# Nikhil Sharma_2k21_phdco_04_Thesis.pdf

Delhi Technological University

## Document Details

Submission ID
trn:oid:::27535:101556471

Submission Date
Jun 19, 2025, 3:26 AM GMT+5:30

Download Date
Jun 19, 2025, 3:35 AM GMT+5:30

File Name
Nikhil Sharma_2k21_phdco_04_Thesis.pdf

File Size
8.8 MB

248 Pages

90,073 Words

507,503 Characters

# 4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

## Filtered from the Report

- Bibliography
- Cited Text
- Small Matches (less than 10 words)

## Exclusions

- 5 Excluded Sources

## Match Groups

**259** Not Cited or Quoted 4%
Matches with neither in-text citation nor quotation marks

**0** Missing Quotations 0%
Matches that are still very similar to source material

**1** Missing Citation 0%
Matches that have quotation marks, but no in-text citation

**0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

## Top Sources

1% Internet sources
3% Publications
1% Submitted works (Student Papers)

## Integrity Flags

**0 Integrity Flags for Review**

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

## Match Groups

- **259** Not Cited or Quoted 4%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **1** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

## Top Sources

1% Internet sources
3% Publications
1% Submitted works (Student Papers)

## Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | | |
|---|---|---|---|
| 1 | Internet | eitca.org | <1% |
| 2 | Publication | Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ... | <1% |
| 3 | Publication | Prashant Giridhar Shambharkar, Nikhil Sharma. "Artificial Intelligence driven Intr... | <1% |
| 4 | Publication | Deb, Dipok. "Application and Analysis of Machine Learning and Deep Learning Al... | <1% |
| 5 | Publication | Pramod R. Gunjal, Satish R. Jondhale, Jaime Lloret, Karishma Agrawal. "Internet o... | <1% |
| 6 | Publication | Ghita Lazrek, Kaouthar Chetioui, Younes Balboul, Said Mazer, Moulhime El bekkal... | <1% |
| 7 | Publication | "Data Science and Applications", Springer Science and Business Media LLC, 2025 | <1% |
| 8 | Publication | "Innovations in Electrical and Electronics Engineering", Springer Science and Busi... | <1% |
| 9 | Publication | "Engineering Applications of Neural Networks", Springer Science and Business M... | <1% |
| 10 | Publication | Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023 | <1% |

11 Publication
"Computational Intelligence in Pattern Recognition", Springer Science and Busin... <1%

12 Publication
H L Gururaj, Francesco Flammini, V Ravi Kumar, N S Prema. "Recent Trends in He... <1%

13 Internet
www.mdpi.com <1%

14 Publication
"Innovations and Advances in Cognitive Systems", Springer Science and Business ... <1%

15 Publication
Anuj Kumar Singh, Sachin Kumar. "Security, Privacy, and Trust in WBANs and E-H... <1%

16 Submitted works
Staffordshire University on 2024-09-16 <1%

17 Submitted works
British University in Egypt on 2024-12-07 <1%

18 Publication
Shalli Rani, Ashu Taneja. "WSN and IoT - An Integrated Approach for Smart Applic... <1%

19 Publication
"Blockchain for Biomedical Research and Healthcare", Springer Science and Busin... <1%

20 Publication
"Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologi... <1%

21 Publication
Singh, Shekhar. "Facial Expression Recognition Using Convolutional Neural Netw... <1%

22 Publication
V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challeng... <1%

23 Publication
Alemu, Shegaw Tiruneh. "A Machine Learning Intrusion Detection System (IDS) T... <1%

24 Publication
Ghubaish, Ali Hussain A.. "A Distributed and Hybrid AI-Based Security Framework ... <1%

25 Publication

Kotoklo, Agbessi. "Enhancing Cyber Threat Detection on the Internet of Healthca... <1%

26 Publication

Niknami, Nadia. "Improving Performance of Intrusion Detection Systems for Soft... <1%

27 Publication

Anurag Tiwari, Manuj Darbari. "Emerging Trends in Computer Science and Its Ap... <1%

28 Submitted works

Eastern Institute of Technology on 2024-09-04 <1%

29 Publication

Pankaj Bhambri, A. Jose Anand. "Handbook of AI-Driven Threat Detection and Pre... <1%

