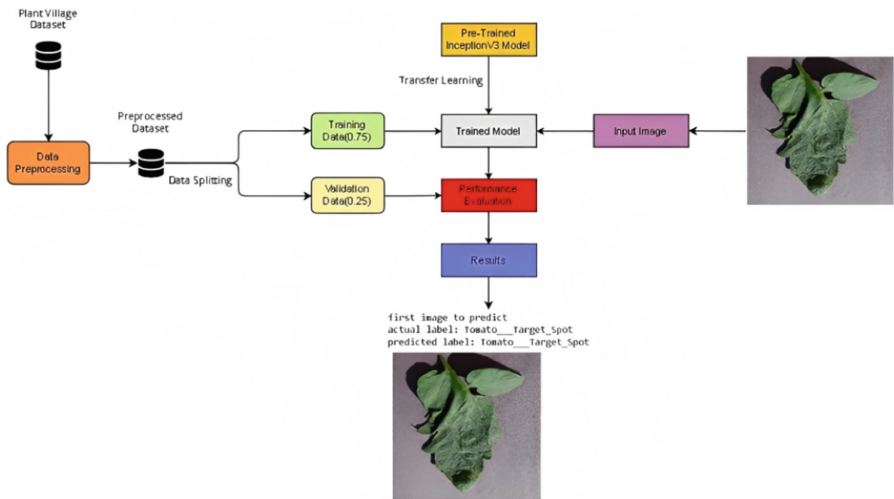


Fig. 4. Proposed Inceptionv3 model with Transfer Learning

5 Results

A Comparison of accuracy and loss of over ten epochs of three models, Resnet50, Alex net, and Lenet has been performed. The graph shows that Inceptionv3 has the highest validation accuracy (0.9535) while Alex net has the lowest (0.5959). However, Resnt50 also had the lowest validation loss (0.1400), while Alex net had the highest (1.1171). Resnet50 performed the best in accuracy and loss, followed by Lenet and Alex net. However, it is essential to note that further testing and analysis would be needed to confirm this conclusion. Figure 5 Shows the graphical representation of accuracy and loss of the proposed model across the state of art models.

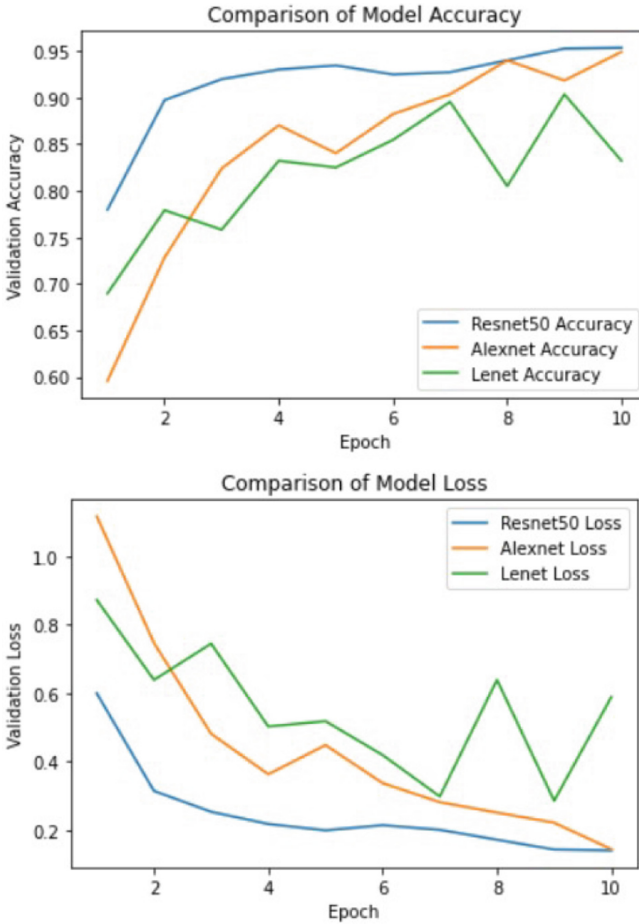


Fig. 5. Results Comparison of Model Accuracy and Loss for Resnet50

The bar graph comparison shows that Inceptionv3 had the highest validation accuracy (0.9876) while Alex net had the lowest (0.5959). However, Inceptionv3 also had the lowest validation loss (0.0414), while Alex net had the highest (1.1171). Overall, Inceptionv3 performed the best in accuracy and loss, followed by Lenet and Alex net. We suggest that Inceptionv3 may be the most effective model for this task. Figure 6 Depicts the performance of the proposed architecture over benchmark models.

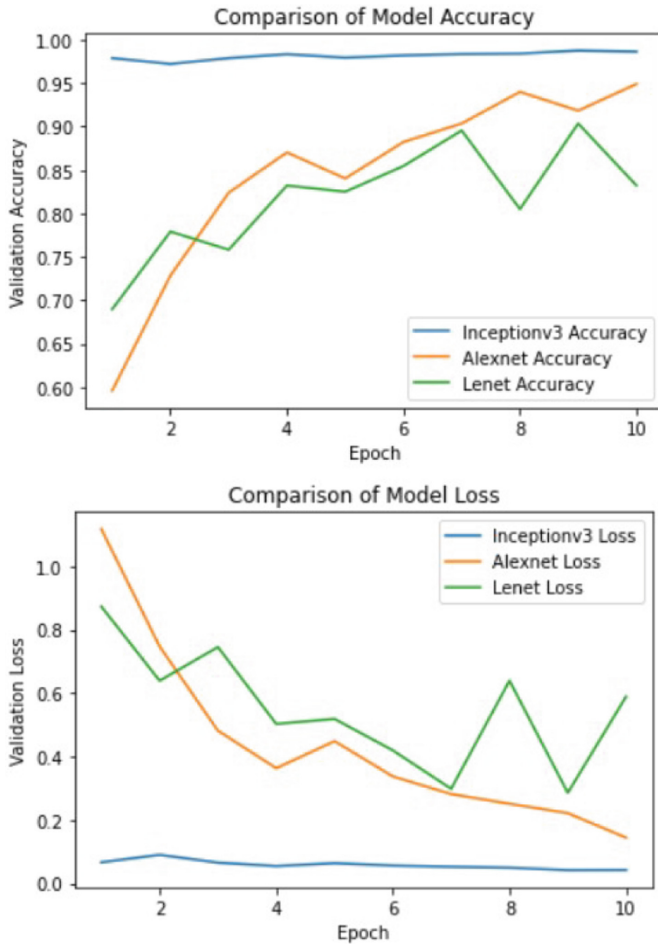


Fig. 6. Results Comparison of Model Accuracy and Loss for InceptionV3

6 Conclusion

Tomato plants are vulnerable to various leaf diseases. Finding these diseases early would save the plant for better yield. The proposed transfer learning models Resnet50 and Inception V3 show the model is effective in disease prediction. In turn, it opens the door to numerous potential applications in agriculture. Transfer learning is on the run, which reduces the training time of a model from scratch as well improves the models' performance. The model can be further enhanced with GAN architectures and extended for predicting various plant diseases.

References

1. N. K. E., K. M., P. P., A. R., V. S.: Tomato Leaf Disease Detection using Convolutional Neural Network with Data Augmentation. In: 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 1125–1132 (2020). <https://doi.org/10.1109/ICCES48766.2020.9138030>
2. Kurup, R.V., Anupama, M.A., Vinayakumar, R., Sowmya, V., Soman, K.P.: Capsule Network for Plant Disease and Plant Species Classification. In: Smys, S., Tavares, J.M.R.S., Balas, V.E., Iliyasa, A.M. (eds.) ICCVBIC 2019. AISC, vol. 1108, pp. 413–421. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-37218-7_47
3. Arunnehr, J., Vidhyasagar, B.S., Anwar Basha, H.: Plant Leaf Diseases Recognition Using Convolutional Neural Network and Transfer Learning. In: Bindhu, V., Chen, J., Tavares, J.M.R.S. (eds.) International Conference on Communication, Computing and Electronics Systems. LNEE, vol. 637, pp. 221–229. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-2612-1_21
4. Salvi, R.S., Labhsetwar, S.R., Kolte, P.A., Venkatesh, V.S., Baretto, A.M.: Predictive analysis of diabetic retinopathy with transfer learning. In: 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), pp. 1–6. IEEE (2021)
5. Jebadas, D.G., Sivaram, M., M, A., Vidhyasagar, B.S., Kannan, B.B.: Histogram Distance Metric Learning to Diagnose Breast Cancer using Semantic Analysis and Natural Language Interpretation Methods. In: Johri, P., Diván, M.J., Khanam, R., Marciszack, M., Will, A. (eds.) Trends and Advancements of Image Processing and Its Applications. EICC, pp. 249–259. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-75945-2_13
6. Xiaoling X., Xu, C., Nan, B.: Inception-v3 for flower classification. In: 2017 2nd International Conference on Image, Vision and Computing (ICIVC), pp. 783–787 (2017)<https://doi.org/10.1109/ICIVC.2017.7984661>
7. Kumar, R., Singh, D., Chug, A., Singh, A.P.: Evaluation of Deep learning based Resnet50 for Plant Disease Classification with Stability Analysis. In: 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, pp. 1280–1287 (2022). <https://doi.org/10.1109/ICICCS53718.2022.9788207>
8. Nawaz, M., Nazir, T., Javed, A., et al.: A robust deep learning approach for tomato plant leaf disease localization and classification. *Sci. Rep.* **12**, 18568 (2022). <https://doi.org/10.1038/s41598-022-21498-5>
9. Chowdhury, E.H., et al.: Tomato leaf diseases detection using deep learning technique. *Technol. Agric.* 453 (2021)
10. Belal A.M.A, Abu-Naser. S.S.: Image-based tomato leaves diseases detection using deep learning (2018)
11. Mosin, H., Tanawala, B., Patel, K.J.: Deep learning precision farming: Tomato leaf disease detection by transfer learning. In: Proceedings of 2nd international conference on advanced computing and software engineering (ICACSE) (2019)
12. Kumari, CU., Jeevan P.S., Mounika, G.: Leaf disease detection: feature extraction with K-means clustering and classification with ANN. In: 2019 3rd international conference on computing methodologies and communication (ICCMC). IEEE (2019)
13. Qiang, Z., He, L., Dai, F.: Identification of Plant Leaf Diseases Based on Inception V3 Transfer Learning and Fine-Tuning. In: Wang, G., El Saddik, A., Lai, X., Martinez Perez, G., Choo, K.-K. (eds.) iSCI 2019. CCIS, vol. 1122, pp. 118–127. Springer, Singapore (2019). https://doi.org/10.1007/978-981-15-1301-5_10
14. Serawork, W., Polceanu, M., Buche, C.: Soybean plant disease identification using convolutional neural network. In: FLAIRS conference (2018)

15. Hasan, M.M.M., et al.: An efficient disease detection technique of rice leaf using AlexNet. *J. Comput. Commun.* **8**(12), 49 (2020)
16. Gunjan, C., et al.: Potato leaf disease detection using inception V3. *Int. Res. J. Eng. Technol (IRJET)* **7**(11), 1363–1366 (2020)
17. Mitra, A., Mohanty, S.P., Kougianos, E.: aGROdet: a Novel framework for plant disease detection and leaf damage estimation. In: *Proceedings of the IFIP International Internet of Things Conference (IFIP-IoT)*, pp. 3–22 (2022)
18. Mitra, A., Mohanty, S.P., Kougianos, E.: A smart agriculture framework to automatically track the spread of plant diseases using mask region-based convolutional neural network. In: *Proceedings of the IFIP International Internet of Things Conference (IFIP-IoT)*, pp. 68–85 (2022)



Video Captioning Based on Sign Language Using YOLOV8 Model

B. S. Vidhyasagar¹✉, An Sakthi Lakshmanan¹, M. K. Abishek¹,
and Sivakumar Kalimuthu²

¹ Department of Computer Science and Engineering,
Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India
bs_vidhyasagar@ch.amrita.edu, {ch.en.u4cse19034,
ch.en.u4cse19001}@ch.students.amrita.edu

² Faculty of Computer Science and Information Technology (FOCSIT),
University Putra Malaysia, Seri Kembangan, Malaysia

Abstract. One of the fastest-growing research areas is the recognition of sign language. In this field, many novel techniques have lately been created. People who are deaf-dumb primarily communicate using sign language. Real-time sign language is essential for people who cannot hear or speak (the dumb and the deaf). Hand gestures are one of the non-verbal communication methods used in sign language. People must be aware of these people's language because it is their only means of communication. In this work, we suggest creating and implementing a model to offer transcripts of the sign language that disabled individuals use during a live meeting or video conference. The dataset utilized in this study is downloaded from the Roboflow website and used for training and testing the data. Transfer Learning is a key idea in this situation since a trained model is utilized to identify the hand signals. The YOLOv8 model, created by Ultralytics, is employed for this purpose and instantly translates the letters of the alphabet (A-Z) into their corresponding texts. In our method, the 26 ASL signs are recognized by first extracting the essential components of each sign from the real-time input video, which is then fed into the Yolo-v8 deep learning model to identify the sign. The output will be matched to the signs contained in the neural network and classified into the appropriate signs based on a comparison between the features retrieved and the original signs present in the database.

Keywords: Sign Language · Roboflow · Transfer Learning · YOLOV8

1 Introduction

Facial expressions, hand gestures, and body movements are among the visual cues utilized in sign language to convey meaning. Sign language is very useful for people who have difficulty hearing or speaking. The process of converting these gestures into words or the rudiments of formally spoken languages is known as recognition of sign language.

Sign language, unlike spoken language, is based on concepts. There are 26 hand symbols used to symbolize each letter of the alphabet in this language, although it cannot be written. These gestures are expressed using the fingers, and words are written out. Words or names are communicated by combining finger spellings and motions. The deaf and dumb use two widely used sign languages that have developed over time: American Sign Language (ASL) and British Sign Language (BSL). A technique called full communication, which incorporates spoken language, lip reading, BSL or ASL, and both, is used in several deaf schools. Hand gestures are a powerful tool for human communication. There are many signs that represent complicated meanings; Therefore, it can be difficult for persons who don't comprehend that language to identify them. Given its increasing popularity, gesture-based communication has been the subject of numerous studies. In the past, sensor-based technology was primarily used for sign language interpretation. Gloves with sensors that link to a recipient on one side are used in this method. However, this strategy has flaws of its own. Convolutional neural networks, deep learning, and AI have all improved signing interpretation communication in this method, to name a few instances. Regional categories for sign language include Indian, American, Chinese, Arabic, and so forth. To improve applications and explain them at the most basic levels, supposedly countries also conduct research on gesture recognition, pattern recognition, and image processing.

2 Literature Survey

The article Real-Time Voice to Sign Language Automatic Translation for the Deaf and Dumb People view the entire approach as a pipeline in which YouTube recordings or videos are delivered as the source and smooth films with sign language are generated and transferred to the appropriate voice [1]. The pipeline was broken up into several smaller sections. The first module uses NLP to process the input video (from YouTube or any other.mp4 file) and captions, extracting the appropriate keyword from the text. The database receives this keyword as 3D symbol signals. Additionally, sign language is represented in the final video. This paper's shortcoming is that it only uses a small-scale framework and the limited gesture-based communication that is available to them. The framework is otherwise adaptable and straightforward to employ in a study hall environment. It features a particular module that makes it simple for the teachers to communicate with the students during exams.

The image frames are first acquired from the video captured via OpenCV. Then the region and the hand gestures are segmented and detected. Finally, the CNN model developed determines the label and converts it to text [2].

The workflow of the model is divided into many phases in the study Real Time Sign Language Recognition and Translation to Text for Vocally and Hearing-Impaired People [3]. When a user uploads a video, it is converted into adequate frames in the first step, which is also where the video procurement takes place. The following stage will preprocess the acquired frames or photos after receiving these frames. In the following step, the frames are preprocessed to get rid of the noise and blur from the video that was shot in poor lighting or under unfavorable circumstances. By using the surrounding entry to replace each one, the median filtering removes the noise and blurriness. The

feature is then derived using the HOG picture processing technique in the following stage. It extracts the feature from the image and outputs vectors, allowing the classification of items using this shape. After features are derived, they are given into the SVM classification, which uses the feature to put them in groups. Utilizing prepared video and group testing video with a specified representation, the characterization supplies the outcome. At the conclusion of the classification procedure, the proportional content portrayal will be provided using the class names that were assigned during the training phase. The sole drawback of this paper is that the algorithm and the picture both play a role in how precisely differentiation is accomplished using this method. Otherwise, it achieves the goal of higher precision and lower processing overhead and has a higher recovery accuracy compared to conventional processing systems.

Sign Language Recognition has been separated into two levels in this research processing level and classification [4]. Real-time images are initially sent to the processing stage, where they are preprocessed by turning from color to grayscale to improve classification. The image is preprocessed by adding noise as needed and eliminating undesired noise with a median filter. The OTSU algorithm is used to extract the features from the image, and the image is then classified into the appropriate sign by comparing it to database images created using the augmentation technique, where each sign will be trained from various angles to produce a classification that is more accurate and precise. SVM (Support Vector Machine), the most well-liked technique for sign language recognition, is the classification approach applied in this case. Also, MATLAB is used to extract the relevant text and voice when the features obtained match the features in the database.

The author insists on a system which involves three equipment: Flex sensors for the hand (5 fingers), Arduino UNO and LCD for the display [5]. All the flex sensors help in collecting real-time data collected from the position of the fingers and the obtained data is sent to Arduino UNO for processing. After the data is processed, the message is displayed on the LCD. Commonly used sentences and phrases are represented by Mode 1, which is triggered by the combination 01000. When switching on Mode-1, the desired message can be selected from the available messages in the database/table. A particular message is associated with each bend of the five flex sensors on the five fingers. For instance, straightening the little finger while bending the other fingers sends the word "Good Morning." Another example is saying "I'm sorry" by straightening the thumb and little finger while bending the other three fingers. Alphabets are represented by Mode-2, which is enabled by the sequence 01100. The desired alphabets can be selected from the English alphabets and few special characters once Mode-2 has been activated. With Mode 2, the user can represent words in addition to the alphabet, removing the limitation found in Mode 1. There were only enough flex sensors to define a maximum of 29 widely used phrases. This mode offered the ability to construct new phrases using different alphabets. Also, it should be noted that only 29 of the possible 32 possibilities can be used for communication.

The study Modelling of Sign Language Smart Glove Based on Bit Equivalent Implementation Using Flex Sensor focuses on the designing a low-cost wearable for the speech-impaired which identifies the sign-language made by the user using Machine learning and produces audio clip for the sign-language interpreted [6]. This system involves three modules: the Gesture sensing module, the data acquisition module and the sign recognition and presentation module. To sense both the orientation and the flexion of fingers, an economical sensing glove that does not disturb the normal functioning of the hands was developed in Gesture sensing module. A bend sensor based on Velostat was created to sense flexion while a three-axis MEMS accelerometer was employed to detect hand orientation. Each bend sensor was connected in series with a $5k\Omega$ resistor to generate a voltage divider. A 5V DC supply was given to this arrangement and voltage V_{bend} is measured as a function of Resistance R_{bend} . Each glove contains eight analogue output lines, of which five are used to transmit bend sensor data and three are used to provide accelerometer data to the data acquisition module. The analogue signals are transformed into digital for further processing by the data acquisition module. The A/D conversion procedure has been carried out using the National Instruments portable data acquisition systems. Using trial and error, it was chosen to set the sampling rate at 50 samples per second to reduce the processing time. The Recognition and Presentation module processes the sensor outputs and identifies the sign being made using the gloves. There are two parts in this module: quantization of sign recognition series and sign recognition and classification. Inside the Quantization part, once the sampled data from the sensors is available, the Symbolic Aggregate Approximation (SAX) algorithm is used to code it into symbols. In the Sign Classification part, the ability of three different classifiers—Naive Bayes, Classification Tree, and Support Vector Machines (SVM)—to recognise and differentiate between ISL gestures was tested. SVM gave the highest accuracy of 93.04% and the only limitation of this system is it is only limited to 56 ISL signs.

3 Proposed Methodology

3.1 Designing a SLR Real-Time Video-Based System Whose Purpose is to Detect the ASL

Designing a SLR (sign language recognition) real-time video-based system involves creating an artificial intelligence model that can accurately detect and recognize American Sign Language (ASL) signs from video input in real-time. The system should be designed to work with a standard camera or webcam and process video input from any angle or orientation. To develop such a system, we would need to start by collecting a large dataset of sign language videos, preferably from multiple signers, with different skin colors and backgrounds. The videos should capture different signs and hand movements from different angles and lighting conditions. Next, the dataset would need to be preprocessed, which includes tasks like removing background noise, segmenting individual signs, and labeling the signs correctly. This would require using computer vision techniques and human annotation.

The American Sign Language Letters dataset is an object detection dataset that includes a bounding box for each ASL letter. This project has a collection of 720 photographs taken with different hand postures held at different places. As this is a tiny dataset, manual labelling with bounding box coordinates was performed using the labelling software in Roboflow, and a probability of transformations function was used to generate many instances of the same picture, each with different bounding boxes. Figure 1 Represents the sample from the dataset with identified sign.

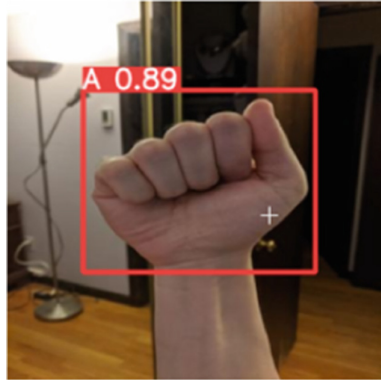


Fig. 1. Sample image from the dataset

Once this dataset is ready, the next step would be to train a deep learning model on the data. The model would need to be optimized for real-time performance to ensure fast processing of video input. To ensure accuracy, the model is trained with a large and diverse set of augmented data and regularly evaluated on test data. The evaluation should measure the model's performance on recognizing signs in different lighting conditions, orientations, and hand shapes.

3.2 Develop a Wrapper that can get Real-Time Video from Online Meetups

Developing a wrapper for obtaining real-time video from online meetups involves creating a software module that can interface with the video streaming capabilities of popular online meeting platforms, such as Zoom, Microsoft Teams, Google Meet, or other similar platforms. The wrapper would be responsible for capturing and processing video streams from online meetups in real-time, allowing developers to build additional functionalities on top of the video streams.

The first step in developing the wrapper would be to understand the APIs (Application Programming Interfaces) provided by the online meeting platforms. These APIs typically provide access to video streams, audio streams, and other data related to online meetings. The wrapper would need to utilize these APIs to capture the video stream from the online meetup platform.

Next, the wrapper would need to process the video stream in real-time. This could involve tasks such as decoding the video stream, extracting frames, and performing image processing operations like resizing, cropping, and color correction. Real-time processing would be crucial to ensure minimal latency and smooth performance.

Once the video stream is processed, the wrapper could provide various functionalities depending on the intended use case. For example, it could perform real-time video analysis, such as face recognition, emotion detection, or object detection, to extract meaningful information from the video stream. It could also overlay additional visual elements, such as annotations, graphics, or subtitles, on top of the video stream to enhance the user experience.

3.3 Recognize the Hand Motions from the Acquired Image Frame

Yolo-v8 is one of the currently leading models in this industry. YOLO can identify hand gestures in image frames because it uses deep neural networks to locate objects in real-time. Image acquisition is followed by pre-processing to assure image quality, object recognition with YOLO, hand gesture classification using further machine learning approaches, post-processing to fine-tune results, and output of the recognised hand gestures. The quality of the training data, the model architecture, and the hyperparameters all have an impact on how accurate the hand motion detection system is. The reliability and sturdiness of the hand motion recognition system must be ensured through careful examination and testing. Also, to extract the exact sign from the frame or the input video, it removes the background and precisely captured the hand from the camera input during real time sign language detection.

Even though Yolo-v8 was not produced by the original YOLO writers, YOLO v8 is believed to be quicker and lighter, with accuracy comparable to YOLO v5, which is widely regarded as the quickest and most accurate real-time object recognition model. 85% of the enhanced photos were used for training, with the remaining 15% set aside for testing and validation. The model was trained for 64 epochs with the YOLOv8 pre-trained weights using transfer learning.

3.4 Translate the Picture to Textual Content

Images of sign language can be converted into text using the YOLO object identification technique and the spellchecker library. The spellchecker library is used to process the labels produced by YOLO, and it can detect and rectify any potential spelling mistakes in the text. As a result, sign language gestures can be converted into written text and the message expressed by the sign language image can be represented textually. The algorithms and dictionaries used in the spellchecker library assist in recommending adjustments depending on context, ensuring that the final literary material is grammatically and linguistically accurate. It is easier for a larger audience to access and comprehend because to this combined use of YOLO and the spellchecker library.



Fig. 2. Sample test

4 Results

4.1 Mean Average Precision

Mean Average Precision (mAP) at 50 is a commonly used evaluation metric for object detection models, including those based on YOLO (You Only Look Once) algorithm. It measures the accuracy of the model in detecting objects across different categories (e.g., person, car, bicycle) at a certain intersection over union (IoU) threshold of 0.5. The mAP@.5 score of the created model is 95.7%. Figure 2 represents the predicted sample test results.

4.2 Confusion Matrix

In object detection with YOLO (You Only Look Once), the confusion matrix can be used to evaluate the performance of the algorithm on a test set of images. The confusion matrix for object detection with YOLO is a table that summarizes the predicted and actual labels for each object in the test set. Figure 3 depicts the Confusion Matrix plotted for each character.

4.3 Precision-Recall Curve

A precision-recall (PR) curve is a graphical representation of the performance of a binary classification algorithm that measures the trade-off between precision and recall at different threshold settings. Figure 4 illustrates the Precision-Recall (PR) curve for our model.

4.4 F1-Confidence Curve

In The F1-Confidence curve is a plot of the F1 score as a function of the confidence threshold for a binary classification model. The confidence threshold is the minimum probability required for the model to make a positive prediction. Figure 5 depicts the

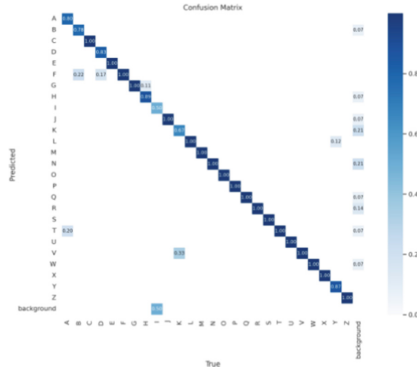


Fig. 3. Confusion matrix plotted for each character.

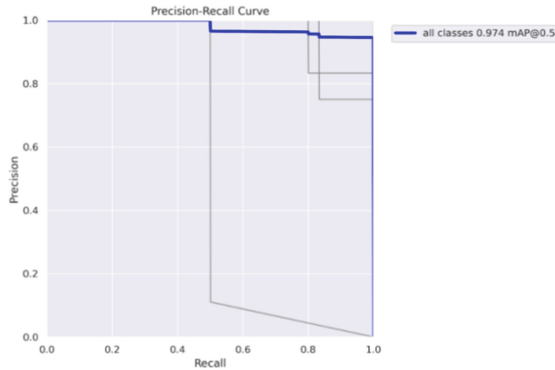


Fig. 4. PR Curve for our model.

F1-Confidence curve, the highlighted one showing the F1-Confidence calculated for 64 epochs. The F1 score is a measure of a model’s accuracy that considers both precision and recall. It is defined as the harmonic mean of precision and recall:

$$F1\ score = 2 * (precision * recall) / (precision + recall).$$

4.5 Challenges Faced

The fact that the model performs admirably with such a tiny dataset cannot be overlooked! Even in fresh locations with different hands, it detects well. There are a couple constraints that can be easily overcome by just providing more data to train on. With a few tweaks and a lot more data, we anticipate having a functional model that can be extended far beyond the ASL alphabet.

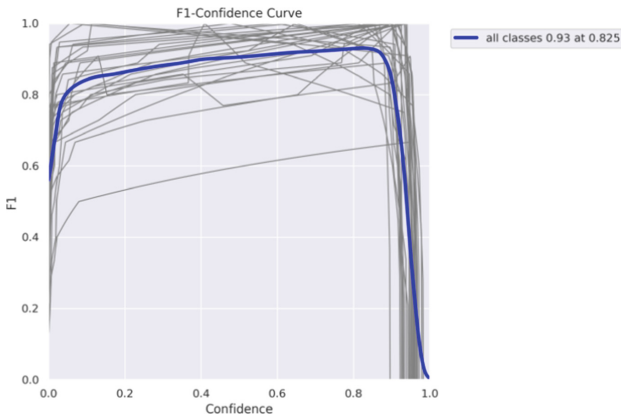


Fig. 5. F1-Confidence curve after 64 epochs

5 Conclusion

Video captioning based on sign language is a crucial solution for improving accessibility for individuals who are deaf or hard of hearing, providing them with equal access to information, education, and multimedia content. With accurate and synchronized captions that convey sign language expressions, video content becomes more inclusive and promotes effective communication, bridging gaps between sign language users and non-sign language users in various settings. Continued advancement and promotion of sign language video captioning can create a more inclusive world, fostering social inclusion and enhancing communication for individuals with hearing loss.

References

1. Mehta, A., Solanki, K., Trupti Rathod, T.: Automatic translate real-time voice to sign language conversion for deaf and dumb people. *Int. J. Eng. Res. Technol. (IJERT) ICRADL – 2021* **9**(5), 174–177 (2021)
2. Rani, R.S., Rumana, R., Prema, R.: A review paper on sign language recognition for the deaf and dumb. *Int. J. Eng. Res. Technol. (IJERT)* **10**(10), (2021)
3. Khallikkunaisa, Kulsoom A.A., Chandan Y.P, Fathima Farheen, F., Halima, N.: Real time sign language recognition and translation to text for vocally and hearing impaired people. *Int. J. Eng. Res. Technol. (IJERT) IETE* **8**(11) (2020)
4. Kodandaram, S.R., Kumar, N., Gl, S.: Sign language recognition. *Turkish J. Comput. Math. Educ. (TURCOMAT)*. **12**, 994–1009 (2021)
5. Muralidharan, N. T. , R. R. S., R. M. R., S. N. M., H. M. E.: Modelling of sign language smart glove based on bit equivalent implementation using flex sensor. In: 2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, pp. 99–104 (2022). <https://doi.org/10.1109/WiSPNET54241.2022.9767137>
6. Singh, A.K., John, B.P., Subramanian, S.R.V., Kumar, A.S., Nair, B.B.: A low-cost wearable Indian sign language interpretation system. In: 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA), Amritapuri, India, , pp. 1–6, (2016) <https://doi.org/10.1109/RAHA.2016.7931873>

7. Kartik, P.V.S.M.S., Sumanth, K.B.V.N.S., Ram, V.N.V.S., Prakash, P.: Sign language to text conversion using deep learning. In: Ranganathan, G., Chen, J., Rocha, Á. (eds) *Inventive Communication and Computational Technologies. Lecture Notes in Networks and Systems*, vol 145. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-7345-3_18
8. Islam, S., Dash, A., Seum, A., et al.: Exploring video captioning techniques: a comprehensive survey on deep learning methods. *SN COMPUT. SCI.* **2**, 120 (2021)
9. Tamilselvan, K.S., Balakumar, P., Rajalakshmi, B., Roshini, C., Suthagar, S.: Translation of sign language for deaf and dumb people. *Int. J. Recent Technol. Eng.* **8**, 2277–3878 (2020). <https://doi.org/10.35940/ijrte.E6555.018520>
10. Juju, R.: Video captioning and sign language interpreting (2022)
11. earn2Sign: Sign Language Recognition and Translation using Human Keypoint Estimation and Transformer Model - September 09, 2020 - Institute for Natural Language Processing University of Stuttgart Pfaffenwaldring 5 bD-70569 Stuttgart
12. Vidhyasagar, B.S., Raja, J., Marudhamuthu, Krishnamurthy.: A novel oppositional chaotic flower pollination optimization algorithm for automatic tuning of hadoop configuration parameters. *Big Data.* **8**, 218–234 (2020). <https://doi.org/10.1089/big.2019.0111>
13. Cihan Camgöz, N., Koller, O., Hadfield, S., Bowden, R.: Sign language transformers: joint end-to-end sign language recognition and translation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 10020–10030 (2020), <https://doi.org/10.1109/CVPR42600.2020.01004>
14. Bantupalli, K., Xie, Y.: American sign language recognition using deep learning and computer vision. In: 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA (2018)
15. Suri, K., Gupta, R.: Convolutional neural network array for sign language recognition using wearable signal processing and integrated networks. In: 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India (2019)
16. Byeongkeun, K., Tripathi, S., Nguyen, T.Q.: Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. arXiv preprint <https://arxiv.org/abs/1509.03001> (2015)
17. Anand, M.S., Kumar, N.M., Kumaresan, A.: An efficient framework for indian sign language recognition using wavelet transform. *Circuits Syst.* **7**, 1874- 1883 (2016)
18. Kumud, T., Baranwal, N., Nandi, G.C.: Continuous dynamic indian sign language gesture recognition with invariant backgrounds. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2211–2216 (2015)
19. Manikandan, A.B.R.K.: Aayush patidar, and pallav walia, hand gesture detection and conversion to speech and text. *Int J. Pure Appl Math.* **120**(6), 1347–1362 (2018)
20. Dutta, K.K., Swamy, S.A.: Double Handed Indian Sign Language to Speech and Text. In: 2015 Third International Conference on Image Information Processing (ICIIP), pp. 374–377 (2017)



Improvement in Multi-resident Activity Recognition System in a Smart Home Using Activity Clustering

E. Ramanujam¹(✉), Sivakumar Kalimuthu², B. V. Harshavardhan¹,
and Thinagar Perumal²

¹ National Institute of Technology Silchar, Silchar 788010, Assam, India
ramanujam@cse.nits.ac.in, bontalakoti21.ug@mech.nits.ac.in

² Universiti Putra, Serdang, Malaysia
gs56792@student.upm.edu.my, Thinagar@upm.edu.my

Abstract. Human Activity Recognition (HAR) integrates ambient assisted living (AAL), leading to smart home automation for monitoring activities, healthcare, fall detection, etc. Various researchers have proposed a single-resident HAR system for ambient-sensor based smart home data, which is simple, and single-resident is not always the case. Multi-resident recognition is slightly complex and time-consuming. The researchers have made several efforts to generate benchmark datasets, such as CASAS, ARAS, vanKasteren, etc., for baseline comparison and performance analysis. However, these datasets have certain limitations, such as data association, annotation scarcity, computational cost, and even with data collection itself. This paper profoundly analyzed these limitations and manually clustered the activity labels to record the improvement in the performance of the system in terms of both recognition rate and computational time on the ARAS dataset.

Keywords: Human Activity Recognition · Smart home · Automation · Multi-Resident Activity

1 Introduction

Human Activity Recognition (HAR) is a system that is built to monitor human activities, especially the elderly person living alone in the house. The HAR integrates ambient assisted living (AAL) into smart environments such as smart homes to recognize the activities automatically [1]. The AAL uses simple pervasive, ubiquitous sensors with artificial intelligence (AI) to connect smart devices with Internet-of-Things (IoT). The sensors used in the AAL are categorized into vision sensors, ambient sensors, wearable sensors/devices, and smartphones for monitoring the simple, complex, and postural transitions [2]. The smart home environment with AAL facility has been primarily used for activity monitoring, healthcare, fall detection, sports tracking, etc. [3]. The vision-based has specific

issues with the person's privacy while monitoring and requires installation in all the actuated areas. Wearable and smartphone devices must be worn for the entire day, which is difficult for the elders. Sometimes, the elders may forget to carry or recharge the battery due to age or other diseases [4]. Recently, the HAR with ambient sensors has attracted most researchers as it is non-invasive and can indirectly alert the family members or caretakers in case of any anomaly in the residents' activity [5].

Various researchers have proposed a HAR system in the last decade that primarily focuses on single resident activity recognition [6]. The single-resident activity is simple and easy to detect. Many state-of-the-art machine and deep learning algorithms have shown promising results for the single-resident activity in the AAL environment [1,3,6]. However, the smart home environment may have more chances for multi-residents such as friends, neighbors, pets, etc., and the single-resident is not always the case. Multi-resident activity recognition in an ambient sensor environment is slightly complex and needs certain improvements in activity recognition and detection. Researchers have developed multi-resident activity recognition benchmark datasets for further research and implementation. However, these datasets have certain limitations in data collection, annotation scarcity, data association, computational cost, etc. This paper solves such a challenge of the benchmark dataset by clustering the activity labels based on the nature of the activity. Experimentation has been carried out by utilizing the ARAS [7] - an ambient sensor-based multi-resident activity recognition dataset.

The remaining sections of the paper are organized as follows. Section 2 deals with the motivation of the challenges discovered in the benchmark datasets. Section 3 discusses the proposed clustering of activity labels, Sect. 4 demonstrates the experimentations and results using the clustered dataset, and Sect. 5 concludes the paper.

2 Motivation

In the last decade, many machine and deep learning techniques have been employed for multi-resident human activity recognition (MRHAR) in an ambient sensor smart environment [8,9]. However, MRHAR has certain limitations that need to be addressed for better improvement in the system performance. The limitations are discussed in subsections as follows.

2.1 Dataset Collection

Training and evaluating deep or machine learning techniques require large data samples. Collecting ambient sensor data on a large scale is costly and time-consuming. In the case of MRHAR, the cost and time will be multi-fold as it involves more sensors to collect the activities in a smart home environment. In the last decade, researchers have spent enormous time and effort collecting and compiling several benchmark datasets as mentioned in the review works [8,9] using object and ambient sensors for the MRHAR system as follows.

ARAS Dataset. [7] - Two pairs of residents in two smart Houses, A and B, perform 27 activities. House A contains two males, while House B is a married couple. The dataset reflects on the natural behaviors of the residents to 30 days of recording for each house. Each house contains 30 days of recording, and each day consists of $22 * 86,400$ instance matrix results in $30 * 22 * 86400$ instances. The first 20 columns are the binary sensor values that say the state of the sensor, either fired or not. Columns 21 and 22 represent the activity label of residents 1 and 2. Annotation of the activities was achieved by the residents using a simple graphical user interface (GUI) placed in the most convenient places of the houses.

CASAS Dataset. [10]- The Centre for Advanced Studies in Adaptive Systems has collected several scripted and unscripted multi-resident activity recognition datasets. “twor.2009”, “twor.summer.2009”, “twor.2010”, “Tulum,” “tulum2”, and “Cairo” are unscripted, and “Multi-residentADLs” is a scripted dataset of CASAS. Activities are annotated by recording the start and end time of the activity via a handwritten diary. The data are delimited in a specific format for recording, and it requires preprocessing before the feature engineering process. Data collection uses 14-21 ambient sensors to carry out 11 activities.

VanKasteren Benchmark. [11] - One resident in three smart houses was considered for the data collection. It has 14-23 different sensors to collect data and demonstrate approximately 10-16 activities. The dataset is available for 25 days but not collected continuously. Activities are annotated using the handwritten diary and a Bluetooth device.

UJA Dataset. [12] - The dataset is collected in the UJAml (University of Jaen Ambient Intelligence, Spain) Smart lab that consists of Single and Multi-Occupancy (SaMO) data. The dataset includes a new generation of sensors with heterogeneous data as a source of information to provide an excellent tool for addressing multi-occupancy in smart home environments. Researchers have utilized different sensor technologies such as binary sensors in objects space, proximity between the inhabitant, and Bluetooth Low Energy, etc. dataset has 10 days of single occupancy data and over 9 days of multi-occupancy data. It contains 25 different types of activities grouped into 7 categories.

SDHAR-Home Dataset. [13] - The dataset has been developed through non-intrusive technology in the multi-resident smart home environment. Set of non-intrusive sensors integrated with activity wristbands to capture the events in the house, positioning the user through triangulation using beacons respectively.

Two months of uninterrupted measurements were obtained on the daily habits of 2 people, along with a pet and friends who has sporadic visits with these two people. Altogether a total of 18 different types of activities were labeled.

MARBLE Dataset. [14] - A novel multi-inhabitant ADLs dataset that combines smartwatch and environmental sensor data. Smartwatches are used to record hand motions to identify ADLs. Environmental sensors such as mats and pressure sensors were used to detect the residents sitting on sofas or sleeping on the couch etc. Indoor locations are identified using Wi-Fi access points and BLE beacons. The detailed summary of the MRHAR smart home datasets is shown in Table 1.

Table 1. Summary of the Multi-resident Smart home datasets

Name of the Dataset	Houses	Residents	Duration	Type and Number of sensors	Activities	Environment
ARAS [7]	2	2	30 Days	Ambient Sensors (20)	27	Real-time
CASAS [10]	7	2	2-8 months	Wearable + Ambient Sensors (20-86)	11	Inlab - Real-time
VanKasteren [11]	3	1	14-25 days	Ambient sensors (14-21)	10-16	Realtime
UJA Dataset [12]	1	2	10 -25 days	Ambient Sensors and Smartphone	7	InLab
SDHAR-Home [13]	1	2	2 months	Wearable + Ambient sensors (35) + Positioning (7)	18	Realtime
MARBLE [14]	1	1	16hrs	Wearable + Ambient Sensors (8)	13	InLab

As per the research survey by [8], collecting two ambient sensing smart home datasets with multi-residents offers a better opportunity to study and compare the activity recognition algorithm more realistically. The vanKasteren dataset may not opt for such a fair comparison as it has only one resident performing various activities in all three smart homes. Similarly, the MARBLE dataset deals with the multi-habitant nature of the single-resident activity and data collected in the Inlab environment. CASAS data are mostly predetermined and were repeatedly performed in a controlled laboratory environment. This collection accounts for inter-subject variability and is thus not suitable for real-world data analysis. Further, the UJA dataset has been collected using multi-model heterogeneous hardware items, and this may be infeasible in multiple cases for the recognition of ADLs. However, the ARAS dataset accounts for intra-subject variability and does not account for inter-subject one. This leads to a better opportunity to study and compare the MRHAR system. UJA, MARBLE, and SDHAR are the recently evolved datasets. They have been created using multi-modal heterogeneous devices for data collection, so it has not been considered here for experimentation.

It has also been proven by collecting the citations of datasets [7, 10–14] from the year of dataset generation (mostly 2011) to till date (April 30, 2023) as per the Scopus and Google scholar records. Alternatively, CASAS has the highest number of citations, 642. VanKasteren has 250 citations, and ARAS has 146 citations. ARAS has fewer citations because the collected data is completely binary and has no discriminating features to easily classify the activities labeled in each instance. The performance of the ARAS dataset still needs to improve due to certain factors of data collected during experimentation. It motivates us to analyze the ARAS dataset to improve its performance apart from implementing hybrid deep learning models to make the system more complex to recognize the activities.

2.2 Data Association

The complexity of data association is the second major challenge in the HAR system, which refers to the number of persons and activities associated with it. Concurrent activity is one of the challenges in activity recognition driven by data association. The activity occurs when the resident participates in multiple activities simultaneously—for instance, sitting and watching TV, conversing while watching TV, etc. A sample challenge with data association is shown in Fig. 1, which represents the activity of watching TV and using the Internet in four different instances. In the first instance, there is no firing of the sensor. In the second instance, the IR sensor (sensor 3) of the TV receiver is fired, which confirms the activity of watching TV. However, there is no sensor firing for the activity using the Internet. In this case, the resident may be standing and using the Internet. In the third instance, the force sensor of the couch (sensor 4) is fired, representing that the resident(s) may be sitting on the couch and watching TV or using the Internet. However, there is no proper sensor firing stating the activity of watching TV and using the Internet. Finally, in the last instance, the force sensor in the couch and chair are fired, representing one resident sitting on the couch and another sitting on the chair. However, the activities are labeled to be again watching TV and using the Internet. This label associated with the binary sensor makes the system complex to recognize the activities.

A similar example is shown in Fig. 2 where the activities labeled are preparing dinner and watching TV. However, it is unclear which resident is preparing the dinner and which is watching TV. Moreover, for watching TV, sensor 4 is fired correctly in all three cases, whereas for preparing dinner (sensor 16), it is not adequately fired in Instances 1 and 4. This makes inevitable proof for improper labeling and association, which makes any machine/deep learning models more complex and consumes time for model generation.

In a specific case, multi-resident cooperation activities are also associated with the complexity of data association. For example, the residents may be in the process of cooking. However, the respective sensor fired relates to cooking and washing dishes. So one resident may cook, and the other may wash dishes. This causes inevitable confusion. Moreover, the system cannot identify who is

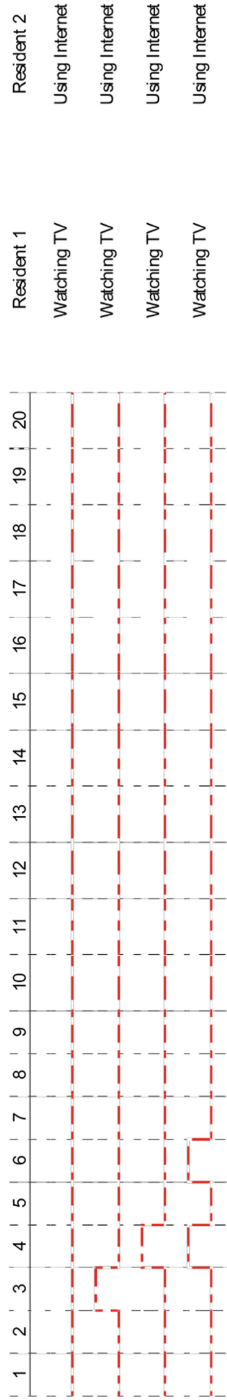


Fig. 1. Sample Annotations on ARAS Dataset-House A

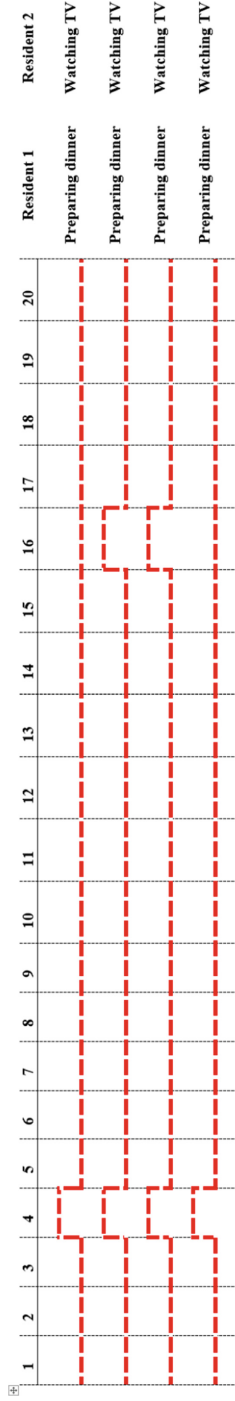


Fig. 2. Sample Annotations on ARAS Dataset-House B

washing the dish and cooking. These limitations must be addressed to improve the performance of any learning model to handle the ARAS dataset.

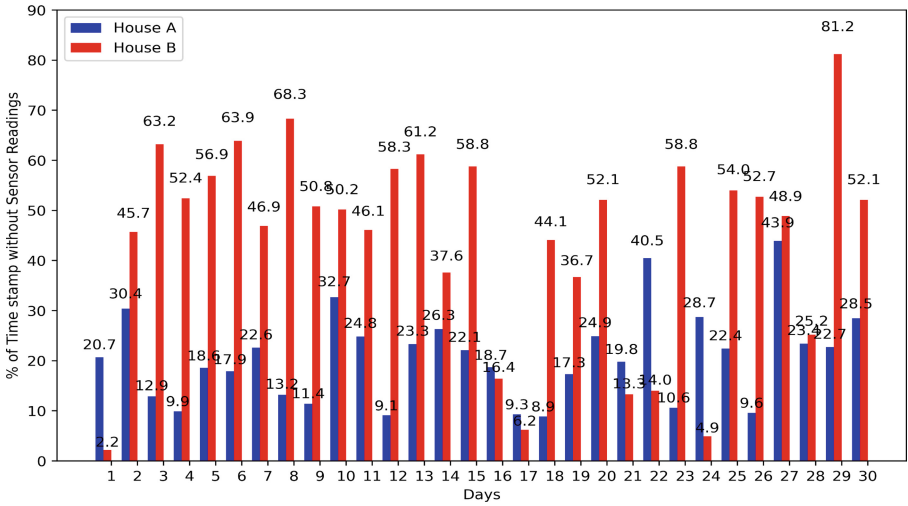


Fig. 3. Percentage of timestamp without any sensor activations for ARAS House A and House B [20]

2.3 Annotation Scarcity

Activities are annotated mainly by either a handwritten diary or using the GUI. Data annotation is expensive and time-consuming, and it is a remarkable challenge for ambient-sensor data, especially for a dataset like ARAS. The residents must properly annotate the activity, and all the sensors must be appropriately fired on all the instances for recording the activities. In certain scenarios, the resident may miss annotating the activity, and in certain other cases sensor may not fire for the activity carried out at a particular instant. In addition, data for some emergency or unexpected activities are hard to obtain, such as toileting, sleeping, etc. A sample annotation scarcity is shown in Fig. 1, which represents the activity of watching TV and using the Internet. In the first instance, there is no firing of a binary sensor, but it represents the activity of watching TV and using the Internet. Similarly, many instances have no firing of data. However, the annotations are labeled for each activity, especially during the night time of recording. The detailed analysis of the Percentage of timestamps labeled with activities without sensor activations for Houses A and B is shown in Fig. 3. Considering the report from Fig. 3, several activities are without the proper firing of sensors.

Another reason for annotation scarcity is that the bathroom resident may annotate toileting before getting into the toilet. Later, there is a chance to have

a shower also. However, the resident may need to annotate the activity correctly, which uses the toileting label even for the shower with the respective sensor fired.

2.4 Computational Cost

The third major factor that needs to be concerned is the feasibility of implementing the MRHAR in real-time. A considerable effort must be made to make the system acceptable to many end-users as it is close to human life. However, the state-of-the-art systems proposed for MRHAR on the ARAS dataset have certain challenges in attaining a preferable recognition rate (Accuracy) or the computational time taken to recognize the event. The research works that considered the ARAS dataset for experimentation, and the results are shown in Table 2. In comparing the performances of the state-of-the-art methods, the accuracy reaches the maximum of 89% for House A and 97% for House B, respectively. However, there is no mention of the execution time for the models represented by various state-of-the-art methods. Alternatively, the research work [19] has attained the accuracy of 86.315% and 87.975% for Houses A and B, respectively, but the execution time is 10-fold high than our previous work [20]. The system should be recourse intensive, so it fits portable devices and can respond instantly. Thus, the computational cost and recognition issue should be addressed.

Table 2. State-of-the-art computational performance of methods on the ARAS dataset

Research Work	Methodology	Accuracy (%)		Computational Time(s)	
		House A	House B	House A	House B
[15]	Transformer with Bi-directional GRU (10-Fold cross validation)	89.48	90.59	–	–
[16]	Random k-label sets approach (Machine Learning - Multiclass classification)	F1- 0.676	F1 - 0.909	–	–
[17]	Generative Adversarial Networks+LSTM	71.45	86.42	–	–
[18]	Classifier Chain method of Multi-Label Classification	88.02	97.13	–	–
[19]	CNN 1D+ LSTM	86.315	87.975	14043.7	14811.3
[20]	MLMO-HSM	89.825	94.95	158.61	146.92

3 Clustering of Activity Labels

The challenges listed above are majorly in the data collection. The collected data can never be wasted as it consumes time and cost. However, the challenges can be rectified through certain other aspects of preprocessing. This section prescribes a manual clustering of activity labels to overcome certain challenges

and to achieve an efficient computation on recognition of the ARAS dataset. The ideology is that the fired binary data can never be changed or altered. However, the activity label annotated can be marked or changed in certain aspects. For instance, consider Fig. 4, which shows the sample label annotations of preparing breakfast, reading a book, preparing lunch, etc., in five different instances at House B.

In comparing all five different instances, the firing of the binary sensor resembles the same for all the activities. Sensors 17 (Armchair) and 19 (kitchen sensor) are fired for preparing breakfast, lunch, and washing dishes. It results that all the activities can be combined into a single activity, reducing the number of activities, so complexity in the class distribution reduces and provides optimal performance using a simple learning algorithm. In addition, most of the annotated activity labels are secondary and can be clustered into a primary activity [21].

- Preparing food is the common primary activity, whereas its secondary activities are washing dishes, cutting vegetables, etc., In addition, the food can be further categorized into breakfast, lunch, and dinner. Thus, the primary and secondary activities are clubbed into one food preparation activity.
- The researchers categorize Eating/having food at all times, namely breakfast, lunch, and dinner, or eating snacks in the evening (secondary activities) based on the timestamp where they recorded the instance. All these activities have been carried out at the dining table kept in the house. Considering this in-depth, the resident may sit in a chair or at the dining table for all eating purposes. There may be fewer chances of firing different sensors for the activity of eating. Thus, the activities are merged into Eating food as a common term.
- Sleeping and napping are carried out on the couch that fires the same binary sensor installed in the house. Thus, the activity has been merged into sleeping, which represents snapping also.
- Watching TV, conversing, and having guests resembles firing the binary sensor of the sofa/couch in the hall where the guest usually arrives. In addition, the cleaning of the house activity closely resembles the resident sitting on the sofa/couch. Thus, watching TV, conversing with guests, and cleaning clubbed into the activity of watching/cleaning.
- Studying and Reading books are carried out by sitting in a chair, and thus respective binary sensors are fired for this activity. Studying is commonly represented here for studying and reading books.
- Having a shower, Toileting, Laundry, shaving, and brushing teeth are the activities that fire the sensors, bathroom cabinet, bathroom door, water closet, etc. As all these activities are secondary activities carried out in the bathroom/washroom, it has been managed in the bathroom/washroom in a single term.
- Using the Internet, talking on the phone, and listening to music represent the entertainment process, and most fire the couch/chair in the hall. Thus, it has been fixed into the activity termed entertainment.

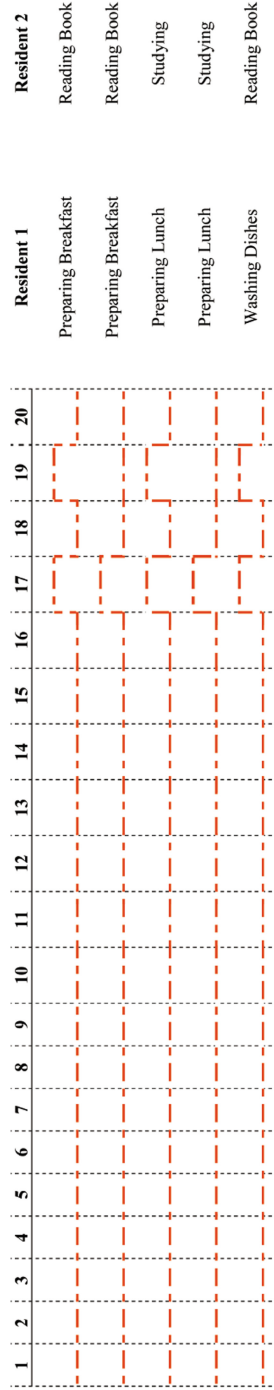


Fig. 4. The resemblance of sensor firing for various activity labels on the ARAS House B dataset

- Finally, changing clothes, going out (outdoors), and other activities have discriminate binary pulse sensors that never mix with any other activities. Thus the same name has been used for the process.

All 27 activities of the ARAS dataset have been clustered into 10 modified clustered label annotations, and the detailed labeling is shown in Table 3. The clustering process is completely manual based on the primary and secondary activities of the dataset.

Table 3. Original and clustered label annotations of ARAS dataset

S.No	Original Annotations	Clustered Annotations
1	Preparing Breakfast, Preparing Lunch, Preparing Dinner, WashingDishes	Preparing Food
2	Having Breakfast, HavingLunch, HavingDinner, Having Snack	Eating Food
3	Sleeping, Napping	Sleeping
4	Watching TV, Cleaning, Having Conversation, Having Guest	Watching/Cleaning
5	Studying, ReadingBook	Studying
6	Having Shower, Toileting, Laundry, Shaving, Brushing Teeth	Bathroom/Washroom
7	Using the Internet, Talking on the Phone, Listening to Music	Entertainment
8	Changing Clothes	Changing Clothes
9	Going out	Going out
10	Other	Other

4 Experimentation and Results

The experimentations have been carried out on the clustered activity labels, as mentioned in Sect. 3, with the deep learning models. In this experimentation, the Multi-layer perceptron (MLP), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional 1D, and Convolutional 1D with LSTM have been used. Standard performance metrics such as Accuracy and computational time have been used to evaluate the performance of clustered activity labels, as mentioned in our previous research work [20].

In all the models, the input layer reads the input in the shape of 20,1, i.e., binary sensor readings with each instance at a time. The output layer classifies the resident activity simultaneously through a multi-label multi-output layer as utilized in [20]. Softmax function in the dense layer with 10 neurons (classes) of output will be inferred for both residents. Table 4 shows the hyperparameters considered in the experimentation.

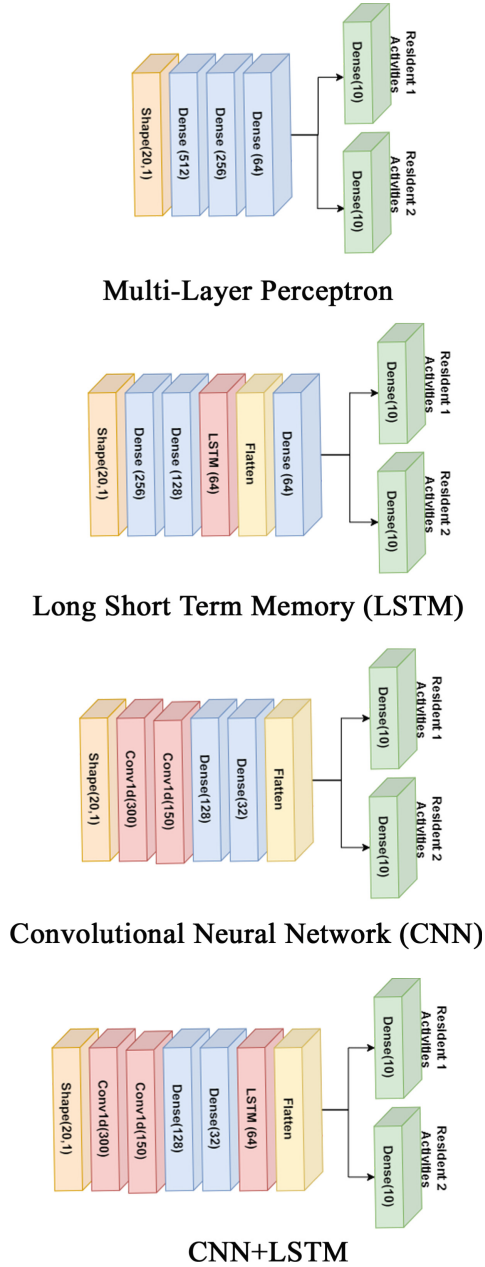


Fig. 5. Architectures used for evaluating clustered activity label dataset

Table 4. Hyperparameters of the experimentation to validate the clustered activity labels

S.No	Hyperparameters	Value
1	Epochs	25
2	Learning rate	0.001
3	Batch size	864
4	Optimizer	Adam
5	Loss function	Categorical cross-entropy
6	Early stopping	Yes

Multi-layer Perceptron. (MLP) [22] is a fully connected class of feed-forward artificial neural network (ANN). MLP consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. This experimentation has been designed with an input layer followed by three dense layers of 512, 256, and 64 neuron units. The Input layer has a shape of (20,1), and the two output layers with a softmax function of 10 units recognize residents 1 and 2 exclusively.

Long-Short Term Memory. (LSTM) [23] is a variant of recurrent neural network (RNN) to address the vanishing gradient descent problem and to maintain long-term dependencies of temporal information. The model has three gates, namely input (i_t), forget (f_t), and output gates (o_t), which make LSTM learn the long-term dependencies adequately. In the experimentation of the LSTM model, the input layer of (20,1) is connected to two dense layers that have 256 and 128 neuron units. The feature extracted from the dense layers is passed into the LSTM layer of 64 units and further connected to the dense layer of LSTM of 64 units by flattening the output of LSTM through the flattened layer. Finally, two dense layers, each of 10 units, are used to exclusively recognize the resident's 1 and 2 activities.

Gated Recurrent Unit. (GRU) introduced by [24] a similar approach of LSTM to solve the vanishing gradient problem, which comes with a standard RNN. GRU is also a variation of LSTM and produces equally excellent results. GRU uses two gates like LSTM: update Z_t and reset r_t gate. In this experiment, GRU uses the same architecture as LSTM. Instead of LSTM cells, GRU cells are used.

Convolutional Neural Network. (CNN) [25] a type of artificial neural network widely used in high-dimensional image and video recognition and text categorization. To preserve the spatial information and the data recorded by the binary sensors and IoT devices, CNN models are used in MRHAR. CNN directly takes

1-dimensional/2-dimensional data as input, extracts the features using convolutional and hidden layers, and finally recognizes the activities using dense layers (fully connected neural network). In this process, 1-dimensional CNN (CNN1D) has been used to extract the patterns from the binary sensor reading. CNN1D has an input layer connected to two Conv1D layers with 300 and 250 filters, respectively, of size 1×1 . The output of the conv1d is fed as input to two dense layers of units 128 and 32, flattened and connected to the output layer.

CNN+LSTM. has the advantage of extracting high-level abstract features from the CNN block and long-term temporal dependencies from the LSTM block. In this experiment, the blocks are concatenated in the dimension of the channel. The CNN1D + LSTM uses the same architecture as Convolutional 1D in which an additional LSTM layer is added in between the dense layer of 32 units and a flattened layer to generate a Conv1D+LSTM model. The deep learning architectures used in this experimentation are shown in Fig. 5 with the details of layers and neurons present with it.

Table 5. Performance of Deep Learning Models on clustered Activity Labels of ARAS House A dataset

S.No	Model	Resident 1	Resident 2	Average	Time(s)
1	MLP	66.8	73.46	70.13	25.387
2	GRU	75.26	86.32	80.79	36.938
3	LSTM	75.3	86.33	80.815	38.08
4	CONV1D	75.8	86.03	80.915	26.4
5	CONV1d+LSTM	75.28	86.34	80.81	39.374

Table 6. Performance of Deep Learning Models on clustered Activity Labels of ARAS House B dataset

S.No	Model	Resident 1	Resident 2	Average	Time(s)
1	MLP	87.31	82.51	84.91	23.822
2	GRU	87.33	83.6	85.465	36.972
3	LSTM	87.33	82.5	84.915	38.285
4	CONV1D	87.75	83.23	85.49	33.922
5	CONV1d+LSTM	87.3	82.52	84.91	39.43

The experimental results on clustered activity labels with various deep learning architectures, as mentioned above, are shown in Tables 5 and 6 for Houses

A and B, respectively. In comparing the performances among the deep learning models, Conv1D performs far better than the other models in House A. It attains the highest accuracy (average of residents 1 and 2) of 80.915% with a computation time of 26.4s, slightly higher than MLP, which has 25.387s. Similarly, in comparing the performance on House B also, conv1D performs better than the other models. Conv1D attains the highest accuracy of 85.49 on an average of residents 1 and 2 accuracies. The results reported in Tables 6 and 4 are after properly tuning hyperparameters.

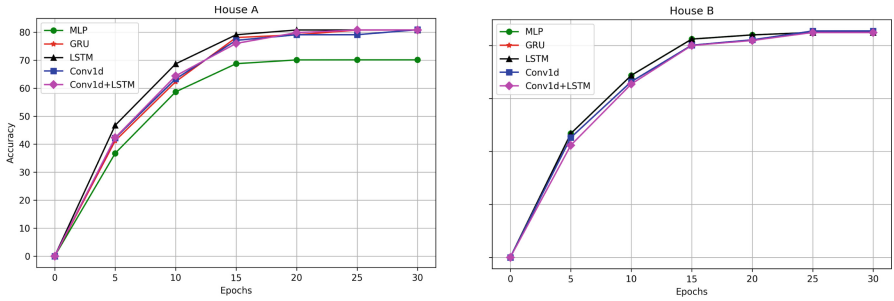


Fig. 6. Accuracy rate with the number of epochs for ARAS Houses A and B respectively

Though parameters are properly tuned, its experimental analysis on specific hyperparameters is shown in Figs. 6 and 7. Figure 6 deals with the accuracy rate with the number of epochs of both houses. On implementing various epochs, the model saturates at the 25th epoch, and in other cases, it provides the same performance values. Since the deep learning models have been performed with early stopping, it mostly stops at the 24 or 25th epoch. For visualization, it has been drawn for more epochs.

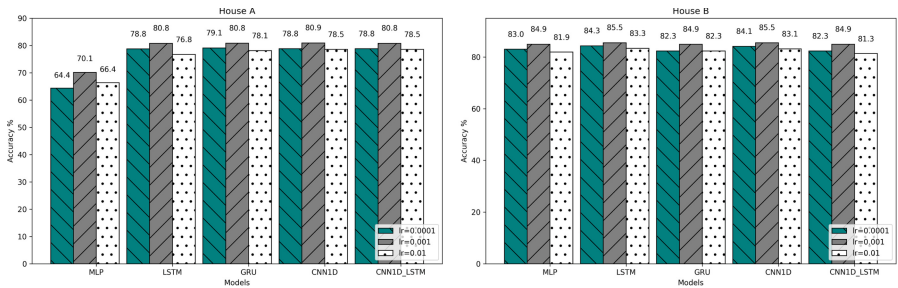


Fig. 7. Performance evaluation of clustered activity label dataset on different learning rates for Houses A and B

Similarly, the learning rate has been analyzed for three values ($lr = 0.0001$, 0.001 , and 0.1), as shown in Fig. 7. Among the three values, the deep learning

models have shown promising results on $lr = 0.001$, and thus, it has been fixed for experimentation as reported in Tables 6 and 4.

4.1 Limitations

Comparing the performance of Houses A and B, the architectures proposed here are very simple. No deep layers of convolution or other operations have been carried out to attain better performance. The computational time taken for the experimentation is nearly 300 to 350 times less than the performance of the models reported in Table 1. However, the performance is very close regarding the accuracy, as reported in Table 1—the experimental behavior changes mainly because of the clustered activity labels, as mentioned in Sect. 3. However, the experimentation can be further improved by integrating some automatic clustering techniques for multi-resident platforms based on the closeness of the activity by measuring the distance and sensor firing for those activities.

5 Conclusion

This paper deals with the limitations of the multi-resident human activity recognition (MRHAR) data collected on ambient sensing smart home environments. The limitations have evolved during the data collection to data association, annotations, and cost of computation. These limitations are profoundly analyzed with the benchmark MRHAR datasets such as ARAS, CASAS, vanKasteren, etc. The clustering of activity labels has been made in this work to analyze the performance variation using simple deep-learning models regarding recognition rate and computation time. The results have also proven that the clustering of activity labels has a multi-fold increase in computational time efficiency and recognition. In the near future, the clustering can be done with certain algorithms such as Expectation-Minimization, Canopy, etc., with the deep convolution models to outperform state-of-the-art works.

References

1. Bianchi, V., Bassoli, M., Lombardo, G., Fornacciari, P., Mordonini, M., De Munari, I.: IoT wearable sensor and deep learning: an integrated approach for personalized human activity recognition in a smart home environment. *IEEE Internet Things J.* **6**(5), 8553–8562 (2019). <https://doi.org/10.1109/jiot.2019.2920283>
2. Sharma, V., Gupta, M., Pandey, A.K., Mishra, D., Kumar, A.: A review of deep learning-based human activity recognition on benchmark video datasets. *Appl. Artif. Intell.* **36**(1), 2093705 (2022). <https://doi.org/10.1080/08839514.2022.2093705>
3. Almeida, A., Mulero, R., Rametta, P., Urošević, V., Andrić, M., Patrono, L.: A critical analysis of an IoT-aware AAL system for elderly monitoring. *Futur. Gener. Comput. Syst.* **97**, 598–619 (2019). <https://doi.org/10.1016/j.future.2019.03.019>

4. Gupta, N., Gupta, S.K., Pathak, R.K., Jain, V., Rashidi, P., Suri, J.S.: Human activity recognition in artificial intelligence framework: a narrative review. *Artif. Intell. Rev.* **55**(6), 4755–4808 (2022)
5. Anikwe, C.V., et al.: Mobile and wearable sensors for data-driven health monitoring system: state-of-the-art and future prospect. *Expert Syst. Appl.* **202**, 117362 (2022). <https://doi.org/10.1016/j.eswa.2022.117362>
6. Babangida, L., Perumal, T., Mustapha, N., Yaakob, R.: Internet of things (IoT) based activity recognition strategies in smart homes: a review. *IEEE Sens. J.* **22**(9), 8327–8336 (2022). <https://doi.org/10.1109/jsen.2022.3161797>
7. Alemdar, H., Durmaz Incel, O., Ertan, H., Ersoy, C.: ARAS human activity datasets in multiple homes with multiple residents. In: *Proceedings of the ICTs for Improving Patients Rehabilitation Research Techniques* (2013). <https://doi.org/10.4108/icst.pervasivehealth.2013.252120>
8. Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., Liu, Y.: Deep learning for sensor-based human activity recognition. *ACM Comput. Surv.* **54**(4), 1–40 (2021). <https://doi.org/10.1145/3447744>
9. Shiri, F.M., Perumal, T., Mustapha, N., Mohamed, R., Ahmadon, M.A.B., Yamaguchi, S.: A survey on multi-resident activity recognition in smart environments. arXiv preprint: [arXiv:2304.12304](https://arxiv.org/abs/2304.12304) (2023)
10. Cook, D.J., Crandall, A.S., Thomas, B.L., Krishnan, N.C.: CASAS: a smart home in a box. *Computer* **46**(7), 62–69 (2013). <https://doi.org/10.1109/mc.2012.328>
11. van Kasteren, T.L.M., Englebienne, G., Kröse, B.J.A.: Human activity recognition from wireless sensor network data: benchmark and software. In: Chen, L., Nugent, C., Biswas, J., Hoey, J. (eds.) *Activity Recognition in Pervasive Intelligent Environments*. Atlantis Ambient and Pervasive Intelligence, vol. 4. Atlantis Press, Amsterdam (2011). https://doi.org/10.2991/978-94-91216-05-3_8
12. De-La-Hoz-Franco, E., Bernal Monroy, E., Ariza-Colpas, P., Mendoza-Palechor, F., Espinilla, M.: UJA human activity recognition multi-occupancy dataset. In: *Proceedings of the 54th Hawaii International Conference on System Sciences* (2021). <https://doi.org/10.24251/hicss.2021.236>
13. Ramos, R.G., Domingo, J.D., Zalama, E., Gómez-García-Bermejo, J., López, J.: SDHAR-HOME: a sensor dataset for human activity recognition at home. *Sensors* **22**(21), 8109 (2022). <https://doi.org/10.3390/s22218109>
14. Arrotta, L., Bettini, C., Civitarese, G.: The MARBLE dataset: multi-inhabitant activities of daily living combining wearable and environmental sensors data. In: Hara, T., Yamaguchi, H. (eds.) *MobiQuitous 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 419, pp. 451–468. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94822-1_25
15. Chen, D., Yongchareon, S., Lai, E.M.-K., Yu, J., Sheng, Q.Z., Li, Y.: Transformer with bidirectional GRU for nonintrusive, sensor-based activity recognition in a multiresident environment. *IEEE Internet Things J.* **9**(23), 23716–23727 (2022). <https://doi.org/10.1109/jiot.2022.3190307>
16. Lentzas, A., Dalagdi, E., Vrakas, D.: Multilabel classification methods for human activity recognition: a comparison of algorithms. *Sensors* **22**(6), 2353 (2022). <https://doi.org/10.3390/s22062353>
17. Natani, A., Sharma, A., Peruma, T., Sukhavasi, S.: Deep learning for multi-resident activity recognition in ambient sensing smart homes. In: *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)* (2019). <https://doi.org/10.1109/gcce46687.2019.9015212>

18. Jethanandani, M., Sharma, A., Perumal, T., Chang, J.-R.: Multi-label classification based ensemble learning for human activity recognition in smart home. *Internet Things* **12**, 100324 (2020). <https://doi.org/10.1016/j.iot.2020.100324>
19. Natani, A., Sharma, A., Perumal, T.: Sequential neural networks for multi-resident activity recognition in ambient sensing smart homes. *Appl. Intell.* **51**(8), 6014–6028 (2021). <https://doi.org/10.1007/s10489-020-02134-z>
20. Ramanujam, E., Perumal, T.: MLMO-HSM: multi-label multi-output hybrid sequential model for multi-resident smart home activity recognition. *J. Ambient. Intell. Humaniz. Comput.* **14**(3), 2313–2325 (2023)
21. Perumal, T., Ramanujam, E., Suman, S., Sharma, A., Singhal, H.: Internet of things centric-based multiactivity recognition in smart home environment. *IEEE Internet Things J.* **10**(2), 1724–1732 (2023). <https://doi.org/10.1109/jiot.2022.3209970>
22. The multilayer perceptron. *Neural Computing - An Introduction* (1990). <https://doi.org/10.1201/9781420050431.ch4>
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
24. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014). <https://doi.org/10.3115/v1/d14-1179>
25. O’Shea, K., Nash, R.: An introduction to convolutional neural networks. arXiv preprint: [arXiv:1511.08458](https://arxiv.org/abs/1511.08458) (2015)

Metaverse for IoT (MIoT)



Forensics Analysis of Virtual Reality Social Community Applications on Oculus Quest 2

Samuel Ho  and Umit Karabiyik  

Purdue University, West Lafayette, IN 47906, USA
{ho176,umit}@purdue.edu

Abstract. As the popularity of virtual reality (VR) applications increases, there is a growing concern for the forensic aspect of privacy and security of user data. This paper aims to investigate the extent to which forensically relevant user data can be recovered from Oculus Quest 2, specifically, from four social community applications: Horizon Worlds, Multiverse, Rec Room, and VRChat. By performing a forensic analysis on Oculus Quest 2, we were able to determine to what extent forensically relevant user data could be recovered from four social-community applications. Existing efforts in this field are limited. Thus, this study adds to the growing body of research on the forensic analysis of VR applications.

Keywords: IoT Forensics · Mobile Forensics · Oculus Quest 2 · Virtual Communities

1 Introduction

Virtual Reality (VR) is currently used in a wide range of industries such as education, healthcare, and social communities. In education, VictoryXR has created immersive learning experiences by using VR to provide training to educators, teaching many subjects [26]. They expect to operate up to 100 digital twin campuses by 2023 [20]. In healthcare, VR goes hand-in-hand with Internet of Medical Things (IoMT) and Internet of Health Things (IoHT) based technologies. In November 2021, the U.S. Food and Drug Administration (FDA) approved a VR system called EaseVRx as back pain treatment for individuals above 18 years old [24]. For social communities, the metaverse has allowed people to meet up in virtual spaces such as restaurants, cinemas, and even concerts. Bouckaert [11] found that cheaper prices, visual effects, and accessibility are some of the motivators for people preferring virtual concerts to live concerts. Social VR platforms such as Meta's Horizon Worlds and Microsoft's AltspaceVR enable people to traverse virtual worlds, play games, and interact with friends as well as strangers online.

While VR brings many benefits, it can also lead to some drawbacks. For example, VR can provide an immersive and interactive learning experience that

can be difficult to replicate in traditional classroom settings. However, it can have negative effects on mental health and physical well-being if used excessively. Social community applications can foster connections and communication between individuals who may not have the ability to do so physically, but they can also contribute to social isolation due to addiction [10]. As of January 2023, 59.4% of the world's population (4.76 billion) use social media [25]. Given such a significant number of social media users, attention should be brought upon ensuring the data collected by social media applications are protected. With VR applications becoming increasingly used and a wide range of applications being available online for free, it is crucial to investigate how much forensically relevant unencrypted user data can be recovered from these devices and applications. The purpose of this research is to examine the privacy implications of VR and social media usage by analyzing the amount of user data that can be recovered from these applications using forensics techniques. Therefore, the research question of this research is "What forensically relevant user data is recoverable from social community applications (Meta Horizon World, Multiverse, Rec Room, and VRChat) on the Oculus Quest 2?"

Previous research focused on conducting forensic analysis to test for traces of artifacts on the HTC Vive and Oculus Rift [27]. However, no forensic analysis involving artifact traces has been conducted on the latest Oculus model - Oculus Quest 2. Additionally, Mahan [22] used mobile forensics to explore ransomware attacks on the Oculus Quest 2. However, the research did not include attempting to recover artifacts from the Oculus Quest 2. At this juncture, this paper attempts to analyze an Oculus Quest 2 using mobile forensic tools to find out if user data can be recovered from the device, and if so, how much user data can be recovered. This paper will focus on four social community applications - Horizon Worlds, Multiverse, Rec Room, and VRChat.

The remainder of this paper is structured as follows: Sect. 2 provides a review of the literature in regard to VR forensics and social community applications. Section 3 discusses all variables, definitions, and validity threats related to the present study. It also discusses the data population, acquisition, and analysis methodology from the current study. Section 4 highlights the results and findings of this study. Section 5 discusses the results and provides future research recommendations. Section 6 summarizes the entire study and the contributions acquired from the analysis.

2 Related Work

The following section discusses a review of the literature that was conducted regarding virtual reality social applications, forensic analysis of virtual reality, and mobile forensic investigations.

2.1 Use in Court

There are previous works that involved the analysis of private messages such as Short Message Service (SMS). Hammond [14] investigated the Brown v. Mayor

of Detroit case where text messages were used as evidence in a court of law. Ultimately, this led to perjury and misconduct charges while Mayor Kwame Kilpatrick and his Chief of Staff Christine Beatty were serving as public officers. The current study forensically analyzes the Oculus Quest 2 for artifacts such as message logs which can be used as possible inculpatory evidence.

2.2 Forensic Analysis

Johnson et al. [17] conducted forensic analysis on social networking applications. The study aimed to demonstrate the forensic need to understand how alternative-tech social applications operate and what they store about their users' personal information and activities. Similar to the current study, Magnet Acquire was used to acquire physical images of both the Android and iOS devices after testing each application. The findings revealed that user information such as usernames, emails, full names, phone numbers, profile pictures, and more could be found, along with posts and comments made, and private messages.

Hutchinson et al. [16] sought to understand and assess the forensic artifacts that can be extracted from IoT devices by performing a comprehensive investigation of the SimpliSafe security system. The authors investigated the interaction of the security system with the SimpliSafe companion app on both Android and iOS devices using Magnet AXIOM as well as the network traffic as the user interacts with the system to identify any security or privacy concerns using Wireshark. The findings revealed the disparity in recoverable artifacts when comparing the Android device to the iOS device. In terms of network traffic, SimpliSafe follows security standards and best practices.

Jones & Winster [18] conducted forensic analysis on smartphones aiming to uncover digital evidence in mobile phones that might be deleted by criminals. The authors did this by performing data acquisition of digital evidence from compromised devices using Oxygen Forensics. The findings revealed that many pieces of deleted forensically relevant user data could be retrieved through this process.

2.3 Oculus Quest 2

Mahan [22] conducted a forensics analysis of ransomware on the Oculus Quest 2. The study aimed to study how applicable Android ransomware is to the Oculus Quest 2's attack surface, due to the Quest 2's usage of Android 10 as a base operating system. The author used Android Debug Bridge (ADB), a SHA-256 hashing tool, and Android Studio for the acquisition of data. This setup allowed the author to test a simple ransomware sample (SRS), WannaLock, and Koler ransomware samples. The author found that the Oculus Quest 2's attack surface does contain the necessary aspects for the successful execution of ransomware. The current study used similar tools in the methodology section, utilizing ADB as well as Android Studio for the acquisition of data.

Yarramreddy et al. [27] conducted a forensic analysis of social applications on the HTC Vive and the Oculus Rift. The study aimed to present the first account for forensically relevant client-side and network-based artifacts generated by the HTC Vive and the Oculus Rift. The authors analyzed SteamVR, BigScreen, Rec Room, AltspaceVR, and Facebook Spaces. The authors used Wireshark for network traffic analysis. They also performed manual examination of all Steam, Oculus, and social application logs. The findings revealed that a large amount of data could be recovered from Bigscreen, Steam, and Facebook Spaces. The current study also analyzes social community applications as well, with the addition of Horizon Worlds, Multiverse, and VRChat. Additionally, the current study was focused on the Oculus Quest 2 instead of the HTC Vive or the Oculus Rift.

Hassenfeldt et al., [15] conducted memory forensics on immersive VR systems, specifically the HTC Vive. The study aimed to conduct the first experimental study to explore digital forensic learning in immersive VR versus a physical learning environment. The authors performed reconstruction of a physical scene using artifacts recovered from the file systems memory. This resulted in the first open-source VR memory forensics plugin for the Volatility Framework. The authors found that VR is more time efficient in the sense that participant completion times were faster. The current study aimed to conduct mobile forensics on the Oculus Quest 2 instead of the HTC Vive. The current study used a similar method in the methodology, accessing artifacts from the file systems memory.

The author from [5] performed forensic extractions on the Oculus Quest 2. The author used Cellebrite UFED 4PC and ADB USB debugging to do so. This resulted in data being successfully pulled from various directories including `/sdcard`, `/bugreports`, and `/storage`. However, most directories were either not pulled or were pulled but did not contain any data. The current study also attempted to use ADB USB debugging to pull files from various directories. Additionally, the current study used Magnet AXIOM instead of Cellebrite UFED 4PC as the forensic tool for imaging and analysis.

3 Methodology

The goal of this study was to discover the amount of recoverable forensically relevant user data from four social community applications on the Oculus Quest 2 through mobile forensic analysis. A virtual reality environment was used to achieve this. A private Wi-Fi network was used for the devices to have internet connectivity. DHCP assignment will be used to assign IP addresses to all devices. All assigned IP addresses follow an RFC 1918 [23] compliant addressing scheme and are in the 10.0.0.0/8 subnetwork. The following sections describe the scenario setup, measures, design, virtual environment, and data analysis methods for the present study.

3.1 Scenario Setup

The physical location of the investigation was within a single research lab. A possible scenario where this investigation would be useful is a case involving child grooming. Choo [12] defines child grooming as “a premeditated behavior intended to secure the trust and cooperation of children prior to engaging in sexual conduct.” Online grooming has become increasingly popular online due to new technologies such as cyberspace [13, 19]. Pedophiles also share information such as tricks to groom children for abuse with other pedophiles [9]. An offender would need a profile to communicate with children online. Given the offender used any of the social community applications investigated, forensically relevant information such as user height, message logs, and interactions can be used as both inculpatory and exculpatory evidence.

3.2 Measures

The single-group posttest-only design was employed for this study as the independent variable is manipulated with a single group. The independent variable is the social community application used. The dependent variable is the amount of recoverable forensically relevant user data from the social community applications. This study attempts to answer the question of “What forensically relevant user data is recoverable from social community applications on the Oculus Quest 2?” A potential threat to internal validity is due to the data population, acquisition, and analysis being done in a controlled virtual environment. This erases any factors and challenges that may influence the study in a real-world situation. To address this, real-world scenarios were replicated as closely as possible within the virtual environment. This involves designing virtual scenarios that mimic the characteristics, interactions, and usage patterns found in real-world VR applications. By creating realistic conditions, we increase the ecological validity of the current study.

3.3 Design

The testing setup consisted of the following hardware components: An Oculus Quest 2, a USB-C cable, a Dell Inspiron 15 laptop, and a Samsung Galaxy Tab S7 tablet. The testing setup consists of the following software components: Android Debug Bridge (ADB) [1], Magnet AXIOM [3], Horizon Worlds [2], Multiverse [6], RecRoom [7], VRChat [8], and Meta Quest [4]. The testing setup consists of connecting the Oculus Quest 2 to a Dell Inspiron 15 laptop that has forensic software Magnet AXIOM Acquire, Process, and Examine installed. Figure 1 shows a visual representation of the network architecture. The box in Fig. 1 represents the components that were forensically analyzed.

Data Population. The social community applications that were used are free software available in the Oculus Quest Store. A test user account (AnnaOup525)

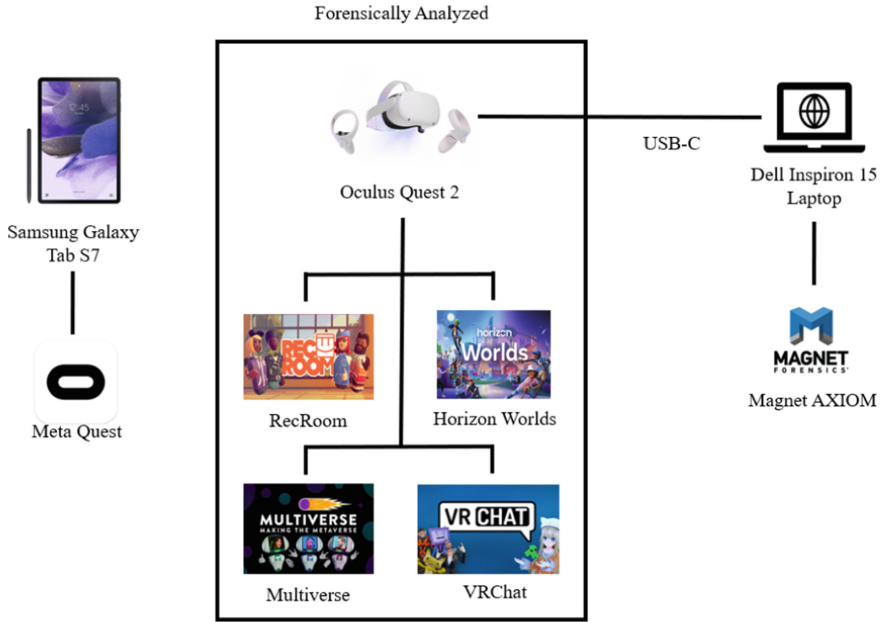


Fig. 1. Virtual Environment Network Diagram

was created and used throughout this study. This account was used to log in to the Meta Quest application on the Samsung Galaxy Tab S7 as well as the Oculus Quest 2. After logging in, the four social community applications were downloaded and installed. A full list of applications and software used can be seen in Table 1.

Table 1. List of the applications and software used in this study

Application Name	Build/Version Number	Usage
Rec Room	20230414	Social community application
VRChat	2023.2.1	Social community application
Multiverse	0.2	Social community application
Horizon Worlds	107.0.0.7.169	Social community application
Meta Quest	212.0.0.2.109	Central application
Android Debug Bridge version 1.0.41	34.0.1-9680074	Acquire evidence
Magnet AXIOM Acquire	2.26.0.20671	Acquire evidence
Magnet AXIOM Process	6.11.0.34807	Acquire evidence
Magnet AXIOM Examine	6.11.0.34807	Analyze evidence

The same activities were done on each of the four social community applications to generate user data on the device. For the purposes of interaction, a

second account (*****hpa) was used. All user interactions were made with the researchers' second account. A short summary of the activities done on the device is as follows:

- Customizing the avatar by changing its appearance, clothing, and accessories
- Joining different communities or rooms
- Interacting with researcher's second account through voice and text chat
- Taking screenshots or pictures within the application
- Using blocking and reporting features to report researcher's second account

Data Analysis. This study used the following methods to analyze the Oculus Quest 2.

Forensics Imaging. The first method was to acquire an image of the Oculus Quest 2. The Oculus Quest 2 was connected to the laptop via a USB-C cable. AXIOM Acquire was used to create a forensically sound image of the device and its storage media. As the Oculus Quest 2 was not rooted, there was no authorized access to the full file system. This resulted in a quick image being made instead of a logical or full image. Hash values were used to verify the integrity of the image and ensure that it is an exact copy of the original data. Once verified, Magnet AXIOM Process was used to load the image into Magnet AXIOM Examine which was then used to extract data from the image. Forensically relevant files and artifacts, such as application data, system logs, and user data were examined to check for any recoverable forensically relevant user data. Examples of forensically relevant user data include personally identifiable information (PII), login credentials, and interaction history.

Log Analysis. The second method was to connect the Oculus Quest 2 to the laptop and manually perform log analysis on available files and folders.

4 Results

The following section describes the results obtained from the analysis section. Results are reported in two parts - general findings and application-specific findings.

4.1 General Findings

As seen in Fig. 2, the latest WiFi scan results were stored in a file called `OculusQuest2QuickImage.zip` `sdcard` `sdcard` `Android` `logs` `wifi-scanner.txt`. We can see that the ***3.0 network was scanned. From Fig. 3, we can see that a successful connection was logged. The connection was logged in with the researcher's username (**76) as well as the domain (*****.edu) which can be seen in Fig. 4.

Latest scan results:

BSSID	Frequency	RSSI	Age(sec)	SSID
8e:2a:a8:81:5b:8f	2412	-50(0:-56/1:-52)	>1000.0	CIT-ADMIN
70:4f:57:15:b6:87	2427	-51(0:-56/1:-53)	>1000.0	AP
70:4f:57:15:b6:86	5745	-58(0:-62/1:-61)	>1000.0	AP_5G
24:81:3b:4e:19:a0	2437	-58(0:-64/1:-59)	>1000.0	PAL3.0
96:2a:a8:81:5b:8f	2412	-60(0:-62/1:-64)	>1000.0	CIT-PI
92:2a:a8:81:5b:8f	2412	-60(0:-62/1:-66)	>1000.0	CIT-CFL
80:2a:a8:81:5b:8f	2412	-61(0:-63/1:-65)	>1000.0	CIT-NET
24:81:3b:4e:19:ae	5240	-62(0:-63/1:-70)	>1000.0	attwifi
24:81:3b:4e:19:af	5240	-63(0:-65/1:-67)	>1000.0	PAL3.0
24:81:3b:4e:19:ad	5240	-63(0:-64/1:-71)	>1000.0	eduroam
6a:17:29:9a:b2:90	2462	-64(0:-66/1:-68)	>1000.0	c240-344g55
62:6c:66:cc:d2:10	2412	-64(0:-66/1:-68)	>1000.0	c240-344g62
a0:0f:37:62:03:83	2437	-66(0:-70/1:-69)	>1000.0	PAL-Recreational
62:6c:66:23:d5:d0	2412	-68(0:-70/1:-74)	>1000.0	c240-344g38
86:2a:a8:82:5b:8f	5180	-69(0:-71/1:-73)	>1000.0	CIT-DEV
8e:2a:a8:82:5b:8f	5180	-69(0:-72/1:-73)	>1000.0	CIT-ADMIN
96:2a:a8:82:5b:8f	5180	-69(0:-71/1:-74)	>1000.0	CIT-PI
8a:2a:a8:82:5b:8f	5180	-69(0:-71/1:-74)	>1000.0	CIT-HPC

Fig. 2. SSID Latest Scan

```

rec[11]: time=04-26 17:43:22.180 processed=ConnectingOrConnectedState
org=L2ConnectingState dest= what=SUPPLICANT_STATE_CHANGE_EVENT
screen=on 0 0 ssid: PAL3.0 bssid: 24:81:3b:4e:19:af nid: 2 state:
GROUP_HANDSHAKE
rec[12]: time=04-26 17:43:22.213 processed=ConnectingOrConnectedState
org=L2ConnectingState dest=L3ProvisioningState
what=NETWORK_CONNECTION_EVENT screen=on 2 false
24:81:3b:4e:19:af nid=2 "PAL3.0" WPA_EAP last=
rec[13]: time=04-26 17:43:22.236 processed=ConnectingOrConnectedState
org=L3ProvisioningState dest=
what=SUPPLICANT_STATE_CHANGE_EVENT screen=on 0 0 ssid: PAL3.0
bssid: 24:81:3b:4e:19:af nid: 2 state: COMPLETED

```

Fig. 3. WiFi Connectivity Success

```

fig: subject_match NULL altsubject_match NULL proactive_k
ey_caching 1 client_cert NULL key_id NULL wapi_cert_suite
NULL plmn NULL domain_suffix_match "purdue.edu" anonymou
s_identity NULL password <removed> decorated_username_pref
ix NULL engine 0 engine_id NULL identity "ho176" ca_path

```

Fig. 4. WiFi Profile Identifier

4.2 Horizon Worlds

The following findings were found from the image created through Magnet AXIOM. There were several forensically relevant user and application data that could be found from Horizon Worlds. Figure 5 shows the last used time and duration used for the Horizon Worlds application. Unlike for Multiverse, Rec Room, and VRChat, the total time used and last time used were 00:00 and 1969-12-31 19:00:00 respectively. As these are default UNIX timestamps, it can be assumed that this data was either not available or it is encrypted.

```
package=com.oculus.horizon totalTimeUsed="00:00" lastTimeUsed="1969-12-31 19:00:00" totalTimeVisible="00:00" lastTimeVisible="1969-12-31 19:00:00" lastTimeComponentUsed="2023-04-26 18:36:02"
totalTimeFS="2:15:32" lastTimeFS="2023-04-26 18:35:58" appLaunchCount=0
```

Fig. 5. Horizon Worlds Last Used Time

Another artifact that was found was a photo that was taken showing the interaction of the two user accounts. This can be seen in Fig. 6.