30 Submitted works

Eastern Institute of Technology on 2024-09-03 <1%

31 Publication

H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Co... <1%

32 Publication

Mireya Lucia Hernandez-Jaimes, Alfonso Martinez-Cruz, Kelsey Alejandra Ramírez... <1%

33 Internet

link.springer.com <1%

34 Publication

Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Intelli... <1%

35 Publication

Zaydi Mounia, Maleh Yassine, Gabriel Chênevert, Hayat Zaydi, Amina El Yaagoubi.... <1%

36 Publication

Ben Othman Soufiene, Saurav Mallik, Abdulatif Alabdulatif. "Using Blockchain Tec... <1%

37 Submitted works

Delhi Technological University on 2025-05-16 <1%

38 Submitted works

The University of the West of Scotland on 2024-08-04 <1%

39 Submitted works
Letterkenny Institute of Technology on 2025-04-23 <1%

40 Publication
Gangisetti, Durgaprasad Venkata. "Securing Critical IoT/IIoT Infrastructures: An I... <1%

41 Submitted works
University of Hertfordshire on 2025-04-28 <1%

42 Submitted works
universititeknologimara on 2025-04-17 <1%

43 Internet
upcommons.upc.edu <1%

44 Publication
Mvita, Meta Jonathan. "Optimisation of the Production of Sodium Dichromate Sal... <1%

45 Publication
R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P... <1%

46 Submitted works
The Robert Gordon University on 2024-08-22 <1%

47 Internet
dokumen.pub <1%

48 Publication
"Proceedings of Third International Conference on Computing and Communicati... <1%

49 Publication
"Proceedings of the Fifth International Conference on Trends in Computational a... <1%

50 Publication
"Securing the Connected World", Springer Science and Business Media LLC, 2025 <1%

51 Publication
Inam Ullah Khan, Salma El Hajjami, Mariya Ouaissa, Salwa Belaqziz, Tarandeep Ka... <1%

52 Publication
Payal Khurana Batra, Pawan Singh Mehra, Sudeep Tanwar. "Network Optimizatio... <1%

53 Publication

S.J. Xavier Savarimuthu, Sivakannan Subramani, Alex Noel Joseph Raj. "Artificial I... <1%

54 Submitted works

Sunway Education Group on 2023-12-05 <1%

55 Internet

download.bibis.ir <1%

56 Publication

"Artificial Intelligence, Data Science and Applications", Springer Science and Busi... <1%

57 Publication

"Intelligent Data Communication Technologies and Internet of Things", Springer ... <1%

58 Publication

"Intelligent Systems Design and Applications", Springer Science and Business Me... <1%

59 Publication

Alghamdi, Rubayyi. "Intrusion Detection System for Internet of Things with Deep ... <1%

60 Submitted works

Glyndwr University on 2023-08-21 <1%

61 Publication

Javaid Iqbal, Alwi M. Bamhdi, Bilal Ahmad Pandow, Faheem Syeed Masoodi. "Appl... <1%

62 Internet

www.scpe.org <1%

63 Publication

Aaron Zimba, Katongo Ongani Phiri, Mwenge Mulenga, George Mukupa. "Blockch... <1%

64 Publication

Chithaluri, Suryapunith. "Advancements and Challenges in the Intersection of Ma... <1%

65 Publication

Saiyed Salim Sayeed, Hemant Kumar Sharma, Pramod Kumar Yadav, Brijesh Mish... <1%

66 Publication

Emeç, Murat. "Increasing Communication Security Among Internet of Things", Do... <1%