The following findings were found through log analysis on the Quest 2 Android data `com.facebook.horizon` Horizon Logs `socialvr_2023-04-24.12.51.01.log` log file.

The first artifact was the username and EPOCH time of the user name tag being created. This can be seen in Fig. 7.

One of the actions taken was sending follow requests. As seen in Fig. 8, the second account's username, ID, avatar head shot picture, event type, and EPOCH time of event were logged.

Additionally, another action is navigating to different rooms. For Horizon World, both accounts navigated to a room called "Arena Clash Winter." As seen in Fig. 9, the name of the room and EPOCH time entered were logged.

4.3 Multiverse

The following findings were found from the image created through Magnet AXIOM. There were several forensically relevant user and application data that could be found in Multiverse. Figure 10 shows the last used time and duration used for the Multiverse application. The total time used and last time used were 32:01 and 2023-04-24 12:48:58 respectively. These timestamps were very accurate as they matched the logged data population time.

In addition, there were Binary Large Object (BLOB) cache files that had signature mismatches. Figure 11 and Fig. 12 were recovered using file carving. These were the profile pictures of the first and second test user accounts in Multiverse.

No useful forensically relevant user data was found through log analysis.

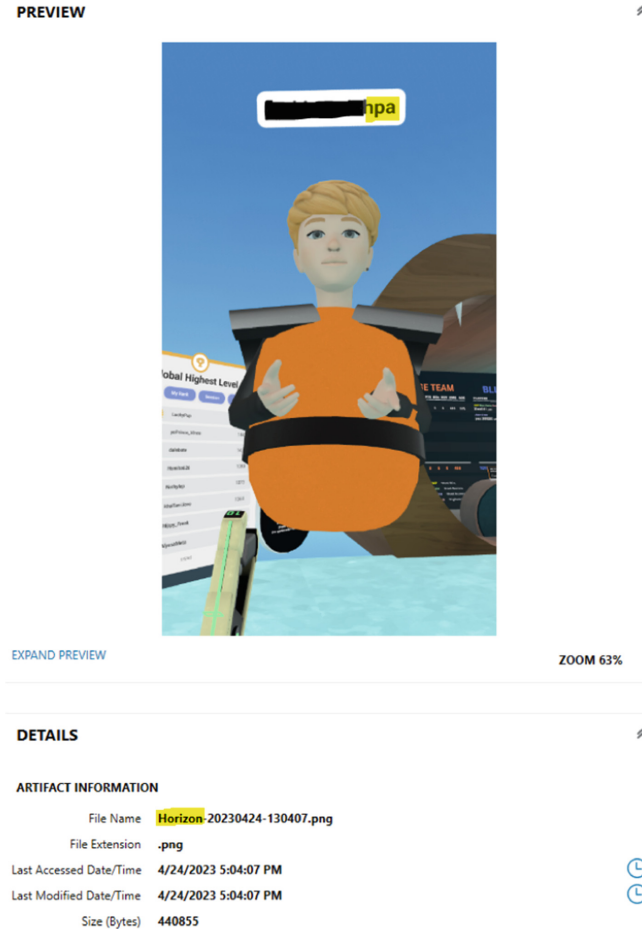


Fig. 6. Horizon Worlds Selfie and Username

```
{"Hash": "1ab71fba-ef49-4d2b-93bc-5973fbc98144", "Time": 1682355070.38, "Route": null, "Uuid": null, "Level": "INFO", "Tag": "playernametag", "Message": "Spawning nametag rvr for player AnnaOup525, togetherAppUserId:107630872312192", "Stack": ""}
```

Fig. 7. Horizon Worlds Player Username

4.4 Rec Room

The following findings were found from the image created through Magnet AXIOM. There were very limited forensically relevant user and application data that could be found in Rec Room. Figure 13 shows the last used time and duration used for the Rec Room application. The total time used and last time used were 32:11 and 2023-04-24 12:29:59 respectively. These timestamps were very accurate as they matched the logged data population time.

```
{
  "Hash": "8c572776-ffbc-416c-b7f2-88df99790093",
  "Time": 1682356036.312,
  "Route": null,
  "Uuid": null,
  "Level": "INFO",
  "Tag": "requeststream [stream_client]",
  "Message": "PayloadCallback received {\\data\\":{\\
    \\horizon_notifications_subscribe\\":{\\__typename\\":\\
    \\HorizonNotificationFollowRequestAccepted\\",
    \\type\\":\\\"FOLLOW_REQUEST_ACCEPTED\\",
    \\message\\":\\\"\\\",
    \\sender_id\\":\\\"1603092963184055\\\",
    \\sender_name\\":\\\"jenkinsmichpa\\\",
    \\sender_together_user\\":{\\avatar_head_shot_picture\\":\\\"https://scontent.xx.fbcdn.net/v/t66.39824-6/310817447_116333714747250_4777672844358370278_n.png?_nc_cat=106&ccb=1-7&_nc_sid=be3241&_nc_ohc=tnag4KnLnAYAX9EECLd&_nc_ad=z-m&_nc_cid=0&_nc_ht=scontent.xx&oh=00_AfCF148kNq452GHkTIqJR5tIREh0z13RS0drasnPk_Iw&oe=64483811\\\",
    \\vr_persona\\":{\\vr_alias\\":\\\"jenkinsmichpa\\\",
    \\id\\":\\\"1603092963184055\\\",
    \\id\\":\\\"1603092966517388\\\",
    \\formatted_message\\":\\\"jenkinsmichpa accepted your follow request\\\",
    \\strong_id\\":null}}},
  "Stack": ""
}
```

Fig. 8. Horizon Worlds Follow Request Accepted

```
{
  "Hash": "8ed38c5d-ff0e-43a0-bd8e-0ebdba6e3ed2",
  "Time": 1682355389.63,
  "Route": null,
  "Uuid": null,
  "Level": "INFO",
  "Tag": "navrvr",
  "Message": "WBDiscoveryWorldNavigationHelper navigating to Arena Clash Winter : together:\\\\world_builder\\\\wb_visit?world_id=151752854252349&snapshot_id=158976103530024 with options SKIP_CONFIRMATION",
  "Stack": ""
}
```

Fig. 9. Horizon Worlds Navigating to Rooms

```
package=itd.ftl.multiverse.oculus.quest totalTimeUsed="32:01"
lastTimeUsed="2023-04-24 12:48:58" totalTimeVisible="32:09"
lastTimeVisible="2023-04-24 12:49:00" lastTimeComponentUsed="2023-04-24
12:38:12" totalTimeFS="00:00" lastTimeFS="1969-12-31 19:00:00"
appLaunchCount=7
```

Fig. 10. Multiverse Last Used Time

No useful forensically relevant user data was found through log analysis.

4.5 VRChat

The following findings were found from the image created through Magnet AXIOM. There were several forensically relevant user and application data that could be found in VRChat. Figure 14 shows the last used time and duration used for the VRChat application. The total time used and last time used were 17:39 and 2023-04-24 13:31:33 respectively. These timestamps were very accurate as they matched the logged data population time.

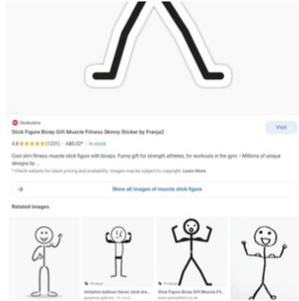
Another artifact was a picture taken using the camera feature in VRChat. This picture shows the interaction between the two user accounts. This can be seen in Fig. 15.

No useful forensically relevant user data was found through log analysis.

5 Discussion and Future Recommendations

By conducting a forensic analysis on the Oculus Quest 2, we were able to determine the extent to which forensically relevant user data could be recovered from

PREVIEW



DETAILS

ARTIFACT INFORMATION

File Name	C0F7CDE7CF0C8FB376D5039B564271320290321B1384CF520C7D2876DBABFE11
Last Accessed Date/Time	4/24/2023 4:40:27 PM
Last Modified Date/Time	4/24/2023 4:40:27 PM
Size (Bytes)	67007

EVIDENCE INFORMATION

Source	Oculus Quest 2 Quick Image.zip\sdcard\sdcard\Android\data\ltd.ftl.multiverse.oculus.quest\files\cache\blob_cache\data\C0\F7\C0F7CDE7CF0C8FB376D5039B564271320290321B1384CF520C7D2876DBABFE11
Recovery method	Carving

Fig. 11. Multiverse First Account Profile Picture

4 social community applications. Based on the scenario provided in Section III-A, the results of our forensic analysis on the Oculus Quest 2 could have significant ramifications in cases involving child grooming. If an offender were to use any of the social community applications investigated in our study to communicate with children, forensically relevant information such as user network connections, message logs, and interactions could be used as both inculpatory and exculpatory evidence. Furthermore, the current study contributes to the development of best practices for conducting forensic analysis of VR applications and helps educate and inform future researchers in this area.

As the base operating system for the Oculus Quest 2 was Android, Android Debug Bridge was needed to create an ADB image. The initial methodology consisted of an attempt at using Magnet Acquire to image the Oculus Quest. However, the imaging process only allowed for a quick image. Therefore, neither a logical image nor a full image was acquired. Future research could focus on more methods of imaging the Oculus Quest 2 to see if more artifacts could be found on the file system. NIST-validated forensic tools [21] such as FTK Imager could be

PREVIEW



EXPAND PREVIEW

ZOOM 100%

DETAILS



ARTIFACT INFORMATION

File Name	9F8B44A7E2E4BF3DA1815BD3422CB52A8BF620ADAC90D56D620766E4092ED2FB_128	
Last Accessed Date/Time	4/24/2023 4:40:25 PM	
Last Modified Date/Time	4/24/2023 4:40:25 PM	
Size (Bytes)	3808	

EVIDENCE INFORMATION

Source	Oculus Quest 2 Quick Image.zip\sdcard\sdcard\Android\data\ltd.ftl.multiverse.oculus.quest\files\cache/blob_cache\data\9F8B44A7E2E4BF3DA1815BD3422CB52A8BF620ADAC90D56D620766E4092ED2FB_128	
Recovery method	Carving	

Fig. 12. Multiverse Second Account Profile Picture

tested to see if an image of the Oculus Quest 2 could be made. Another limitation of the current study is the small sample size of social community applications that were analyzed ($n = 4$). Future research could expand upon these limitations to gain a more comprehensive understanding of the forensic artifacts present in VR applications and the potential risks to user privacy.

```
package=com.AgainstGravity.RecRoom totalTimeUsed="32:11"
lastTimeUsed="2023-04-24 12:29:59" totalTimeVisible="32:28"
lastTimeVisible="2023-04-24 12:30:06" lastTimeComponentUsed="2023-04-24
12:24:36" totalTimeFS="00:00" lastTimeFS="1969-12-31 19:00:00"
appLaunchCount=8
```

Fig. 13. Rec Room Last Used Time

```
package=com.vrchat.oculus.quest totalTimeUsed="17:39"  
lastTimeUsed="2023-04-24 13:31:33" totalTimeVisible="18:02"  
lastTimeVisible="2023-04-24 13:31:39" lastTimeComponentUsed="2023-04-24  
13:27:25" totalTimeFS="00:00" lastTimeFS="1969-12-31 19:00:00"  
appLaunchCount=7
```

Fig. 14. VRChat Last Used Time

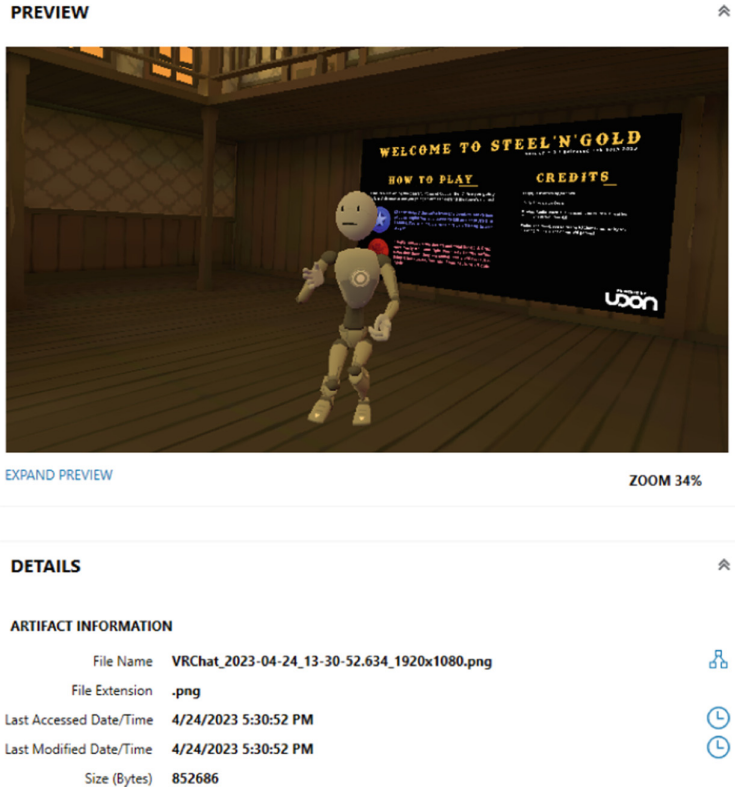


Fig. 15. VRChat Camera Interaction

6 Conclusions and Recommendations

The current study analyzed four social community applications - Horizon Worlds, Multiverse, Rec Room, and VRChat on the Oculus Quest 2 using mobile forensic tools to determine if forensically relevant user data could be recovered from the device. The results of the study showed that some forensically relevant user data, such as timestamps, usernames, follow requests, and profile pictures could be recovered from the file system of the device. These findings suggest that there are potential artifacts that could be used in cases such as child grooming

via virtual reality social community applications, which can help support the investigation and prosecution of crimes involving the use of these technologies.

Overall, this study highlights the importance of considering privacy concerns when using VR applications. Developers need to prioritize user data protection in their designs. As VR technology continues to advance and become more integrated into our daily lives, it is crucial to ensure that our personal information remains secure.

References

1. Android debug bridge. <https://developer.android.com/studio/command-line/adb>
2. Horizon worlds. <https://www.meta.com/horizon-worlds/>
3. Magnet axiom forensics. <https://www.magnetforensics.com/products/magnet-axiom>. Accessed 03 Feb 2023
4. Meta quest. https://play.google.com/store/apps/details?id=com.oculus.twilight&hl=en_US&gl=US&pli=1
5. Meta quest 2 forensic extraction (testing). <https://revo4n6.com/blog-posts/f/meta-quest-2-forensic-extraction-testing>. Accessed 18 May 2023
6. Multiverse. <https://www.oculus.com/experiences/quest/4140152679403866/>
7. Rec room. <https://recroom.com/>
8. Vrchat. <https://www.oculus.com/experiences/quest/1856672347794301/>
9. Adam, A.: Cyberstalking and internet pornography: gender and the gaze. *Ethics Inf. Technol.* **4**(2), 133 (2002)
10. Adamski, D.: The influence of new technologies on the social withdrawal (hikikomori syndrome) among developed communities, including poland. *Soc. Commun.* **4**(1), 58–63 (2018)
11. Bouckaert, L.: Virtual reality as new concert space. Ph.D. thesis, Universiteit Gent, Belgium (2021)
12. Choo, K.K.R.: Online child grooming: a literature review on the misuse of social networking sites for grooming children for sexual offences (2009)
13. Choo, K.K.R., Smith, R.G., McCusker, R., Choo, K.K.R.: Future directions in technology-enabled crime: 2007–09. *Citeseer* (2007)
14. Hammond, T.: Text messages and the detroit mayor: Kwame kilpatrick. *Online J. Commun. Media Technol.* **3**(3), 146–177 (2013)
15. Hassenfeldt, C., Jacques, J., Baggili, I.: Exploring the learning efficacy of digital forensics concepts and bagging & tagging of digital devices in immersive virtual reality. *Forensic Sci. Int. Digit. Invest.* **33**, 301011 (2020)
16. Hutchinson, S., Stanković, M., Ho, S., Houshmand, S., Karabiyik, U.: Investigating the privacy and security of the simplisafe security system on android and iOS. *J. Cybersecur. Priv.* **3**(2), 145–165 (2023)
17. Johnson, H., Volk, K., Serafin, R., Grajeda, C., Baggili, I.: Alt-tech social forensics: forensic analysis of alternative social networking applications. *Forensic Sci. Int. Digit. Invest.* **42**, 301406 (2022)
18. Jones, G.M., Winster, S.G.: Forensics analysis on smart phones using mobile forensics tools. *Int. J. Comput. Intell. Res.* **13**(8), 1859–1869 (2017)
19. Kierkegaard, S.: Cyberbing, online grooming and ageplay. *Comput. Law Secur. Rev.* **24**(1), 41–55 (2008)
20. Kshetri, N., Rojas-Torres, D., Grambo, M.: The metaverse and higher education institutions. *IT Prof.* **24**(6), 69–73 (2022)

21. Lyle, J.: Computer forensic tool testing at NIST (2007). <http://www.cftt.nist.gov/documents/Amalfi-04.ppt>
22. Mahan, M.E.: Exploring ransomware on the oculus quest 2. Ph.D. thesis, Louisiana Tech University (2022)
23. Rekhter, Y., Moskowitz, B., Karrenberg, D., Groot, G.D., Lear, E.: RFC 1918: address allocation for private internets (1996). <https://www.rfc-editor.org/rfc/rfc1918.html>. Accessed 01 Mar 2022
24. Sato, T., Ishimaru, H., Takata, T., Sasaki, H., Shikano, M.: Application of internet of medical/health things to decentralized clinical trials: development status and regulatory considerations. *Front. Med.* **9**, 903188 (2022)
25. Statista: Number of internet and social media users worldwide as of January 2023 (in billions). <https://www.statista.com/statistics/617136/digital-population-worldwide/#:~:text=As%20of%20January%202023%2C%20there,population%2C%20were%20social%20media%20users>. Accessed 17 Apr 2023
26. Wang, M., Yu, H., Bell, Z., Chu, X.: Constructing an edu-metaverse ecosystem: a new and innovative framework. *IEEE Trans. Learn. Technol.* **15**(6), 685–696 (2022)
27. Yarramreddy, A., Gromkowski, P., Baggili, I.: Forensic analysis of immersive virtual reality social applications: a primary account. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 186–196. IEEE (2018)



Objective Emotion Quantification in the Metaverse Using Brain Computer Interfaces

Anca O. Muresan¹(✉)(iD), Meenalosini V. Cruz²(✉)(iD),
and Felix G. Hamza-Lup²(✉)(iD)

¹ Florida Atlantic University, Boca Raton, USA
amuresan2023@fau.edu

² Georgia Southern University, Savannah, USA
{mvimalcruz, fhamzalup}@georgiasouthern.edu

Abstract. Certain human emotions can be quantified by processing electroencephalography (EEG) data. Recent advances in Brain Computer Interfaces (BCI) allow us to record, process and determine user functional intent and emotional implication from such data. The Metaverse captures an extensive spectrum of multi-modal content on the Internet including social media, games, videos, and more complex VR, AR, MR platforms. We propose an objective method to quantify user emotion using EEG data collected through non-invasive BCIs during user interaction. BCI's qualify as IoT sensors that record EEG data in real-time as users are exploring multimedia content through several emotion-generating scenarios.

Keywords: Brain Computer Interfaces · IoT · Emotion Assessment · Metaverse · Affective Computing

1 Introduction

The advent of better and inexpensive virtual reality (VR) consumer hardware has stimulated (neuro) psychological attention and emotional research with multimedia based immersive environments. Particularly, the Metaverse-based VR experiences offers opportunities for individuals to express emotions in diverse ways which provide a range of expressions that contribute to a richer emotional data for analysis using classification algorithms. The multimedia environments are dynamic and customizable and emotion classification may contribute to emotional analytics for virtual spaces. In this study, we analyze human emotion by capturing EEG data using BCI's (IoT real-time sensors) while the users are exploring various multimedia scenarios suitable for the metaverse. By assessing emotional cues and patterns during virtual interactions, emotion classification algorithms can support features such as emotion-based matchmaking, emotional chat-bots, or emotional sentiment analysis of online communities. Understanding the feelings of other people is part of successful social interactions in humans'

© IFIP International Federation for Information Processing 2023

Published by Springer Nature Switzerland AG 2023

D. Puthal et al. (Eds.): IFIPIoT 2023, IFIP AICT 683, pp. 353–361, 2023.

https://doi.org/10.1007/978-3-031-45878-1_24

daily life. Automatic emotion recognition is advantageous in various fields such as healthcare and mental health, market research, security, surveillance, education and learning etc. Emotions can be represented by various sensory cues, such as facial expressions, hand gestures, body movements, and vocal intonations [9]. One of the categorization criteria of emotions is valence (pleasantness-unpleasantness) and arousal (level of activation); these are the building blocks of emotions and are instrumental while understanding the psychological construct of emotions [11]. The primary emotions tracked in affective computing applications are happiness, sadness, anger, fear, disgust, and surprise [3]. There are several technologies and methods used to quantify human emotion such as facial expression analysis, gesture and body movement analysis, and physiological signal analysis [6]. Scrutinizing physiological signals, such as heart rate, electrodermal activity (EDA), or brain activity, can provide more reliable clues into emotional states. In recent years, the development of dry sensor technology capable of sensing human brain activity led to the employment of BCIs in various fields ranging from rehabilitating stroke victims [1] allowing pelagic people access to forms of competition [2]. BCIs enables data collection regarding brain activity and can be connected to the Internet as IoT devices. BCI technology has the potential to connect internal human predilection under the form of EEG data with external devices, enabling seamless communication between the humans and the computers [7]. BCI technology is classified as invasive, semi-invasive, or non-invasive [4]. Electroencephalography (EEG) is a non-invasive technology that measures the electrical activities of the human brain through electrodes placed on the scalp area [5]. These electrodes record the electrical impulses of the brain. The frequency bands of interest are Delta (<4 Hz), Theta (4–8 Hz), Alpha (8–13 Hz), Beta (13–30 Hz), and Gamma (>30 Hz). These physiological signals gathered from EEG must be preprocessed through denoising, filtering, and amplification techniques. The electrodes provide the voltage difference data that is converted to frequency data using Fourier Transform or Wavelet Transform techniques based on the application. Affective studies employed the Self-Assessment Manikin (SAM) system [1] to standardize the emotions that subjects can report. The SAM system has the respondents gauge emotions based on an effective scale of 1 to 5 for 3 different categories: Happiness, Excitement, and the sense of Control. The rating of these 3 factors is positively correlated to previously used emotional reporting practices [2]. When interpreting EEG data with the goal of emotion detection, there are two primary models for representing them: a Discrete and a Dimensional model. The Discrete model will predict a clear emotion, such as Happiness or Disgust. A Dimensional model will utilize two or three different dimensions to plot the model in space [8]. Valence (V) is a measure of happiness, arousal (A) is used to measure the excitement level, dominance is used to distinguish between emotions that have similar VA measures, such as fear and anger, and it ranges from the feeling of being controlled to being in control. These three domains directly correspond to the three feelings that are surveyed in the SAM. Implicit emotional attention might be a more meaningful metric compared with explicit emotional attention because emotional arousal is

seen as a crucial element for research in VR. Evidence suggests that immersion, presence, and emotion are related only to arousing, but not to non-arousing neutral content. However, the exact relationship between immersion, presence, and emotion remains unclear [2, 4], hence the importance of research on emotion and attention effects in VR. Alternatively, in the field of human-computer interaction, the correlation of attention and emotion can tell us about the quality of the Metaverse experience. It has been shown that emotional responses, such as arousal ratings, are enhanced in Metaverse experiences, e.g., by using either a “low immersion” or a “high immersion” with the surrounding presentation. The following is the outline of the remaining segments of this paper. Section 2 provides a literature review on EEG-based techniques for human emotion classification both in virtual and non-virtual environments. Section 3 elaborates on the methodology and experimental setup. Section 4 presents the experimental results while Sect. 5 presents the overall findings and conclusion.

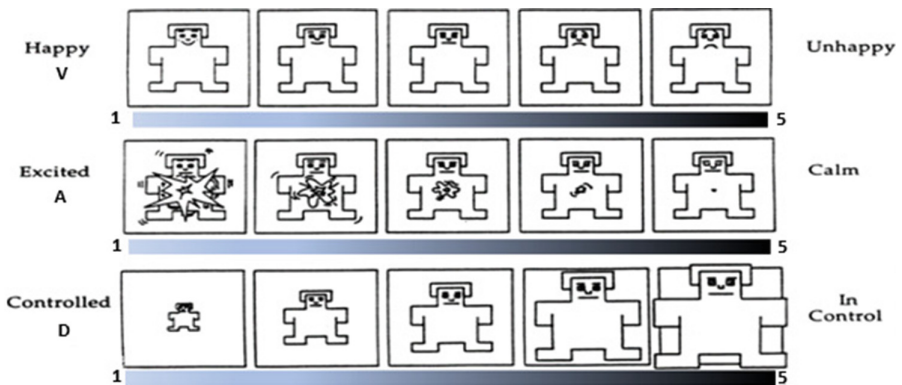


Fig. 1. Self-Assessment Manikin System VAD Parameters

Problem Statement: Human emotion appraisal is a challenging assignment, both subjectively by self-understanding one’s emotions and objectively aided by software. Brainwave data is an exciting resource in researching emotions, carrying the potential to liaise the knowledge gaps and predict emotional reactions to certain stimuli. By analyzing users’ emotional responses to virtual environments correlated with brainwave data, developers may gain insights into the emotional impact of specific UI design elements, interactions, or content leading to positive ramifications in various fields, such as online medical assistance, learning, counseling.

2 Related Work

Several researchers worked on emotion recognition utilizing the signals from EEG data by combining the Virtual Reality environment. Longo B et al. [1]

integrated BCI technology with VR to enable direct brain control for stroke patients. The results demonstrated promising outcomes in motor performance improvements, functional gains, and changes in brain activity and connectivity. Yan et al. [5] examined the existing research landscape, identifies key trends, and themes, and proposes future research directions in the field of emotion recognition using BCIs. The findings and research agenda contributed to the advancement of emotion recognition and provide directions for further exploration and development in the field. Özerdem et al. [10] focused on emotion recognition using EEG features extracted from movie clips and employing channel selection techniques. The results demonstrate the potential of EEG-based emotion recognition in response to multimedia stimuli. Javier Marín-Morales et al. [12] developed a system that can accurately detect and classify emotions based on physiological signals obtained from individuals immersed in a VR environment. The study described the experimental setup, which involved participants wearing wearable sensors while engaging in a VR scenario designed to elicit specific emotions. The captured EEG data were preprocessed and analyzed with Machine Learning algorithms, such as Support Vector Machines (SVM) or Deep Neural Networks (DNN). The findings suggested that wearable sensors can provide valuable insights into emotional states during VR experiences. Similarly, David Schubring et al. [13] investigated the effects of multimedia data on alpha/beta brain oscillations in relation to emotions and cognitive tasks. Their findings have implications for understanding the neural processes underlying emotions and cognition in VR environments. Hongyu Guo et al. [14] developed an emotion-based analysis method for designing English language lessons in the Metaverse. They have put forward a compelling argument on integrating emotional factors into the instructional design process would help to enhance learner engagement, motivation, and language acquisition outcomes. Their work emphasized the potential benefits of the metaverse in language instruction and provides pedagogical considerations for successful implementation. Eduardo Perez-Valero et al. [15] used a VR and EEG combo to quantitatively analyze the stress level during a stress-relaxation session. They used changes in power spectral density and coherence within these frequency bands were examined to identify stress and relaxation patterns. Similarly, Rhaíra Helena Caetano E Souza et al. [16] explored the feasibility of utilizing EEG to measure the attentional states of individuals immersed in virtual scenarios. Techniques such as Power Spectral Analysis (PSA), Time-Frequency (TFA) Analysis, or ERP components extraction were employed to identify attention-related patterns. The International Affective Picture System (IAPS) is a standardized set of images widely used in psychological and neuroscience research to elicit emotional responses. Lang et al. [17] worked on preparing the manual providing comprehensive information on the IAPS, including detailed instructions on its use and a compilation of affective ratings for the image stimuli. The manual outlines the procedures for image selection, and normative ratings, and provides an extensive database of affective ratings for the stimuli. Yan et al. [18] successfully implemented an emotion recognition model that utilizes EEG data, focusing on the rhythm and time characteristics

of brain-wave signals. They used features such as power spectral density, signal entropy, and wavelet transform coefficients to capture the unique rhythmic and temporal properties of the EEG signals. The data were classified using Rhythmic Time EEG Emotion Recognition Model generating the highest average recognition accuracy 0.69 which is 0.07 higher than the traditional SVM and KNN models. Nazmi Sofian Suhaimi et al. [19] examined the current trends and opportunities in the field, focusing on the use of EEG signals to detect and classify human emotions.

3 Experimental Setup: Emotion Induction Method and Data Collection

The data acquisition phase of the experimental setup is accomplished by utilizing the Neuro-Sky MindwaveTM BCI headset, and the Self-Assessment Manikin (SAM) technique. The resulting dataset has been Z-score normalized and includes the users' brainwave activity under the form of various frequency signals(recordings), their associated attention and meditation levels and the SAM rating for the respective recordings. During the brainwave data acquisition phase, the subjects have been requested to wear the headset while watching videos triggering sadness, happiness, anger, and fear. Subjects have been exposed to both visual and auditive stimuli as well as requested to rank and report their emotional intensity after each video using the standardized SAM technique, on a scale from 1 to 5, for Valence (Happiness), Arousal (Excitement), and Dominance (sense of Control), as depicted in Fig. 1. To facilitate a clear distinction between the targeted emotions, the videos are separated by a black screen pause.

4 Data Analysis and Results

Both neural network evaluations consist of 8 input nodes, 3 hidden nodes, 2 output nodes. The first type of data analysis utilizes Tensor Flow Keras Sequential Neural Networks Software to objectify and forecast possible VAD values against the SAM reported values (outputs) and considers the users' brainwaves as inputs, materializing the experiment with correlation indexes for each VAD parameter, ranging from 0 to 1. The methodology outcomes suggest that there might be a connection between brainwave data and VAD parameters that can be predicted with 89% accuracy for Dominance, and around 70% accuracy for Valence and Arousal, which may lead to promising results if confirmed by enlarging the data pool. Figure 2 showcases the performance for each VAD parameter.

The second type of data analysis implements MatLab Neural Networks and still considers the collected brainwave data as predictor variables (inputs), and the attention and meditation levels provided by the headset as outputs. The outcome of this appraisal are the regression indexes, showcasing the degree of correspondence between the predictor variables and response variables (outputs). The dataset is split in 3 main categories: training, validation and test; each

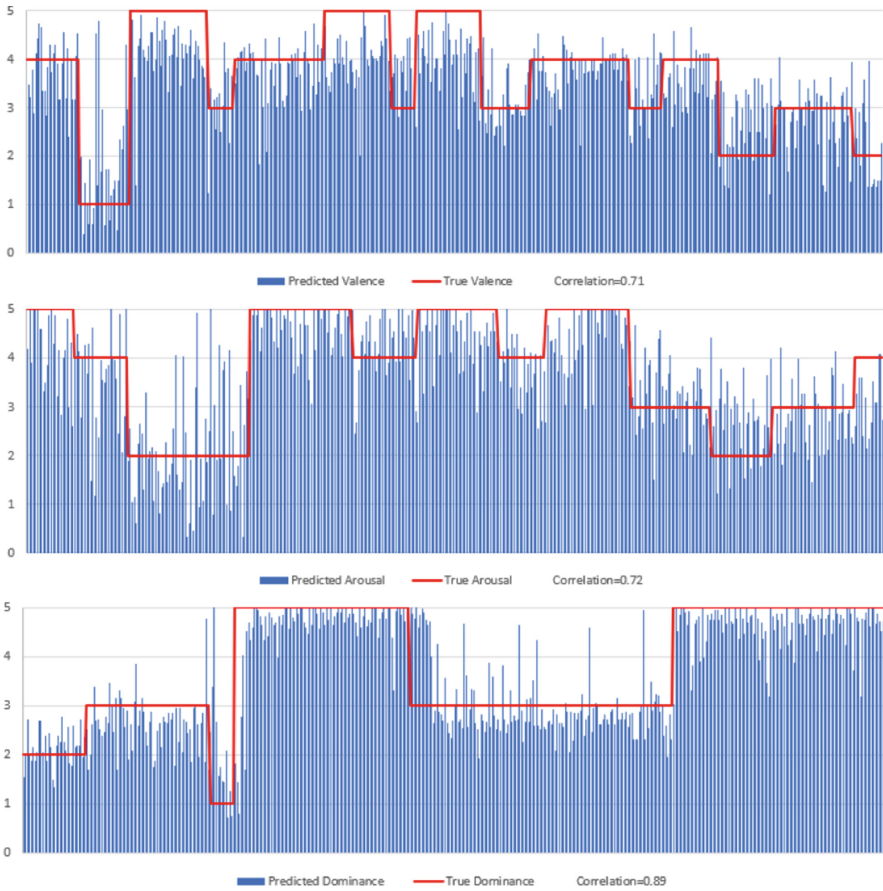


Fig. 2. Juxtaposition of actual and forecasted VAD parameters using TFKS Neural Networks

of them has its respective regression value, as well as computing the overall regression index, all ranging from 0 to 1. In examining the results, one may observe that the training regression level is quite high at 0.85, this might suggest that the SAM subjective analysis and both the meditation and attention levels are intertwined to some degree. The overall results infer that the Dominance emotion is slightly more intensely experienced by the participants in the study. The neural network-based analysis utilized in the present setup supports this theory, however for a higher accuracy analysis, a larger pool of subjects may be required. Nevertheless, our results prove that emotion recognition can be objectively quantified using BCI's data. Moreover, a hybrid approach where EEG data is combined with facial expression or other form of emotion detection may significantly improve prediction accuracy. The data process is implemented by two investigative pathways, both utilizing neural networks as an engine for

data analysis and considering the collected brainwave data as inputs. Both which are trained to recognize patterns and classify emotions based on the extracted features (Fig. 3).

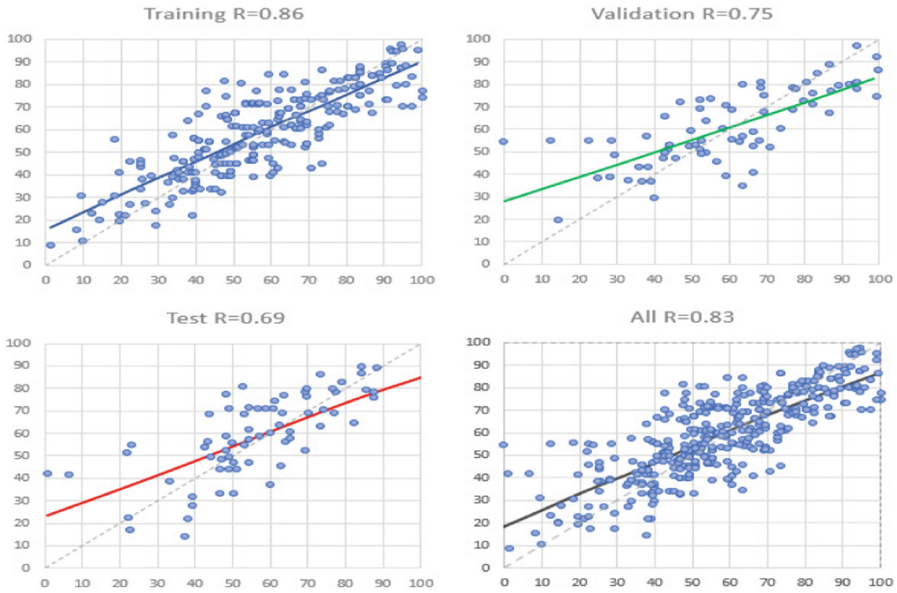


Fig. 3. EEG Data Regression Analysis using Matlab Neural Networks

5 Conclusion

Emotion is commonly associated with logical decision-making, perception, human interaction, and to a certain extent, human intelligence itself. With the growing interest of the research community towards establishing some meaningful “emotional” interactions between humans and computers, the need for reliable and deployable solutions for the identification of human emotional states is required. The purpose of this study was to investigate the use of BCIs to quantify users’ emotion during their interaction with multimedia content that is specific to the Metaverse. The Metaverse represents an all-encompassing concept that integrates digital experiences with social presence and can be deployed as a real time data provider if utilized in conjunction with sensor filled devices with online connectivity. The present paper aims to identify appropriate creative compositions between multimedia systems, and the Metaverse while integrating the social, civilizational key aspect of emotional response and awareness.

References

1. Longo, B., Castillo, J., Bastos, T.: Brain-computer interface combined with virtual reality environment (VRE) for inferior limbs rehabilitation in post-stroke subjects. In: SBBIOTEC, BMC 2014, vol. 8 (2014). <https://doi.org/10.1186/1753-6561-8-S4-P18>
2. Statthaler, K., Schwarz, A., Steyrl, D.: Cybathlon experiences of the Graz BCI racing team Mirage91 in the brain-computer interface discipline. *J. Neuro Eng. Rehabil.* **14**, 1–16 (2017). <https://doi.org/10.1186/s12984-017-0344-9>
3. Torres, E., Hernández-Álvarez, M., Yoo, S.G.: EEG-based BCI emotion recognition: a survey. *Sensors* **20**, 5083 (2020). <https://doi.org/10.3390/s20185083>
4. NeuroSky Homepage. <https://neurosky.com/2016/11/bci-what-is-it-and-where-is-it-going/>. Accessed 3 Nov 2016
5. Yan, W., Liu, X., Shan, B., Zhang, X., Pu, Y.: Research on the emotions based on brain-computer technology: a bibliometric analysis and research agendas. *Front. Psychol.* **12**, 771591 (2021). <https://doi.org/10.3389/fpsyg.2021.771591>
6. UWA. <https://online.uwa.edu/news/emotional-psychology/>. Accessed 22 June 2020
7. Liu, Y., Zhang, Y., Tao, D.: Wearable sensors for emotion recognition: a comprehensive review. *IEEE Trans. Affect. Comput.* **10** (2019)
8. Ekman, P.: An argument for basic emotions. *Cogn. Emot.* **6**, 169–200 (1992). <https://doi.org/10.1038/s41598-021-85163-z>
9. Özerdem, M.S., Polat, H.: Emotion recognition based on EEG features in movie clips with channel selection. *Brain Inform.* **4**(4), 241–252 (2017). <https://doi.org/10.1007/s40708-017-0069-3>
10. Russell, J.A.: Core affect and the psychological construction of emotion. *Psychol. Rev.* **110**, 145 (2003). <https://doi.org/10.1037/0033-295X.110.1.145>
11. Baumeister, R.F., Stillwell, A.M., Heatherton, T.F.: Guilt: an interpersonal approach. *Psychol. Bull.* **115**, 243 (1994). <https://doi.org/10.1037/0033-2909.115.2.243>
12. Llinares, C., et al.: Affective computing in virtual reality: emotion recognition from brain and heartbeat dynamics using wearable sensors. *Sci. Rep.* **8**, 13657 (2018). <https://doi.org/10.1038/s41598-018-32063-4>
13. Schubring, D., et al.: Virtual reality potentiates emotion and task effects of alpha/beta brain oscillations. *Brain Sci.* **10**, 537 (2020). <https://doi.org/10.3390/brainsci10080537>
14. Hongyu, G., Wurong, G.: Metaverse-powered experiential situational english-teaching design: an emotion-based analysis method. *Front. Psychol.* **13**, 859159 (2022). <https://doi.org/10.3389/fpsyg.2022.859159>
15. Perez-Valero, E., et al.: Quantitative assessment of stress through EEG during a virtual reality stress-relax session. *Front Comput. Neurosci.* **15**, 684423 (2021). <https://doi.org/10.3389/fncom.2021.684423>
16. Souza, R., et al.: Attention detection in virtual environments using EEG signals: a scoping review. *Front. Physiol.* **12**, 727840 (2021). <https://doi.org/10.3389/fphys.2021.727840>
17. Lang, P.J., et al.: Technical manual and affective ratings of the International Affective Picture System (IAPS), NIMH Center for the Study of Emotion and Attention (1997)

18. Yan, J., Chen, S., Deng, S.: An EEG-based emotion recognition model with rhythm and time characteristics. *Brain Inf.* **6**, 1–8 (2019). <https://doi.org/10.1186/s40708-019-0100-y>
19. Suhaimi, N.S., Mountstephens, J., Teo, J.: EEG-based emotion recognition: a state-of-the-art review of current trends and opportunities. *CIN* **2020** (2020). <https://doi.org/10.1155/2020/8875426>



MetaHap: A Low Cost Haptic Glove for Metaverse

S. Sibi Chakkaravarthy¹(✉), Marvel M. John¹, Meenalosini Vimal Cruz²,
R. Arun Kumar¹, S. Anitha¹, and S. Karthikeyan³

¹ Centre of Excellence, Artificial Intelligence and Robotics (AIR) and School of Computer Science and Engineering, VIT-AP University, Andhra Pradesh, India
chakkaravarthy.sibi@vitap.ac.in

² Allen E. Paulson College of Engineering and Computing, Georgia Southern University, Statesboro, USA

³ Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, India

Abstract. This paper presents the design and development of a low cost haptic glove equipped with a range of affordable sensors, including gyroscopes, accelerometers, GPS, servos, and encoders, for use in meta-verse environments. The primary focus of this research is to create an immersive and interactive Virtual Reality (VR) experience by incorporating haptic feedback into the glove. This research provides a detailed overview of the glove's design and construction, highlighting the integration of Arduino micro-controllers with the various sensors and actuators. The techniques employed to ensure accurate and synchronized data capture are also discussed. Furthermore, the haptic feedback system integrated into the glove is thoroughly explained, including the mechanisms for generating the haptic feedback, which will allow the user to determine the shape and size of the objects in the virtual environment. By utilizing the servos and encoders, the glove can provide users with a tactile experience by simulating the sensation of touching virtual objects or environments. The potential applications of the haptic glove in gaming, virtual training, and medical simulations are explored, emphasizing the benefits of incorporating haptic feedback for enhanced user immersion and engagement. The glove's versatility and affordability make it a viable solution for a wide range of VR applications. In conclusion, this research presents an Small Board Computer (SBC) - based haptic glove that combines affordable sensors with haptic feedback capabilities, providing users with an immersive and tactile VR experience. The findings contribute to the advancement of VR technology, particularly in the field of haptic interfaces, and open avenues for further exploration and customization of haptic glove applications.

Keywords: Virtual Reality · Metaverse · Reality portal · Haptic Glove · Glove

1 Introduction

Virtual reality (VR) technology has come a long way since its inception [1]. It is now being used in a wide range of applications, from gaming and entertainment to education, healthcare, and industrial training [2]. One of the main benefits of VR is its ability to provide an immersive and interactive experience to users, allowing them to feel as if they are part of the digital environment. However, this sense of immersion can be limited by the lack of tactile feedback in most VR systems. This is where haptic gloves come in [3].

Haptic gloves are a type of wearable device that use sensors and actuators to simulate the sense of touch, allowing users to feel virtual objects and textures as if they were real. These gloves are equipped with small motors or other types of actuators that apply pressure or vibrations to the user's fingers, creating the illusion of touch [4]. This technology can enhance the immersion of VR experiences and make them more realistic and interactive [5].

The development of VR haptic gloves has been an active area of research for several years. There have been numerous prototypes developed, each with its own unique design and set of features [6]. The early versions of haptic gloves were bulky, expensive, and not very practical for most applications. However, recent advancements in technology have led to the development of more compact and affordable gloves, making them more accessible to a wider range of users [7].

In recent years, there has been growing interest in the potential applications of haptic gloves in various fields, such as gaming, music, education, and healthcare. In gaming, haptic gloves can be used to provide a more immersive and realistic experience by allowing users to feel the impact of virtual objects and interactions. In music, haptic gloves can simulate the experience of playing different types of instruments, making it easier for users to learn and practice. In healthcare, haptic gloves can be used to simulate medical procedures and training scenarios, allowing medical students to gain hands-on experience in a safe and controlled environment. Overall, haptic gloves have the potential to enhance the immersion and realism of VR experiences, making them more engaging and interactive for users. The development of haptic gloves is an exciting area of research that has the potential to revolutionize the way we experience and interact with digital content [8–10].

1.1 How is Our Model Used in MetaHap Differs from the Existing Ones

Its an easy to design and implement model that provides the user accurate and reliable experience of the virtual world. This model will let the user experience the virtual world, in the same way they experience the real world with their own hands. This model has been implemented with efficient and accurate position tracking algorithms along with finger tracking and haptic feedback algorithms, that will allow the user to freely move their hands around in the virtual world. However this model will not be as advanced as the ones available out their as this is being implemented on a single processor system, but this model will be

able to replicate the capabilities that are capable by an advanced VR haptic glove, and will also serve as a base model for further development in SBC based VR haptics.

Motivation and Applicability. The main goal of this research is to develop a VR haptic glove that revolutionizes the virtual reality experience in MetaVerse and other reality platforms. We envision a glove that can simulate a wide range of sensations and scenarios, allowing users to feel the intricate details of their virtual environment. To achieve this, our research focuses on designing and developing a glove that operates on simple and affordable microcontrollers, leveraging inexpensive and easily accessible sensors.

The primary aim of our research is to overcome the barriers of cost and complexity associated with existing VR haptic gloves, while still delivering a realistic and immersive user experience. To achieve a high level of realism and fidelity, our VR haptic glove incorporates advanced algorithms and feedback mechanisms. Sensors such as encoders, accelerometers, gyroscopes, and magnetometers are utilized to precisely measure the hand's position, orientation, movement, and force. Actuators like servos are employed to provide haptic feedback, simulating the sensations felt in the hand's muscles, tendons, and joints.

The glove seamlessly communicates with a VR headset or computer through Bluetooth or USB, enabling real-time data transmission for an immersive experience [11]. In summary, our research aims to develop a cost-effective and accessible VR haptic glove with a myriad of applications. By leveraging innovative technologies and algorithms, we strive to provide users with a highly realistic and immersive virtual reality experience, paving the way for new possibilities in gaming, education, entertainment, and healthcare.

Contributions in this Paper

- The proposed system implements on board position trackers, allowing it to achieve non optical position tracking for the fingers and the hand, making the system highly portable, and easy to setup.
- The proposed system has the capability to be integrated with any existing virtual reality environments or games that are compatible with off-the-shelf VR gloves.
- The complete system is built with economically viable, off-the-shelf sensors, leading to a substantial cost reduction of the final product when juxtaposed with comparable market offerings.

2 Literature Review

Optical methods for position tracking provide diverse and effective approaches for accurately determining the position and movement of objects. Camera-based tracking utilizes computer vision techniques and image analysis to track objects based on visual data captured by cameras. Marker-based tracking relies on

distinctive markers or fiducial markers attached to objects to enable precise identification and tracking. Laser-based tracking systems utilize lasers to measure distances and angles, providing accurate position information. Structured light methods project patterns onto objects and analyze their deformations or reflections to calculate object position and shape. Infrared tracking uses infrared light and sensors to track objects based on the detection of emitted or reflected infrared signals.

However these methods have some major limitations and they're, requirement of a clear line of sight between the sensors and the tracked objects, which hinders their effectiveness in obstructed or occluded environments. Optical tracking systems are also sensitive to variations in lighting conditions, making them susceptible to inaccuracies caused by changes in ambient lighting or shadows. Another drawback is the reliance on markers or fiducial markers, which can be time-consuming to attach or place on objects. Additionally, optical tracking systems have a limited effective range, with accuracy and reliability decreasing as the distance between the sensors and objects increases. Furthermore, some optical tracking methods, such as camera-based tracking, can be computationally demanding due to the need for image processing, feature extraction, and object tracking algorithms. This computational requirement may pose challenges for resource-limited devices or real-time applications [8, 12]. Hence there is a pressing need for a non-optical tracker based solution.

To address the above mentioned issues, this paper presents a system which utilizes the non-optical tracker. Firstly, the proposed system eliminates the requirement for a clear line of sight, making it highly suitable for tracking objects in obstructed or occluded environments. This allows for accurate tracking even when objects are hidden from view. Secondly, non-optical trackers are often less sensitive to changes in lighting conditions, making them reliable in diverse lighting environments without compromising accuracy. They can operate effectively in low-light conditions or environments with dynamic lighting changes. Additionally, non-optical tracking systems can offer extended range capabilities, allowing for tracking over larger distances. This makes them ideal for applications that require coverage across expansive areas. Moreover, non-optical methods are typically less affected by environmental factors such as dust, smoke, or reflective surfaces, providing robust tracking performance in challenging conditions. Lastly, non-optical trackers may require lower computational power compared to optical trackers, making them suitable for resource-constrained devices or real-time applications. These advantages make non-optical trackers a compelling choice for scenarios where line of sight, lighting conditions, range, and environmental factors present challenges to optical tracking methods.

3 MetaHap - A VR Based Haptic Glove

MetaHap, a VR-based haptic glove, has been developed to redefine the interaction with virtual environments. MetaHap is equipped with strategically placed sensors and actuators, the glove accurately simulates touch sensations and provides precise feedback to the user's fingertips.

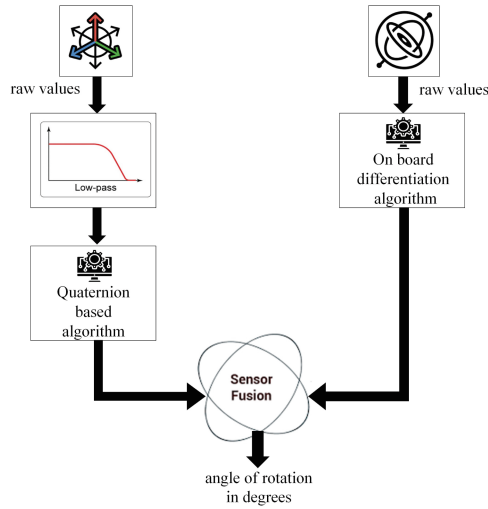


Fig. 1. Workflow of the proposed MetaHap

The MetaHap model (Fig. 1 showcases the utilization of math-based algorithms with high efficiency and reliability. These algorithms enable the system to seamlessly integrate, differentiate, and conduct quaternion-based calculations, thereby facilitating real-time motion tracking. Additionally, the MetaHap incorporates diverse map functions that enable simultaneous tracking of finger poses and provision of haptic feedback. This comprehensive approach ensures a cohesive and immersive user experience within the system, enhancing its overall functionality and versatility. Analysis of each component and the algorithms associated with them are Position Tracking Unit, Finger Pose Tracking and Haptic Feedback.

3.1 Position Tracking Unit (PTU)

Figure 2 shows the workflow of PTU algorithm. The position tracking unit consists of three sensors Accelerometer, Gyroscope and GPS. To track the yaw, roll, and pitch angles, a fusion of accelerometer and gyroscope sensors is employed [13]. This fusion is achieved through a meticulously designed on-board algorithm based on quaternions. Additionally, on-board differentiation algorithms are utilized to extract the desired values. These calculated angles are then merged and filtered using a combination of filters, ensuring the output attains a high level of reliability and stability. By employing these techniques, the system achieves accurate and consistent tracking of orientation, enhancing the overall performance and user experience.

The combined utilization of accelerometer and GPS sensors enables both absolute and relative position tracking of the glove within a three-dimensional space. The accelerometer readings undergo processing through an on-board inte-

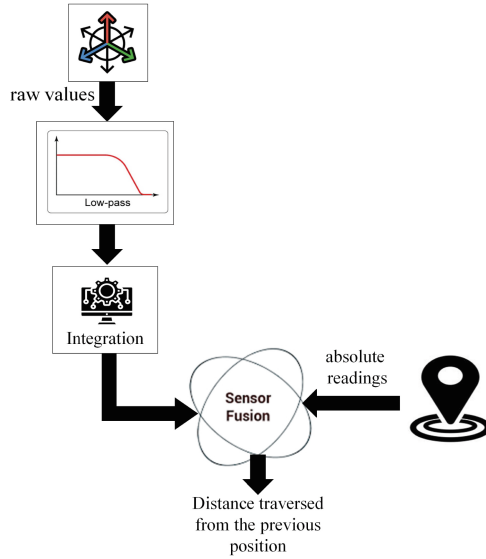


Fig. 2. Workflow of position tracking algorithm

gration algorithm. Subsequently, these processed readings are merged with absolute position data obtained from the GPS module, employing sensor fusion algorithms. The resulting merged data is then subjected to filtering techniques, enhancing the accuracy and reliability of the readings. This approach empowers the system to accurately calculate the real-time position of the glove in three-dimensional space, relative to an absolute reference point.

Finger Pose Tracking (FPT). FPT is accomplished through the utilization of rotary encoders (See Fig. 4). The system employs a total of five rotary encoders, with each encoder connected to an individual finger. These encoders monitor the rotational direction of the fingers, generating a high signal whenever they rotate by 1°. A counter is employed to keep track of the number of high and low signals produced by each encoder. By associating each finger with a rotary encoder, the system is able to calculate the angle of rotation of the lowest finger joint (i.e., the angle of the proximal phalanx relative to the metacarpal). Figure 3 shows the workflow of FPT algorithm. This angle is crucial in determining the finger pose. Subsequently, a forward kinematics model is implemented directly within the virtual environment to predict the finger pose accurately.

$$Current_finger_pose = -0.5 * counter_value \tag{1}$$

$$Here, 0 \leq Current_finger_pose \leq 90, 0 \leq counter_value \leq 180 \tag{2}$$

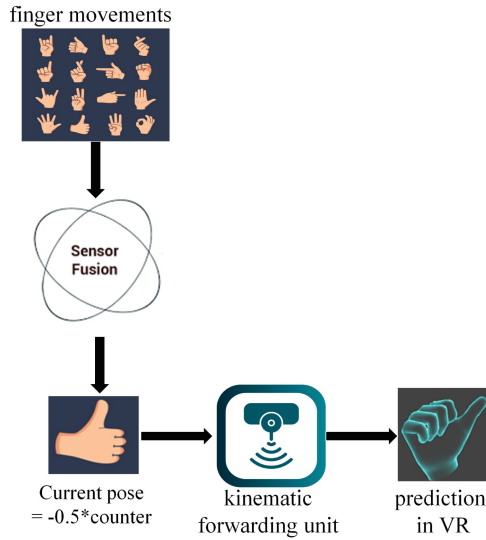


Fig. 3. Workflow of finger tracking algorithm

Haptic Feedback. The achievement of collision detection and prevention is facilitated through the use of servos. Each servo is directly mounted over its corresponding encoder. When a collision is detected in the virtual environment on any of the fingers, the computer sends a stop signal to the SBC. Upon receiving this stop signal, the servo associated with the affected finger acts accordingly. For instance, if a collision is detected on the index finger at a finger pose angle of 40°C, a stop signal is generated specifically for the index finger. Consequently, the servo attached to the index finger rotates in the opposite direction of the encoder’s rotation by an angle calculated as follows:

$$servoAngle = Max_{Fingerpose} - current_{fingerpose} \tag{3}$$

In our current scenario, we are using a maximum finger pose of 90°, this means that the servo will rotate 50° in the opposite direction of the encoder’s rotation, effectively locking the encoder’s position at 40°.

This mechanism ensures that collisions are detected and promptly addressed, preventing any unintended movements or incorrect finger poses. By dynamically adjusting the servo’s position based on the detected collision, the system enhances the overall safety and accuracy of the glove’s movements in the virtual environment.

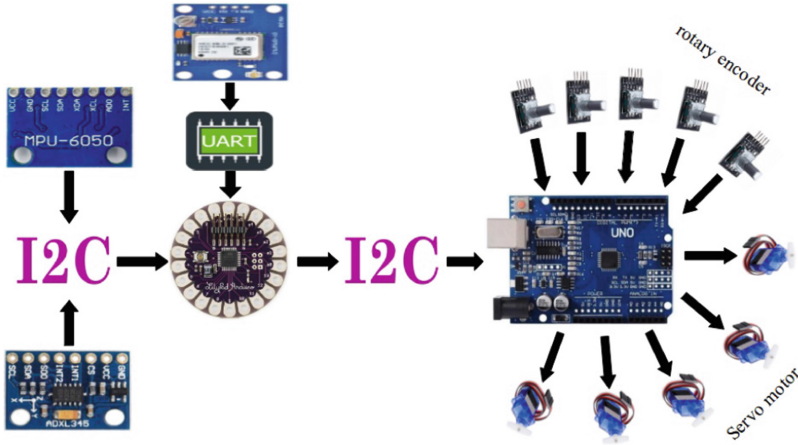


Fig. 4. System design of the proposed VR based haptic glove (MetaHap)

4 Overall System Design

The overall design of the model comprises two distinct components: the primary system and the secondary system. This design allows for flexibility and extensibility, enabling users or developers to modify or enhance the model with ease. If desired, additional features can be incorporated by simply programming and attaching a separate secondary system to the primary system via the I2C protocol. Alternatively, lightweight programs or features can be directly added to the primary system.

For example, consider the model described in Fig. 4, which encompasses finger pose tracking and haptic feedback. If a developer wishes to introduce pressure sensing for each finger to determine the user's grip strength on virtual objects, they can implement this functionality using an additional secondary system. This secondary system would be connected to the appropriate sensors and subsequently linked to the primary system via the I2C protocol. The primary system would then collect data from this new secondary system and transmit it to the virtual environment to generate the desired output, such as causing an object to break if held too tightly. Also notice in the provided diagram, the position tracking unit has been implemented on a separate secondary system to accommodate heavy algorithms that may be impractical to execute on a single system. This modular approach enhances the model's scalability and adaptability to evolving requirements.

5 Experimental Setup and Results

In this section, the development of the VR haptic glove using SBC and other various sensors, such as gyroscopes, accelerometers, GPS, servos, and encoders, has

been successfully achieved. The glove incorporates haptic feedback, finger pose tracking, and motion tracking capabilities, providing users with a highly immersive and realistic virtual reality experience [6]. Figure 5 shows the experimental hardware setup of MetaHap.

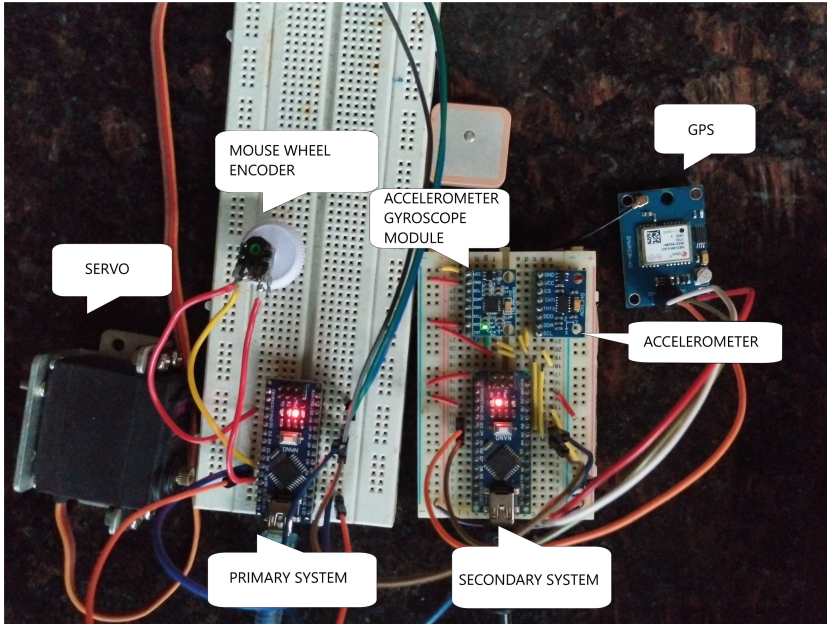


Fig. 5. Basic hardware setup

The model's finger pose tracking mechanism, utilizing rotary encoders and advanced algorithms, accurately calculates the angles of finger joints, enabling precise tracking and realistic interactions within the virtual environment. The integration of accelerometer and gyroscope sensors allows for accurate motion tracking, enhancing the user's immersion and control. The fusion of accelerometer and GPS sensors enables absolute and relative position tracking in three-dimensional space, providing users with an enhanced sense of spatial presence. The implementation of collision detection and prevention using servos ensures the safety and stability of the glove's movements. The servo-based mechanism effectively responds to collision signals, freezing the movement of the affected finger and preventing any unintended actions. This feature enhances the overall user experience and reduces the risk of accidental collisions or improper finger poses.

6 Conclusion

The proposed MetaHap, a VR haptic glove demonstrates promising capabilities in providing a highly immersive and interactive virtual reality experience. The integration of various sensors, along with advanced algorithms, allows for accurate finger pose tracking, motion tracking, and position tracking within the virtual environment. The implemented features, such as collision detection and prevention, ensure the safety and stability of the glove's movements, providing users with a seamless and reliable interaction experience. The modularity of the design allows for easy expansion and customization, enabling users and developers to incorporate additional features or modify the glove according to specific requirements. MetaHap serves as a foundation for further advancements in VR haptics using simple microcontrollers. The use of cost-effective components and the integration of efficient algorithms have the potential to reduce overall costs and increase the number of features in future iterations. With continuous research and development, the VR haptic glove holds a promising development for a wide range of applications, including gaming, virtual musical instrument simulation, medical training, and beyond.

References

1. Helmold, M.: Extended reality (XR) in QM. In: Helmold, M. (ed.) *Virtual and Innovative Quality Management Across the Value Chain: Industry Insights, Case Studies and Best Practices*, pp. 21–26. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30089-9_3
2. Wider, W., Jiang, L., Lin, J., Fauzi, M.A., Li, J., Chan, C.K.: Metaverse chronicles: a bibliometric analysis of its evolving landscape. *Int. J. Hum.-Comput. Interact.* 1–14 (2023)
3. Qi, J., Gao, F., Sun, G., Yeo, J.C., Lim, C.T.: HaptGlove-untethered pneumatic glove for multimode haptic feedback in reality-virtuality continuum. *Adv. Sci.* **10**, 2301044 (2023)
4. Kuhail, M.A., Berengueres, J., Taher, F., Alkuwaiti, M., Khan, S.Z.: Haptic systems: trends and lessons learned for haptics in spacesuits. *Electronics* **12**(8), 1888 (2023)
5. Sinciya, P.O., Orethu, J.A., Philip, M.A., Prakash, N., Jacob, J.: Multipurpose immersive virtual reality environment. In: *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pp. 855–860. IEEE (2023)
6. Civelek, T., Arnulph, F.: Virtual reality learning environment with haptic gloves. In: *2022 3rd International Conference on Education Development and Studies*, pp. 32–36 (2022)
7. Pezent, E., Agarwal, P., Hartcher-O'Brien, J., Colonnese, N., O'Malley, M.K.: Design, control, and psychophysics of tasbi: a force-controlled multimodal haptic bracelet. *IEEE Trans. Rob.* **38**(5), 2962–2978 (2022)
8. Dangxiao, W., Yuan, G., Shiyi, L., Zhang, Y., Weiliang, X., Jing, X.: Haptic display for virtual reality: progress and challenges. *Virtual Real. Intell. Hardw.* **1**(2), 136–162 (2019)

9. Wu, C.M., Hsu, C.W., Lee, T.K., Smith, S.: A virtual reality keyboard with realistic haptic feedback in a fully immersive virtual environment. *Virtual Reality* **21**, 19–29 (2017)
10. Perret, J., Vander Poorten, E.: Touching virtual reality: a review of haptic gloves. In: *ACTUATOR 2018; 16th International Conference on New Actuators*, pp. 1–5. VDE (2018)
11. Edwards, B.I., Bielawski, K.S., Prada, R., Cheok, A.D.: Haptic virtual reality and immersive learning for enhanced organic chemistry instruction. *Virtual Reality* **23**, 363–373 (2019)
12. Antonov, V.O., Arustamov, D.A., Zavalokina, U.V., Apurin, A.A.: A method for controlling groups of unmanned aerial vehicles in a virtual environment using haptic gloves. In: *IOP Conference Series: Materials Science and Engineering*, vol. 1069, no. 1, p. 012043. IOP Publishing (2021)
13. Li, F., Chen, J., Ye, G., Dong, S., Gao, Z., Zhou, Y.: Soft robotic glove with sensing and force feedback for rehabilitation in virtual reality. *Biomimetics* **8**(1), 83 (2023)

Technologies for Smart Agriculture (TSA)



CroPAiD: Protection of Information in Agriculture Cyber-Physical Systems Using Distributed Storage and Ledger

Sukrutha L. T. Vangipuram¹ , Saraju P. Mohanty¹ , and Elias Kougianos² 

¹ Department of Computer Science and Engineering, University of North Texas, Denton, USA
{1t0264, saraju.mohanty}@unt.edu

² Department of Electrical Engineering, University of North Texas, Denton, USA
elias.kougianos@unt.edu

Abstract. The agricultural domain has had a significant role throughout history in human societies across the globe. With the fast growth of communication and information systems, the structure of farming procedures has evolved to new modern standards. Although multiple features helped gain from these advancements, there are many current and rising threats to security in the agricultural domain. The present paper gives novel methods and architectural designs and implements distributed ledger through the Tangle platform. Initially, the article discusses the threats and vulnerabilities faced in the farming sector and presents an extensive literature survey, and later conducts an experiment for distributing data through a tangle distributed ledger system. The authors highlight the limitations of central, cloud, and blockchain and suggest mitigation measures through distributed IOTA systems and distributed storage facilities for data and the possible influence these solutions can bring in the aspects of data security in the agricultural sector.

Keywords: Smart Agriculture · Precision Agriculture · Precision Farming · Agriculture Cyber-Physical Systems (A-CPS) · Internet-of-Agro-Things (IoAT) · Cybersecurity · Blockchain · Distributed Ledger Technology (DLT) · IOTA Tangle · Distributed Storage · InterPlanetary File System (IPFS)

1 Introduction

Smart agriculture is designed as Agriculture Cyber-Physical Systems (A-CPS) using Internet-of-Agro-Things (IoAT). IoAT collect data from multiple sensors installed on farming fields for data analysis and decision-making. The sensors and communication devices record the statistics and understand the machine-to-machine and machine-to-human interactions. With the Internet of Things, the model of agriculture has shifted to precision agriculture on farming fields, including in areas of planting, feeding, lessening water and fertilizer use, and supporting intelligent systems with reduced energy consumption [1]. The Fig. 1 illustrates how modern equipment and IoAT are helping to collect critical data from the fields for agricultural research and science institutes and farmers. These IoAT things require additional features such as real-time data streaming and end-to-end data security. From the agricultural point of view in data security,

© IFIP International Federation for Information Processing 2023

Published by Springer Nature Switzerland AG 2023

D. Puthal et al. (Eds.): IFIP IoT 2023, IFIP AICT 683, pp. 375–394, 2023.

https://doi.org/10.1007/978-3-031-45878-1_26

there are two main concerns, including the collection and storage of farming data. As there is a variety of data residing in fields, conventional monitoring methods cannot be applied, and also the data can be exposed to human errors. With the increase in information and communications systems, the number of cyber crimes have also increased worldwide, stealing and harming a variety of assets. Existing cloud systems have multiple limitations of central storage, continuous connectivity, security risks through different providers, bandwidth constraints, and faint backup procedures. The cyber security procedures combine different techniques to give high protection against attacks on data [2]. Blockchain (BC) is a Distributed Ledger Technology (DLT), a new way to share data securely through cryptographic hashes in a distributed platform that is being applied in various domains nowadays. The agricultural field is also taking advantage of the decentralized features of blockchain, but these systems are going through complex steps to generate blocks and consume energy with on-chain storage [3]. To face issues in blockchain, the applications are designed with off-chain storage solutions and exercise other methods to avoid cost, latency, and energy consumption.

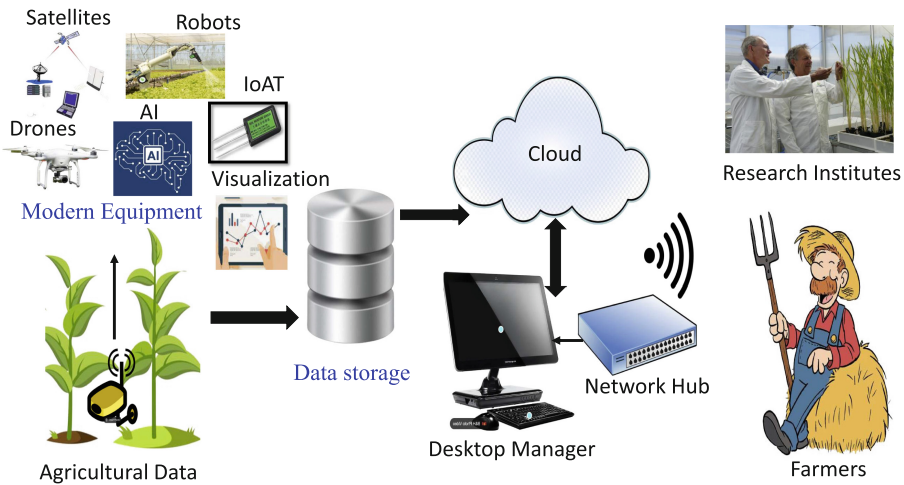


Fig. 1. Data collected from modern equipment's in agriculture.

Tangle is another Distributed Ledger Technology (DLT) that does not require fees and validates the transaction nodes at full speed. The transactions are said to be valid if the previous two transaction's history does not conflict with the current transaction. The consensus mechanism, proof of Work (PoW), is not used for validating the transaction but for keeping the network secure from spam. The overall throughput of the tangle is infinite, and the consensus mechanism is used for defining the limits of the throughput. A distributed storage (IPFS) system is a platform for storing data, websites, applications, and accessing files [4] and does everything a central system does but without a central storage system. Some of the motivations for implementing current paper CroPAiD are listed in the Fig. 2. Combining both tangle and IPFS can bring more security and privacy to sensitive agricultural data. The information stored on the IPFS network generates a cryptographic hash through a content identifier to retrieve the data later securely.

Motivations				
Central Limitations <ul style="list-style-type: none"> • Single point failure. • Security Breaches. • Data Confidentiality Issues. • Unresponsive for massive real-time data. • Increase in costs • Bottlenecks in data access. 	Cloud Drawbacks <ul style="list-style-type: none"> • Data loss or theft. • Insecure Interfaces. • Denial of service attack. • Data Leakage. • Vulnerabilities through different technologies. 	Cybersecurity Issues <ul style="list-style-type: none"> • Malware attacks. • Phishing Attacks. • Ransomware Attacks. • Internet anonymity. • Attack on middleware, network & application layers. 	Sensor Problems <ul style="list-style-type: none"> • Communication problems. • Security Breaches. • Use of Different Technologies. • Challenges in Storage. 	Blockchain Disadvantages <ul style="list-style-type: none"> • Scalability • Storage Issues. • Security. • Privacy. • Cost. • Energy consumed. • Private-key visibility during wallet creations.

Fig. 2. Motivations for the CroPAiD.

The paper follows the given order: We discuss various prior works using conventional and modern methods for transmitting and storing sensitive agricultural data in Sect. 2. The Sect. 3 elaborates on problems raising through modern and traditional methods and list the novel solutions provided through current system. In Sect. 4 and Sect. 5, we define various components, provide a state of the architecture for CroPAiD and give algorithm steps for navigating the data over IPFS and Tangle, respectively. The implementation of the system and the results obtained are shown in Sect. 6 followed by the conclusions and further research aspects in Sect. 7.

2 Related Works

For agricultural data storage, usually, the data is collected in conventional local databases or cloud systems. In the current agricultural 4.0 era, many researchers and scholars are conducting profound studies on how modern data storage methods can be introduced. Launching ledger technology into IoT for data security can be done in two scenarios; the first is making use of off-chain data storage with the help of distributed storage-IPFS or traditional local databases. The second is the direct storage of data on distributed ledger systems.

The paper [5] G-DaM sends the data collected from the Internet of Things to the near edges for storing the data in distributed platforms and public blockchain technology. The application overcomes the traditional data sharing and limitations of central and cloud systems and increases the quality and integrity of the data. The agroString [6] proposed an intelligent IoT-based edge system for the management of data through a private corDapp application. The system sends the information collected from the IoT edge sensors through the private blockchain to avoid traditional public blockchain systems' costs and energy consumption and evade bottlenecks of central and cloud systems. The application implements an IoAT-edge for collecting temperature and humidity datasets and sends those readings to the corDapp to bring integrity, trust, visibility, and data quality to each supply chain stakeholder.

The paper [7] uses blockchain for fruit and vegetable traceability to overcome traditional centralized systems. With the help of a dual storage structure- “database + blockchain” the system designs the application using an on-chain and off-chain storage technique to reduce the load pressures and increase the data integrity throughout the supply chain. The results of the system exhibit improved security of secluded information and further enhance the authenticity and reliability of data. Information modifications and tampering with the sensitive data collected in a supply chain may lead to serious issues regarding the quality and safety of the end product. The article [8] makes use of blockchain for time stamping, traceability, and tamper-proofing of data with the help of smart contracts. The solidity language contract manages the agricultural product transactions with access control and improves the upload and response times.

Crop monitoring is essential to keep a check for pests, weeds, and diseases in the crops. Monitoring is done using different sensors to see the current state of the product to project and predict what will be the next state and issues arising in the crops. The farmer takes preventive measures accordingly based on the information collected in the crop monitoring. Field monitoring plays a vital role in increasing crop yield, and modern IoT technology and communication systems are beneficial in fulfilling this requirement. An efficient crop monitoring system is proposed in sFarm [9] through a sensor to collect the data and share the real-time data securely using IOTA Tangle distributed ledger platform. With the help of IOTA, the central, cloud, public, and private blockchain limitations are overcome, saving energy and time for uploading and validation. Many distributed access control technologies through blockchain are already in practice for dealing with centralized and cloud network limitations, but they, too, inherit some drawbacks, such as high fee transactions and low throughput. The paper [10] proposes a novel access control framework based on IOTA that enables free transactions with higher throughput.

Using Ciphertext-Policy Attribute-Based Encryption (CP-ABE) technology, access rights are encrypted to provide access control and store the data on the distributed ledger Tangle. IOTA Tangle has some disadvantages and security threats, such as a parasite chain attack that is a common double-spending attack. To decrease these types of attacks, the paper [11] gives an efficient method for detecting a parasite chain. The authors measure a score function at each IOTA transaction to see the importance level. Any change in this importance is reflected in the 1st and 2nd order of the derivatives, thus giving accurate results in detecting the parasite chain attack. All the above-discussed prior works try to improve the security in transmitting and storing agricultural data, but

Table 1. Comparing Prior works with Current application CroPAiD.

Application	Storage and Sharing	Cost	Platform	Energy Consumption
G-DaM [5]	IPFS+Public BC	Low	Distributed+Decentralized	High
agroString [6]	Private BC-corDapp	Zero	Decentralized	High
Traceability [7]	Database+BC	Low	Decentralized	High
Traceability [8]	BC	High	Decentralized	High
Crop Monitoring [9]	IOTA Tangle	Zero	Distributed ledger	Low
Access Control [10]	IOTA Tangle	Zero	Distributed ledger	Low
CroPAiD [Current Paper]	IPFS + IOTA Tangle	Zero	Distributed Storage + ledger	Low

the current system adds additional features of distributed storage of IPFS to the IOTA distributed ledger platform to overcome conventional and modern limitations as given in the Table 1.

3 Novel Contributions of the Current Paper

3.1 Modern Communication Technologies in Agriculture

With the increase in demand for food in the global market, the need for reduced costs and increased agricultural production has given way to using new technologies, which is an attractive choice for farmers and companies [12]. Some of the advantages of IoT applications in the agricultural sector include crop health monitoring, pest infestation, water management, frost protection, and decision support. This new method of using novel communication technologies in agriculture is denoted as precision agriculture or precision farming (PF) [13]. The use of satellites and GPS in farming helps in digitizing agricultural measurements to see the accuracy and efficiency of the crop. Based on the measurements collected from these precision agriculture tools, the farmers and the experts in the field, study and analyze the variations of crops and livestock data. To collect different types of information from the fields, the farmer would use IoT nodes that come with specific features that make them useful in limited domains [14].

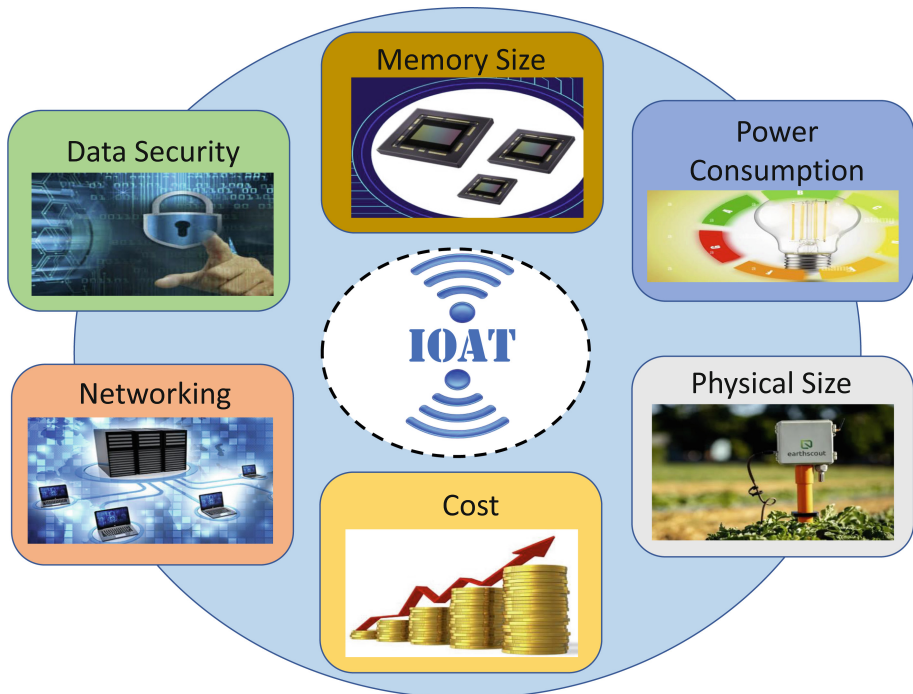


Fig. 3. Challenges in IoAT.

3.2 Data Threats Through Contemporary Systems in Farming

With the emergence of new communication systems and the addition of the Internet of Things (IoT) in farming, unknown security risks and data threats arise in the cyber-physical environment. These data risks are mainly related to cyber security, data integrity, and data loss disturbing the stakeholder businesses [15]. The constraints of the IoAT are given in Fig. 3. Precision Farming uses vast modern machinery in the fields, leading to higher consequences and threats. Farming is possible in open grounds where weather and environmental conditions are inconsistent, leading to malfunctioning of the machinery and technical equipment in the fields, resulting in wrong measurements and hence wrong analysis [16]. Additionally, the temperature and humidity conditions can affect sensor things for communication, which can lead to data loss [17]. The cyber-security issue is a worldwide severe threat activity that uses a smart device to access sensitive personal and government information. Although strict restrictions have been implanted through law enforcement, the hackers take advantage of internet anonymity and attack middleware, network, and application layers [2].

3.3 Novel Solutions Proposed

The novel contributions of the current paper CroPAiD include:

- A unique system is designed with Tangle to increase the quality of data and avoid drawbacks of sensor things.
- To move bulk data to IOTA and avoid double spending issues of Tangle, the current system uses distributed storage systems near the edges.
- The imitations of conventional storage databases, cloud, and central systems are circumvented using the IOTA distributed ledger platform.
- Increasing security, data integrity, and evading data tampering by the IOTA system.
- Overcoming blockchain high transaction fees and energy usage through distributed ledger system of Tangle.
- Using Double hashing procedure for the agricultural data through IPFS and Tangle to increase security and privacy of data.
- A state-of-the-art architecture is presented for the current system CroPAiD.
- Designing a Cost-efficient infrastructure and presenting results with zero transaction fees and secured hashes.

4 Overview of the Proposed Framework - The CroPAiD

4.1 Agriculture Cyber-Physical Systems (A-CPS)

The cyber-physical systems (CPS) combines the software and hardware components to execute a well-defined task. A system that connects and manages the physical attributes towards its computing capabilities and a design that connects and controls the physical organizations with virtual structures through networks. The combination of wireless sensor networks that supervise the physical entities can enhance itself in real-time scenarios. The CPS are applied in multiple domains to help in substituting conventional

methods and integrating various platforms and technologies together [18]. Smart Agriculture is one of the domains that can benefit from CPS due to its modern and smarter applicability in monitoring and controlling farming activities and gathering the information associated with crops, soil, livestock hygiene, and weather in real-time, along with maintaining the environment and preserving energy. Figure 4 gives different layers of cps and their connectivity in physical systems through smart devices to control and manage the data in an intelligent way.

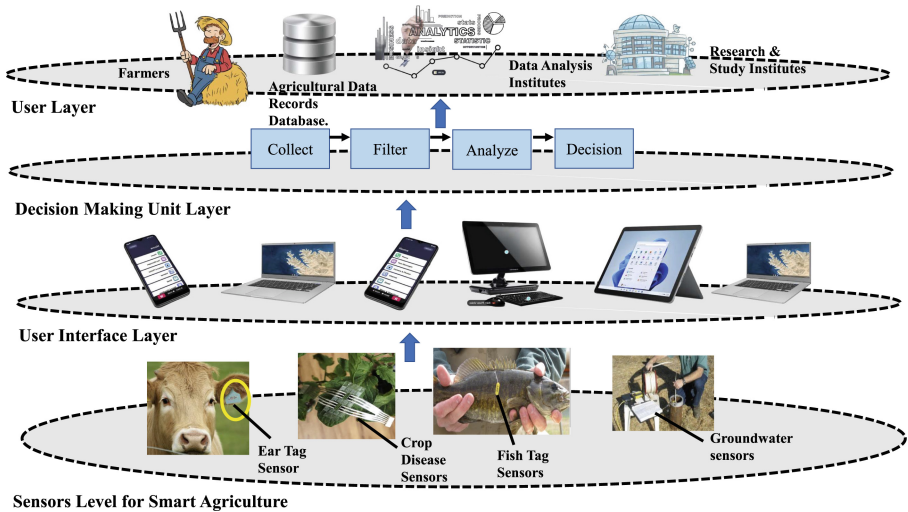


Fig. 4. Agriculture Cyber-Physical Systems (A-CPS).

4.2 Distributed Storage - IPFS

One of the limitations of Tangle is that the attackers can implant several duplicates of the data that can lead to double-spending transactions [19]. A user can create and spend the same digital asset multiple times, which must be checked and prevented. A distributed storage-IPFS or Interplanetary File System is an internet protocol mainly to store data, avoid data or asset duplicates across the network, and collect the addresses of the data in the network. By stopping asset duplication, the IPFS can help in avoiding double spending issues. By using IPFS as off-chain storage for IOTA tangle, the information is stored in a distributed platform, reducing local database, central, and cloud constraints. The data is recognized through content, and every piece of information is divided into 256 kb maximum length blocks. Every block is labeled with a unique identifier for the content through the cryptographic hash. The distributed hash table of IPFS is based on the principle of distributed key-value store. Both node identifiers and distance metrics strategies in IPFS help in storing and retrieving the data quickly. When reading or writing the data from or to the edges, the end devices search for the nodes close to the key attribute values using buckets inside the networks to identify the nodes [4,20]. The

S/Kademlia algorithm is used for DHT in IPFS to register the nodes whenever a file gets uploaded and links to nodes through an identifier for file retrieval.

4.3 Data Security Through IOTA Tangle

The IOTA Tangle has built two-layer solutions called L2 for dealing with the data. The first one is built for MAM, and the second is with STREAMS. The Tangle with MAM has two protocols to traverse and authenticate the data in the distributed ledger network. These protocols mainly work on the principle of cryptography. With the help of Masked Authenticated Messaging (MAM) in the IOTA Tangle network, it allows any device to publish data in the transactions but only read the data of authorized devices. The IOTA introduces the concept of zero-value transactions, here, the first protocol is responsible for transactions, but the data transactions are authenticated. MAM is a second protocol that helps in protecting the data and verifies its authenticity. With MAM, data channels can avoid malicious attempts or fake data because only the owner has the right to publish data into the channel. As data is published into its respective channel, a channel ID is received that acts as the identifier, which allows other devices to connect to retrieve the data. There are three different channel modes: public, private, and restricted. In the public channel mode, the transaction uses the root of the Markle tree as the address; therefore, whichever device gets access to the channel ID can decrypt the data using the address as the decryption key. In private mode, the Markle tree's root is hashed; hence, only those devices with the original root can decrypt the data. Lastly, restricted channels will include both pre-shared keys and the root of the Markle tree. Only devices with information regarding both pre-shared keys and Markle tree roots can decrypt the data. The application for IOTA tangle can be developed using quantum-proof cryptography, and javascript language [21]. The second is the STREAMS [22] tool that helps in structuring and navigating the data securely through Tangle. It is basically a framework to develop applications through secure cryptographic messaging and allows any device to order messages with integrity and immutability. A publisher device takes control of the messages to be sent by everyone else and makes the messages private by using public key encryption. Any device, called a subscriber, can consume and pull and publish the information from the Tangle as opposed to MAM where only a channel owner can publish the data [23].

4.4 Novel Architecture for CroPAiD

An IoT device used to collect data from agricultural fields is referred to as the Internet-of-Agro-Things (IoAT). The IoAT is equipped with all the capabilities of networking (WiFi, LoRa, Bluetooth) to communicate the data through IOTA Tangle. There are several endpoints between the target device and the IOTA gateway. Figure 5 shows the state-of-the-art architecture design for the current system CroPAiD. The edge layer is responsible for fetching the data from the internet things and sharing the agricultural data among servers. The edges interface with higher networking, space, and energy supplies and provide management and monitoring services with multiple sensor nodes and other gateways. The servers store, process, and visualize data moving from edges. The architecture presents an edge layer as a communication medium between sensors and

servers. A distributed storage technology-IPFS is embedded into the edge along IOTA Tangle that serves as a gateway between IoT devices and servers. With distributed storage on edge, the limitations of IOTA, such as double spending and other attacks, are evaded. Each agricultural sensor data file is transmitted to IPFS to generate a hash, as explained in Subsect. 4.2, through its unique content identifier. A Merkle-directed acyclic graph (Merkle-DAG) calculates a root that can retrieve the original file from the segments. The distributed storage hash of the crop’s sensor data is then moved toward the Tangle residing in the edge. The IOTA node receives the hash from the IPFS and further secures it by generating Tangle hashes using MAM and STREAMS tools, as discussed in Subsect. 4.3 above. The distributed ledger technology is feasible for point-point, point-multipoint, and multipoint-multipoint communications between various sensor devices on the field and multiple servers.

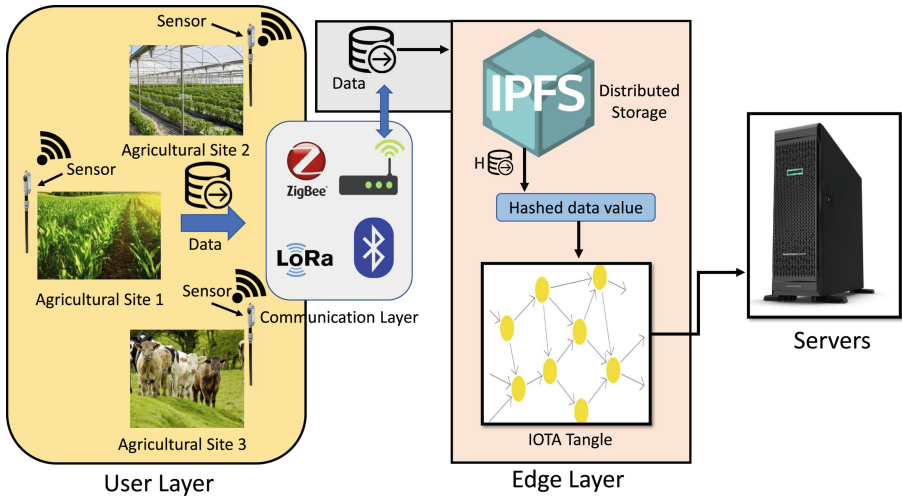


Fig. 5. CroPAiD Novel Architecture with IPFS and IOTA Tangle.

5 Proposed Algorithms for CroPAiD System

The data from the Internet of Things moves toward the edge layer that has both Distributed Storage-IPFS (DS) and IOTA Tangle systems implemented. Algorithm 1 presents phases in transferring crop data(C_d) to IPFS and generating 256 kb buffer files to give a root hash at the end. In the edge distributed storage system (DS_E), both private (DS_{pr}) and public keys (DS_{pu}) are generated to incorporate access control through digital signatures and signing crop data files. The IPFS converts the crop data (C_d) into a 256 kb buffer file (C_d) and signs the buffer file ($C_{dbf256KB}$) to get a root hash file of the crop data ($H(C_dC_{bf})$) where H denotes the hash of the crop data file.

Algorithm 1. Crop Data File to IPFS.

-
- 1: Inside the Edge layer the Distributed storage (DS_E) generate both Public and Private Keys (DS_{pu} , DS_{pr}) for the Crop Data.
 - 2: $DS_E(C_d) \rightarrow DS_E(C_{dbf265KB})$.
 - 3: The file gets hashed through cryptography method using SHA 256/SHA 3 to give unique id represented as C_{Id} (Content Identifiers).
 - 4: $Encr(DS_{pu})S = H(DS_{pr} * A)$, where A is a constant, * is a mathematical operation that is calculated in single direction and H is the secured hash function.
 - 5: **if** C_d is equal $H(DS_{pr} * A)$ is equal $H(DS_E(C_{dbf265KB}))$ **then**
 - 6: Publishing $H(C_{dbf265KB}) \rightarrow IPFS$.
 - 7: **else**
 - 8: Process End.
 - 9: **end if**
 - 10: Repeat the steps from 1 through 10 whenever a file is uploaded in the edge layer.
-

Each input data present in the Tangle creates the following fields: data-length, data, public key, private key, index, index-next, sign, and auth-sign. The IOTA tangle generates a seed (S_d) from a random source and produces a key pair for input data using the edwards25519 curve algorithm. Each input data calculates the index and the index-next via private and public keys. The hash of the public key is the index, and the hash of the public key for the following input data is the index-next. A different key pair is generated for the next input data from another random source, and for hashing, the algorithm used in IOTA is BLAKE2b [24]. Computing index and index-next are significant because they help in continuous data streaming, data ownership, verification, and authentication. A digest 'd' is given by hashing the data, data-length, public key, and index-next. The sign field is then calculated by signing the digest with the private key. This will be helpful in verification later for the user. If the user has to verify the data, compare the hash and sign field values with the public key in the input data. If both are equal, then the data is verified correctly. The sign field helps in only verification of the data but does not give authenticity or the author's identity. The field auth-sign is calculated by the key pair associated with the sensor device. This authorization signature is calculated by the private key of the IoT device and stored in a hardware source along with the public key certificate. To validate and see the authentication of the data, the user compares the signature with the public key through a trusted third-party certificate authority. The algorithm 2 and the Fig. 6 show the flow of Crop data in the Edge layer between IPFS and IOTA Tangle in detail and explains how the data is moved, verified, and authenticated in Tangle.

' $H(C_{dbf265KB})$ ' is taken as the input data to the Tangle ledger System. We represent that input data ' $H(C_{dbf265KB})$ ' in the following algorithm as ' In_{iota} ,' data-length as ' $In_{iota}len$,' the public key as ' $In_{tangle}Pukey$,' private key as ' $In_{tangle}Prkey$,' index as 'I' and next-index as 'n-I,' sign field as 'sign' and auth-sign as ' $auth_{sign}$.' For Hashing and digest, we represent with letters 'H' and 'd'.

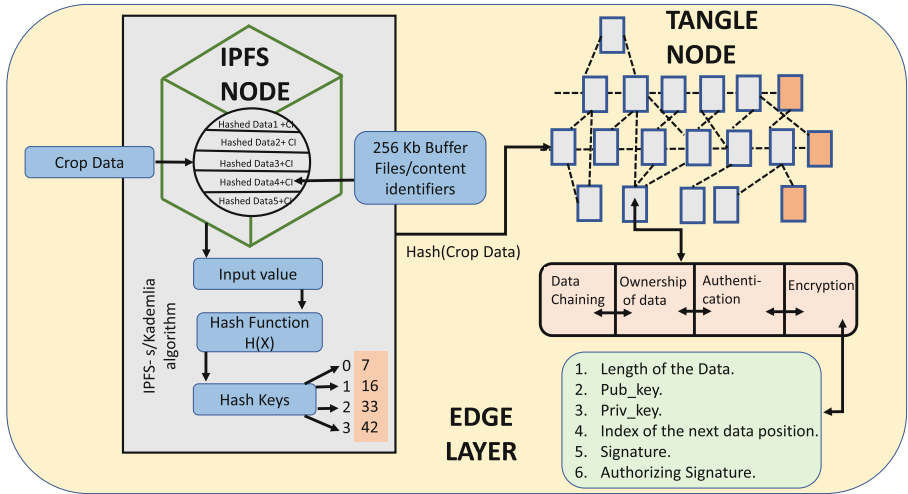


Fig. 6. CroPAiD detailed Data flow in the Edge layer between IPFS and IOTA Tangle.