67 Publication
Tusharkanta Samal, Ambarish Panda, Manas Ranjan Kabat, Ali Ismail Awad, Suve... <1%

68 Publication
Faheem Syeed Masoodi, Alwi M. Bamhdi, Majid A. Charoo, Zubair Sayeed Masoodi... <1%

69 Publication
Saha, Sajal. "Toward Building an Intelligent and Secure Network: an Internet Traf... <1%

70 Submitted works
University of Hertfordshire on 2024-03-22 <1%

71 Publication
"Proceedings of the Third International Conference on Innovations in Computing ... <1%

72 Publication
"Healthcare Recommender Systems", Springer Science and Business Media LLC, 2... <1%

73 Submitted works
Australian Institute of Higher Education on 2025-03-25 <1%

74 Publication
Namvar, Anahita. "Adversarial Machine Learning in IoT: Vulnerability Analysis an... <1%

75 Submitted works
Università degli studi di Salerno on 2023-07-11 <1%

76 Publication
Yu, Jeffrey C.. "Assessing Industrial Internet of Things Security at the Network Ed... <1%

77 Publication
"Artificial Intelligence for Security", Springer Science and Business Media LLC, 2024 <1%

78 Publication
"Data Protection", Springer Science and Business Media LLC, 2024 <1%

79 Publication
Azizian, Sasan. "A Data-Driven Discovery System for Studying Extracellular Micro ... <1%

80 Publication
D. Lakshmi, Ravi Shekhar Tiwari, Rajesh Kumar Dhanaraj, Seifedine Kadry. "Explai... <1%

81 Publication

Minghao Wang. "Utilizing Blockchain for Privacy Preservation in Internet of Thing... <1%

82 Publication

Sotirios Messinis, Nikos Temenos, Nicholas E. Protonotarios, Ioannis Rallis, Dimitr... <1%

83 Publication

Wreh, Kimma. "Federated Learning-Based Intrusion Detection System (IDS) With ... <1%

84 Publication

"Advances in Cyber Security", Springer Science and Business Media LLC, 2021 <1%

85 Publication

"Internet of Things. Advances in Information and Communication Technology", S... <1%

86 Publication

Al-Sumaidaee, Ghassan. "Blockchain Tokens as Universal Encrypted Access : A Co... <1%

87 Submitted works

Chester College of Higher Education on 2025-05-15 <1%

88 Publication

Jiang, Yiqun. "Advance in Healthcare Analytics: Electronic Health Record-Based Pr... <1%

89 Publication

S. Kannadhasan, R. Nagarajan, Alagar Karthick, V. Kumar Chinnaiyan. "Technolog... <1%

90 Publication

Tufekci, Burak. "Intrusion Detection System for Drones", University of North Texas <1%

91 Submitted works

University of Glamorgan on 2024-09-01 <1%

92 Internet

aip.vse.cz <1%

93 Internet

www.dsu.edu.in <1%

94 Publication

"AI Applications in Cyber Security and Communication Networks", Springer Scien... <1%

95 Publication
"Blockchain-Assisted Technologies for Sustainable Healthcare System", Springer ... <1%

96 Publication
"Fourth Congress on Intelligent Systems", Springer Science and Business Media L... <1%

97 Publication
"Proceedings of International Conference on Information Technology and Applic... <1%

98 Publication
"Proceedings of International Conference on Recent Innovations in Computing", ... <1%

99 Publication
Alanazi, Sami. "Improving Aspect-Based Sentiment Analysis Through Large Langu... <1%

100 Publication
Alhussien, Nour. "A Comprehensive Framework for Evaluating and Mitigating Ad... <1%

101 Publication
Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artific... <1%

102 Submitted works
Beirut Arab University on 2024-01-18 <1%

103 Publication
D. Jeya Mala, Anto Cordelia Tanislaus Antony Dhanapal, Saurav Sthapit, Anita Kha... <1%

104 Submitted works
George Washington University on 2024-07-11 <1%

105 Publication
Guimarães, Tiago André Saraiva. "Improving Veracity and Value of Data in Health... <1%

106 Publication
Neha Goel, Ravindra Kumar Yadav. "Internet of Things enabled Machine Learning... <1%

107 Publication
Nishu Gupta, Sandeep S. Joshi, Milind Khanapurkar, Asha Gedam, Nikhil Bhave. "... <1%

108 Submitted works
Tilburg University on 2025-04-10 <1%

109 Submitted works

University of Surrey on 2023-06-23 <1%

110 Publication

Yassine Maleh, Mohammad Shojafar, Mamoun Alazab, Imed Romdhani. "Blockch... <1%

111 Publication

Zipperle, Michael. "Provenance-Based Intrusion Detection for Automated Rule Ge... <1%

112 Internet

d197for5662m48.cloudfront.net <1%

113 Internet

papers.academic-conferences.org <1%

114 Internet

www.theseus.fi <1%

# *Design and Development of Smart and Secure Healthcare System*

64

A Thesis Submitted
in partial fulfillment of the requirements
For the award of the degree of
**Doctor of Philosophy**
in
**Department of Computer Science and Engineering**
*by*