Algorithm 2. Crop Data File in IOTA Tangle.

- 1: We represent $H(C_d)_{ipfs}$ coming from ipfs as input data to IOTA as $Tangle(In_{iota})$.
- 2: $In_{iota} \rightarrow In_{iota}, In_{iota}len, In_{tangle}Prkey, In_{tangle}Pukey, ind,$
- 3: $nex-ind, sign auth_{sign}$.
- 4: Random Source $\rightarrow S_d$.
- 5: $S_d \rightarrow In_{tangle}Prkey, In_{tangle}Pukey$.
- 6: $H(In_{tangle}Pukey) \rightarrow I$.
- 7: A different key pair is generated for the next input data (Next- In_{iota}) from another random source.
- 8: The key pair from the next input data is (Next- $In_{tangle}Prkey$) and (Next- $In_{tangle}Pukey$).
- 9: $H(Next-In_{tangle}Pukey) \rightarrow n-I$.
- 10: A digest d is calculated for signature.
- 11: $d = H((In_{iota}) + (In_{iota}len) + (In_{tangle}Pukey) + (n-I))$.
- 12: $sign = signature(d + In_{tangle}Prkey)$
- 13: **if** $H(In_{iota}) == sign + In_{tangle}Pukey$ **then**
- 14: Verification Success.
- 15: **else**
- 16: Process End.
- 17: For authorization, we need the public(IoT_{Pukey}) and private keys (IoT_{Prkey}) of the IoT device.
- 18: $auth_{sign} = signature(IoT_{Prkey})$
- 19: **if** $auth_{sign} == signature(IoT_{Pukey})$ **then**
- 20: Authentication Success.
- 21: **else**
- 22: Process End.
- 23: **end if**
- 24: **end if**
- 25: Repeat the steps from 1 through 25 whenever a file is moved from IPFS in the edge layer.

6 Implementation and Validation

To implement the current system, we have taken the source code from Github and modified the code to our needs for the CroPAiD application. The application is designed using javascript; hence we use Node.js as an environment for executing our JS programs. In this application, it is mainly used to create, open, read, write, delete, and close files that reside on the server. Node.js is installed in both the application program interface and client programs to test and deploy the application. We modified and configured the local .json file of the application program interface (api) with the network settings of the node provider, IPFS node, and the database using dynamoDbConnection services provided through Amazon web services. Once the api is configured, the API server starts in the development mode as shown in Fig. 7. For the client mode to execute, we installed the node.js inside the client directory and configured the local.json file with the required fields of API endpoint URL, ipfs gateway URL, and the URL for tangle explorer. After client configuration, the client connects to the API server to open the front-end web browser.

The front end of the application is designed using React javascript. The user interface of the CroPAiD application is given in Fig. 8(a) and the Fig. 8(b) shows the front-end design for uploading the crop data files. A hash is generated once the file gets uploaded to IPFS as shown in the Fig. 9(a) and files can be retrieved from IPFS and IOTA Tangle hashes as shown in Fig. 9(b). Thus, we implement and validate the application and record the DDS and IOTA hash results.



```

[12:04:43 PM] Starting compilation in watch mode...
[nodemon] 2.0.6
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting mode /dist/index.js
Started API Server on port 4000
Running Config 'local'
  
```

Fig. 7. Connecting to api and Client Programs.

6.1 Datasets

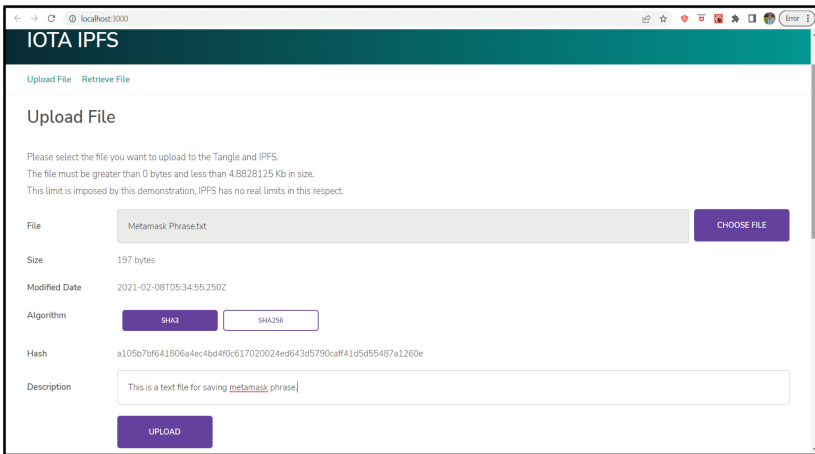
The agricultural datasets we used are from the Kaggle [25] source. Each data belonged to different vegetables and fruits containing images of healthy and diseased crops. These data collected are sensitive and usable for further research and analysis in bringing improvements in farming and are also beneficial in the field of agricultural science. We uploaded the crop data in the current application to test and validate. Table 2 demonstrates different crop statistics we used for the present paper.

The Fig. 10 shows some sample dataset images we used for storing and sharing in the CroPAiD application through IPFS and IOTA. When a crop gets infected, it damages and changes all the primary functions of the food that can harm humans when consumed. This type of crop infected data is beneficial in predicting future crop damage

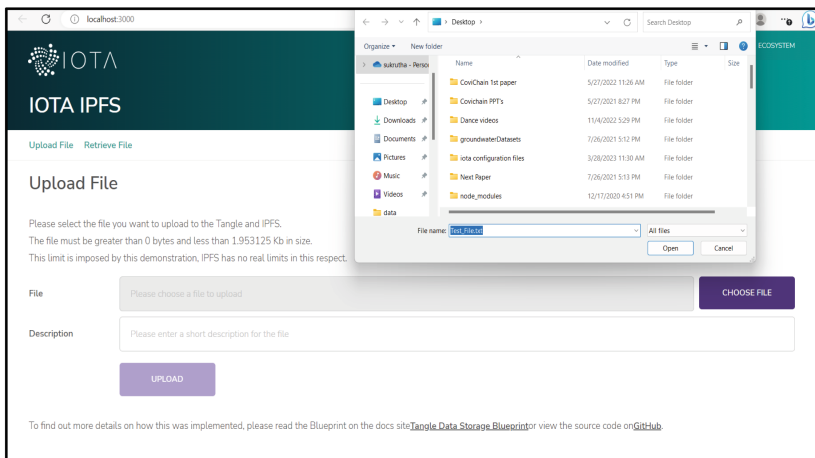
and helps improve crop yield. Therefore, such data is crucial for farmers and scientists to take precautions and perform research and study. This data need to be transmitted in a secure manner without any tampering for correct analysis. The Fig. 10(a), 10(b), 10(c), 10(d), 10(e), 10(f) show pictures of a healthy crop and a diseased crop of apple, potato, cherry, corn, grape and tomato correspondingly.

6.2 Experimental Results

To obtain the results for the current application, we have used Intel(R) Core(TM) i9-10885H CPU @ 2.40 GHz, 32.0 GB RAM as the edge layer. We have deployed the application logic of IPFS and IOTA tangle in this edge system. We upload the crop data file to the IPFS node to get the hash of the file, as shown in the Table 3. The IPFS hash file generated does not have the time stamp but avoids duplicates and double-spending attacks on the data transferred. The application further takes the IPFS hash as an input to the IOTA node to give another hash from the tangle platform. The Table 3 shows the double hashes produced by both IPFS and Tangle. The application has been tested with different sizes of crop data to produce two hashes with both technologies. Once both the hashes were received from the application, we used the message unique ID to retrieve the original file. The time to upload the files was very minimal, and the data transactions costs were zero compared to blockchain latencies and transaction fees. The paper we implement combines distributed storage IPFS and IOTA Tangle successfully, resulting in higher data security with reduced energy consumption nullifying the limitations of central, cloud, conventional database, and blockchain systems.

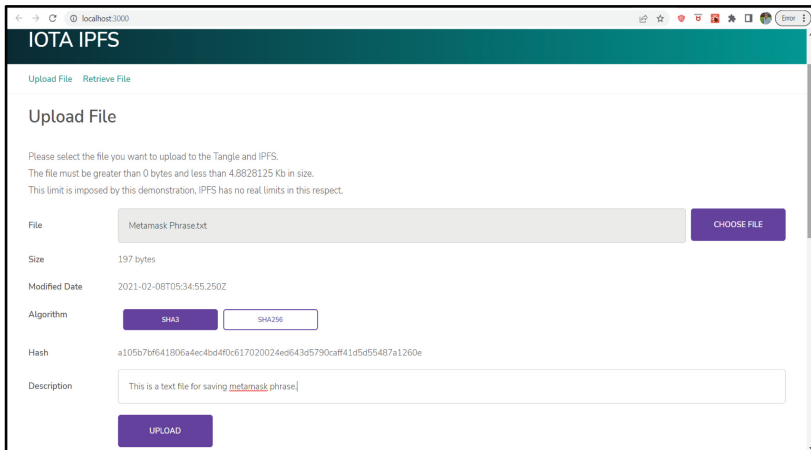


(a) User Interface.



(b) File Uploading

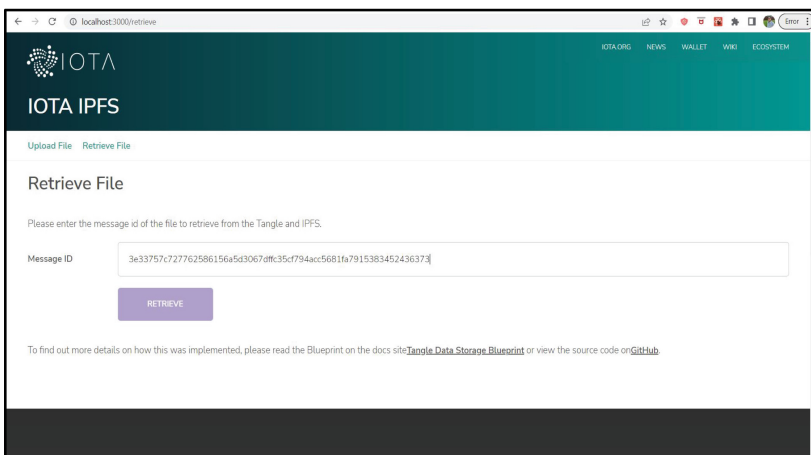
Fig. 8. CroPAiD Application User Interface.



The screenshot shows the 'Upload File' page of the IOTA IPFS application. The page has a dark teal header with the IOTA logo and 'IOTA IPFS' text. Below the header, there are two tabs: 'Upload File' (active) and 'Retrieve File'. The main content area is titled 'Upload File' and contains the following information:

- A message: "Please select the file you want to upload to the Tangle and IPFS. The file must be greater than 0 bytes and less than 4.8828125 Kb in size. This limit is imposed by this demonstration, IPFS has no real limits in this respect."
- A file input field containing 'Metamask Phrase.txt' and a 'CHOOSE FILE' button.
- File details: Size (197 bytes), Modified Date (2021-02-08T05:34:55.250Z).
- Algorithm selection: Two buttons, 'SHA3' (selected) and 'SHA256'.
- Hash: a105b7bf641806a4ec4bd4f0c617020024ed643d5790caff41d5d55487a1260e
- Description: This is a text file for saving metamask phrase
- An 'UPLOAD' button.

(a) Hash of the File.



The screenshot shows the 'Retrieve File' page of the IOTA IPFS application. The page has a dark teal header with the IOTA logo and 'IOTA IPFS' text. Below the header, there are two tabs: 'Upload File' and 'Retrieve File' (active). The main content area is titled 'Retrieve File' and contains the following information:

- A message: "Please enter the message id of the file to retrieve from the Tangle and IPFS."
- A 'Message ID' input field containing the value: 3e33757c727762586156a5d3067affc35c7f94acc5681fa7915383452436373
- A 'RETRIEVE' button.
- A footer note: "To find out more details on how this was implemented, please read the Blueprint on the docs site [Tangle Data Storage Blueprint](#) or view the source code on [GitHub](#)."

(b) File Retrieving

Fig. 9. Implementation of CroPAiD Application.

Table 2. Datasets for CroPAiD

Data Name	Dataset Size	Compressed Data Size	Dataset Link
Apple-healthy	25.7 MB	23.8 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Apple-Cedarapplerust	3.25 MB	2.9 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Cherry-healthy	15.1 MB	14.06 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Cherry-Powderymildew	12.8 MB	11.41 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Corn-healthy	14.9 MB	13.39 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Corn-Commonrust	18.4 MB	16.72 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Grape-healthy	6.87 MB	6.29 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Grape-Esca (Black-Measles)	28.6 MB	27.30 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Peach-healthy	6.16 MB	5.74 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Peach-Bacterialsplot	32.8 MB	29.89 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Potato-healthy	3.17 MB	3.05 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Potato-Lateblight	17.5 MB	16.5 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Tomato-healthy	37.0 MB	35.29 MB	https://www.kaggle.com/datasets/divumarcus/plant-health
Tomato-Bacterialsplot	30.5 MB	27.5 MB	https://www.kaggle.com/datasets/divumarcus/plant-health

Table 3. Hashes generated through IPFS and Tangle

File Name	Reduced Size	IPFS Hash	Tangle Hash	Txn Time (Sec)
Apple-healthy	23.8 MB	QmXWpe6Q5v9qH7Wwgr 5HH5BmB78Q2u4wP WFd7NkvoofZrP	SKJYF76R3947IRYREIU59 8475FHKEUR834759IFKR3fOW PWEKDSVLDKFRoIRFHDKJ	35
Apple-Cedarapplerust	2.9 MB	QmYuERUHbu8fuXRa b7RkWwDqDZKHCn8Dp kUwpopaNMjAB3	GZSDUAYR87R675RWRYGJDH FU9586ERUFJBLDIR43950 35RTHGKVJS579048EOIHK	3.45
Cherry-healthy	14.06 MB	QmPRkovGVUgYx2ue hy1g5QhWqECXpd1No CXAsUehznjU5t	JHSGFUY5R635RWGFJSH VET875985WIGDShVLUSP5T98 FHDVJDOYW8R76487RITHK	21
Cherry-Powdery mildew	11.41 MB	QmTy9g2ENwSP66D V2qkUP7XchCd9AQ maznM8saZbvz1xcY	CMVNGGF653RFHHKJLLOU ERWEQSCCBJH87966453FDJ GHKJUyRTEESXZVFMHKKO	7
Corn-healthy	13.39 MB	QmZkM4ymQCXKThL hY6igBMPxcjwaNa uGj6Khvnr1rfuHNh	LQREWRR5473FCVvNGH67 892DHGNCSK53FH5FFKJOIW RW9345FDGERSBHYUKIOUQW	14
Corn-Commonrust	16.72 MB	QmVCm8uXgyvnQEfvC bDpPxZ95XNuTyS ir7thRMMLfoNzFi	FR5476HYHKHNCVZSA3386 87UYKJNGGFTR544333DEH GJUHPKNMNBFFVDFSEW4YU	12.34
Grape-healthy	6.29 MB	QmX1ohMDQqRqtvdG PYVVGZjyFvX3zuVEK TXxKRmj6VJxc75	MNXBFYO5173RGKLD78 79HSJRY764934UTWJHEUFQJO 7GDAPOLKLKOIUSDWKN- MND4	5.13
Grape-Esca(Black-Measles)	27.30 MB	QmZw4X69QyptuNWj bA.3o6NwAK6x9ve eb3CcXZdPWQV6qcY	D564837HTYCBHGDJDUR7 595HFYE54658THG84658HRI 746595RHF176HJGDTYRIR	37
Peach-healthy	5.74 MB	QmUCANWk22uX6JC Bew8SCRXXDbMfru XyfCj7YJmSjesmYz	MNZCJHARU8473EIDHKSJ FLJG9485029QPWADJSLKFWOR IAJFKZJFKSDJLLKPLSKJ	4.61
Peach-Bacterial spot	29.89 MB	QmdWXdT8LaTHaL wFAPe49FCBd5eii jaM43kMd16yj13S7	BVKJSDYFIWUR23OUOQFH SKDLSEORIQPOWASJCDKFLKI KDFIY98T4OIP4O549TIDH	43.2
Potato-healthy	3.05 MB	QmenHxheRqXnXE57D mL6Ncgv3pTJ9Ed g9KFXW58ei5R6z	XJSTF346TIUWFH7W6457V1 SU6WILQURW87RIF18479WR IUFLSJKAS511OQALSJWP	3.52
Potato-Lateblight	16.5 MB	QmbY6uwYER8WYXbz C8ES9xS6iXumS yK2oy757EgUp2gcxR	U6785GHFVDBXDSEWR5687I JKGNBMCVXDSWQTIUOPIK BMNVVDGTR6E4R7T8987JJ	23.4
Tomato-healthy	35.29 MB	QmTozqarvDLCzaqX rt2895H9jBVPsiFx l2JedBc9Jy4NFA	VDFER4557YHGDDSXZMKN JOU865GGJLDVXVGSAWUWO IWNVHZFQSE7TIUGVJHIFJH	61.3
Tomato-Bacterial spot	27.5 MB	Qmbzvc2Pk4qN9vR l3vuuFhWiDnhWjh TiMEB12PucDZwGP	MBSJAOEUGD7847KI387HOW SKDHGVXMSLE- DUR6E6R9TUWSBxKBO FIF8EE6RWFsBVk	38.6



(a) Apple-healthy and Apple-Cedarapplerust.

(b) Potato-healthy and Potato-Lateblight



(c) Cherry-healthy and Cherry-Powderymildew.

(d) Corn-healthy and Corn-Commonrust.



(e) Grape-healthy and Grape-Esca (Black-Measles).

(f) Tomato-healthy and Tomato-Bacterialspt.

Fig. 10. Sample Images of Crop Condition Dataset.

7 Conclusions and Future Research

The paper suggests a state-of-the-art model that combines distributed storage-IPFS and the IOTA Tangle for managing the quality and integrity of the agricultural crop sensor data. The paper resolves various issues raised by traditional database, cloud, central, and blockchain storage systems, that include data security, privacy, integrity, and overcoming bottlenecks and latencies of conventional platforms. The Tangle uses tools such as MAM and STREAMS for communication and to secure the data received from the distributed storage system. In this paper, we also propose a novel architecture using an edge between the sensor things and the servers. The system can further be improvised with automation for taking in real-time data towards the edge IOTA Tangle systems.

References

1. Farooq, M.S., Riaz, S., Abid, A., Abid, K., Naeem, M.A.: A survey on the role of IoT in agriculture for the implementation of smart farming. *IEEE Access* **7**, 156237–156271 (2019). <https://doi.org/10.1109/ACCESS.2019.2949703>
2. Ivanov, I.: Cyber Security and Cyber Threats: Eagle VS “New Wars”? *Academia.edu* (2018). https://www.academia.edu/38462737/CYBER_SECURITY_AND_CYBER_THREATS_EAGLE_VS_NEW_WARS_
3. Henry, R., Herzberg, A., Kate, A.: Blockchain access privacy: challenges and directions. *IEEE Secur. Priv.* **16**(4), 38–45 (2018). <https://doi.org/10.1109/MSP.2018.3111245>
4. Musharraf, M.: What is InterPlanetary File System (IPFS)? (2021). <https://www.ledger.com/academy/what-is-ipfs>
5. Vangipuram, S.L.T., Mohanty, S.P., Kougianos, E., Ray, C.: G-DaM: a distributed data storage with blockchain framework for management of groundwater quality data. *Sensors* **22**, 8725 (2022). <https://doi.org/10.3390/s22228725>
6. Vangipuram, S.L.T., Mohanty, S.P., Kougianos, E., Ray, C.: agroString: visibility and provenance through a private blockchain platform for agricultural dispense towards consumers. *Sensors* **22**(21), 8227 (2022). <https://doi.org/10.3390/s22218227>, <https://www.mdpi.com/1424-8220/22/21/8227>
7. Yang, X., Li, M., Yu, H., Wang, M., Xu, D., Sun, C.: A trusted blockchain-based traceability system for fruit and vegetable agricultural products. *IEEE Access* **9**, 36282–36293 (2021). <https://doi.org/10.1109/ACCESS.2021.3062845>
8. Yi, W., Huang, X., Yin, H., Dai, S.: Blockchain-based approach to achieve credible traceability of agricultural product transactions. *J. Phys. Conf. Ser.* **1864**(1), 012115 (2021). <https://doi.org/10.1088/1742-6596/1864/1/012115>
9. Bapatla, A.K., Mohanty, S.P., Kougianos, E.: sFarm: a distributed ledger based remote crop monitoring system for smart farming. In: Camarinha-Matos, L.M., Heijenk, G., Katkoori, S., Strous, L. (eds.) *Internet of Things. Technology and Applications. IFIPIoT 2021*, vol. 641. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-96466-5_2
10. Nakanishi, R., Zhang, Y., Sasabe, M., Kasahara, S.: IOTA-based access control framework for the Internet of Things. In: *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pp. 87–95 (2020). <https://doi.org/10.1109/BRAINS49436.2020.9223293>
11. Ghaffaripour, S., Miri, A.: Parasite chain attack detection in the IOTA network. In: *2022 International Wireless Communications and Mobile Computing (IWCMC)* (2022). <https://doi.org/10.1109/IWCMC55113.2022.9824318>
12. Calicioglu, O., Flammini, A., Bracco, S., Bellù, L., Sims, R.: The future challenges of food and agriculture: an integrated analysis of trends and solutions. *Sustainability* **11**, 222 (2019). <https://doi.org/10.3390/su11010222>
13. Wolf, S.A., Wood, S.D.: Precision farming: environmental legitimation, commodification of information, and industrial coordination. *Rural. Sociol.* (1997). <https://doi.org/10.1111/j.1549-0831.1997.tb00650.x>
14. Demestichas, K., Peppes, N., Alexakis, T.: Survey on security threats in agricultural IoT and smart farming. *Sensors* **20**, 6458 (2020). <https://doi.org/10.3390/s20226458>
15. West, J.: A prediction model framework for cyber-attacks to precision agriculture technologies. *J. Agric. Food Inf.* **19**(4), 307–330 (2018). <https://doi.org/10.1080/10496505.2017.1417859>
16. Devendra, C.: *Climate Change Threats and Effects: Challenges for Agriculture and Food Security*. ASM Series on Climate Change (2012). https://pdf.usaid.gov/pdf_docs/PBAAK550.pdf

17. Elijah, O., et al.: Effect of weather condition on LoRa IoT communication technology in a tropical region: Malaysia. *IEEE Access* **9**, 72835–72843 (2021). <https://doi.org/10.1109/ACCESS.2021.3080317>
18. An, W., Wu, D., Ci, S., Luo, H., Adamchuk, V., Xu, Z.: Chapter 25 - Agriculture cyber-physical systems. In: *Cyber-Physical Systems, Intelligent Data-Centric Systems*, pp. 399–417. Academic Press (2017). <https://doi.org/10.1016/B978-0-12-803801-7.00025-0>
19. Popov, S.: The Tangle. Tangle White Paper (2018). https://assets.ctfassets.net/r1dr6vzfxhev/4i3OM9JTleiE8M6Y04Ii28/d58bc5bb71cebe4adc18fadea1a79037/Tangle_White_Paper_v1.4.2.pdf
20. Lundkvist, D.C., Lilic, J.: An Introduction to IPFS (2016). <https://medium.com/@ConsenSys/an-introduction-to-ipfs-9bba4860abd0>
21. Foundation, I.: mam.js (2021). <https://github.com/iotaedger/mam.js>
22. Foundation, I.: IOTA Streams (2021). <https://www.iota.org/solutions/streams>
23. Palmieri, A., Vilei, A., Castanier, F., Vesco, A., Carelli, A.: Enabling secure data exchange through the IOTA Tangle for IoT constrained devices. *Sensors* **22**, 1384 (2022). <https://doi.org/10.3390/s22041384>
24. Saarinen, M.J., Aumasson, J.P.: The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC) (2015). <https://www.rfc-editor.org/rfc/rfc7693.html>
25. Srivastava, D.: Plant Health (2020). <https://www.kaggle.com/datasets/divumarcus/plant-health>



Sana Solo: An Intelligent Approach to Measure Soil Fertility

Laavanya Rachakonda^(✉)  and Samuel Stasiewicz

Department of Computer Science, University of North Carolina Wilmington, Wilmington, USA
{rachakonda1, bk3723}@uncw.edu

Abstract. Worm castings (Worm Excretion) are one the richest natural fertilizers on earth, making earthworms a very important and applicable soil health indicator. According to an article published in the Polish journal of Environmental studies, the most important chemical components of worm castings are pH, total organic carbon (TOC), total nitrogen (N), plant available phosphorus (P), plant available potassium (K), and calcium water soluble (Ca). These chemical components of worm castings, paired with soil temperature, humidity and electric conductivity, are all measurable values that can indicate the overall health and fertility of soil. Furthermore, these physical-chemical properties can also be measured and analyzed to estimate worm populations in soil, making traditional manual extraction techniques obsolete. The proposed project, Sana Solo, is a device that uses machine learning to estimate worm populations based on the quantities of the physical-chemical properties listed above. Being able to estimate earthworm populations in a timely manner, without the use of extraction techniques, can be used in farms and gardens to evaluate soil fertility.

Keywords: Worm Castings · Soil Fertility · Soil Health · Smart Agriculture · Internet of Things · Edge Computing

1 Introduction

Soil is considered as the upper layer of earth in which plants grow. It is a mixture of organic remains, clay and rocky particles and has black or brown color. Soil serves as a medium for filtration of wastes, serves as the reservoir to hold water and nutrients.

The properties of soils exhibit significant variations due to differences in geology and climate over both space and time. Even a basic attribute like soil thickness can vary greatly, ranging from a few centimeters to several meters. This variability is influenced by factors such as the intensity and duration of weathering, soil deposition and erosion events, as well as the patterns of landscape changes. Despite these differences, soils possess a distinct structural feature that sets them apart from ordinary earth materials and forms the foundation for their classification: a vertical arrangement of layers formed through the combined effects of water percolation and the actions of living organisms [3]. The thematic representation of the proposed Sana solo is represented in Fig. 1.

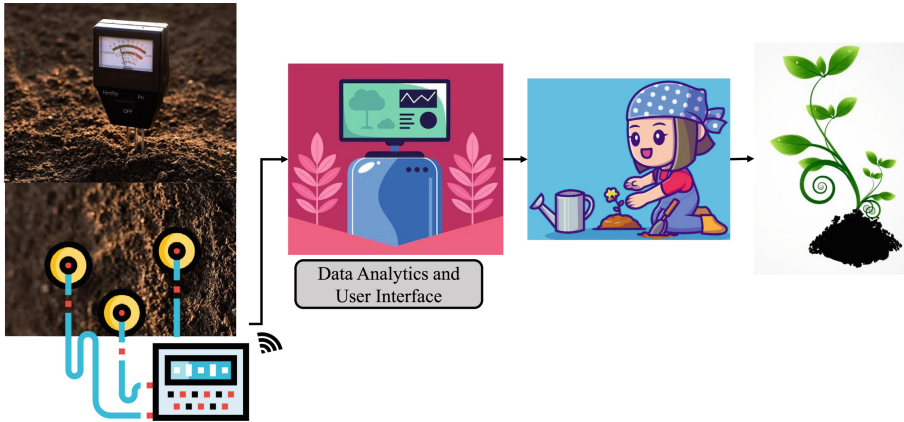


Fig. 1. Thematic Representation of Sana Solo System.

1.1 Soil Erosion and Its Impacts

The actions of flowing water, wind, ice, and gravity constantly disturb soil profiles. These erosive processes remove soil particles from the topsoil and expose underlying horizons to the process of weathering. As a result, there is a loss of essential components such as humus, plant nutrients, and beneficial soil organisms. These losses are particularly crucial for agriculture and forestry. Furthermore, the removal, movement, and subsequent deposition of soil can have significant economic implications [2].

In the past 150 years, approximately 50% of the Earth's topsoil has been depleted or eroded. According to a report by the Intergovernmental Panel on Climate Change (IPCC), soil erosion is occurring at a rate up to 100 times faster than the natural process of soil formation when cultivated without conservation practices. This alarming finding highlights the current imbalance in soil health. Moreover, the risk of erosion is expected to increase further in the future due to temperature changes driven by emissions. This can lead to detrimental consequences such as reduced agricultural production, declining land value, and negative impacts on human health [4].

Soil erosion has implications beyond environmental concerns; it also results in significant economic losses. The global economic losses due to soil erosion are estimated to be approximately \$8 billion. These losses stem from reduced soil fertility, decreased crop yields, and increased water consumption [5]. Soil erosion accounts for a 2% reduction in total agricultural GDP, considering both direct losses faced by farmers and downstream losses affecting others [6]. Another study revealed that in Sleman, a district in Java, soil erosion costs an average farmer 17% of their net income per hectare of agricultural land [7].

Each year, soil erosion inflicts significant economic losses on the agricultural sector. In the United States, the impact amounts to approximately \$44 billion, encompassing reduced productivity as well as the adverse effects of sedimentation and water pollution [8]. This erosion-induced loss extends further to an estimated \$100 million in farm income annually. European countries face substantial agricultural productivity

losses totaling \$1.38 billion per year, alongside a decrease of \$171 million in their gross domestic product [9]. Similarly, South Asia experiences a staggering annual cost of \$10 billion due to soil erosion [10].

2 State of Art and Its Advancement Through the Current Paper

There are methods that involve soil-friendly agricultural practices like terraced farming. The presence of manure enhances soil organic matter, leading to the prevention of erosion. Likewise, implementing a rotation of crops that include both deep-rooted and shallow-rooted varieties enhances soil structure and simultaneously decreases the occurrence of erosion. This prevents erosion and allows more water flow to crops [4].

Crop recommendations based on soil quality are proposed in [11]. Precision agricultural practices were proposed in [12] to enhance crop yields. GIS and GPS technologies were used to monitor crops and weather conditions in [13]. Hydroponic techniques were used to monitor and accelerate plant growth in [14]. Hyper-spectral sensing techniques to access nutrients in soil, mainly nitrogen is performed in [15]. Using features like temperature, humidity, pH and rainfall, an enhanced genetic algorithm has been proposed to predict the nutrients of soil in [16]. Hydroponic techniques were used to propose a vertical farming method to improve the crop fertility in [17].

Decision Trees and Random Forest algorithms were used to predict the crops for soil in [18]. A web application to monitor the environment quality of an area is proposed in [19]. A vertical gardening technique has been proposed using edge computing in [20]. A smart plant monitoring system has been proposed in [21] using environmental conditions. A site specific nutrient management system has been proposed in [22]. A GIS based spatial detection analyses has been proposed to control soil erosion in [23].

2.1 Motivation

As mentioned, most of the solutions support plant growth but the focus on soil health is very minimal. The proposed Sana Solo project focuses more on soil health and its fertility. Monitoring the health of soil by macro-fauna and determining their relationship with respect to soil fertility is the main objective. There are a few devices and products which monitor soil fertility using macro-fauna as mentioned in Table 1.

3 Proposed Sana Solo System

The proposed Sana Solo system measures soil fertility in relationship with the macro-fauna present in the soil. The broad perspective of the proposed system is represented in Fig. 2.

Two different scenarios are presented in the above figure. Scenario A where soil is healthy enough to grow a plant and Scenario 2 which needed worm castings to improve the fertility for efficient plant growth.

The proposed Sana Solo project uses Internet of Things to acquire, monitor and maintain the soil health. The Internet of Things (IoT) refers to the network of physical objects or “things” embedded with sensors, software, and connectivity capabilities, enabling them to collect and exchange data over the internet. These connected

Table 1. Existing Solutions To Measure Soil Fertility using Macro-fauna

Device/Prototype	Notable Features	Main Operation	Drawbacks/Problems
MicroBIOMETER [25]	On site testing kit for microbial biomass and fungal to bacterial ratio	measures microbial biomass to determine health of soil	-nothing to do with macro-fauna, only microbes; Simply takes measurements, gives no recommendations as of what to add/subtract from soil, takes 20 min to get results
Weyers, et al. [26]	8 soil probes positioned in small (.22 m ²) octagon which emits an electrical field into the soil	Electrical field causes earthworms to emerge from ground, which are then collected and counted to determine density in electrified area	could have possible effects on earthworm health
Kempson Extractor [27]	Automated arthropod extractor from soil	Uses heating and cooling to move arthropods into a collecting vessel	gives no evaluations on soil health based on extractions
Ismayilov Amin, et al. [28]	Uses Vis-NIR Spectroscopy technique to determine optical properties of carbon in the soil to give	High levels of SOC indicate high levels of soil organic matter which indicates biological activity in soil and overall soil health	soil organic matter is difficult to measure



Fig. 2. The Broad Perspective of Sana Solo System.

devices can be anything from everyday objects like household appliances and vehicles to industrial equipment and infrastructure components. The IoT allows these objects to communicate and interact with each other, as well as with users or systems, creating a vast ecosystem of interconnected devices. This network of devices enables data gathering, automation, remote monitoring, and control, leading to increased efficiency, convenience, and potential innovation across various industries and domains [32].

Concepts of IoT have been used in Agricultural fields in Sana Solo. Here, a precision farming techniques are being implemented to monitor the soil health alongside the health of macro-fauna present in the soil as represented in Fig. 3.

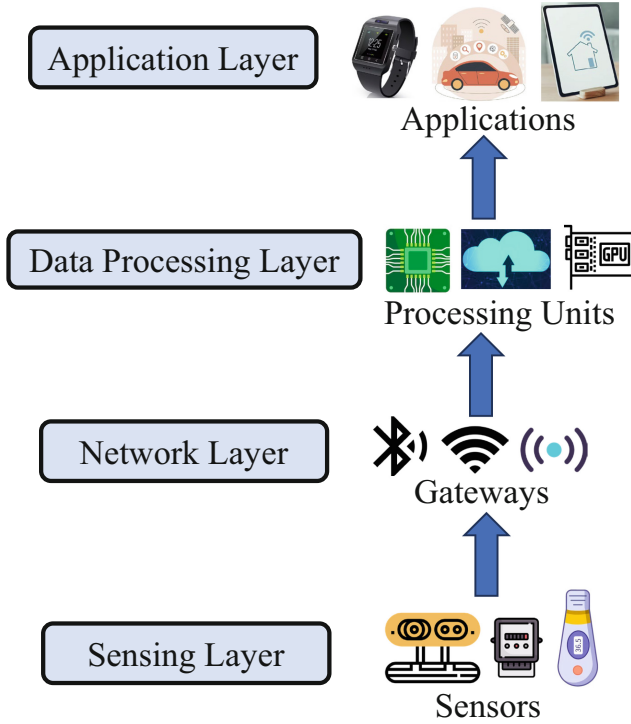


Fig. 3. Architectural View of Internet of Things Network.

Alongside IoT, Edge computing is performed in Sana Solo to make the system efficient and robust. Edge computing refers to the decentralized processing and storage of data at or near the source of its generation, rather than sending it to a centralized cloud or data center for processing. In edge computing, data is processed locally on devices or edge servers, situated closer to where the data is produced, such as IoT devices, sensors, or edge gateways [33] as represented in Fig. 4.

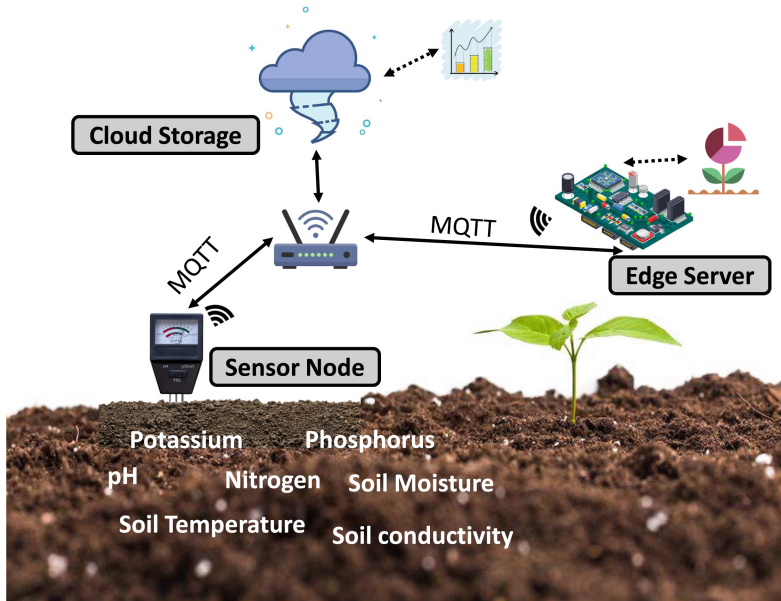


Fig. 4. Edge Computing Paradigm used in Sana Solo System.

3.1 Relationship Between Macro-Fauna and Soil Fertility

The diversity of macro-fauna plays a crucial role in the decomposition and mineralization processes, especially when there are fluctuations in food availability and quality. Therefore, a rich diversity of macro-fauna ensures a consistent and reliable supply of nutrients in soil to support the growth of crops [29]. The humus found in earthworm castings contributes to enhanced water retention in the soil, improved soil aeration, and the retention of plant nutrients that would otherwise be washed away with water. Additionally, earthworm castings provide nourishment to beneficial soil microorganisms, which play a role in producing, storing, and gradually releasing essential plant nutrients into the soil, thereby serving as a source of sustenance for plants [30].

The three main components of worm casting with major implications in soil fertility are plant available, nitrogen (N), phosphorous (P) and potassium (K). The ratio of these in worm castings are 3-1-1. (Synthetic fertilizers have a ratio of 10-10-10 for N-P-K.) [30].

There has been a significant change in chemical properties for soil with and without worm castings. There is an increase in the pH, P, Na, N, Mg, K, Ca values [31].

4 Implementation of Sana Solo System

For the implementation of the system, an Edge processing system has been proposed. The features of the soil are monitored and are transmitted to the user interface for the farmer or care taker to monitor the nutritional value of the soil as shown in Fig. 5.



Fig. 5. Design Flow of the Sana Solo System.

Following features are considered to monitor to determine the soil fertility:

- Soil conductivity
- Soil Moisture
- Soil Temperature
- pH
- Nitrogen
- Phosphorus
- Potassium

The IoT system used in Sana Solo is represented in Fig. 6.

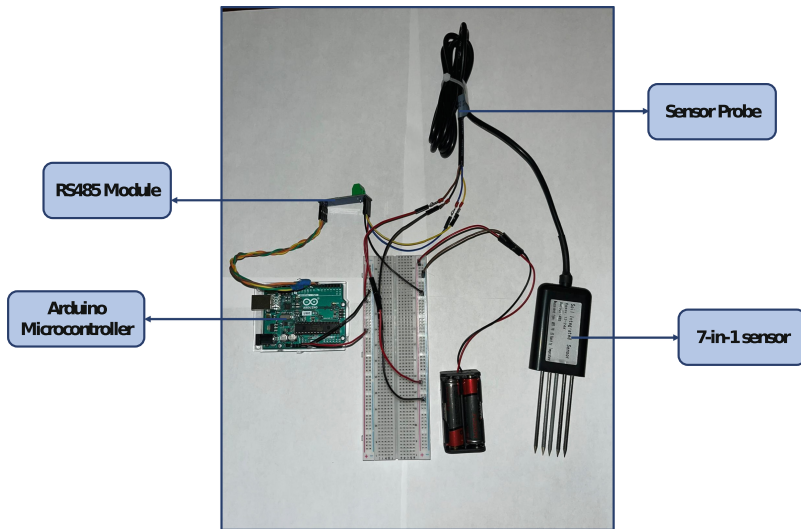


Fig. 6. IoT system used in Sana Solo System.

5 Conclusions and Future Research

Sana Solo System represents the importance of macro-fauna and its growth in the soil. This system also allows to analyze the soil fertility and provides mechanisms to manage the deficiencies. Using this system, farmers or care takers of a particular field can analyze the total number of warm castings and can decide if the land/soil needs more.

For the future research, this system will be placed in multiple test beds. These test beds will be with various configurations - with and without castings to monitor the growth of the plants. The system will also be configured in a way that it can communicate with the devices in the network to share and transfer information to the farmer using an user interface.

Acknowledgements. This version of the project is funded by the College of Arts and Sciences in the University of North Carolina Wilmington.

References

1. Rachakonda, L.: ETS: a smart and enhanced topsoil health monitoring and control system at edge using IoT. In 2022 IEEE International Symposium on Smart Electronic Systems (iSES), Warangal, India, pp. 689-693 (2022). <https://doi.org/10.1109/iSES54909.2022.00153>.
2. Gregory, A.S., et al.: A review of the impacts of degradation threats on soil properties in the UK. *Soil Use Manag.* **31**(Suppl. 1), 1–15 (2015). Epub 12 October 2015. PMID: 27667890; PMCID: PMC5014291. <https://doi.org/10.1111/sum.12212>
3. Liu, H., Li, B., Ren, T.: Soil profile characteristics of high-productivity alluvial cambisols in the North China Plain. *J. Integr. Agric.* **14**(4), 765–773 (2015). ISSN 2095-3119, [https://doi.org/10.1016/S2095-3119\(14\)60789-9](https://doi.org/10.1016/S2095-3119(14)60789-9)
4. Dede, S., Thomas, W.: The Causes and Effects of Soil Erosion, and How to Prevent It. World Resources Institute, February 2000. <https://www.wri.org/insights/causes-and-effects-soil-erosion-and-how-prevent-it#:~:text=A%20report%20from%20the%20Intergovernmental,times%20quicker%20than%20it's%20forming>
5. Martina, S., et al.: A linkage between the biophysical and the economic: assessing the global market impacts of soil erosion. *Land Use Policy* **86**, 299–312 (2019). ISSN 0264–8377, <https://doi.org/10.1016/j.landusepol.2019.05.014>
6. Strutt, A.: Trade Liberalisation and Soil Degradation in Indonesia. Indonesia in a Reforming World Economy: Effects on Agriculture, Trade and the Environment, edited by Kym Anderson et al., pp. 40–60. University of Adelaide Press. JSTOR (2009). <http://www.jstor.org/stable/10.20851/j.ctt1sq5w4j.10> Accessed 21 June 2023
7. Möller, A., Ranke, U.: Estimation of the on-farm-costs of soil erosion in Sleman, Indonesia. *WIT Trans. Ecol. Environ.* **89**, 43–52 (2006)
8. Shepard, K.C.: Oklahoma Farm Report. <https://www.oklahomafarmreport.com/okfr/2023/04/>. Accessed May 2023
9. Panagos, P., Standardi, G., Borrelli, P., Lugato, E., Montanarella, L., Bosello, F.: Cost of agricultural productivity loss due to soil erosion in the European Union: from direct cost evaluation approaches to the use of macroeconomic models. *Land Degrad. Dev.* **29**, 471–484 (2018). <https://doi.org/10.1002/ldr.2879>
10. Third World Network Berhad. <https://twn.my/title/land-ch.htm#:~:text=Its%20shocking%20conclusion%20was%20that,losses%20resulting%20from%20land%20degradation>
11. Kumar, P., Bhagat, K., Lata, K., Jhingran, S.: Crop recommendation using machine learning algorithms. In: International Conference on Disruptive Technologies (ICDT), Greater Noida, India, pp. 100–103 (2023). <https://doi.org/10.1109/ICDT57929.2023.10151325>
12. Saha, P., Kumar, V., Kathuria, S., Gehlot, A., Pachouri, V., Duggal, A.S.: Precision agriculture using Internet of Things and Wireless Sensor Networks. In: International Conference on Disruptive Technologies (ICDT), Greater Noida, India, pp. 519–522 (2023). <https://doi.org/10.1109/ICDT57929.2023.10150678>

13. Prabha, C., Pathak, A.: Enabling technologies in smart agriculture: a way forward towards future fields. In: International Conference on Advancement in Computation and Computer Technologies (InCACCT), Gharuan, India, pp. 821–826 (2023). <https://doi.org/10.1109/InCACCT57535.2023.10141722>
14. Kumar, A., Savaridassan, P.: Monitoring and accelerating plant growth using IoT and Hydroponics. In: International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, pp. 1–6 (2023). <https://doi.org/10.1109/ICCCI56745.2023.10128383>
15. Fleming, K., Gardner, A., Nagel, P., Miao, Y., Mizuta, K.: Hyperspectral sensing for soil health. In: IEEE Conference on Technologies for Sustainability (SusTech), Portland, OR, USA, pp. 1–5 (2023). <https://doi.org/10.1109/SusTech57309.2023.10129629>
16. Irene Monica, N., Pooja, S.R., Rithiga, S., Madhumathi, R.: Soil NPK prediction using enhanced genetic algorithm. In: 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 2014–2018 (2023). <https://doi.org/10.1109/ICACCS57279.2023.10113121>
17. Anuradha, B., Pradeep, R., Ahino, E., Dhanabal, A., Gokul, R.J., Lingeshwaran, S.: Vertical farming algorithm using hydroponics for smart agriculture. In: International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, pp. 432–437 (2023). <https://doi.org/10.1109/ICISCoIS56541.2023.10100527>
18. Rohini, V., Meghana, K., Sowmya, R.K., Krishna, K.S., Srikrishna, B.: Application of SMAP images in predicting Crops by using Decision Tree and Random Forest. In: International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, pp. 1–6 (2023). <https://doi.org/10.1109/ICECONF57129.2023.10083570>
19. Verma, M., Kumar, A., Garg, M., Juneja, S.: Environment quality assessment web application. In: International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, pp. 1339–1342 (2023). <https://doi.org/10.1109/AISC56616.2023.10085252>
20. Bhowmik, A., Sannigrahi, M., Dutta, P.K., Bandyopadhyay, S.: Using edge computing framework with the Internet of Things for intelligent vertical gardening. In: 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jeddah, Saudi Arabia, pp. 1–6 (2023). <https://doi.org/10.1109/ICAISC56366.2023.10085507>
21. Nehra, V., Sharma, M., Sharma, V.: IoT based smart plant monitoring system. In: 13th International Conference on Cloud Computing, Data Science and Engineering (Confluence), Noida, India, pp. 60–65 (2023). <https://doi.org/10.1109/Confluence56041.2023.10048792>
22. Pallavi, C.V., Usha, S.: IoT based site specific nutrient management system for soil health monitoring. In: International Conference on Smart and Sustainable Technologies in Energy and Power Sectors (SSTEPS), Mahendragarh, India, pp. 166–170 (2022). <https://doi.org/10.1109/SSTEPS57475.2022.00050>
23. Sui, X., Lin, C., Zhou, S.: Spatial decision analysis on soil erosion control measures research based on GIS: taking Changting country as an example. In: Third World Congress on Software Engineering, Wuhan, China, pp. 119–122 (2012). <https://doi.org/10.1109/WCSE.2012.29>
24. Naik, T.R., et al.: Environmental testing methodology for real-time soil health monitoring system. In: IEEE Applied Sensing Conference (APSCON), Bengaluru, India, pp. 1–3 (2023). <https://doi.org/10.1109/APSCON56343.2023.10101082>
25. Microbiometer. <https://microbiometer.com/>. Accessed March 2023
26. Weyers, S.L., Schomberg, H.H., Hendrix, P.F., Spokas, K.A., Endale, D.M.: Construction of an electrical device for sampling earthworm populations in the field. *Appl. Eng. Agric.* **24**(3), 391–397 (2008). <https://doi.org/10.13031/2013.24492>
27. Kempson Extractor. https://www.ecotech.de/en/product/kempson_extractor_1. Accessed May 2023

28. Ismayilov, A., Feyziyev, F., Elton, M., Maharram, B.: Soil organic carbon prediction by Vis-NIR spectroscopy: case study the Kur-Aras Plain, Azerbaijan. *Commun. Soil Sci. Plant Anal.* **51**(6), 726–734 (2020)
29. Tauro, T.P., Mtambanengwe, F., Mpepereki, S., Mapfumo, P.: Soil macrofauna response to integrated soil fertility management under maize monocropping in Zimbabwe. *Heliyon* **7**(12), e08567 (2021). PMID: 34917826; PMCID: PMC8666646. <https://doi.org/10.1016/j.heliyon.2021.e08567>
30. Adomako, M.O., Xue, W., Roiloa, S., Zhang, Q., Du, D.L., Yu, F.H.: Earthworms modulate impacts of soil heterogeneity on plant growth at different spatial scales. *Front Plant Sci.* **23**(12), 735495 (2021). PMID: 35003149; PMCID: PMC8732864. <https://doi.org/10.3389/fpls.2021.735495>
31. Huang, M., et al.: Rice yield and the fate of fertilizer nitrogen as affected by addition of earthworm casts collected from oilseed rape fields: a pot experiment. *PLoS ONE* **11**, e0167152 (2021). <https://doi.org/10.1371/journal.pone.0167152>
32. Rachakonda, L., Bapatla, A.K., Mohanty, S.P., et al.: BACTmobile: a smart blood alcohol concentration tracking mechanism for smart vehicles in healthcare CPS framework. *SN Comput. Sci.* **3**, 236 (2022). <https://doi.org/10.1007/s42979-022-01142-9>
33. Rachakonda, L.: Agri-Aid: an automated and continuous farmer health monitoring system using IoMT. In: Camarinha-Matos, L.M., Ribeiro, L., Strous, L. (eds.) *Internet of Things. IoT Through a Multi-disciplinary Perspective. IFIPIoT 2022. IFIP Advances in Information and Communication Technology*, vol. 665. Springer, Cham (2022). <https://doi.org/10.1007/978-3-031-18872-54>



Smart Agriculture – Demystified

Alakananda Mitra¹ , Saraju P. Mohanty² , and Elias Kougianos³ 

¹ Nebraska Water Center, Institute of Agriculture and Natural Resource, University of Nebraska
-Lincoln, Lincoln, USA
amitra6@unl.edu

² Department of Computer Science and Engineering, University of North Texas, Denton, USA

³ Department of Electrical Engineering, University of North Texas, Denton, USA
{saraju.mohanty, elias.kougianos}@unt.edu

Abstract. To tackle the adverse effects of climate change, unprecedented population growth, natural calamities, and natural resource depletion and to ensure food security, smart agriculture is the future of agriculture. This extended abstract for this invited talk is focused on some of the important points of smart agriculture to raise conscientiousness among the future research community.

1 Introduction

Throughout history, agriculture has been crucial to human survival, and it continues to be the backbone of the economies of many countries today. Agriculture’s significance has grown alongside the global population and economy. It now encompasses not just farming but also livestock, poultry, forestry, fisheries, food supply chain, and so on. Unprecedented population growth, climate change, depletion of natural resources, urbanization, over-farming, and deforestation are the crucial factors that are affecting crop yield, disrupting the food supply chain, and threatening human civilization with food scarcity and high prices.

The food and agricultural industries embrace technological advancements, giving birth to “Agriculture 4.0,” a green and smart revolution. Conventional agriculture is transforming into “smart agriculture” and becoming more productive and sustainable by optimizing human labor and natural resources. As a result, crop yield and food production are increasing. Figure 1 shows the various areas of “smart agriculture.” In this article, we highlighted the key factors of “smart agriculture.”

2 Smart Agriculture and Related Terms

Traditional agriculture, which relied on manual labor and produced low yields, is evolving to efficient, sustainable, and eco-friendly “smart agriculture” a.k.a. “smart farming” with the help of technologies like Sensors and Actuators, Internet-of-Things (IoT) [1], Artificial Intelligence (AI) [2], Robotics, and Unmanned Aerial Vehicles. The goal of “smart agriculture” is to maximize both crop quality and output while simultaneously decreasing the amount of effort required to grow the food [3].

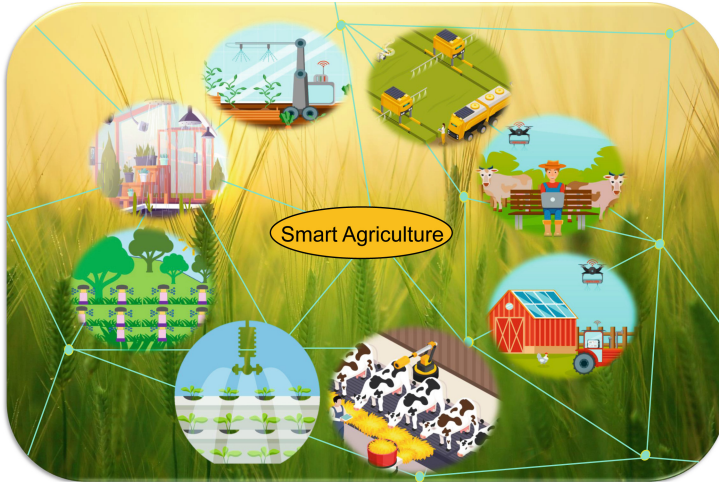


Fig. 1. Smart Agriculture

“Smart agriculture” differs from “precision agriculture” in that it does not prioritize metric precision. Instead, “smart farming” relies on data collection and analysis enabled by modern computing systems to improve the predictability and efficiency of agricultural processes. Both “smart agriculture” and “precision agriculture” together are the two branches of “digital farming” with different focuses. The evolution of “digital farming” also defines the fourth stage of the agricultural revolution, “Agriculture 4.0.”

In this context, a new hybrid system, Cyber-Physical System (CPS), which originated from the IoT deployment in physical systems, is gaining popularity. CPSs connect physical things and infrastructure to the internet as well as to each other by integrating sensing, processing, and networking into these physical objects and infrastructure. The National Science Foundation (NSF) is a pioneer in fostering advancements in the foundational knowledge and technologies necessary to bring cyber-physical systems into existence [4]. Figure 2 shows the three parts of an A-CPS: physical systems, cyber systems, and network fabric. CPSs enable precision and improve functionality, scalability, resilience, safety, security, and usability over simple embedded systems [5]. “Agriculture Cyber-Physical Systems (ACPSs)” can collect meteorological, soil, and crop data to improve agricultural management. ACPSs may monitor water, humidity, and plant health and employ actuators and infrastructure to control temperature and humidity.

Another important and relevant term is “climate smart agriculture (CSA)” [6]. As climate change has already been started, efforts to overcome the adverse effects of climate change are being included in agriculture for sustainability. Smart agriculture has started to transform to climate-smart agriculture to fight against the aftermath of climate change.

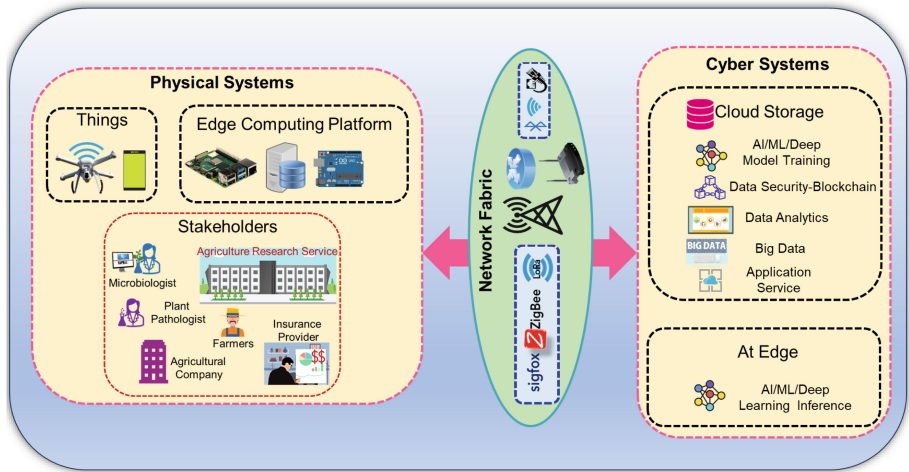


Fig. 2. Elements of a typical Agriculture Cyber Physical System [7]

3 Climate Smart Agriculture and Food Security

The current ramifications of anthropogenic global warming are presently observable, and their impact on humans is irreversible. Furthermore, these consequences are expected to exacerbate in proportion to the continued emission of greenhouse gases into the atmosphere by human activity [8]. By 2100, sea level in the U.S. will increase to 6.6ft. Hurricanes will be much more powerful and destructive. Heat waves will cover a large area of the earth, causing drought and a longer wildfire session. The precipitation pattern will also change. The deserts may see more rain, and fertile land can have no rain. The Arctic will be ice-free as global temperature rises [8].

Climate change impacts crop yield and food production more negatively than positively. Traditional agriculture itself is a major contributor to global warming by emitting 12% of the total greenhouse gases emitted by human activity. Enteric fermentation, manure deposited on pasture, synthetic fertilizer, paddy rice cultivation, and biomass burning are considered to be the agricultural categories with the highest emissions [6].

The CSA emphasizes the significance of collecting actual findings to discern feasible alternatives and essential facilitating actions [6]. It assesses the implications of technology and practices for national development and food security in the context of climate change's site-specific repercussions. It stresses sustainable agriculture, which increases productivity. It focuses on practices such as less tillage, planting different cultivars and cover crops, efficient fertilizer and treatment use, smart water management, increasing the water retention capability of soil, limiting agricultural waste, precise weather forecasting that can optimize the use of irrigation and fertilizers in farming, and so on.

CSA also focuses on communication between policymakers and producers. As it stresses the collective effort from all the communities at each level, starting from the national level to individual stakeholders. Advances in Information and Communication Technologies (ICT) and their large-scale adaptation can build a resilient system. Figure 3 describes the goals of CSA.

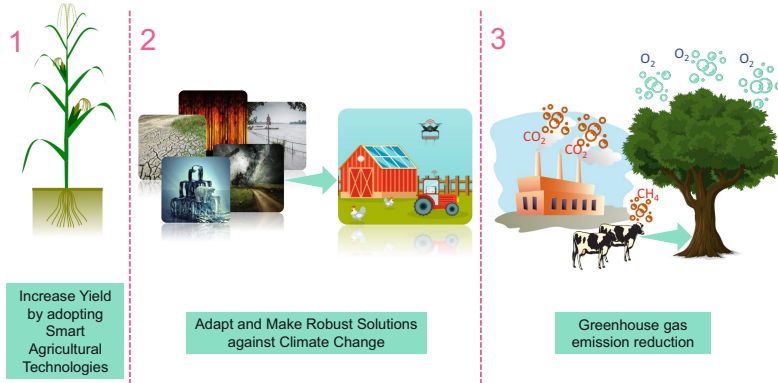


Fig. 3. Goals of Climate Smart Agriculture

Various efforts are being proposed to limit the causes of climate change; e.g., the AgSTAR program [9] has been introduced to help the agricultural industry cut down on methane emissions from livestock manure. Producers concerned with soil health should reduce soil disturbance while increasing cover crops, biodiversity, and the number of plants with roots in the ground. These practices work together to lessen the impact on the environment by decreasing emissions and increasing carbon sequestration. They also benefit the environment by decreasing soil erosion, decreasing the need for costly inputs like fertilizer, increasing water infiltration, boosting nutrient cycling, and constructing more resilient soils over time [10].

4 Smart Agriculture Technologies

All the efforts for sustainable agriculture are possible because of the rapid growth in technologies, especially in the hardware and IC industries, Graphic Processing Units (GPU) and Tensor Processing Units (TPU), computing platforms, and last but not least, Information and Communication Technology (ICT). Industries in different sectors are eagerly embracing digital, smart, green, and sustainable ecosystems to meet the challenges of climate change. Because of this, the relationship between “man” and “machine” is being rethought. Changes are happening in the agricultural sector. “Agriculture 5.0” [11] is knocking at the door.

Artificial Intelligence (AI), Machine Learning (ML), and IoT are playing a major role here, along with UAVs and robotics, as they provide decision-making automation.

Remote sensing through satellite monitoring and cloud computing are two established advanced technologies used for data gathering and decision-making. Different new concepts, like edge computing in agriculture and Software as a Service (SaaS), are emerging. Distributed ledger technology is showing promise and can play an important role in the agricultural industry because of its ability to store immutable data.

Farms are being equipped with sensors and actuators. These IoT sensors and actuators generate huge amounts of data, or big data,” which demands a new stream of data analysis, “big data analysis, in data science.

Farmers can now monitor how far along their crops are in their distinct growth cycles thanks to drone technology. In addition, growers can use UAVs to provide treatments for infected plants. The concept of urban farming, like hydroponics, aeroponics, aquaponics, vertical farming, smart greenhouses, and livestock monitoring, is revolutionizing today’s agriculture and ensuring sustainable agriculture.

5 Smart Agriculture Challenges

Smart Agriculture has simplified and updated the traditional agricultural industry. However, many problems are still to be solved before widespread technological adoption may occur.

- Smart agriculture uses power-hungry, massive machine automation. Farms are large and require many electronic components; therefore, power requirements are often considerable. This has hindered extensive agricultural automation. Renewable energy sources like solar, wind, geothermal, and hydroelectric are being used. However, the storage and transmission of such power are always complex.
- One of the most prevalent features of “smart” farming is machine-to-machine (M2M) communication. To accomplish their goal, they utilize a variety of network and communication protocols to exchange information and coordinate their activities, such as ZigBee, Wi-Fi, LoRA, SigFox, and GPRS. However, due to the chances of physical damage, farms cannot afford such pricey networks over vast open lands.
- High-bandwidth internet connections are not always available in remote rural areas. Unavailability of the internet makes smart agricultural services unavailable.
- Data privacy and security are another bottleneck for smart agriculture. IoT devices generate huge amounts of data, and moving that data from the user or the origin is not always permissible. So, the solution is to move the service near the location of the data.
- Hardware security is another major aspect of IoT devices. The demand for inexpensive and easy-to-use hardware undermines hardware safety. Because of the prevalence of Hardware Trojans and Side Channel Attacks, the widespread adoption of the IoT network in mission-critical applications is being hampered.
- We don’t have any global standards for units and technologies in agriculture. Uniformity will standardize the available services and prices in agrobusiness across the globe.
- Installation of sensors, actuators, or other edge devices, such as drones and agrobots, requires initial capital investment. Investing in that automation is not always easy for small-holder farmers who have small margins of revenue.

- As the field size varies from small-holder farms to large farms, scalability of solutions is needed. It optimizes all the efforts. Along with scalability, the reliability of the solutions will optimize the number of devices. A smaller number of redundant devices that replace faulty devices will minimize the cost.
- To modernize agriculture, one of the biggest challenges is the communication gap between the research community and stakeholder farmers. The issues the farmers need to address do not always reach the researchers, and the agricultural industry cannot fully utilize the benefits of modern technologies.

6 Smart Agriculture Research Problems

As the challenges suggest, there are various areas in agriculture where more research is necessary. For example, research on microgrid structures, power distribution strategies based on requirements and load, the supply of electricity without interruption, and energy smart automation can solve the power issues. Affordable and robust communication technologies can provide better communication between devices and systems. More research on data compression techniques, extreme temperature sensors, publicly accessible datasets, data privacy and security aspects, hardware security, and robust networking is also necessary to accelerate the progress of smart agriculture. Research on federated learning and edge computing-based solutions, robust cryptography, and network protocols for tinyML devices is needed to address data privacy and security issues. Publicly accessible dataset availability is another dire need of the AI community for agricultural research.

7 Conclusions

Today, we live in a world where we cannot deny irreversible climate change. Technological progress and the rapid development of ICT have already boosted the digitization and modernization of agriculture, which results in an increase in agricultural productivity and yields, a decrease in ecological footprints, improved water conservation, increased climate smart efforts, and a decrease in operational costs. Overall, agriculture advances in quality and quantity. However, more climate-smart efforts are needed. In the United States, \$19.5 billion has been sanctioned via the Inflation Reduction Act to support climate change alleviation efforts from 2023 to 2027 [12]. Common Agricultural Policy 2023-2027 of the European Commission aims to form a sustainable, resilient, and contemporary European agriculture economy. It also has a focus on efforts for climate change mitigation [13]. In 2011, India launched *National Innovations in Climate Resilient Agriculture (NICRA)* with \$42.7 million to make Indian agriculture-crops, livestock, and fisheries-more resilient to climate change and unpredictability.

Acknowledgment. This article is an analytic synopsis of [3].

References

1. Farooq, M.U., Waseem, M., Mazhar, S., Khairi, A., Kamal, T.: A review on internet of things (Iot). *Int. J. Comput. Appl.* **113**(1), 1–7 (2015)

2. Winston, P.H.: Artificial intelligence. Addison-Wesley Longman Publishing Co., Inc (1984)
3. Mitra, A., et al.: Everything You Wanted To Know About Smart Agriculture. CoRR, abs/2201.04754 (2022)
4. Cyber-Physical Systems: Enabling a Smart and Connected World. https://www.nsf.gov/news/special_reports/cyber-physical/. Accessed May 25 2023
5. An, W., Wu, D., Ci, S., Luo, H., Adamchuk, V., Xu, Z.: Chapter 25 - agriculture cyber-physical systems. In: Song, H., Rawat, D.B., Jeschke, S., Brecher, C. (eds.) Cyber-Physical Systems. Intelligent Data-Centric Systems, pp. 399–417. Academic Press, Boston (2017)
6. Lipper, L., et al.: Climate-smart agriculture for food security. *Nature Climate Change* **4**(12), 1068–1072 (2014)
7. Mitra, A., Mohanty, S.P., Kougianos, E., et al.: aGROdet: A Novel Framework for Plant Disease Detection and Leaf Damage Estimation. In: Proceedings of the 5th IFIP International Internet of Things Conference (IFIP-IoT), pp. 3–22 (2022)
8. The Effects of Climate Change. <https://climate.nasa.gov/effects/>. Accessed May 25 2023
9. AgSTAR Accomplishments. <https://www.epa.gov/agstar/agstar-accomplishments> Accessed June 22 2023
10. NRCS Climate-Smart Mitigation Activities. <https://www.nrcs.usda.gov/conservation-basics/natural-resource-concerns/climate/climate-smart-mitigation-activities>. Accessed June 23 2023
11. Saiz-Rubio, V., Rovira-Más, F.: From smart farming towards agriculture 5.0: A review on crop data management. *Agronomy* **10**(2), 207 (2020)
12. Climate-Smart Agriculture and Forestry. <https://www.farmers.gov/conservation/climate-smart> Accessed June 23 2023
13. Key policy objectives of the CAP 2023–27. <https://agriculture.ec.europa.eu/common-agricultural-policy/cap-overview/cap-2023-27/key-policy-objectives-cap-2023-27.en> Accessed June 2023

Student Research Forum (SRF)



WeedOut: An Autonomous Weed Sprayer in Smart Agriculture Framework Using Semi-Supervised Non-CNN Annotation

Kiran Kumar Kethineni¹ , Alakananda Mitra² , Saraju P. Mohanty¹  ,
and Elias Kougianos³ 

¹ Department of Computer Science and Engineering, University of North Texas, Denton, USA
{kirankumar.kethineni, saraju.mohanty, elias.kougianos}@unt.edu

² Nebraska Water Center, Institute of Agriculture and Natural Resource, University of
Nebraska-Lincoln, Nebraska, USA
amitra6@unl.edu

³ Department of Electrical Engineering, University of North Texas, Denton, USA

Abstract. With rising challenges and depleting resources, many automation solutions have been developed in agriculture. Integration of Internet-of-Agro-Things (IoAT) and Artificial Intelligence (AI) helped gain better yields while maximizing utilization of minimal resources. Weed management being a task affecting quality and yield of crop attracted attention of automation. However, due to the diverse nature of agriculture, same crop from various geographical locations in different growth stages exhibit different features. Additionally, unknown weeds might also exist in the farm rendering feature based supervised CNN solutions not suitable for weed classification. The current paper presents a weed management Agriculture Cyber-Physical System (A-CPS) called WeedOut with a novel methodology enabling it to work in feature variant environments. WeedOut uses a Semi-Supervised methodology that classifies crops by their shapes and labels them as primary crop and weed crop with minimal inputs from farmer. An autonomous weed sprayer uses outputted labeled images to spray herbicide at weed locations and save primary crop.

Keywords: Smart Agriculture · Agriculture Cyber-Physical System (A-CPS) · Internet-of-Agro-Things (IoAT) · Artificial Intelligence · Computer Vision · Semi-Supervised Learning · Weed Pressure · Weed management

1 Introduction

Agriculture is the primary source of food for all the human beings across the world. Various factors like rapid growth in human population, reduction of farmland, depletion of natural resources and advances in Internet-of-Agro-Things (IoAT) [1] paved path to new paradigm in agriculture named “Smart Agriculture” [2] to automate agriculture routines with help of Artificial Intelligence (AI). Weeds are unwanted plants that grow along with the crop being cultivated and compete with primary crops for resources like

sunlight, water, nutrients, space. Weeds can also serve as a habitat for pests and diseases that can infect crops, provide shade which promotes the growth of fungi. These factors present weed management as a significant part of cultivation in agriculture [3]. Manually spraying herbicide to suppress weeds is easier when farm area is small. In cases of large-scale farming where area of farmland ranges from tens of acres to hundreds of acres manual weeding needs lot of physical labor and inappropriate usage of herbicide can have several negative effects on environment [4]. To reduce manpower and use right amount of herbicide there ought to be a system which can scout through farm to identify weeds and spray at locations of weeds so that weed growth is suppressed without affecting primary crop. Such systems where multiple IoAT devices and AI technologies like Computer Vision are deployed in agriculture infrastructure to automate a specific task are referred to as Agriculture Cyber-Physical System (A-CPS). Current article Weed-Out is a weed management A-CPS that follows a semi-supervised approach as depicted in Fig. 1 to identify weeds and suppress their growth.

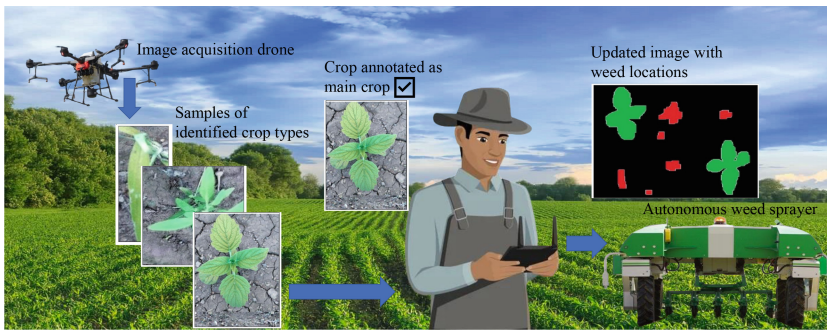


Fig. 1. Overview of proposed WeedOut.

The Rest of the paper is organized as follows: Sect. 2 presents novel contributions of this article followed by discussion on related works in Sect. 3. Section 4 demonstrates the working of proposed solution while experimental results are discussed in Sect. 5. Section 6 concludes the article.

2 Novel Contributions of the Current Paper

2.1 Problem Statement

There have been many solutions using Convolution Neural Networks (CNN) to detect weeds with high accuracy and efficiency. Any CNN needs ample amount of data (images) to train, and such trained networks will be able to only detect any new images with the help of features learned. But, appearance of same crop varies with growth cycle and geographic locations due to various factors. In addition, there could be new kind of weeds which the model has not learned. So, CNN models will have to be trained with lot of images to encompass all possible cases which would not be available in some

cases [5]. In such cases solutions that consider other properties like shape of the crop, area occupied by the crop, patterns in sowing the crop can help us distinguish between primary crop and weed.

2.2 Proposed Solution of the Current Paper

Proposed WeedOut tries to differentiate crops by their shape (profile plots) and clusters similar crops (similar shapes) together. This approach makes proposed method unaffected by differences in appearances/features of the crop due to geographical and aging factors. Knowledge of farmer is utilized in identifying the clusters that represent primary crop to classify crops as weeds and primary crop in semi-supervised fashion.

2.3 Novelty and Significance of the Proposed Solution

The following are novel contributions of this article.

1. No prior data or training is required by WeedOut: Proposed methodology does not need any training involving lot of images and manual labeling effort.
2. Provides insights on weeds present and weed pressure: In addition to classifying crops as primary crops and weeds, proposed method also provides the farmer a list of all kind of weed crops in the farm, percentage of their contribution to total vegetation and weed pressure.
3. Simple and computationally low intensive solution: Proposed algorithm pass through image only 2 times to classify and cluster which is quick and simple. Thus, it can run on end devices like mobile or tablet.

3 Related Prior Works

There have been multiple solutions that do not use CNN for identifying weeds in farmland like [6] which detect rows of plantation, row orientation to know crop margins and label crops outside of crop margins and with lower NDVI as weeds. Whereas, in [7] crop rows are detected by help of depth data and crops lying between crop margins are clustered to 2 clusters by their geometric properties and KNN algorithm. Assuming the number of weeds is greater than primary crop, the smaller cluster is marked as primary crop. In [8] authors proposed a method where crop lines are derived and super pixels (obtained by SLIC) that are in contact with crop lines will be classified as crops, super pixels that are not in contact with crop lines are classified by comparing with neighbors.

In contrast to the above solutions that rely on practice of cultivating in rows, some solutions classify crops by the area they occupy. Authors of [9,10] proposed methods where area covered (number of pixel occupied) by individual crop is computed and the one whose area is below a threshold is classified as weed while the one whose area is above the threshold is classified as primary crop by assuming individual primary crop occupies more area than individual weed crop. But in [11,12] the classification is performed the other way assuming individual primary crop occupies less area than individual weed crop. Article [13] proposes use of Active Shape Models (ASM) for

classification, which calculates shapes of crops present in the image and compares them to shapes of primary crops in memory (training data) to know if its a primary crop or weed. A brief summary of these works are presented in Table 1.

Unlike the above approaches, current approach makes no assumptions on pattern in cultivation or differences in area occupied by individual crops. Instead, WeedOut utilizes shapes of the crops similar to [13] to cluster similar crops and farmers inputs to classify them.

4 Proposed Method - WeedOut

The solution is an A-CPS comprising of multiple devices/machines like drones, weed sprayers and phone/tablet engaged in weed management as presented in Fig. 2. Entire work flow starts with a rover/drone scouting the farm [14] to capture a grid of photos which when stitched together represent the entire farmland.

Table 1. A brief summary of relevant literature.

Work	Year	Assumptions made	Features considered	Remark
Louargant et al. [6]	2019	Cultivation of crops is performed in rows.	Spatial and spectral properties of crop.	Specific to crops which vary in vegetation indices.
Ota et al. [7]	2022	Cultivation of crops is performed in rows.	Spatial and geometric features of crop.	Needs more number of weeds for better classification.
Bah et al. [8]	2017	Cultivation of crops is performed in rows.	Position of crop in farmland and orientation of super pixels.	Specific for crops that are cultivated in rows.
Rani et al. [9]	2017	The average area of a primary crop is greater than that of a weed.	Area occupied by individual crop.	Weeds larger in size may be classified as primary crops.
Irías Tejada et al. [10]	2019	The average area of a primary crop is greater than that of a weed.	Area occupied by individual crop.	Weeds larger in size may be classified as primary crops.
Aravind et al. [11]	2015	The average area of a primary crop is lesser than that of a weed.	Area occupied by individual crop.	Weeds smaller in size may be classified as primary crops.
Siddiqi et al. [12]	2009	The average area of a primary crop is lesser than that of a weed.	Area occupied by individual crop.	Weeds smaller in size may be classified as primary crops.
Maria Persson et al. [13]	2008	NA	Shape of the crop.	Needs to be trained with shapes of primary crop at various orientations.
WeedOut	2023	NA	Shape of the crop.	No training needed, works for all types of crops and all patterns of cultivation.

4.1 Detection and Identification of Individual Crops in Images

Algorithm of crop detection proceeds by processing one image at a time from the set images in the sequence they have been captured. In-order to classify crops, first task is separation of crops from soil by eliminating background. So, image is transformed into HUE color space which represents colors based on hue, saturation and value parameters. Thresholding is performed on image with prior defined limits for green color to detect objects that are in green color (crops) [15]. Image is then resized to 250×250 for ease of computing and converted to binary image as in Fig. 3.

In order to classify crops in the image as primary crops and weeds, individual crops in the image have to be identified and labeled uniquely. Two-pass Connected Component Labeling is a Computer Vision algorithm, which essentially identifies and uniquely labels all the objects in a image by just passing over the image twice. When ever a binary

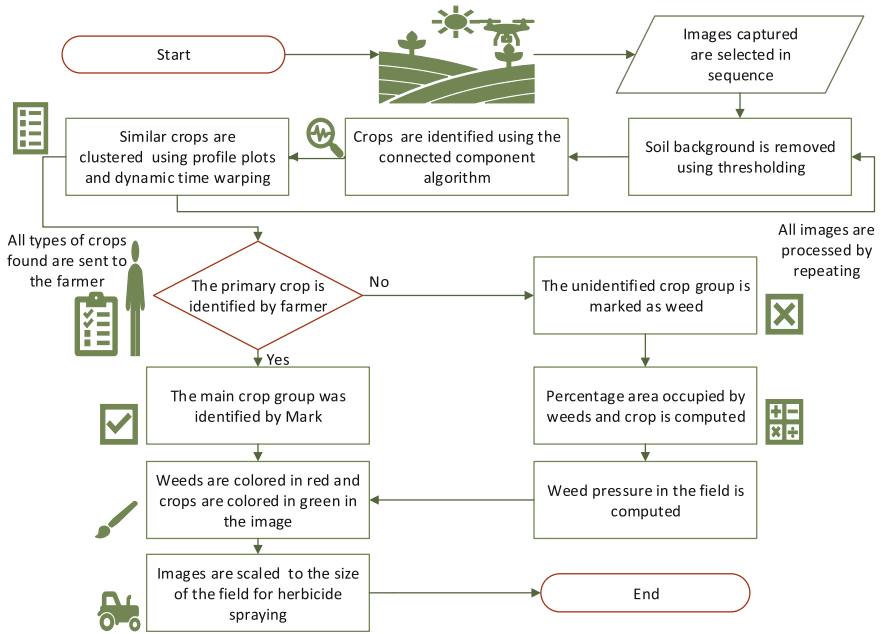


Fig. 2. Working of WeedOut.

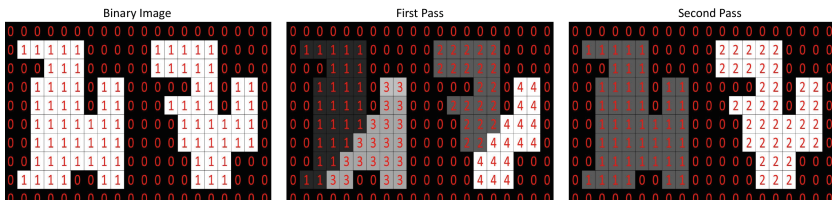


Fig. 3. Demonstration of 2 Pass Connected Component algorithm.

image is presented, the algorithm starts to process each pixel of the image column after column in each row. In the first pass, whenever it reads a pixel that is bright it looks for its neighbor pixels that are bright. If there are any neighbors available, highest of their labels would be assigned to the current pixel. If no neighbors are found a unique label is assigned to the pixel and equivalence between neighboring labels is saved. This process of labeling continues till all the pixels in current image are assigned a label. Second pass identifies various labels assigned to a single object and replaces them with label that is unique to every object as represented in Fig. 3.

4.2 Grouping Identical Crops into Clusters

Every crop essentially differs with others in properties like length of leaves, width of leaves, number of leaves, orientation of leaves. All these features effect how the whole crop looks and how width of crop changes with its length from tip to tip. A plot describing variation in width of a plant with length is termed as Profile Plot, Fig. 4 shows profile plots of two crops demonstrating how profile plots can help differentiating crops.

After computing profile plots for all crops identified in the image, they are extrapolated to length of 250 for ease of visualization. All these profile plots are compared with one another by Dynamic Time Warping (DTW). Dynamic Time Warping of two signals is finding best alignment between them by stretching and compressing one of them along time axis while distance between corresponding points is being minimized as in Fig. 4. DTW distance is the minimum distance required to align the signals. In simple terms, signals those are highly similar would have low DTW distance and thus, can be used as similarity measure. This helps in finding pairs of crops that are similar and all pairs that have a common element are merged to form clusters in iteration till no clusters have a common element. DTW is performed between identified crops for couple images at the initial stage to detect all kinds of crops present. Later on instead of performing DTW between crops identified in a image, DTW is performed between each identified crop and identified crop types to group with similar ones.

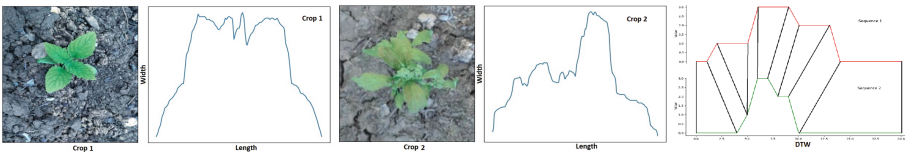


Fig. 4. Visualization of profile plots and Dynamic Time Warping.

4.3 Classification and Targeted Herbicide Application

Once all possible clusters are identified, contribution of each cluster to the entire vegetation is computed by dividing area occupied by each cluster with area occupied by all clusters. Results are presented to farmer with image of one instance from each cluster. From the list of images presented to him, he classifies/labels the image that is similar to his primary crop as primary crop. The label is propagated throughout the cluster to classify crops in that cluster as primary crops. Rest all crops from other clusters are considered as weeds. Thus labeling is performed in semi-supervised fashion with minimal manual intervention. After classifying clusters as primary crops and weeds, area occupied by primary crops and weeds are calculated to determine the percentage of contribution by weeds to the total vegetation known as weed pressure.

All the crops in the image classified as primary crop are marked green and the ones classified as weeds are marked red. Results are now presented to user/farmer and updated images are sent to autonomous weed sprayer. Autonomous weed sprayer is a rover that can travel across the field with provision to carry herbicide. The weed sprayer starts processing each pixel of the image scaled to actual size of farm with a spray nozzle moving correspondingly. When the processor finds a red color pixel belonging identified weed, nozzle sprays herbicide at the location.

5 Experimental Results

Proposed solution was implemented with python and a Computer Vision library OpenCV on a data set from kaggle [16]. To create an image of a farmland multiple images were combined and results of one of such image are discussed below. Thresholding and Connected Component Algorithm are performed on inputted image to identify individual crops, profile plots are plotted for 8 individual crops identified shown in different shade of gray in Fig. 5. DTW is then performed to detect and group similar crops to 3 clusters in Fig. 6.

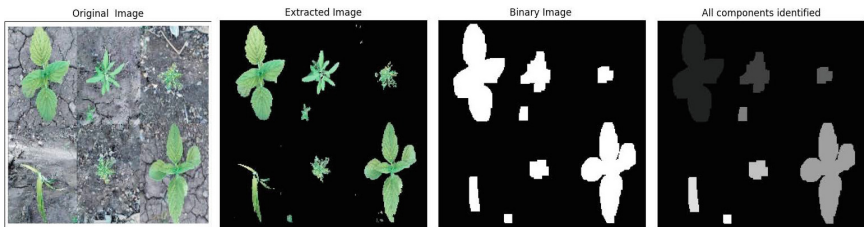


Fig. 5. Different stages in crop identification.

A sample from each cluster is now presented with percentage of contribution of that cluster to the total vegetation to farmer as in Fig. 7. In this experiment farmer selected cluster 1 as his primary crop. All other clusters except cluster 1 are marked weeds and colored red while primary crops are colored green. Final results are presented in Fig. 7

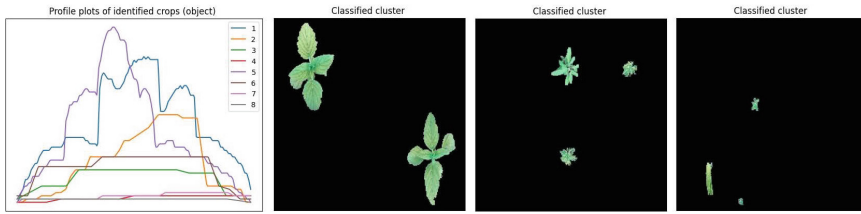


Fig. 6. Clustering of similar crops.

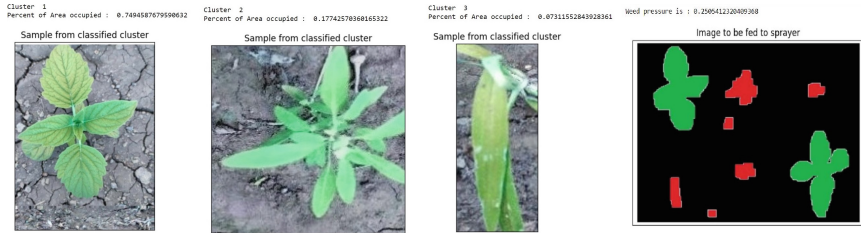


Fig. 7. Results of WeedOut presented to farmer.

along with computed weed pressure, and can be used as an input to autonomous weed sprayer to spray herbicide at weed location.

The same algorithm is fed with 20 of such images to simulate a small sized farmland and calculate its performance metrics. The proposed clustering method showed an accuracy of 93% while F1 score for primary crops and weeds were 0.80, 0.95 respectively indicating that the proposed method was particularly effective at identifying weeds.

6 Conclusion

Current article proposed a novel methodology for an A-CPS delegated with weed management utilizing shape of crops and domain knowledge of farmer to detect weeds in the farmland instead of CNN methods which depend on visual features of crops. WeedOut identifies various crops in the image using Connected Component Labeling Algorithm which checks if any pixel has a directed connection or connected path to other pixel of a object to decide if it belongs to same object or not. This assumption leads to two crops with some overlap be considered as single crop, which means this solution only works for non-overlapping crops in farmland. Proposed method classifies crops by their shape which poses chances of misclassification if two crops have similar profile. Methods to distinguish crops even in cases of overlap with help of edge detection and considering some additional geometrical features that help in more accurate identification can be explored as future works for the proposed solution.

References

1. Mohanty, S.P.: Internet-of-agro-things (IoAT) makes smart agriculture. *IEEE Consum. Electron. Mag.* **10**(4), 4–5 (2021)
2. Mitra, A., et al.: Everything you wanted to know about smart agriculture (2022)
3. Ekwealor, K.U., Echereme, C.B., Ofobeze, T.N., Okereke, C.N.: Economic importance of weeds: a review. *Asian J. Plant Sci.* **3**, 1–11 (2019)
4. Kudsk, P., Streibig, J.C.: Herbicides-a two-edged sword. *Weed Res.* **43**(2), 90–102 (2003)
5. Slaughter, D.C., Giles, D.K., Downey, D.: Autonomous robotic weed control systems: a review. *Comput. Electron. Agric.* **61**(1), 63–78 (2008)
6. Louargant, M., et al.: Unsupervised classification algorithm for early weed detection in row-crops by combining spatial and spectral information. *Remote Sens.* **10**(5), 761 (2018)
7. Ota, K., Kasahara, J.L.Y., Yamashita, A., Asama, H.: Weed and crop detection by combining crop row detection and k-means clustering in weed infested agricultural fields. In: *Proceedings IEEE/SICE International Symposium on System Integration (SII)*, pp. 985–990 (2022)
8. Bah, M.D., Hafiane, A., Canals, R.: Weeds detection in UAV imagery using SLIC and the hough transform. In: *Proceedings Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6 (2017)
9. Rani, K.A.A., Supriya, P., Sarath, T.V.: Computer vision based segregation of carrot and curry leaf plants with weed identification in carrot field. In: *Proceedings International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 185–188 (2017)
10. Tejada, A.J.I., Castro, R.C.: Algorithm of weed detection in crops by computational vision. In: *Proceedings International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pp. 124–128 (2019)
11. Aravind, R., Daman, M., Kariyappa, B.S.: Design and development of automatic weed detection and smart herbicide sprayer robot. In: *Proceedings of IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 257–261 (2025)
12. Siddiqi, M.H., Ahmad, I., Sulaiman, S.B.: Weed recognition based on erosion and dilation segmentation algorithm. In: *Proceedings of International Conference on Education Technology and Computer*, pp. 224–228 (2009)
13. Persson, M., Åstrand, B.: Classification of crops and weeds extracted by active shape models. *Biosys. Eng.* **100**(4), 484–497 (2008)
14. Mitra, A., Singhal, A., Mohanty, S.P., Kougiannos, E., Ray, C.: eCrop: a novel framework for automatic crop damage estimation in smart agriculture. *SN Comput. Sci.* **3**(4), 319 (2022)
15. Mitra, A., Mohanty, S.P., Kougiannos, E.: aGROdet: a novel framework for plant disease detection and leaf damage estimation. In: *Proceedings of the 5th IFIP International Internet of Things Conference (IFIP-IoT)*, pp. 3–22 (2022)
16. Dabhi, R., Makwana, D.: Crop and weed detection data with bounding boxes (2020). <https://www.kaggle.com/datasets/ravirajsinh45/crop-and-weed-detection-data-with-bounding-boxes>. Accessed 20 Jan 2023



ALBA: Novel Anomaly Location-Based Authentication in IoMT Environment Using Unsupervised ML

Fawaz J. Alruwaili¹ , Saraju P. Mohanty¹  , and Elias Kougianos² 

¹ Department of Computer Science and Engineering, University of North Texas, Denton, USA
fawazalruwaili@my.unt.edu, saraju.mohanty@unt.edu

² Department of Electrical Engineering, University of North Texas, Denton, USA
elias.kougianos@unt.edu

Abstract. Smartphones have become essential components in the Internet of Medical Things (IoMT), providing convenient interfaces and advanced technology that enable interaction with various medical devices and sensors. This makes smartphones serve as gateways for sensitive data that could potentially affect patients' health and privacy if compromised, making them primary targets for cybersecurity threats. Authentication is crucial for IoMT security, as its effectiveness relies on its resistance to any conditions of environment, device, or user. In this paper, we propose the Anomaly Location-based Authentication (ALBA) method using GPS technology and a lightweight unsupervised ML algorithm with more stable features. Our experimental results showed that the model successfully identified anomalous locations across three distinct datasets, demonstrating the adaptability of ALBA.

Keywords: Healthcare Cyber-Physical System (H-CPS) · Internet of Medical Things (IoMT) · Intelligent Security · Cybersecurity · Location-Based Authentication

1 Introduction

The growth of IoT embedded systems and biosensors, has introduced the IoMT as a branch that integrates medical devices, applications, and networks to enhance the efficiency of healthcare system [1, 2]. The rapid advancements in mobile technology have enabled smartphones to become an important component of the IoMT network and a source of information due to the increasing complexity of software and hardware components and multiple interfaces in medical devices [3]. However, smartphones also introduce new security challenges due to the sensitive nature of medical data that they collect, making them a valuable target for cybersecurity threats [4, 5]. Therefore, ensuring the security of IoMT is crucial to mitigate risks and enhance the sustainability of healthcare.

Artificial Intelligence (AI) technologies have been advanced significantly and can be used to monitor and predict the behavior of entities within an IoT environment. However, data quality is crucial in machine learning (ML) for achieving accurate results.

© IFIP International Federation for Information Processing 2023

Published by Springer Nature Switzerland AG 2023

D. Puthal et al. (Eds.): IFIP IoT 2023, IFIP AICT 683, pp. 424–432, 2023.

https://doi.org/10.1007/978-3-031-45878-1_30

While many studies on behavioral authentication for smartphones have contributed valuable insights, research expectations have not met in terms of accuracy or considered IoT device security requirements under different conditions related to environment, device, and user. Therefore, effective IoMT security solutions require holistic security considerations, while maintaining user convenience.

In this paper, we propose a behavior-based authentication method for smartphones in IoMT network using GPS sensors and an unsupervised ML model, which can be utilized as a additional security layer without requiring user intervention, and with more stable features. Figure 1 depicts the basic overview of our proposed method. The method’s efficacy was evaluated using three distinct datasets. The results demonstrate its adaptability to various realistic conditions, indicating its potential to be implemented in IoMT.

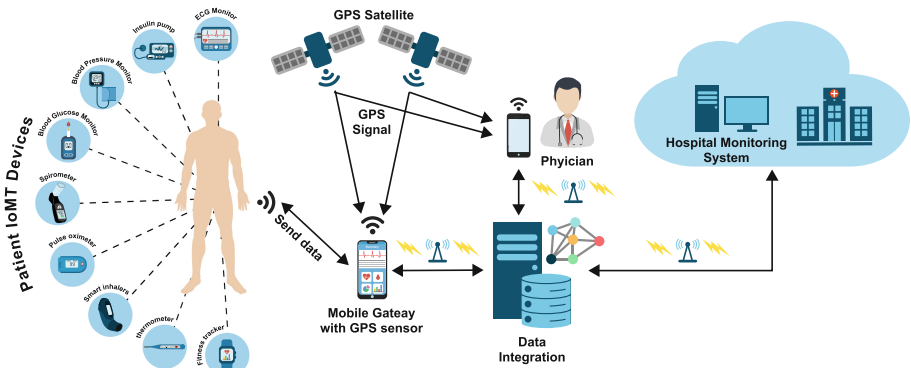


Fig. 1. Overview of the Proposed ALBA for IoMT.