**Nikhil Sharma**
**2K21/PhDCO/04**

Under the Guidance of

**Dr. Prashant Giridhar Shambharkar**
**(Supervisor)**
Assistant Professor
12
Department of Computer Science and Engineering
Delhi Technological University


**Delhi Technological University**
**Shahbad Daulatpur, Main Bawana Road**
**Delhi-110042**
**June, 2025**

ॐ कृष्णाय वासुदेवाय हरये परमात्मने॥
प्रणतः क्लेशनाशाय गोविंदाय नमो नमः॥

37


# DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

## CANDIDATE'S DECLARATION

I certify that the dissertation titled "Design and Development of Smart and Secure Healthcare System" submitted for the Doctor of Philosophy degree is my work and has not been submitted

87 for the award of any degree or diploma to any other University or Institute. The work done in the thesis is original and has been done by me under the supervision of my supervisors.

I also mention that the research work is original and has not been submitted by me, in part or completely, to any other University or Institution for the award of any degree or diploma.

**Nikhil Sharma**

**(Ph.D. Research Scholar)**

12 **Department of Computer Science and Engineering,**

**Delhi Technological University, Delhi**

37

# DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

## CERTIFICATE

This is to certify that the work contained in the thesis entitled "Design and Development of Smart and Secure Healthcare System" submitted by Nikhil Sharma (2K21/PHDCO/04) for the award of the degree of Doctor of Philosophy to Delhi Technological University, India contains original research work carried out by him under my supervision.

He has fulfilled all the requirements as per the required standard for the submission of the thesis. I hereby confirm the originality of the work and certify that the thesis has not been submitted for the award of any degree or diploma at this or any other institution.

74

**Dr. Prashant Giridhar Shambharkar**

**(Supervisor)**

**Assistant Professor**

**Department of Computer Science and Engineering**

**Delhi Technological University**

# ACKNOWLEDGMENT

I would first like to express my deepest gratitude to my supervisor **Dr. Prashant Giridhar Shambharkar**, for his unwavering guidance, insightful feedback, and constant encouragement throughout my Doctoral journey. His expertise and dedication have profoundly shaped both this research and my development as a scholar. I am grateful to **Prof. Manoj Kumar**, Professor and Head of the Department, Department of Computer Science and Engineering, Delhi Technological University, for their scholarly input and encouragement at critical stages of this work. My sincere thanks also go to **Prof. Rahul Katarya,** DRC Chairperson, Department of Computer Science and Engineering, Delhi Technological University. It is a privilege to submit this thesis under their guidance, constructive suggestions and steadfast support have been invaluable. I am also grateful to **Prof. Shailender Kumar, Professor,** Associate Head and SRC expert from within the department, and **Dr. N. Jayanthi,** Associate Professor, SRC expert from outside the Department, for their scholarly input and encouragement at critical stages of this work. I would like to thank my colleagues with whom I have had the pleasure of working throughout my time at DTU. Their support, encouragement, and camaraderie have made my experience in the Doctoral program truly memorable. I am indebted to my family and friends for their unwavering love and support, even during the most challenging times. My Parents, Brother, Sister, Wife, and Family & Friends have always been the pillars of my strength, and their constant support has helped me reach this stage in life. Without the collective support of these individuals, this research would not have been possible. Finally, I thank the Almighty for giving me the strength to carry out the present research work.

**Nikhil Sharma**

**2K21/PHDCO/04**

# ABSTRACT

The digital transformation of healthcare through the Internet of Medical Things (IoMT) has introduced unprecedented opportunities for real-time monitoring, remote diagnosis, and intelligent health management. However, the integration of interconnected medical devices and the continuous exchange of sensitive health data have also exposed IoMT systems to a wide range of cyber threats. Data confidentiality, integrity, and availability in resource-constrained environments remain challenging.

This thesis presents a novel security framework that combines deep learning-based intrusion detection with blockchain technology to provide comprehensive protection for healthcare data. First, an intelligent intrusion detection system (IDS) is developed using advanced deep learning architectures that integrate convolutional and recurrent neural networks with attention mechanisms. These models are designed to capture both spatial and temporal network traffic features and detect sophisticated attack patterns in heterogeneous IoMT environments.