The paper is organized as follows: Section 2 presents the literature review, while Sect. 3 introduces the novel contributions of this paper. The proposed method is presented in Sect. 4, and data preprocessing is detailed in Sect. 5. The ML model used in our method is described in Sect. 6. Experimental results are provided in Sect. 7, and the conclusion with future work are in Sect. 8.

2 Related Research on Behavioral Authentication

Existing research on behavioral authentication has provided valuable insights using various techniques and sensors, such as Keystroke Dynamics (KD) [6, 7], Touch Gestures (TG) [8, 9], and Gait Behavior [10, 11]. However, some studies may not have fully met expectations in terms of accuracy and stability of authentication data, nor considered the holistic security needs covering diverse environmental, device, and user conditions. This limits their accuracy, suitability and effectiveness for the IoT devices, especially smartphones.

The limitations of these techniques are mainly due to internal and external factors, such as the variety of devices, where smartphones have touchscreens or keyboards with different shapes, layouts, and sizes [12]. Also, the specific language in which technique is applied affects the tested interval time between touches, where the user may be unfamiliar with some vocabularies. Additionally, there are external factors that affect these techniques, such as, environment, clothing, sickness, injuries, fatigue, emotional or mental status, and smartphone position. These limitations make the extracted features insufficient for behavioral-based authentication. Based on the above discussions, we conclude that existing approaches to behavioral authentication in IoMT are still lacking and have limitations. Therefore, ALBA method aims to address these limitations and improve authentication data stability to be more accurate and usable in IoT devices.

3 Novel Contributions

3.1 Problem Addressed and Proposed Solution

Smartphones have revolutionized healthcare access due to their advanced technology, where they used to collect and transmit sensitive medical data, making them vulnerable to cyberattacks that compromise patient privacy and have life-threatening consequences. Therefore, securing smartphones within the IoMT network is essential. Various behavioral authentication methods for smartphones have been proposed to address vulnerabilities in traditional authentication factors. However, these methods face challenges in performance and accuracy due to factors impacting authentication data stability. Therefore, authentication methods must consider holistic security considerations, the nature of devices, targeted environments, and their applicability to available technologies.

ALBA exploits GPS technology in smartphones to authenticate users based on their behavior of their locations utilizing ML technology for analyzing and detecting anomalous locations, ensuring faster response times to security threats. ALBA overcomes limitations of behavioral features used in previous studies, and provides more stable behavioral features under different conditions related to environment, device, and user. GPS sensors can be embedded in multiple IoMT devices without requiring specific hardware design or size.

3.2 Novelty of the Proposed Solution

ALBA method provides several contributions: robustness by being less sensitive to internal/external factors and countering for GPS inaccuracies; increasing efficiency as GPS requires less features and a lightweight iForest algorithm that has low memory requirements, reducing computational demand and power consumption compared to existing behavioral methods; scalability and applicability with GPS integration in various IoMT devices without specific hardware requirements; enhancing user convenience as our method can be an additional security layer along with existing authentication factors, reducing their constraints. These contributions make our proposed method more

suitable for device technologies and more comprehensive in terms of security considerations in an IoMT environment. To the best of our knowledge, we have proposed the first behavioral authentication method integrating GPS and unsupervised ML technologies for IoMT security.

4 Proposed Authentication Mechanism

When user credentials are validated, authentication is based on comparing the current location with historical locations stored in the database within a given time frame. If the behavior of current location matches the behavior of the historical locations, it will be considered normal location. Otherwise is anomaly.

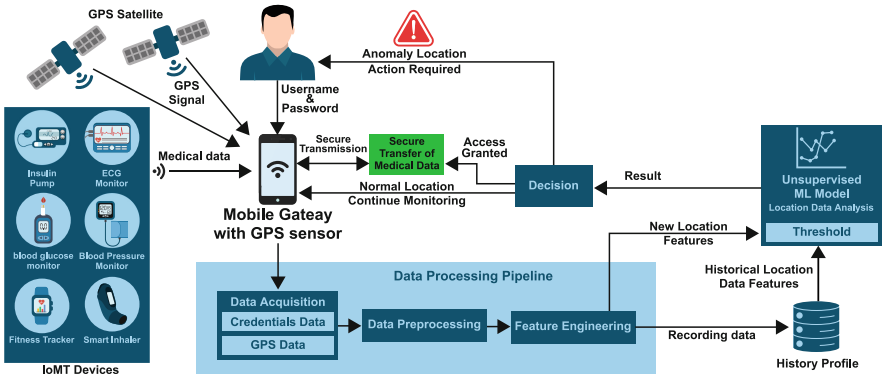


Fig. 2. ALBA Method Workflow for IoMT.

Figure 2 illustrates the proposed authentication method, where the mobile device collects data from different resources (user, medical devices, and GPS satellites) and transmits it to the medical server’s IoMT system for verification. Specifically, when users connects their medical devices to the server, the IoMT system prompts the user for username and password. The GPS data are then verified and analyzed using ML algorithms to detect whether the current location is anomalous or not. If the location is normal, the verification process is successful, and user’s medical devices will be connected to the medical server, allowing secure transmission of medical data for doctor diagnosis. The user also will be able to access health record. Historical location data is pre-processed before being stored to reduce computational time and resource consumption, which positively impacts power consumption during future authentication processes. The result of data analysis determines whether to continue monitoring user’s current location or take appropriate action in case of any deviation from the expected behavior, such as limiting system functionality until additional authentication is provided or sending alerts through other channels.

4.1 Data Collection

The real-world dataset was collected over 27 days using the Google Maps app on an iPhone 11 Pro, with 359 locations visited during various times of the day and using different navigation modes. Figure 3 (a) illustrates a sample of the recorded locations density, with reduced clutter to improve readability. Figure 3 (b) shows the recorded locations individually.

Data accuracy is crucial in ML, and significantly impact model performance. Therefore, data collection process was monitored daily to ensure the accuracy.

4.2 Datasets Description

The effectiveness of ALBA was evaluated using three distinct datasets: (1) a real-world dataset which was collected for this study with 359 observations recorded at irregular intervals over 27 d using an iPhone 11 Pro to evaluate ALBA under real-world scenarios, (2) a public dataset that was utilized to evaluate the performance of ALBA on different real-world data and scenarios, and to ensure its generalizability. It was obtained from Kaggle [13] with 40,603 observations collected in October 2014 using an Android device, and (3) a virtual dataset which was created with 3,359 observations using Python programming language, including 5 anomalous locations with different regular patterns and sudden changes. It was utilized for evaluation under specific realistic scenarios not clearly present in the previous datasets.

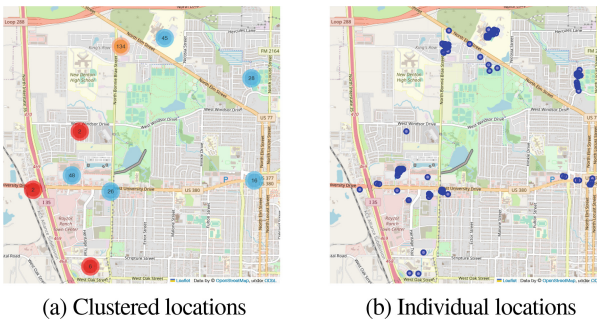


Fig. 3. Sample of collected locations - Real-world Dataset.

5 Data Pre-processing

In our experiment, location features, such as latitude, longitude, and timestamp features are selected from the original dataset, while others are redundant. Day and hour features were extracted from timestamps to improve anomaly detection by identifying deviations during specific times or days. In addition, latitude and longitude features are standardized to be on comparable scales. These features are then combined using PCA to create a single location feature, enabling easier analysis and visualization of location patterns and anomalies.

6 Isolation Forest Model

The iForest is an unsupervised ML algorithm used for anomaly detection without pre-labeled data. It has several advantages that make it suitable for our proposed method, including its low memory requirements, reducible model sensitivity, adaptability to data distribution changes.

Anomalies are isolated by building decision trees (DTs), which are combined to produce prediction. DTs are constructed by recursively selecting a random feature and a split value within the range of the selected feature. The iForest has a linear time complexity $O(n)$, as it isolates anomalies instead of normal observations [14], where anomalies are expected to be fewer [15], tending to be closer to the root. The number of DTs affects the model performance and accuracy, but also impacts computational time and resource requirements. The optimal choice relies on the dataset, available resources, and multiple experiments.

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (1)$$

The base 2 in the exponential function is to ensure the score is between 0 and 1. The number of observations is indicated by n , where $h(x)$ is the path length of a point x , and $E(h(x))$ its expected average path length. The constant $c(n)$ is the average path length of terminal nodes in DTs, used to scale and normalize scores. Utilizing a suitable threshold value is essential for accurately identifying security threats, especially in the IoMT where false positives can disrupt medical operations and compromise patient safety. Identifying an optimal threshold requires iterations, evaluating results, and refining the value with domain knowledge and expert input.

7 Experimental Results

7.1 Real-World Dataset Results

In our experiment, the iForest model was trained on location data in the real-world dataset. The results showed that the model successfully calculated anomaly scores as shown in Fig. 4 (a), identifying 11 anomaly scores represented in red dots as negative values, while positive values (blue dots) represent the normal scores. The farther from 0, the more anomalous (or normal) a location is. The model's prediction is shown in Fig. 4 (b) as a binary series of -1 for anomalies and 1 for normal points.

The model's sensitivity was controlled by utilizing an anomaly threshold of -0.05 after conducting multiple experiments and evaluating performance, leading the model to identify 6 real anomalous locations with a significant deviation indicated by dark red in Fig. 4 (c). For contextual anomalies, they occurred during certain hours or days, which makes them different from other locations. However, the model considered them as normal based on the determined threshold as they have slight deviation. For more insight on the spatial distribution of anomalous and normal locations, they were projected on the map depicted in Fig. 4 (d).

7.2 Public Dataset Results

The model was trained on the public dataset, and all anomaly scores were calculated successfully as shown in Fig. 5 (a). There were 121 anomalous scores (red dots), where 58 of them were above the utilized threshold of -0.015 , representing real anomalous locations as shown in Fig. 5 (b) represented by dark red dots.

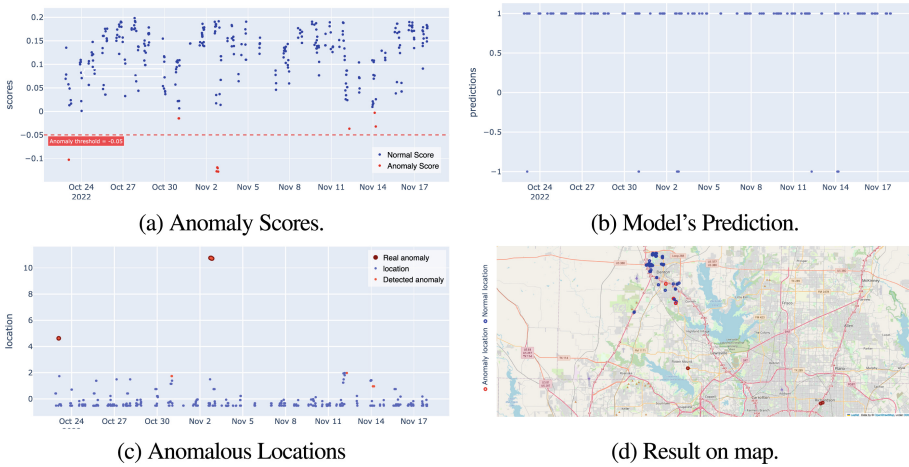


Fig. 4. Real-world dataset Results - A threshold of -0.05

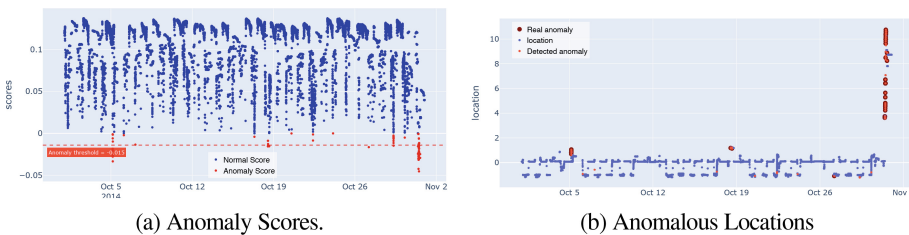


Fig. 5. Public Dataset Results - A threshold of -0.015

We can notice that some of dark red locations have a significant deviation from the behavior of other locations, while the red locations have a slight deviation, and which considered normal based on the determined threshold. However, the public dataset locations are shown in a clear daily pattern with some significant deviation, especially on the right side. Most of the deviations occur at the beginning or end of the week or during the weekends, which is a reasonable pattern.

7.3 Virtual Dataset Results

In the virtual dataset, there were 7 anomaly scores as depicted in Fig. 6 (a). Based on the calculated anomaly scores, the model successfully identified 7 anomalous locations as illustrated in Fig. 6 (b)

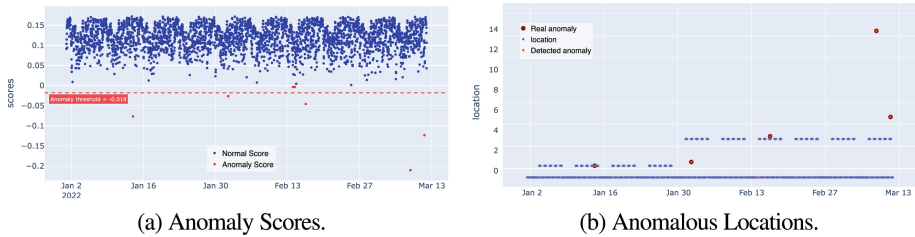


Fig. 6. Virtual Dataset Results - A threshold of -0.018

The model effectively detected all 5 known anomalous locations, which are represented by dark red dots. However, there were 2 false positives identified by the model, which were due to the adjustment of the model's parameters to optimize accuracy for the virtual dataset. Despite this, by adjusting the threshold value to 0.018 , the model effectively reclassified these false positives as normal locations (red dots), demonstrating its ability to adapt and perform well on the given dataset.

8 Conclusion and Future Work

Integrating GPS and ML technologies can enhance the security of traditional authentication factors. The behavioral patterns in proposed ALBA are more stable and accurate compared to previous studies, which depend on other behavioral patterns that can be affected by various factors. The experimental results across diverse datasets validate the model's ability to detect location deviations from normal patterns, ensuring effective authentication. For future work, we suggest exploring the integration of additional behavioral patterns and data types to further improve effectiveness and robustness of authentication process.

References

1. Mohanty, S.P., Choppali, U., Kougianos, E.: Everything you wanted to know about smart cities: the internet of things is the backbone. *IEEE Consum. Electron. Mag.* **5**(3), 60–70 (2016)
2. Ghubaish, A., Salman, T., Zolanvari, M., Unal, D., Al-Ali, A., Jain, R.: Recent advances in the internet-of-medical-things (IoMT) systems security. *IEEE Internet Things J.* **8**(11), 8707–8718 (2021)

3. Mutyara, A.G., Farras, B.A., Sari, L.P., Achmad, S., Sutoyo, R.: The influence of smartphone applications on human healthcare. In: International Conference on Informatics Electrical and Electronics (ICIEE), pp. 1–6 (2022)
4. ur Rahman, G.M.E., Chowdhury, R.I., Dinh, A., Wahid, K.A.: A smart sensor node with smartphone based IoMT. In IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), pp. 92–95 (2019)
5. Iqbal, W., Abbas, H., Daneshmand, M., Rauf, B., Bangash, Y.A.: An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security. *IEEE Internet Things J.* **7**(10), 10250–10276 (2020)
6. T. L. Lin and Y. S. Chen. A chinese continuous keystroke authentication method using cognitive factors. In IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), pp. 1–2 (2019)
7. Herath, H.M.C.K.B., Dulanga, K.G.C., Tharindu, N.V.D., Ganegoda, G.U.: Continuous user authentication using keystroke dynamics for touch devices. In: 2nd International Conference on Image Processing and Robotics (ICIPRob), pp. 1–6 (2022)
8. Ouadjer, Y., Adnane, M., Bouadjenek, N.: Feature importance evaluation of smartphone touch gestures for biometric authentication. In: 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH), pp. 103–107 (2021)
9. Suharsono, A., Liang, D.: Hand stability based features for touch behavior smartphone authentication. In: 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII), pp. 167–170 (2020)
10. Musale, P., Baek, D., Choi, B.J.: Lightweight gait based authentication technique for IoT using subconscious level activities. In: IEEE 4th World Forum on Internet of Things (WF-IoT), pp. 564–567 (2018)
11. He, L., Ma, C., Tu, C., Zhang, Y.: Gait2vec: Continuous authentication of smartphone users based on gait behavior. In: IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 280–285 (2022)
12. Roy, S., et al.: A systematic literature review on latest keystroke dynamics based models. *IEEE Access* **10**, 92192–92236 (2022)
13. SimonD, J.: Mobile location history of 10/2014. Kaggle (2017). Accessed 28 Dec 2022
14. Zhang, L., Liu, L.: Data anomaly detection based on isolation forest algorithm. In: International Conference on Computation, Big-Data and Engineering (ICCBDE), pp. 87–89, Yunlin, Taiwan (2022)
15. Marcelli, E., Barbariol, T., Savarino, V., Beghi, A., Susto, G.A.: A revised isolation forest procedure for anomaly detection with high number of data points. In: IEEE 23rd Latin American Test Symposium (LATS), pp. 1–5, Montevideo, Uruguay (2022)



A Configurable Activation Function for Variable Bit-Precision DNN Hardware Accelerators

Sudheer Vishwakarma¹, Gopal Raut², Narendra Singh Dhakad²,
Santosh Kumar Vishvakarma², and Dhruva Ghai¹(✉)

¹ Oriental University, Indore, India

{vsudheer062,dhruvaghai}@orientaluniversity.in

² Indian Institute of Technology Indore, Indore, India

{phd1901202004,skvishvakarma}@iiti.ac.in

Abstract. This paper introduces a configurable Activation Function (AF) that utilizes ROM/ Cordic architecture to generate sigmoid and tanh with varying bit precision. Two design strategies are explored: a ROM-based approach for low-bit precision and a Cordic-based approach for high-bit precision. The accuracy of the configurable AF is assessed on LeNet and VGG-16 DNN models, revealing minimal accuracy loss (less than 1.5%) compared to the tensorflow-based model. Experimental results on the Zybo Evaluation kit-Xilinx, using a ‘fixed<9, 6>’ arithmetic representation, demonstrate the ROM-based approach’s memory efficiency, achieving 86.66% LUT savings for 4-bit precision and 80.95% LUT savings for 8-bit precision compared to the Cordic-based approach. The Cordic-based approach, on the other hand, shows $\approx 93\%$ LUT savings for 16-bit precision, compared to the ROM-based approach. The proposed AF utilizes the robustness of ROM and Cordic architectures for appropriate bit precision to enhance the overall performance of Deep Neural Networks (DNNs).

Keywords: DNN accelerators · Cordic architecture · Configurable AF · fixed-point · FPGA

1 Introduction and Contributions

DNN hardware accelerator implementation poses significant challenges due to intensive computational demands and hardware resource requirements [8]. The minimum precision for accurate neural networks with reduced complexity has been discussed in [11]. Major components of DNN include arithmetic and computational units, which are Multiply-Accumulate (MAC) unit and Activation Function (AF) [10]. The AF applies non-linear transformations to the MAC output as shown in Fig. 1(a). It depicts a single neuron having MAC with inputs (x_1 to x_n), weights (w_1 to w_n), and a bias b . A conventional design for configurable AF is shown using the select line `AF_select`, which is needed for Field

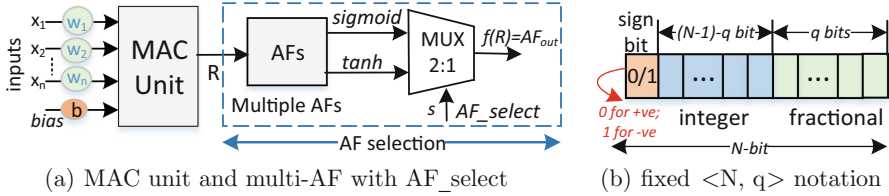


Fig. 1. Neuron architecture with MAC and multi-AFs, utilizing fixed-point computation format $\langle N, q \rangle$.

Programmable Gate Array (FPGA) and Application-Specific Integrated Circuit (ASIC) implementations. The AF should also support multi-bit precision. A programmable AF is preferred for multiple applications [9]. Implementing multiple AFs requires additional resources, leading to higher critical delays. The MAC unit generates an output (R) (Eq. 1), which the AF then processes to produce the final output $f(R)$ (Fig. 1(a)).

$$AF_{out} = f(R) = f\left(\sum_{i=1}^n (x_i \times w_i) + b\right) \tag{1}$$

The fixed-point representation (Fig. 1(b)), is denoted as ‘fixed $\langle N, q \rangle$ ’. It includes an MSB as a sign bit (0 for positive, 1 for negative), $(N-1) - q$ integer bits, and q fractional bits. Targeting hardware implementation with this fixed-point notation, ROM-based AFs are not suitable for high-bitwidth applications due to their significant resource utilization (i.e., LUT in FPGA and memory elements in ASIC implementations). The LUT-based approach involves splitting non-linear input ranges into regions and storing their data in LUTs as straight-line segments. FPGA-based customizable hardware designs for AFs have been proposed in [6], allowing configurability, but consuming more on-chip area compared to ASICs. FPGAs use BRAM for efficient AF access, reducing computation overhead. However, increased BRAM utilization trades memory usage for bit precision [10]. A library of VLSI implementations for various AFs is presented in [7] to design hardware-efficient neural network accelerators.

A linear function has been utilized to approximate the log sigmoid function, while [2] proposes a polynomial model for the fractional exponent part of the tanh AF implementation. These approaches aim to achieve accurate approximations with minimal resource usage, but configurability in AF has not been discussed. An energy-efficient DNN accelerator, which incorporates variable precision support, improved performance, and reduced energy consumption, is evaluated at the MAC level in [3], but the investigation of the AF is warranted for further enhancement. The Cordic method, originally introduced by Volder and later modified by Walther, performs circular, linear, and hyperbolic operations and finds applications in various fields such as multimedia and digital signal processing [4]. To address the issues related to additional resources and higher critical delays associated with configurability, a resources reused Cordic-based

architecture has been proposed in [9]. This design efficiently realizes sigmoid and tanh AFs using the same logic resources. However, it has two main drawbacks: low accuracy and high LUT utilization for bit-precision ≤ 8 . To overcome these limitations, we propose a new approach that combines the Cordic algorithm for high bit-precision AF and ROM for low bit-precision AF (≤ 8). The *distinct contributions* of this paper are as follows:

- We introduce a configurable AF based on Cordic and ROM, capable of generating sigmoid and tanh functions for variable bit-precision.
- The accuracy of the proposed AF is compared with the tensor-based model, demonstrating an accuracy loss of less than 1.5%.
- The proposed design demonstrates reduced LUT utilization.

The remaining sections are organized as follows: Sect. 2 introduces the Cordic architecture and the configurable AF that combines ROM and Cordic approaches. Section 3 reports the performance analysis. The paper concludes in Sect. 4.

2 Proposed Configurable Architecture for Variable Bit Precision AFs Using ROM/Cordic Approach

This section presents the configurable AF design, using a combination of the Cordic algorithm and ROM-based approach. First, the Cordic algorithm is discussed, providing insights into its functionality and application. Subsequently, the configurable AF is introduced, highlighting its key features and design considerations.

2.1 Cordic Algorithm for High-Bit Precision

The Cordic algorithm operates by iteratively rotating vector coordinate components (P_i, Q_i) to (P_{i+1}, Q_{i+1}) at each iteration. Equation 2 represents the computation underlying the Cordic algorithm. In hyperbolic mode, it is utilized to generate hyperbolic sine and cosine [9].

$$P_{i+1} = P_i \cdot \cosh\theta_i - Q_i \cdot \sinh\theta_i \quad (2a)$$

$$Q_{i+1} = Q_i \cdot \cosh\theta_i + P_i \cdot \sinh\theta_i \quad (2b)$$

$$R_{i+1} = R_i - \theta_i \quad (2c)$$

The scaling factor $\cosh\theta_i = 0.8281$ is factored out from Eq. 2 as the pseudo-rotation scaling factor, while $\frac{1}{\cosh\theta_i} = 1.2075$ is applied at P_i as an offset. To derive Eq. 3, θ_i is substituted with $d_i \cdot E_i$, where $d_i \in \{-1, 1\}$ determines the negative (-1) or positive (1) rotational direction and E_i represents the memory element (Lookup Table) for the i^{th} iteration (shown in Fig. 2). The values of E_i and mode m depend on the type of coordinate system (linear, circular, or hyperbolic) being employed [5]: $E_i \in \{2^{-i}, \tan^{-1}(2^{-i}), \tanh^{-1}(2^{-i})\}$, and $m \in$

$\{0, 1, -1\}$, respectively. Therefore, for hyperbolic mode, $E_i = \tanh^{-1}(2^{-i})$, and $m = -1$.

$$P_{i+1} = P_i - m \cdot d_i \cdot Q_i \cdot 2^{-i} \tag{3a}$$

$$Q_{i+1} = Q_i + d_i \cdot P_i \cdot 2^{-i} \tag{3b}$$

$$R_{i+1} = R_i - d_i \cdot E_i \tag{3c}$$

Algorithm 1. P_{out} and Q_{out} generation using Cordic

- 1: **Objective:** Generate P_{out} and Q_{out} using Cordic.
 - 2: **Input Factors:** P_{in} , Q_{in} , Hyperbolic angle R_{in} .
 - 3: **Output Responses:** P_{out} and Q_{out} .
 - 4: Initialize N = bit-precision, $P_0 = P_{in} = 1.2075$, $Q_0 = Q_{in} = 0$, $R_0 = R_{in}$, $m = -1$, $d_0 = R_i[N-1]$ (sign bit).
 - 5: **for** i **in** range(0: $N-1$) **do**
 - 6: $P_{i+1} = P_i - (\bar{d}_i - d_i) \cdot m \cdot \frac{Q_i}{2^i}$
 - 7: $Q_{i+1} = Q_i + (\bar{d}_i - d_i) \cdot \frac{P_i}{2^i}$
 - 8: $R_{i+1} = R_i - (\bar{d}_i - d_i) \cdot E_i$
 - 9: **end for**
 - 10: $P_{out} = P[N-1 : 0]$, $Q_{out} = Q[N-1: 0]$, $R_{out} \rightarrow 0$.
 - 11: **return** P_{out} and Q_{out} .
-

Algorithm 1 demonstrates the iterative calculations required to compute the hyperbolic functions cosh and sinh, resulting in the generation of P_{out} and Q_{out} , using the hardware architecture depicted in the ROM/ Cordic Block (Fig. 2). P_i and Q_i are calculated for N iterations until R_{out} converges to 0. The MAC output serves as the input to the AF (AF_{in}) and is denoted as R_{in} in algorithm 1.

2.2 Configurable AF Architecture with ROM/ Cordic Block

The core of the configurable AF, as depicted in Fig. 3, consists of the ROM/ Cordic Block (Fig. 2). The ROM/ Cordic Block incorporates adder/subtractors, shifters, and memory elements. The most significant bit (MSB) of $R_{in}[N-1]$ (sign bit) generates the directional signal d_i [5], determining whether addition or subtraction is performed such that R_{out} converges to 0. Here $d_i \in \{0, 1\}$ as the sign bit $R_{in}[N-1] \in \{0, 1\}$. Equation 3 has been modified to the equations presented in algorithm 1 for hardware realization. The Add/Sub block utilizes the 2's complement form for subtraction operations. The 1:2 DeMux, controlled by the `precision_select` signal, selects the input $R_{in}[N-1:0]$ for either ROM or Cordic operation. In the ROM-based approach (Fig. 2), the value at the R_{in} address is accessed as $ROM[R_{in}]$. Depending on the implementation of the ROM, the AF's output for sigmoid or tanh is obtained. Thus, $R_{out} = ROM[R_{in}]$ for the ROM-based approach, while R_{out} converges to 0 for the Cordic-based approach

(Fig. 2). Additionally, the state machine in Fig. 2 generates control signals for input and feedback based on the `clock` and `reset` signals. The output of the Cordic block in Fig. 2 produces the values $\cosh(R_{in})$ and $\sinh(R_{in})$ at P_{out} and Q_{out} , respectively. These outputs are used for exponential calculation, as described in Eq. 4.

$$e^{R_{in}} = \cosh(R_{in}) + \sinh(R_{in}) \tag{4}$$

The proposed configurable AF architecture, as shown in Fig. 3, incorporates select signals `precision_select` and `AF_select` to determine the outputs using either ROM or Cordic, as summarized in Table 1. The input data R_{in} serves as AF_{in} to the proposed block and produces the output $ROM[R_{in}]$ in the subsequent clock cycle or converges to 0 after the N^{th} Cordic iteration. The ROM/Cordic Block has three outputs: $\sinh(R_{in})$, $\cosh(R_{in})$, and $0/ ROM[R_{in}]$, as depicted in Fig. 3, with `AF_select` controlling MUX1 and MUX2 for Cordic-based sigmoid or tanh AF selection. $\sinh(R_{in})$ and $\cosh(R_{in})$ are sent to the ADDER1 block. The output of ADDER1 is $e^{R_{in}} = \sinh(R_{in}) + \cosh(R_{in})$ which is input to the ADDER2. The output of ADDER2 is $1 + e^{R_{in}}$. MUX1 has inputs $e^{R_{in}}$, $\sinh(R_{in})$, and MUX2 has inputs $\cosh(R_{in})$, $1 + e^{R_{in}}$ with the select line as `AF_select`. The outputs of MUX1 and MUX2 are processed in the divider to calculate $Cordic[R_{in}]$ for sigmoid/ tanh evaluation. Subsequently, MUX3 is used to select $ROM[R_{in}]$ or $Cordic[R_{in}]$ based on the `precision_select` signal. The state of select signals for generating tanh and sigmoid AFs using ROM/ Cordic approaches are presented in Table 1.

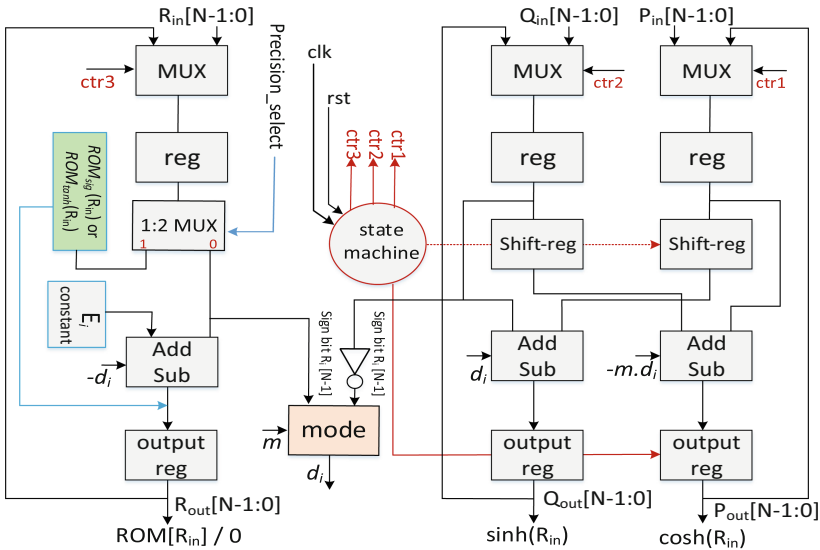


Fig. 2. ROM/Cordic Block. The `precision_select` line allows for the selection of either the $ROM[R_{in}]$ output (sigmoid or tanh) or the Cordic-based output i.e., $\sinh(R_{in})$ and $\cosh(R_{in})$

3 Performance Analysis of Proposed Configurable AF

This section presents the performance analysis of the configurable AF. The experimental setup encompasses both software and hardware implementations of the configurable AF. In the software-based accuracy evaluation, a Python implementation of the configurable AF replicates the behavior of the hardware design and is compared against the standard *TensorFlow* computation [1]. In the hardware-based evaluation, resource utilization is assessed by implementing the configurable AF using Verilog-HDL, and the corresponding parameters are extracted using the *Vivado-Xilinx* tool. The proposed design is implemented on the Zybo Evaluation Kit, with a specific focus on the sigmoid AF, which effectively utilizes all the hardware resources within the configurable architecture.

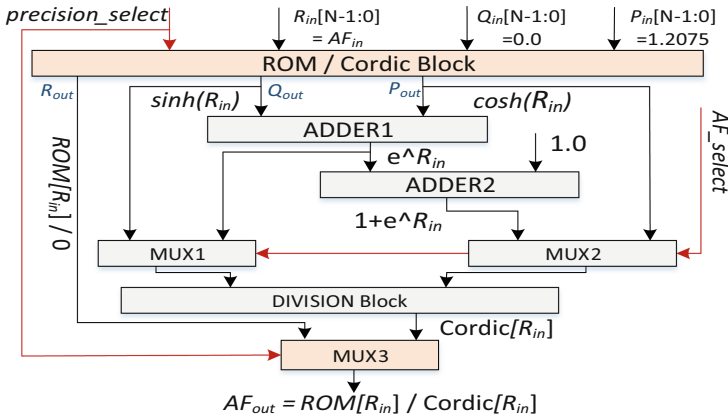


Fig. 3. The architecture of the configurable AF for variable precision, comprising of the ROM/ Cordic AF as depicted in Fig. 2 and Fig. 3

Table 1. AF selection using *AF_select* and *precision_select* signals for ROM/ Cordic AF as depicted in Fig. 2 and Fig. 3

<i>AF_select</i>	<i>precision_select</i>	R_{out}	AF_{out}
0	0	0	$Cordic[R_{in}] = \text{sigmoid}(R_{in}) = \frac{e^{R_{in}}}{1+e^{R_{in}}}$
1	0	0	$Cordic[R_{in}] = \text{tanh}(R_{in}) = \frac{\sinh(R_{in})}{\cosh(R_{in})}$
X	1	$ROM[R_{in}]$	$ROM[R_{in}] = \text{sigmoid/ tanh}$

To evaluate the accuracy of the AF, LeNET and VGG-16 [1] DNN models, along with the MNIST, CIFAR-10, and CIFAR-100 datasets are utilized. The inference accuracy results for Tensor (T) and proposed AF (P) using varying-bit

Table 2. Comparison of accuracy of proposed configurable AF (P) with Tensor-based AF model (T) [1] for sigmoid implementation on LeNET and VGG-16 DNN models

DNN Arch.	LeNET				VGG-16			
Datasets	MNIST		CIFAR-10		CIFAR-10		CIFAR-100	
Precision	T	P	T	P	T	P	T	P
Infer. Accuracy (%) for Proposed AFs with ROM								
4-bit	96.8	96.1	55.4	54.7	65.1	63.2	24.1	23.6
8-bit	98.5	97.9	64.1	62.3	82.7	81.3	50.1	48.4
16-bit	99.1	97.9	65.3	63.5	83.8	81.9	52.2	50.8
32-bit	—	—	—	—	—	—	—	—
Infer. Accuracy (%) for Proposed AFs with Cordic								
4-bit	96.8	88.3	55.4	48.6	65.1	52.7	24.1	22.8
8-bit	98.5	96.8	64.1	62.3	82.7	80.9	50.1	48.4
16-bit	99.1	97.9	65.2	64.3	86.1	84.8	55.3	54.1
32-bit	99.1	98.2	66.9	66.4	87.2	85.8	57.1	55.9

precision (4, 8, 16, and 32-bit) are presented in Table 2. This paper uses the terms ‘fixed <9, 6>’ and 8-bit precision interchangeably, as the 9th bit (MSB) is the sign bit. The same applies for all bit precisions presented in this paper. The ROM-based approach demonstrates superior accuracy for low-bit precision computation (≤ 8), while the Cordic-based approach offers considerable accuracy improvements for higher-bit precision. Across all bit widths, the proposed configurable AF achieves accuracy levels comparable to the tensor-based model [1], with an accuracy loss of less than 1.5%.

A comparative analysis of resource utilization is presented in Table 3 for ROM, Cordic, and BRAM-based approaches across different bit-precisions. For 4-bit precision, the ROM-based design employs 6 LUTs, while the Cordic-based design utilizes 45 LUTs and 37 flip-flops (FFs). This results in a notable LUT saving of 86.66% for the ROM-based design compared to Cordic. Similarly, in the case of 8-bit precision, the ROM-based design requires 16 LUTs, whereas Cordic utilizes 84 LUTs and 72 FFs, achieving a LUT saving of 80.95%. Remarkably, the ROM-based design exclusively relies on LUTs and does not require any FFs. However, as the precision increases to 16-bit, the ROM-based design demands a substantial number of 2111 LUTs, in contrast to the Cordic-based design’s requirement of 140 LUTs and 126 FFs. Furthermore, it is observed that implementing a 32-bit ROM-based design on smaller FPGAs is infeasible due to the exponential increase in resource requirements (2^N memory elements). In contrast, Cordic requires 257 LUTs and 221 FFs for 32-bit precision. Additionally, the evaluation includes the BRAM-based approach, which reveals a significant increase in BRAM utilization as precision increases. Precisely, for 4, 8, and 16-bit precisions, the corresponding BRAM requirements are 0.5, 0.5, and 17 BRAMs,

respectively. Overall, the Cordic-based technique offers better LUT utilization for higher precision computations, making it suitable for accuracy-driven applications. Additionally, leveraging pipeline architecture can enhance its throughput performance. Conversely, the ROM-based implementation of AFs demonstrates superior performance at lower precision levels. These findings are useful for selecting appropriate AF implementations based on precision requirements and resource constraints in DNN design.

Table 3. Resource Utilization of configurable AF for different bit-widths evaluated on Zybo-board.

AF Type	ROM		Cordic		BRAM
Precision	LUTs	FFs	LUTs	FFs	—
4-bit	6	0	45	37	0.5
8-bit	16	0	84	72	0.5
16-bit	2111	0	140	126	17
32-bit	—	0	257	221	—

4 Conclusions and Future Research

This paper introduces a novel approach for designing a configurable AF using ROM and Cordic architectures. The proposed AF achieves accurate inference with reduced memory requirements, catering to variable bit precision needs. Extensive evaluations on LeNet and VGG-16 DNN models, employing MNIST, CIFAR-10, and CIFAR-100 datasets, demonstrate its competitive performance with less than a 1.5% accuracy loss compared to tensor-based models. The ROM-based design excels in low-bit precision, offering high accuracy and significant LUT savings. On the other hand, for higher bit precision, the Cordic-based design outperforms the ROM-based design by leveraging the Cordic algorithm to minimize memory requirements. The suitable AFs selection contributes to reduced power usage, making it particularly advantageous for AI-enabled IoT applications. The future work for this research will involve designing a configurable neuron for variable bit-precision using the proposed AF and a configurable MAC.



References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org (2015)
2. Aggarwal, S., Meher, P.K., Khare, K.: Concept, design, and implementation of reconfigurable CORDIC. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **24**(4), 1588–1592 (2015)

3. Lee, J., et al.: Unpu: an energy-efficient deep neural network accelerator with fully variable weight bit precision. *IEEE J. Solid-State Circuits* **54**(1), 173–185 (2018)
4. Lin, C.H., Wu, A.Y.: Mixed-scaling-rotation CORDIC (MSR-CORDIC) algorithm and architecture for high-performance vector rotational DSP applications. *IEEE Trans. Circuits Syst. I Regul. Pap.* **52**(11), 2385–2396 (2005)
5. Mehra, S., Raut, G., Das, R., Vishvakarma, S.K., Biasizzo, A.: An empirical evaluation of enhanced performance softmax function in deep learning. *IEEE Access* **11**, 34912–34924 (2023)
6. Mohamed, S.M., et al.: FPGA implementation of reconfigurable CORDIC algorithm and a memristive chaotic system with transcendental nonlinearities. *IEEE Trans. Circuits Syst. I Regul. Pap.* **69**(7), 2885–2892 (2022)
7. Prashanth, H., Rao, M.: Somalib: Library of exact and approximate activation functions for hardware-efficient neural network accelerators. In: 2022 IEEE 40th International Conference on Computer Design (ICCD), pp. 746–753. IEEE (2022)
8. Raut, G., Karkun, S., Vishvakarma, S.K.: An empirical approach to enhance performance for scalable cordic-based deep neural networks. *ACM Trans. Reconfigurable Technol. Syst.* (2023)
9. Raut, G., Rai, S., Vishvakarma, S.K., Kumar, A.: RECON: resource-efficient CORDIC-based neuron architecture. *IEEE Open J. Circuits Syst.* **2**, 170–181 (2021)
10. Raut, G., et al.: Data multiplexed and hardware reused architecture for deep neural network accelerator. *Neurocomputing* **486**, 147–159 (2022)
11. Sakr, C., Kim, Y., Shanbhag, N.: Analytical guarantees on numerical precision of deep neural networks. In: International Conference on Machine Learning, pp. 3007–3016. PMLR (2017)



Authentication and Authorization of IoT Edge Devices Using Artificial Intelligence

Muhammad Sharjeel Zareen^(✉) , Shahzaib Tahir^(D) , and Baber Aslam

College of Signals, National University of Sciences and Technology,
Islamabad, Pakistan

mzar.phdismcs@student.nust.edu.pk, {shahzaib.tahir, ababer}@mcs.edu.pk

<https://mcs.nust.edu.pk/about-us/>

Abstract. The field of Internet of Things (IoT) has experienced rapid growth, but it has also introduced significant security and privacy challenges. In particular, the authentication and authorization of edge devices pose major concerns due to their limited resources. While various solutions have been proposed, most of them rely on increasing the computing power, storage, and power capabilities of edge devices. However, these solutions are not practical because of the constraints imposed by the small size and cost-effectiveness requirements of IoT edge devices. Some suggestions involve the use of lightweight cryptographic primitives, but not all edge devices have the necessary resources to implement such solutions. This paper presents a novel approach to addressing the authentication and authorization challenges in edge devices by leveraging artificial intelligence (AI). The proposed solution adopts a fog computing model within the framework of a smart home, but it does not depend on the computational or storage capabilities of the edge devices.

Keywords: Internet of Things (IoT) · Fog computing · Artificial Intelligence (AI) · Smart devices

1 Introduction

The Internet of Things (IoT) is rapidly advancing, and it is projected that by 2020, the number of connected IoT devices will reach 20.4 billion, contributing significantly to the global economy [1, 2]. Various sectors, including healthcare, living, supply chain, factories, and agriculture, are being revolutionized by IoT technologies. The current benefits of IoT are already substantial, and they are expected to grow further with the emergence of innovative technologies.

However, the widespread adoption of IoT also brings security challenges. As the number of IoT devices grows, so does the potential for malicious attacks. The IoT industry is vulnerable to various security threats, and protecting IoT systems and data has become a critical concern.

Furthermore, the rapid growth of IoT has a significant impact on internet traffic. It is projected that IoT will contribute to a compound annual growth

rate of 14.4% from 2017 to 2021 [3]. This increased consumption of internet resources further emphasizes the need for robust security measures to protect IoT infrastructure from potential attacks.

Since the late 1990s, IoT devices have been manufactured with little emphasis on security [4]. The limited resources and constrained nature of end devices present a significant obstacle to implementing robust security measures in IoT [5]. Security and privacy remain ongoing challenges in the IoT landscape [5]. Various solutions have been proposed to address these issues, however, most of them involve adding some computational power and storage in IoT devices, which are not practical. To tackle these challenges, fog computing has been introduced as a potential solution [6]. However, authentication and authorization continue to be key security issues, especially due to the resource limitations of end devices [6]. Use of Artificial Intelligence (AI) to address these challenges under fog environment has not been proposed so far. This paper focuses on addressing these authentication and authorization challenges using AI technology within a fog computing model. By leveraging AI and machine learning techniques, a proposed solution aims to enhance the security of IoT devices in a resource-efficient manner. Authors have already proposed the model in [7]. However, in this paper, specific AI techniques to be used in proposed model, have been suggested, thus taking the model a step further.

The primary contribution of this paper is the utilization of AI for authentication and authorization of edge devices, without requiring additional computational capacity, storage, or power from the edge devices. The structure of the paper is as follows: Sect. 2 presents a comprehensive literature review, discussing the research on the use of AI for authentication, authorization, and enhancing security in IoT. In Sect. 3, an AI-based framework for the authentication and authorization of end devices is proposed. In Sect. 4, implementation of AI techniques for authentication of IoT devices has been discussed. Finally, Sect. 5 concludes the paper.

2 Literature Review

Computational, storage, and end-device power limitations in the Internet of Things pose serious challenges in implementing workable solutions that address security concerns in the Internet of Things. This section describes some of the latest research in addressing end device security issues.

Roman et al. We evaluate authentication and access control issues in research and analysis of security threats and challenges for mobile edge computing [8]. They recommend studying the applicability of other edge network distributed authentication mechanisms for IoT. It has also been concluded that edge device security is in its infancy.

To tackle the limitations of resource-constrained edge devices, the concept of fog computing was introduced by Cisco in 2012 [6]. Fog computing extends the capabilities of the cloud by providing computing, storage, and networking services between cloud services and edge devices. This approach aims to address challenges such as latency, location awareness, mobility, and the large number of end devices in the cloud and IoT. However, the original fog computing model does not explicitly address the authentication issues of end devices. Subsequent studies have further refined the concept of fog computing, leading to the development of robust fog nodes that facilitate the creation of a unified infrastructure for collaboration across different fog environments [9].

Numerous solutions have been proposed that involve adding computational power and storage to edge devices. Ibrahim presented a novel authentication approach where edge devices authenticate themselves with fog servers, along with the use of basic cryptographic tools [10]. However, this solution is not feasible for all edge devices due to their limited computation and storage capabilities. Jia et al. proposed a mutual authentication mechanism for edge devices that ensures anonymity and intractability [11]. However, this solution relies on various cryptographic functions, such as elliptic curve encryption, bilinear pairing, and Diffie-Hellman algorithm, which require computational resources that may not be available on constrained edge devices.

Xiong et al. proposed a privacy-aware authentication scheme for edge devices, which utilizes cryptographic primitives such as binary pairing, hashing, and computation [12]. However, this solution requires computational power on the edge device, making it unsuitable for all IoT devices with limited resources. Chiang et al. concluded that remote attestation solutions using add-ons for authentication are not feasible for IoT due to the inability of a large number of edge devices to support such add-ons [13]. While fog computing has been proposed as a solution for resource-constrained devices, the issue of authentication has not been addressed in that solution.

Several studies have investigated the integration of AI into IoT, exploring different aspects and applications. Sezer et al. emphasized the importance of security in IoT, considering the diverse range of IoT devices and frameworks being developed [14]. Mougy et al. proposed a scalable personalized IoT network that leveraged AI for context awareness, improving mobility prediction, device duty cycle, and cognitive networking [5]. However, their research did not specifically address the authentication and authorization aspect. Similarly, Wan et al. focused on enhancing the performance of smart factories through AI, targeting areas such as flexibility, efficiency, and intelligence, but did not incorporate AI for authentication and authorization purposes in their model [15].

Several studies have addressed the security challenges in the context of IoT. For instance, Chin et al. proposed a context-aware network infrastructure that supports security, diversity, and virtualization of networks for IoT services [16]. Although their solution uses context awareness for traffic routing, it does not cover authentication of edge devices. Meanwhile, Blazek et al. proposed a device security model for IoT that includes a Central Authentication Module (CAM)

[17]. However, CAM uses hardware-based authentication, which requires costly add-ons like Raspberry Pi 3 board, RFID readers, power supply, and sensors to be installed on end devices, making it impractical for large-scale IoT implementations.

Based on the reviewed literature, it can be concluded that research on the use of AI for authentication and authorization of edge devices is limited. This paper proposes such a solution, which is detailed in Sect. 3.

3 AI Based Authentication and Authorization Model

The proposed solution in this paper is focused on the smart home scenario, where the limited computational capacity, storage, and power of edge devices make it challenging to incorporate authentication and authorization directly into them. As mentioned in Sect. 2, adding extra computation or storage to edge devices is not practical. To overcome this limitation, the paper suggests adding an additional hardware component at the fog layer in the user's premises for authentication and authorization purposes. An AI-based solution is proposed to run on this additional hardware to authenticate and authorize edge devices. Clustering techniques of unsupervised learning and classification techniques of supervised learning are recommended to be used for this purpose. The proposed solution effectively addresses the long-standing issues of authentication and authorization of edge devices in IoT. The modified architecture of Fog Computing, along with the addition of the proposed AI-based computer, is depicted in Fig. 1.

3.1 Communication/Data Exchange

In order to utilize the IoT devices of a specific manufacturer, the user is required to download and install a user application on their computer or smartphone. This application allows the user to create an account with the device manufacturer. When adding a new smart device to the smart home, the user needs to configure it for communication with its cloud-based application server and to enable remote management by the owner. The user completes this setup process through the user application. During the installation process, the smart device is configured and assigned a password to access the cloud-based application server managed by the manufacturer through a fog gateway router. The device's unique ID is then linked to the user account on the application server, allowing the user to access it using their login credentials from their computer or smart device. Occasionally, IoT devices can also be accessed through a web browser; however, the user must still provide login credentials. Each type and manufacturer of IoT device has a unique format and data structure for exchanging data.

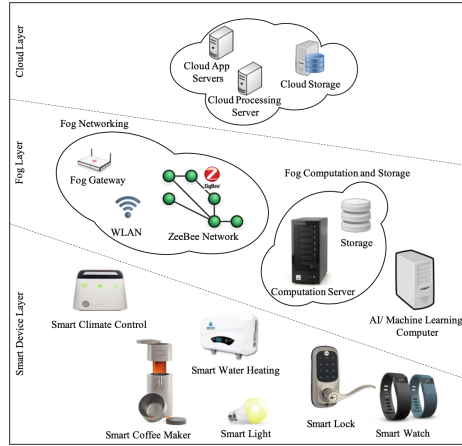


Fig. 1. Architecture of Fog Computing

Smart devices have two-way communication, where the owner can communicate with the device and the device can communicate with its owner. However, the format and structure of data exchange vary for each edge device, depending on its functionality. The data exchange process is illustrated in Fig. 2.

Smart devices rely on two-way communication between the device owner and the device itself. The specific type of data exchange varies depending on the functionality of each individual edge device. As illustrated in Fig. 2, this exchange typically involves the owner sending update requests to the device and the device responding with status notifications and other relevant information via a fog router. To understand the data exchange mechanism, few examples are discussed in succeeding paras.

For example, a smart bulb may receive update requests from its owner regarding its status, color settings, and other customizable features. In response, the bulb sends notifications to its owner through the fog router indicating its current status (on or off) and color setting (in the case of an RGB bulb), as well as other relevant information. Upon receiving these updates, the owner can then provide instructions to the bulb to modify its status, color, or other settings as needed.

Smart devices have distinct types of communication: from the owner to the smart device and from the smart device to its owner. The data exchange procedure is unique for each edge device in IoT, depending on its functionality. To illustrate the data exchange process, several examples are provided below.

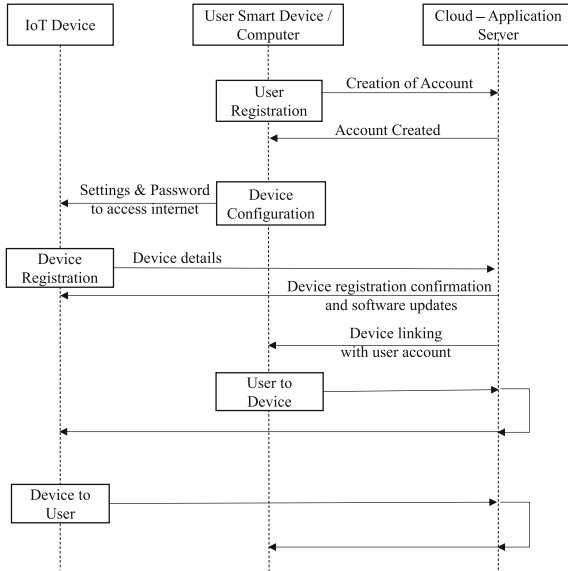


Fig. 2. Data Exchange Flowchart

For example, communication from the owner to a smart bulb involves requests for status updates, color settings, and other relevant information, according to the manufacturer’s offerings. The smart bulb communicates its status notification (on or off) and color settings (in the case of an RGB bulb), and other relevant information to its owner through a fog router. After receiving an update about the bulb, the owner can give instructions to modify the status, color, or related information/settings.

Communication from the owner to a smart water heating system involves update requests on status, current temperature, desired water temperature, and other relevant information, as per the manufacturer’s offerings (general settings triggering its auto on/off, record of its previous activities, etc.). The smart water heating system communicates its status notification, current water temperature, desired water temperature, and other relevant information to its owner. Subsequently, the owner may give instructions regarding changing the status and related settings.

Communication from the owner to a smart coffee maker involves update requests on status, supplies needed to make coffee, and other information, as per the manufacturer’s offerings (general settings of auto-making coffee, etc.). The smart coffee maker communicates its status notification, the state of water, coffee beans, and other necessary supplies needed to make coffee, and other related information to its owner. The owner may give instructions on preparing the coffee or changing related settings.

In summary, each smart edge device has its unique set of communications and data exchange procedures depending on its type and the services offered by the device manufacturer.

3.2 Use of AI in Authentication and Authorization of IoT Devices

One potential use of AI in IoT security is for labeling and associating datasets with known device profiles. By identifying characteristics unique to each edge device, such as its communication protocols and data exchange procedures, AI algorithms can authenticate devices and grant authorization based on their pre-defined profiles. An AI-based authentication model is proposed in Fig. 3, which will be discussed in more detail in the following paragraphs. The specific AI techniques used in this model will be explained in Sect. 4.

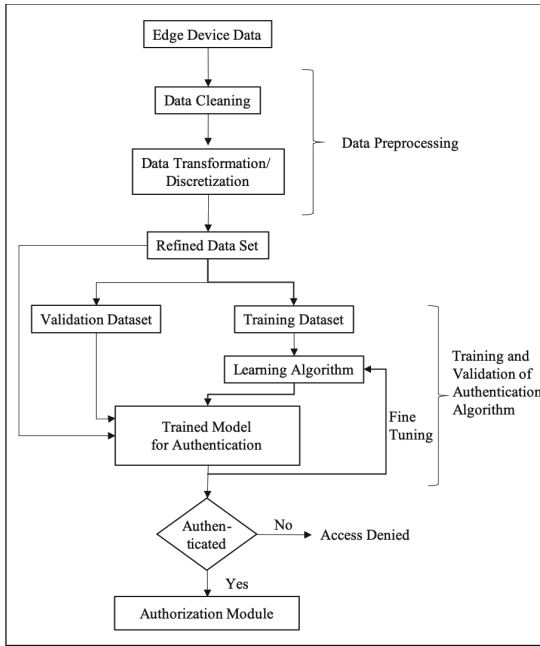


Fig. 3. Proposed AI Model for Authentication

- Data Preprocessing. The first step in the data preprocessing phase is to clean the raw data generated by the edge device [18]. This involves filling in missing data, smoothing out noise, removing outliers and inconsistencies, and other techniques to ensure that the data is accurate and consistent. However, data integration is not necessary in the proposed model, as it is assumed that the same type of IoT device (e.g., a smart bulb) from the same manufacturer

will have a similar data structure. Therefore, there is no need to combine data from different sources. In addition, data reduction is not required in the proposed model because the feature space of each type of IoT device is not too large, which makes it unnecessary to reduce it. Finally, the cleaned data will be transformed into a refined form suitable for the AI algorithm used in subsequent steps.

- Data Preprocessing. In the data preprocessing phase, the raw data generated by the edge device is refined for use in authentication with AI techniques [18]. The first step is to clean the data, which may contain errors due to wireless transmission or inconsistency in the edge device’s performance. Cleaning involves filling in missing data, smoothing noise, removing outliers and inconsistencies. Data integration is not required in the proposed model, as we assume that devices of the same type from the same manufacturer have the same communication and data exchange patterns. Data reduction is also not performed, as the feature space of each IoT type is not large enough to require reduction. The cleaned data is then transformed or consolidated into the appropriate form of mining, referred to as refined data, which is the most efficient for the AI algorithm used in subsequent steps.
- Division of Refined Data. The refined data is divided into two parts: 70% for training the learning algorithm and 30% for validating/testing the trained model. After training and validation, live input of the refined dataset will be provided to the trained model for the authentication of edge devices.
- Training and Validation of Authentication Algorithm. The training dataset is used to train the authentication algorithm. The desired output of the trained model is the ability to authenticate edge devices using clustering and classification techniques. Clustering is used to segregate the type of edge devices into respective categories, and classification is used to authenticate the edge device. The results of the validation testing are used to fine-tune the training algorithm. Once the model is trained and validated, it can be used to authenticate live input of refined dataset for edge devices.
- Authentication and Authorization using Trained Model. The trained model will be used for authentication of edge devices based on the live feed of refined dataset. If the edge device is successfully authenticated, its refined data will be forwarded to the authorization model. The authorization model will grant appropriate access to the authenticated device as defined by the user. There are numerous authorization models available in literature and research papers and hence are not discussed in this paper.

3.3 Authentication and Authorization of Device Owner

The device owner typically uses a smartphone or laptop to access their smart devices, and these devices have ample processing capability, storage, and power. Therefore, incorporating authentication and authorization of device owners in

IoT is not a significant challenge. Several solutions have been proposed in various research papers, as discussed in Sect. 2. This paper does not propose a new solution for device owner authentication and authorization, as existing solutions can be used for this purpose.

3.4 Prerequisites of Proposed Solution

The proposed model assumes that the AI computer can read data exchanged with smart edge devices. However, in some cases, edge devices use a proprietary format for data exchange, and data is encrypted to ensure privacy. To address this, one solution is to install the manufacturer's application for the smart device on the AI computer to process the data before passing it to the pre-processing stage. Another solution is to configure smart devices to allow data exchange in a generic format such as XML. However, this may compromise data privacy within the smart home. Once edge devices and device owners are authenticated and authorized through the AI computer, communication can continue in a proprietary format with encryption, but the manufacturer's application must be installed on the computer. Therefore, for both the outside world (device owner/cloud-based application server) and inside world (within the smart home), the AI computer will appear transparent as an application layer device.

4 Implementation of AI Techniques for Authentication of IoT Devices

The selection of AI techniques for the proposed model is based on the scenario of smart home, where the authentication and authorization of IoT devices are desired. The preprocessing phase includes the following AI techniques:

4.1 Preprocessing Phase

Since smart homes offer a relatively controlled environment with well-defined IoT devices that generate data with small feature space, the steps of data integration and data reduction are not required in the proposed model. The AI techniques used for the proposed steps are as follows:

- Data Cleaning. Data cleaning will be carried out in first phase. Cleaning is carried out for data which becomes dirty in shape of outliers, inconsistencies, noise and missing values. In context of smart home, wireless communication media is assumed to be main cause of making the data dirty. As data is coming live, hence “ignoring the tuple” technique will be adopted for dirty as it will have no affect on volume of data.
- Data Transformation and Discretization. Data discretization will be used to transform continuous attributes to categorical attributes to speed up the implementation of AI techniques during authentication and authorization phases. Some of the examples of continuous attributes are colour settings

and luminosity of smart bulb, temperature settings in smart water heating system, quantity of various materials in smart coffee maker. Categorical data will be converted to binary data. Example of categorical data is status of smart bulb, smart water heating system and coffee maker (ON, OFF and standby).

4.2 Authentication Phase

This phase comprises two main activities; Clustering and classification.

- Clustering. Clustering will be used to segregate the data of smart devices into different groups. K-mean clustering technique will be used to hard cluster the items to specific groups based on similarity. As data packet of each type of IoT device has unique format/structure, Jaccard Similarity will prove efficient in measuring the similarity between the items.
- Classification. Decision tree classification technique will be used to classify the items into classes (type of IoT device) as it can handle both categorical as well as numerical data. Best attribute having highest information gain will be selected. Information gain will be calculated using C4.5 algorithm.

4.3 Validation Phase

Validation of trained model will be carried out on already classified dataset comprising 30% of the data. Based on validation results, model will be fine tuned. Confusion matrix will be used to distribute classified results into True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Typical confusion matrix is shown in Fig. 4.

Predicted Actual	Yes	No
Yes	TP	FN
No	FP	TN

Fig. 4. Typical Confusion Matrix

Following measures will be calculated to validate the performance of trained model.

$$Precision = TP / (TP + FP) \quad (1)$$

Precision will return the percentage of correctly classified items are actually correct.

$$Recall = TP / (TP + FN) \quad (2)$$

Recall will return the percentage of correctly identified items of a type from total population of that item in the dataset.

$$Accuracy = (TP + TN)/(TP + FN + FP + TN) \quad (3)$$

Accuracy will return the percentage of total correctly classified items from complete dataset.

5 Conclusion

Proposed solution offers several advantages to IoT by leveraging AI. AI can help in addressing the challenges and problems that act as bottlenecks for the widespread adoption of IoT. The proposed solution can be a practical manifestation of this and can prove beneficial for IoT in general. Additionally, the solution can be extended beyond smart homes to other application areas of IoT such as smart industries, smart cities, and smart roads.

References

1. Iot connected devices to reach 20.4 billion by 2020, says gartner. <https://which-50.com/iot-connected-devices-reach-20-4-billion-2020-says-gartner/>
2. Columbus, L.: (2017) Internet of things market to reach \$267b by 2020. <https://www.forbes.com/sites/louiscolombus/2017/01/29/internet-of-things-market-to-reach-267b-by-2020/#1a073d14609b>
3. Framingham, M.: Idc forecasts worldwide spending on the internet of things to reach \$772 billion in 2018 (2017). <https://www.idc.com/getdoc.jsp?containerId=prUS43295217>
4. Zareen, M.S., Tariq, M.: Internet of things (IoT): the next paradigm shift but whats the delay? In: 17th IEEE International Multi Topic Conference 2014, pp. 143–148, December 2014
5. El-Mougy, A., Al-Shiab, I., Ibnkahla, M.: Scalable personalized IoT networks. Proc. IEEE **107**(4), 695–710 (2019)
6. Bonomi, F., Milito, R.: Fog computing and its role in the internet of things. In: Proceedings of the MCC Workshop on Mobile Cloud Computing, August 2012
7. Zareen, M.S., Tahir, S., Akhlaq, M., Aslam, B.: Artificial intelligence/machine learning in IoT for authentication and authorization of edge devices. In: 2019 International Conference on Applied and Engineering Mathematics (ICAEM), pp. 220–224. IEEE (2019)
8. Roman, R., Lopez, J., Mambo, M., Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Gener. Comput. Syst. **78**, 680–698 (2018). <https://www.sciencedirect.com/science/article/pii/S0167739X16305635>
9. Iorga, M., Feldman, L., Barton, R., Martin, M.J., Goren, N., Mahmoudi, C.: Nist special publication 500–325 - fog computing conceptual model (2018) . <https://doi.org/10.6028/NIST.SP.500-325>
10. Ibrahim, M.: Octopus: An edge-fog mutual authentication scheme. Int. J. Netw. Secur. **18**, 1089–1101 (2016)

11. Jia, X., He, D., Kumar, N., Choo, K.K.R.: A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. *IEEE Syst. J.* **14**(1), 560–571 (2019)
12. Xiong, L., Peng, D., Peng, T., Liang, H.: An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services. *KSII Trans. Internet Inf. Syst.* **11**, 6169–6187 (2017)
13. Chiang, M., Zhang, T.: Fog and IoT: an overview of research opportunities. *IEEE Internet Things J.* **3**(6), 854–864 (2016)
14. Sezer, O.B., Dogdu, E., Ozbayoglu, A.M.: Context-aware computing, learning, and big data in internet of things: a survey. *IEEE Internet Things J.* **5**(1), 1–27 (2018)
15. Wan, J., Jun, Y., Zhongren, W., Qingsong, H.: Artificial intelligence for cloud-assisted smart factory. *IEEE Access* **6**, 55419–55430 (2018)
16. Chin, W.S., soo Kim, H., Heo, Y.J., Jang, J.W.: A context-based future network infrastructure for IoT services. In: *Procedia Computer Science*, vol. 56, pp. 266–270, 2015, the 10th International Conference on Future Networks and Communications (FNC 2015)/The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015) Affiliated Workshops. <https://www.sciencedirect.com/science/article/pii/S1877050915016889>
17. Blazek, P., Krejcar, O., Jun, D., Kuca, K.: Device security implementation model based on internet of things for a laboratory environment. *IFAC-PapersOnLine* **49**(25), 419–424 (2016). 14th IFAC Conference on Programmable Devices and Embedded Systems PDES 2016. <https://www.sciencedirect.com/science/article/pii/S2405896316327240>
18. Han, J., Kamber, M., Pei, J.: *Data Mining: Concepts and Techniques*, 3rd ed. 225 Wyman Street, Waltham, MA 02451, Elsevier, USA (2012)



Secure Dynamic PUF for IoT Security

Shailesh Rajput and Jaya Dofe^(✉)

California State University, Fullerton, CA, USA

shaileshrajput@csu.fullerton.edu, jdofe@fullerton.edu

Abstract. This student research forum paper is based on our accepted work [1]. The widespread adoption of the Internet of Things (IoT) has brought many benefits to our lives. Still, the low-power, heterogeneous, and resource-constrained nature of IoT devices makes it difficult to ensure secure communication and authenticity. Physical Unclonable Functions (PUFs) provide a promising solution by generating a unique and device-specific identity through manufacturing process variations without requiring additional resources. However, recent advances in machine learning algorithms like artificial neural networks and logistic regression have made it possible to predict PUF responses by training the model. Machine learning models can use multiple challenges and responses to predict accurate results from the PUF. To address this concern, we propose integrating a dynamically configurable PUF structure into the design to counteract machine learning attacks. The dynamicity of the PUF makes it challenging for machine learning models to predict PUF responses.

Keywords: Hardware Security · Physical Unclonable Functions (PUF) · Machine Learning Attacks · Dynamic PUF

1 Introduction

In the era of digital advancements, guaranteeing the security and integrity of Internet of Things (IoT) devices and data is of paramount importance. The need for safeguarding IoT devices has become critical, with sensitive information being shared online, such as in the banking and defense sectors. These devices, including monitoring systems, automated cars, medical equipment, home automation, and smart infrastructure, facilitate data transmission over the Internet. Many IoT devices verify an individual's identity and may store personal, social, and banking data. Consequently, any compromise in the security of these devices can result in significant harm. Cryptography secures critical information by encrypting and decryption of data by using cryptography keys. Cryptography and authentication protocols rely on secure key storage in nonvolatile electrically erasable programmable read-only memory and static random-access memory. This approach has significant resource and area overhead and is susceptible to invasive attacks such as Side-Channel Attacks (SCA) and non-invasive attacks

proposed in [2]. A compromised cryptographic key can jeopardize the authentication process of a device or user, potentially leading to the exposure of critical information. Furthermore, these devices can be breached due to the widespread deployment of IoT devices in public access areas. Storing a key identifier in a device helps identify the device. However, there is a need to enhance privacy, authentication, and authorization methods to address these security issues effectively. One proposed solution is the use of Physical Unclonable Functions (PUFs) as a more secure and cost-effective alternative to conventional key storage methods for device authentication [3,4]. PUFs leverage the inherent manufacturing variations in nanoscale Integrated Circuits (ICs), making them nearly impossible to replicate even by the original IC manufacturer. These unique variations produce distinct challenge-response pairs (CRPs) used in device authentication. It was assumed that the CRPs were unpredictable [5–8] and unknown to potential attackers. However, unfortunately, the advancement of machine learning algorithms/models such as Neural Networks, SVM, and Random Forest has led to the cracking of PUF responses through model training [9–11]. Therefore, before considering PUFs as a trusted security feature in the IoT paradigm, it is crucial to evaluate their resilience against machine learning modeling attacks.

To address the challenge of machine learning modeling attacks on PUFs, previous solutions have relied on software-based dynamic behavior [12]. To address this limitation, we propose a dynamic PUF that proves resilient to modeling attacks compared to the other PUF architectures, including 4-XOR and Arbiter PUF (APUF).

2 Background

2.1 PUF Designs

Arbiter PUF (APUF) is the most researched PUF because of its easy implementation and ability to produce more CRPs. APUF is derived from racing conditions between two identical paths. Depending on the time the Multiplexer chain takes, the flipflop used as arbiter at the end triggers 1 or 0. The APUF design demands a completely identical path for race conditions to avoid biased responses due to delay differences added by wires. The Arbiter PUF generates a random response that is not predictable, and due to manufacturing variation, the adversary can't clone the APUF. This property of PUF makes it secure against side-channel attacks. However, several modeling attacks have been proposed earlier to predict the PUF response [13]. Several complex architectures of PUFs, such as XOR PUF, Double Arbiter PUF(DA PUF), and Anderson PUFs, have been proposed. Despite the complex architecture, machine learning-based modeling attacks can clone the behavior of PUF.

To enhance the resilience of the arbiter PUF, Suh and Devadas proposed using an XOR PUF [14]. This approach involves merging several arbiter PUFs, with the response from each arbiter being XORed to produce the output response of the XOR PUF. While this introduces nonlinearity to the design, it also

increases system complexity. Despite this additional complexity, machine learning techniques have been shown to predict the response of the XOR PUF [15].

2.2 Machine Learning Attacks

In this work, we consider two prominent attack models on PUF—logistic regression (LR) model and multi-layer perception (MLP). LR is a robust statistical tool that effectively identifies patterns and relationships between input data and output labels. This is especially true in the context of PUF, where the CRPs do not display linear relationships. MLP modeling attacks exploit manufacturing variations by training a multilayer neural network to predict PUF responses. The trained MLP model can then replicate or predict responses for unauthorized access or cloning purposes. Despite the complexity of PUF architectures, LR and MLP modeling attacks have demonstrated robustness in accurately predicting PUF response bits, posing a significant threat to the security and unclonable nature of PUFs.

3 Experimental Analysis

3.1 Experimental Setup

The APUF architecture was implemented on the Xilinx Artix 7 100T board using the Xilinx Vivado tool, and all challenges are captured at room temperature. Thousands of CRPs are required to obtain performance metrics. As shown in Fig. 1, a control unit is designed to generate random challenge bits using Linear Feedback Shift Register (LFSR) and pass them to the APUF. The control unit consists of three main modules: LFSR, PUF, and RAM, which generate a random response bit based on their delay characteristics. Xilinx’s Integrated Logic Analyzer (ILA) tool captures approximately 131K randomly generated responses from the PUF and LFSR-generated challenge bits in one round. The ILA tool made debugging the APUF, real-time designing, and recording CRPs easier. Floor planning is performed using the Xilinx Vivado tool to ensure accurate response pairs for the APUF. The Vivado tool automatically redesigns the synthesis according to behavioral logic. However, the PUF functionality cannot be distinguished by behavioral logic as, ideally, the circuit output should be the same for the identical path. It is necessary to ensure that the auto-synthesis tool of Vivado does not change the path design of PUF according to behavioral logic. To avoid auto-optimization by Vivado DONT_TOUCH attribute is used. The APUF and its variants can generate a higher number of CRPs due to the switching of the cross-coupled and parallel paths due to the select line of the multiplexer. The implementation of this multiplexer is depicted in Fig. 2 more precisely.

Approximately 1 Million CRPs are recorded to evaluate the resiliency of modeling attacks against PUF. The RAM block is used to provide identical challenge bits multiple times to evaluate the performance of the PUF.

3.2 Proposed Dynamic Arbiter Skip APUF (DPUF)

We propose Arbiter-Skip Dynamic PUF (DPUF) that effectively mitigates the modeling attack mentioned earlier. The proposed DPUF incorporates an additional chain of APUFs with fewer multiplexer units alongside the main chain of multiplexers, as illustrated in Fig. 3. The challenge bits are selected from the intermediate response generated by the arbiter PUF, which is then XORed with one of the final inputs to introduce bias into the result. Having fewer multiplexers in the additional arbiter skip module makes the results biased when the intermediate response is 1. On the other hand, if the intermediate response is 0, no bias is introduced, and the PUF functions like a standard arbiter PUF. When the intermediate response bit one is obtained from the Arbiter skip chain, assuming that path 1 (as shown in Fig. 3) has a longer delay, it will be biased towards an active high state instead of zero and vice versa. This intermediate response can

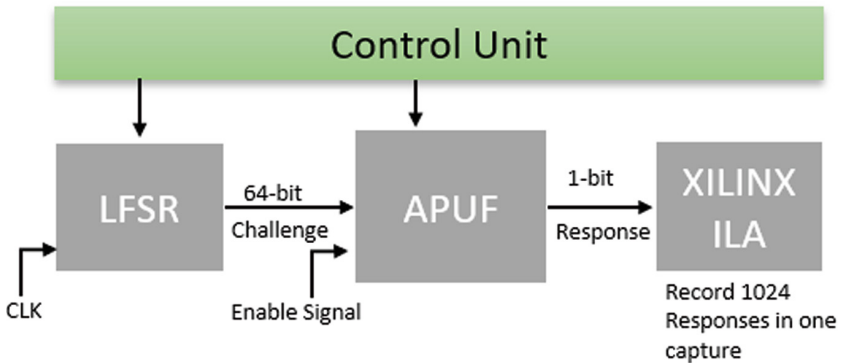


Fig. 1. Block Diagram of Control Unit.

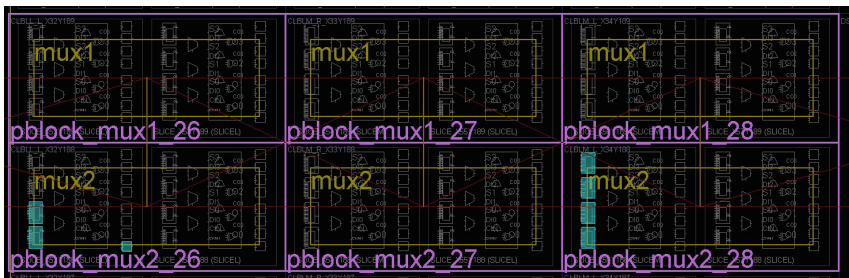


Fig. 2. Implementation of Multiplexer Switch on Artix-7 FPGA

be XORed with path 2 for multiple response bits to alter the pattern of multiple responses. Assuming the randomness of the PUF in generating an equal distribution of 50% 1's and 0's for both the intermediate and main arbiter PUF, approximately 12–15% of the final response will be biased, resulting in a pattern that thwarts various modeling attacks.

4 Results and Discussion

First, we evaluate the resilience of the proposed DPUF against the most successful modeling attacks on PUF functions—MLP and LR and compare it with APUF. Table 1 shows the prediction accuracy of these attacks on the existing APUF and proposed DPUF design for 10k, 100k, 500k, and 1 Million CRPs. The accuracy is determined by applying the formula: (Number of correct predictions / Total number of predictions). We observed that the maximum accuracy for the MLP and LR attack on DPUF was 81.11% and 77.97%, respectively, for 1 Million CRPs, whereas that of APUF was over 98%.

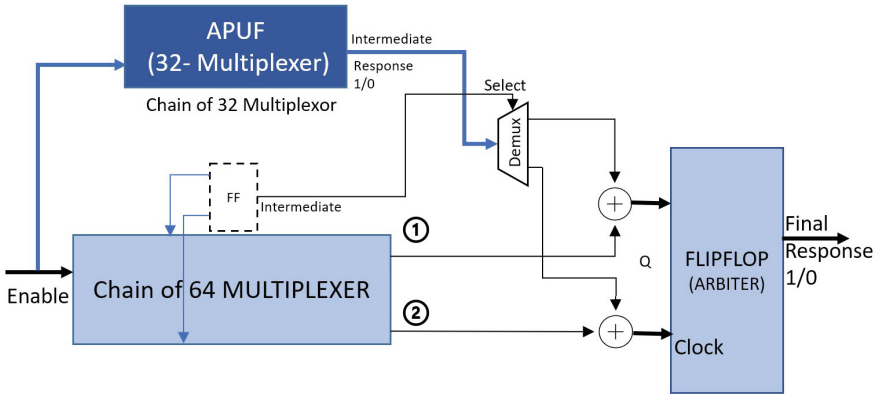


Fig. 3. Proposed Dynamic Arbiter Skip APUF

Table 1. Prediction of Proposed DPUF with APUF on Test Data for MLP and LR Attack Models.

CRPs	MLP		LR	
	APUF	DPUF	APUF	DPUF
10k	72.90 %	64.00 %	88.00 %	51.00 %
100k	88.40 %	79.99 %	94.61 %	75.20 %
500K	98.28 %	81.09 %	95.22 %	75.02 %
1M	98.99 %	81.11 %	98.36 %	77.97 %

Next, we analyze the impact of the MLP attack on the DPUF for 10k, 100k, and 1 Million CRPs of 4-XOR, 5-XOR, and 6-XOR with DPUF, as depicted in Fig. 4. The figure shows that the prediction accuracy for all PUF designs, except 6-XOR PUF, was around 60% for 10k challenges, with 6-XOR PUF having a prediction accuracy of around 50%. When trained on 100k CRPs, the model predicted the 4 & 5-XOR variants with over 95% accuracy, DPUF at around 78%, while 6-XOR remained close to 55%. However, when the MLP model was trained with 1 Million CRPs, it successfully predicted the response of all mentioned XOR variants with over 95% accuracy. Nonetheless, DPUF's accuracy remained close to 80%, even with one Million CRPs. This demonstrates that DPUF poses a more significant challenge to MLP-based algorithms, making it more difficult to predict response bits than XOR PUFs. Finally, we compare the proposed PUF's resilience with n-XOR PUF against the LR attack in Fig. 5. The results reveal that DPUF outperforms 4-XOR PUF in terms of resistance to LR attack. However, 5 and 6-XOR PUFs exhibited greater immunity toward LR attack than the proposed dynamic PUF.

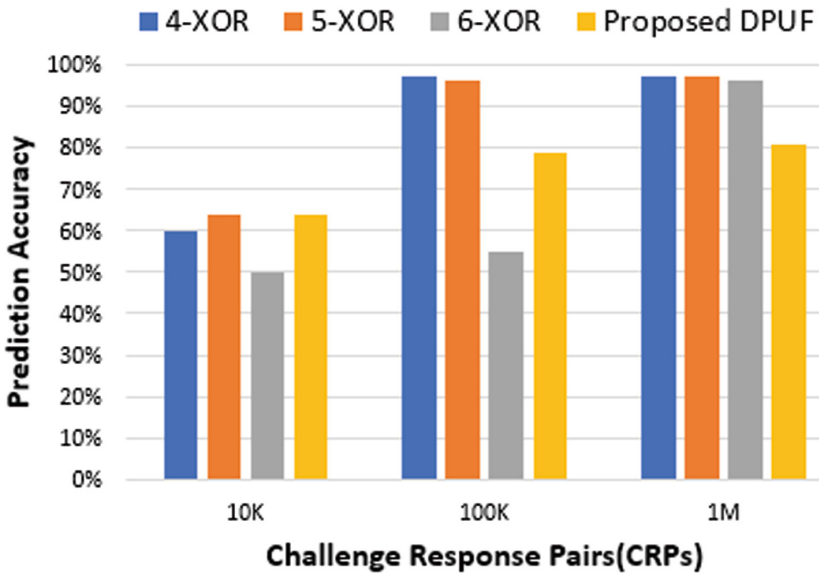


Fig. 4. Comparison of Prediction Accuracy on XOR and Proposed PUF using MLP Attack.

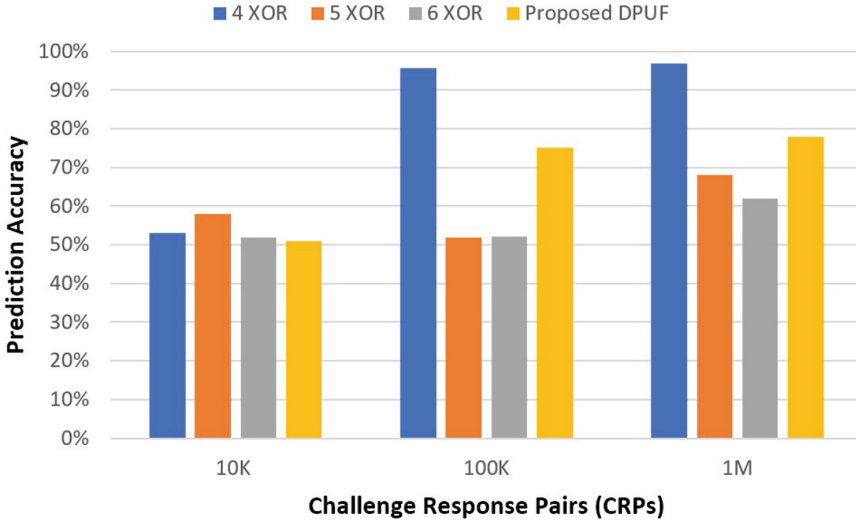


Fig. 5. Prediction Accuracy of XOR and Proposed PUF using LR Attack.

The analysis clearly showed that the proposed DPUF offers significantly higher resilience against both types of modeling attacks compared to the APUF and 4-XOR PUF. Additionally, it is worth mentioning that the proposed PUF requires significantly fewer resources compared to previously suggested complex PUF architectures.

5 Conclusion and Future Work

Physical Unclonable Functions (PUFs) are commonly used to generate unique cryptographic keys that rely on the manufacturing variation of integrated circuits, producing a unique challenge-response pair (CRP) to authenticate and identify devices. However, with the advancement of machine learning algorithms, predicting response bits generated by PUF devices has become possible, undermining their unclonable nature. This research paper presents dynamic PUF as a countermeasure against machine learning modeling attacks. While previous approaches have primarily relied on adding dynamic behavior through software architecture, our research proposes a hardware-based Dynamic PUF as a promising alternative. Among the demonstrated attacks, the MLP attack outperformed LR based modeling attack in terms of training time and prediction accuracy for all PUF designs. The accuracy of predicting one-bit response using MLP or LR attacks was around 96% on the previously proposed complex PUF architectures. However, the accuracy results of DPUF show that attack accuracy has decreased to 81.1% for MLP attacks and 77.97% for LR attacks on DPUF.

We will comprehensively evaluate the proposed PUF architecture's quality metrics in future research. This evaluation will involve multiple devices under

various temperature conditions to assess the uniqueness property of different FPGA devices. Furthermore, we will explore the integration of alternative PUF architectures that can be investigated to introduce reliable noise, increase complexity, and optimize resource utilization.

References

1. Rajput, S., Dofe, J.: Counteracting modeling attacks using hardware-based dynamic physical unclonable function. In: 2023 IEEE International Conference on Cyber Security and Resilience (CSR), Venice, Italy, pp. 586–591 (2023). <https://doi.org/10.1109/CSR57506.2023.10224914>
2. Anderson, R., Kuhn, M.: Tamper resistance - a cautionary note new. In: 2nd USENIX Workshop on Electronic Commerce (EC 96). Oakland, CA, USENIX Association, November 1996. <https://www.usenix.org/conference/2nd-usenix-workshop-electronic-commerce/tamper-resistance-cautionary-note>
3. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Silicon physical random functions. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, Ser. CCS 2002. New York, NY, USA, Association for Computing Machinery, pp. 148–160 (2002). <https://doi.org/10.1145/586110.586132>
4. Yu, M.-D., Sowell, R., Singh, A., M’Raihi, D., Devadas, S.: Performance metrics and empirical results of a PUF cryptographic key generation ASIC. *IEEE Int. Symp. Hardware-Oriented Secur. Trust* **2012**, 108–115 (2012)
5. Khalfaoui, S., Leneutre, J., Villard, A., Gazeau, I., Ma, J., Urien, P.: Security analysis of machine learning-based PUR enrollment protocols: a review. *Sensors* **21**, 8415 (2021)
6. Strieder, E., Frisch, C., Pehl, M.: Machine learning of physical unclonable functions using helper data - revealing a pitfall in the fuzzy commitment scheme, *Cryptology ePrint Archive*, Paper 2020/888 (2020). <https://eprint.iacr.org/2020/888>
7. Tobisch, J., Aghaie, A., Becker, G.T.: Combining optimization objectives: new modeling attacks on strong PUFs. *IACR Trans. Cryptographic Hardware Embedded Syst.* **2021**(2), 357–389 (2021). <https://tches.iacr.org/index.php/TCHES/article/view/8798>
8. Santikellur, P., Bhattacharyay, A., Chakraborty, R.S.: Deep learning based model building attacks on arbiter PUF compositions. *IACR Cryptol. ePrint Arch.* **2019**, 566 (2019)
9. Wisiol, N., et al.: Splitting the interpose PUF: a novel modeling attack strategy. *Cryptology ePrint Archive*, Paper 2019/1473, 2019, <https://eprint.iacr.org/2019/1473>
10. Li, G., Mursi, K.T.: A subspace pre-learning strategy to break the interpose PUF. *Electronics* **11**(7) (2022). <https://www.mdpi.com/2079-9292/11/7/1049>
11. Zhuang, H., Xi, X., Sun, N., Orshansky, M.: A strong subthreshold current array PUF resilient to machine learning attacks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **67**(1), 135–144 (2020)
12. Xiong, W., Schaller, A., Katzenbeisser, S., Szefer, J.: Dynamic physically unclonable functions. In: Proceedings of the 2019 on Great Lakes Symposium on VLSI, ser. GLSVLSI 2019. New York, NY, USA, Association for Computing Machinery, pp. 311–314 (2019). <https://doi.org/10.1145/3299874.3318025>
13. Rührmair, U., et al.: PUF modeling attacks on simulated and silicon data. *IEEE Trans. Inf. Forensics Secur.* **8**(11), 1876–1891 (2013)

14. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: 2007 44th ACM/IEEE Design Automation Conference, pp. 9–14 (2007)
15. Tobisch, J., Becker, G.T.: On the scaling of machine learning attacks on PUFs with application to noise bifurcation. In: Mangard, S., Schaumont, P. (eds.) RFIDSec 2015. LNCS, vol. 9440, pp. 17–31. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24837-0_2

Author Index

A

Abdalla, Hassan II-87, II-101
Abdelgawad, Ahmed II-246
Abishek, M. K. I-306
Agarwal, Swaraj II-402
Ahmad, Riyaz II-396
Ahmed, Samir I-199, II-410
Al Neyadi, Darwish II-227
Alruwaili, Fawaz J. I-424
Alruwaill, Musharraf II-381
Amsaad, Fathi I-97, II-347, II-357
Anitha, S. I-362
Arun Kumar, R. I-362
Aslam, Baber I-442

B

Baba, Asif I. II-198
Baligodugula, Vishnu Vardhan I-97, II-357
Banerjee, Sourav I-243
Bapatla, Anand Kumar II-283, II-381
Barik, Sudip I-243
Bhargavi, Gouravajjula Lakshmi Sai II-149
Bhatta, Niraj Prasad II-347
Bhattacharyya, Shuvra S. I-182
Bhunia, Suman I-272
Bipin, K. C. II-390
Boolchandani, Dharmendar II-396

C

Candell, Richard I-182
Charan, Vandavasi Satya II-3
Chatterjee, Pushpita II-309
Chinara, Suchismita II-36
Chishti, Mohammad Ahsan II-198
Choi, Baek-Young I-48
Cruz, Meenalosini V. I-353

D

Damiani, Ernesto II-227
Das, Debashis I-243, II-309
Dascalu, Sergiu M. I-62

De Nardi, Adriel Monti I-18
Dhakad, Narendra Singh I-433
Dhibar, Kunal I-261
Diviya, M. I-293
Dockendorf, Catherine II-371
Dofe, Jaya I-454
Dutta, Joy II-227

E

Egala, Bhaskara Santhosh II-114

F

Feil-Seifer, David I-62

G

Gawali, Shubhangi K. I-218
Geng, Jing I-182
Ghai, Dhruva I-433
Ghimire, Ashutosh II-347, II-357
Ghosal, Prasun II-74
Ghosh, Uttam I-243, II-309
Gnanesh, Divi II-149
Goveas, Neena I-218
Gudino, Lucy J. I-218
Gupta, Shubham II-114

H

Hamza-Lup, Felix G. I-353
Harris Jr., Frederick C. I-62
Harshagnan, Koganti I-293
Harshavardhan, B. V. I-316
Ho, Samuel I-337
Hossain, Al Amin II-347
Hua Tsai, I I-135

I

Islam, Md Tajul I-48

J

Jain, Prem Chand II-3
 Jamali, Hossein I-62
 Jean, Marc I-151
 John, Marvel M. I-362
 Joshi, Amit Mahesh II-396
 Joshi, Rajeev I-34, I-76, I-168

K

Kalimuthu, Sivakumar I-293, I-306, I-316
 Kalyanam, Lakshmi Kavya I-34, I-76
 Karabiyik, Umit I-337
 Karam, Robert I-199, II-410
 Karthikeyan, S. I-362
 Kashef, Mohamed I-182
 Katkooori, Srinivas I-34, I-76, I-168
 Keller, Myles I-199
 Kethineni, Kiran Kumar I-415
 Khatik, Sadhvi II-212
 Kougianos, Elias I-375, I-405, I-415, I-424,
 II-283, II-371, II-381

L

Lakshmanan, An Sakthi I-306
 Liu, Ying I-3, II-14, II-179, II-333

M

Mahapatra, Kamalakanta I-229
 Mahmud, Md Ishtyaq II-246
 Mahmud, Shakil I-199, II-410
 Maji, Prasenjit I-261
 Mandke, Pranav I-104
 Mishra, Alekha Kumar II-212
 Mitra, Alakananda I-405, I-415, II-371
 Mohanty, Prajnyajit I-229
 Mohanty, Saraju P. I-375, I-405, I-415,
 I-424, II-283, II-371, II-381
 Mondal, Hemanta Kumar I-261
 Mondal, Rishabh II-74
 Monteiro, Maxwell Eduardo I-18
 Morshed, Bashir I. I-135
 Mukhopadhyay, Debajyoti I-104, II-64
 Muresan, Anca O. I-353
 Musale, Vinayak I-104

N

Naga Nithin, G. II-133, II-160
 Naik, Varsha II-64
 Narayan, Panigrahi II-49

Narayanan, Varun I-114
 Negi, Vipul Singh II-36
 Nithin, G. Naga II-149

O

Owen, Kylie II-302

P

Panda, Satyajit II-36
 Panigarhi, Narayan II-402
 Pati, Umesh Chandra I-229
 Patra, Jayaprakash II-36
 Patra, Ramapati I-261
 Perry, Nicholas I-272
 Perumal, Thinaganan I-316
 Pradhan, Ashok Kumar II-114, II-133
 Puthal, Deepak II-212, II-227

Q

Qi, Xiaowen I-182

R

Rachakonda, Laavanya I-395, II-302, II-390
 Rajesh, M. A. II-402
 Rajput, Shailesh I-454
 Ramanujam, E. I-316
 Raut, Gopal I-433
 Roy, Swapnoneel I-104, II-64

S

Sadhu, Pintu Kumar II-246, II-262
 Sahith, Chitumalla II-3
 Sankaran, Sriram I-114
 Shah, Iqra Amin II-198
 Shah, Sneha II-64
 Shill, Ponkoj Chandra I-62
 Shillingford, Nadine II-309
 Shivam, Pant II-49
 Sibi Chakkaravarthy, S. I-362
 Siddiqui, Arish II-87, II-101
 Singh, Aniket I-104
 Singh, Harshdeep I-97
 Singh, Rohit II-3
 Somesula, Raaga Sai I-168
 Song, Sejun I-48
 Stasiewicz, Samuel I-395
 Swain, Gandharba II-133

T

Tahir, Shahzaib [I-442](#)
Tansen, Kazi [II-87, II-101](#)

U

Utsha, Ucchwas Talukder [I-135](#)

V

Vangipuram, Sukrutha L. T. [I-375](#)
Vidhyasagar, B. S. [I-293, I-306](#)
Vimal Cruz, Meenalosini [I-362](#)

Vishvakarma, Santosh Kumar [I-433](#)
Vishwakarma, Sudheer [I-433](#)

Y

Yanambaka, Venkata P. [II-246, II-262](#)
Yuksel, Murat [I-151](#)

Z

Zambare, Pallavi [I-3, II-14, II-179, II-333](#)
Zareen, Muhammad Sharjeel [I-442](#)