The research further introduces a blockchain-based security architecture to address the challenges of data tampering, centralized trust, and unauthorized access. This framework incorporates dynamic encryption schemes, robust access control policies, zero-knowledge proofs for privacy preservation, and a practical Byzantine Fault Tolerant consensus mechanism. Additionally, decentralized storage is enabled using the InterPlanetary File System (IPFS) to ensure immutability and high availability of medical records.

A comprehensive experimental evaluation is conducted to assess the performance and scalability of the proposed solutions. The Comparative analyses with existing methodologies are carried out to demonstrate improvements in detection accuracy, cryptographic efficiency, and overall system robustness.

The integrated approach developed in this thesis offers a resilient and scalable security solution designed for modern healthcare systems. Combining the predictive power of deep learning with the trustless and immutable nature of blockchain significantly advances the state of the art in IoMT security. It provides a practical foundation for secure healthcare applications in real-world deployments.

# Table of Content

# List of Abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| IoT | Internet of Things |
| IoMT | Internet of Medical Things |
| ML | Machine Learning |
| DL | Deep Learning |
| FL | Federated Learning |
| XAI | Explainable AI |
| IDS | Intrusion Detection System |
| EHRs | Electronic Health Records |
| RPM | Remote Patient Monitoring |
| DDoS | Distributed Denial of Service |
| PKI | Public Key Infrastructure |
| OTA | Over-the-Air |
| RBAC | Role-Based Access Control |
| ABAC | Attribute-Based Access Control |
| MitM | Man-in-the-Middle |
| VPNs | Virtual Private Networks |
| MFA | Multi-Factor Authentication |
| ECC | Elliptic Curve Cryptography |
| ABE | Attribute-Based Encryption |
| ABSE | Attribute-Based Searchable Encryption |
| IDS/IPS | Intrusion Detection and Prevention Systems |
| DLT | Distributed Ledger Technology |
| PoW | Proof of Work |
| PoS | Proof of Stake |
| PBFT | Practical Byzantine Fault Tolerance |
| M2M | Machine-to-machine |
| DoS | Denial-of-Service |
| AIDS | Anomaly Intrusion Detection Systems |
| RFE | Recursive Feature Elimination |
| FAR | False Alarm Rates |
| NID | Network Intrusion Detection |
| RF | Random Forest |

| | |
|---|---|
| SDN | Software-Defined Networking |
| AdaBoost | Adaptive Boosting |
| GNN | Graph Neural Networks |
| Bi-LSTM | Bidirectional Long Short-Term Memory |
| LSTM | Long Short-Term Memory |
| DNN | Deep Neural Network |
| SHAP | Shapley additive explanations |
| LIME | Local Interpretable Model-agnostic Explanations |
| DT | Decision Tree |
| SHS | Smart healthcare systems |
| DPoS | Delegated Proof-of-Stake |
| HIoT | Healthcare Internet of Things |
| EHMS | Electronic Health Monitoring System |
| NLP | Natural Language Processing |
| SMOTE | Synthetic Minority Oversampling Technique |
| LR | Linear regression |
| XGB | Extreme Gradient Boosting |
| GBT | Gradient-Boosted Trees |
| KNN | K-nearest neighbour |
| SVM | Support Vector Machine |
| tanh | Hyperbolic tangent activation functions |
| FCL | Fully connected layer |
| SAF | Sigmoid Activation Function |
| MI | Mutual information |
| mGRU | stacked modified Gated Recurrent Units |
| MSCSL | Multi-Step Convolutional Neural Network Stacked Long Short-Term Memory |
| LSO | Light Spectrum Optimizer |
| LeLeLU | Leaky Learnable ReLU |
| DA-DRL | Dynamic Adaptive Deep Reinforcement Learning |
| NIZKPs | Non-Interactive Zero Knowledge Proof |
| SMPC | Secure Multi-Party Computation |
| SHA-512 | Secure Hash Algorithm 512-bit |
| Recon | Reconnaissance |
| BLE | Bluetooth Low Energy |
| TPS | Transactions per Second |
| RTIDS | Robust Transformer-based Intrusion Detection System |

| | |
|---|---|
| CFMT | Clustering-enabled federated meta-training |
| SCAE | Stacked Contractive Autoencoder |
| CART | Classification and Regression Trees |
| CDBN | Conditional Deep Belief Network |
| DR | Detection rates |
| RNNs | Recurrent Neural Networks |
| sklearn | scikit-learn |

# List of Table

# List of Figures

# Chapter 1. Introduction

## 1.1 Overview

The global healthcare sector is undergoing a profound transformation driven by technological innovation. With the convergence of emerging technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), blockchain, and cloud computing, the traditional paradigms of healthcare delivery are being reshaped into intelligent, interconnected, and patient-centric ecosystems [1]. A major contributor to this transformation is the Internet of Medical Things (IoMT), a specialized segment of IoT that integrates medical devices and applications into healthcare networks, enabling real-time monitoring, remote diagnostics, and predictive care [2].

While these developments present immense opportunities to improve patient outcomes, operational efficiency, and accessibility to healthcare services, they also introduce significant security and privacy risks [3]. The sensitivity of healthcare data, combined with the diversity of connected devices and heterogeneous communication protocols, creates a complex threat landscape where even minor vulnerabilities can lead to catastrophic breaches [4]. From unauthorized data access and identity theft to manipulation of critical medical equipment, the consequences of cyberattacks in the healthcare domain can be far-reaching, potentially endangering lives and undermining trust in digital health systems [5].

Security and privacy concerns have, therefore, become central to the design and deployment of IoMT systems. Despite the development of various security techniques, such as password-based authentication, digital certificates, and access control mechanisms, many of these are no longer sufficient in countering advanced persistent threats and ensuring compliance with regulatory requirements [6]. In response to these challenges, there is a growing recognition of the need for smart, secure, and scalable frameworks that integrate advanced technologies like blockchain and deep learning to ensure end-to-end protection of healthcare data [7].

Blockchain, with its decentralized and tamper-proof architecture, offers a promising solution for establishing trust, transparency, and immutability in healthcare data transactions [8]. Meanwhile, deep learning, especially when applied to intrusion detection, which enables proactive threat detection through pattern recognition and anomaly detection, outperforming traditional signature-based approaches [9]. The fusion of these two technologies creates a potent framework capable of addressing both data integrity and cybersecurity concerns across IoMT ecosystems [10].

This PhD research aims to explore and develop such an integrated approach. The study investigates existing literature on healthcare data security, proposes an Intrusion Detection System (IDS) tailored for healthcare environments, designs a blockchain-based security framework, and performs comparative analysis with current state-of-the-art techniques to validate its effectiveness. The predominant goal is to contribute to a resilient, trustworthy, and secure healthcare infrastructure that not only safeguards sensitive data but also enhances the quality-of-care delivery.

## 1.2 Motivation

The exponential growth of the digital healthcare ecosystem, powered by technologies such as the Internet of Medical Things (IoMT), Artificial Intelligence (AI), and cloud computing, has fundamentally transformed the

healthcare delivery model. Today, patient care is no longer confined to hospital walls. Remote monitoring, wearable sensors, and interconnected medical devices provide continuous health insights, enabling early diagnosis, personalized treatment, and proactive disease management. While these advancements have vastly improved patient outcomes and system efficiency, they have also introduced unprecedented security and privacy concerns.

Sensitive healthcare data, such as electronic health records (EHRs), diagnostic images, and real-time sensor data, are being generated and shared across networks at an unparalleled scale. These datasets, if intercepted or tampered with, can lead to identity theft, insurance fraud, or even life-threatening medical errors. Furthermore, the diverse and distributed nature of IoMT devices presents numerous points of vulnerability. Devices are often resource-constrained, lack standardized security protocols, and are prone to physical and cyber tampering. These factors make IoMT ecosystems prime targets for cyberattacks.

Traditional security methods like password-based authentication, symmetric encryption, and rule-based intrusion detection systems fall short in this dynamic environment. They cannot efficiently scale to accommodate the millions of interconnected devices or detect advanced persistent threats that evolve over time. Moreover, centralized data storage models create single points of failure, making healthcare systems susceptible to ransomware attacks and data breaches.

This pressing situation motivates the need for a paradigm shift in how we approach healthcare data security. The integration of blockchain technology and deep learning models represents a significant leap forward. Blockchain's decentralized and immutable architecture offers enhanced data integrity, auditability, and access control. When combined with deep learning techniques, especially in the context of intrusion detection, this hybrid approach can effectively identify anomalies, predict threats, and secure patient information against sophisticated attacks.

The motivation behind this research is thus twofold: to address the growing security and privacy threats in modern healthcare systems and to design a scalable, intelligent, and robust framework that ensures data integrity, confidentiality, and system availability. By proposing a comprehensive security model that integrates blockchain and deep learning, this research aims to fortify the healthcare ecosystem against current and future cyber threats.

## 1.3 Internet of Things (IoT)

The Internet of Things (IoT) represents a paradigm shift in how physical devices interact, communicate, and process data. It involves a vast ecosystem of interconnected devices embedded with sensors, actuators, software, and communication technologies that enable them to gather, transmit, and sometimes process data autonomously [11]. IoT has gained prominence due to its ability to enhance efficiency, productivity, and quality of life across a wide range of sectors, including agriculture, smart cities, industry, energy, and particularly healthcare [12].

### 1.3.1 IoT Architecture

The IoT architecture is typically structured in four key layers:

- **Perception Layer (Sensing Layer):** This layer consists of sensors and actuators that detect, measure, and collect data from the physical environment. Devices in this layer are responsible for recognizing physical parameters like temperature, heart rate, motion, and more.

- **Network Layer:** The collected data is transmitted through wired or wireless communication networks. This layer ensures that the information is relayed to other layers efficiently and securely. It often uses communication protocols like Wi-Fi, Bluetooth, ZigBee, LTE, and 5G.
- **Processing Layer (Middleware Layer):** This layer involves data processing, filtering, and analysis. It typically uses cloud computing or edge/fog computing paradigms to perform complex computations. It can include decision-making mechanisms based on AI/ML algorithms.
- **Application Layer:** The application layer offers specific services to users based on the data analysis. This can include remote patient monitoring in healthcare, automated irrigation in agriculture, smart metering in utilities, etc.

Each layer plays a vital role in the overall functionality, scalability, and security of an IoT system. A secure IoT system must account for vulnerabilities and threats that may arise at each of these layers [13].

### 1.3.2 Applications of IoT in Healthcare

IoT has led to the development of smart healthcare systems where devices such as wearable fitness bands, smart pills, remote monitoring systems, and implanted sensors can track patient vitals, monitor medication adherence, and detect abnormalities in real-time [14]. Data collected by these devices are transmitted to cloud servers for analysis and decision-making, facilitating early diagnosis and timely medical intervention [15]. The applications of IoT in healthcare are explained and depicted in **Figure 1.1** as follows:

- **Smart Healthcare Infrastructure:** IoT enables the creation of intelligent healthcare ecosystems by integrating connected devices into clinical and home care settings.
- **Real-Time Monitoring via Smart Devices:** Wearable fitness bands, smart pills, and implanted sensors continuously monitor patient vitals such as heart rate, glucose levels, and body temperature.
- **Medication Adherence Tracking:** IoT-based solutions track whether patients take prescribed medications on time, helping reduce human error and improve treatment outcomes.
- **Remote Patient Monitoring (RPM):** IoT devices support continuous remote monitoring of chronic or elderly patients, reducing the need for frequent hospital visits.
- **Cloud-Based Data Management:** The data collected from IoT sensors is transmitted to cloud servers, where it is stored, processed, and analyzed using AI/ML algorithms for diagnostics and recommendations.
- **Facilitating Early Diagnosis and Intervention:** Real-time data analysis aids in early disease detection and timely intervention, improving prognosis and reducing healthcare costs.
- **Critical Role of Data Security:** Ensuring the confidentiality, integrity, and availability of transmitted medical data is essential to avoid incorrect diagnoses and protect patient privacy.
- **Risk of Security Breaches:** Compromised data can lead to privacy violations, insurance fraud, and even fatal treatment errors, highlighting the need for secure architectures.
- **Need for Robust Security Mechanisms:** To mitigate these risks, IoT-based healthcare systems must incorporate strong security protocols such as encryption, authentication, and access control.

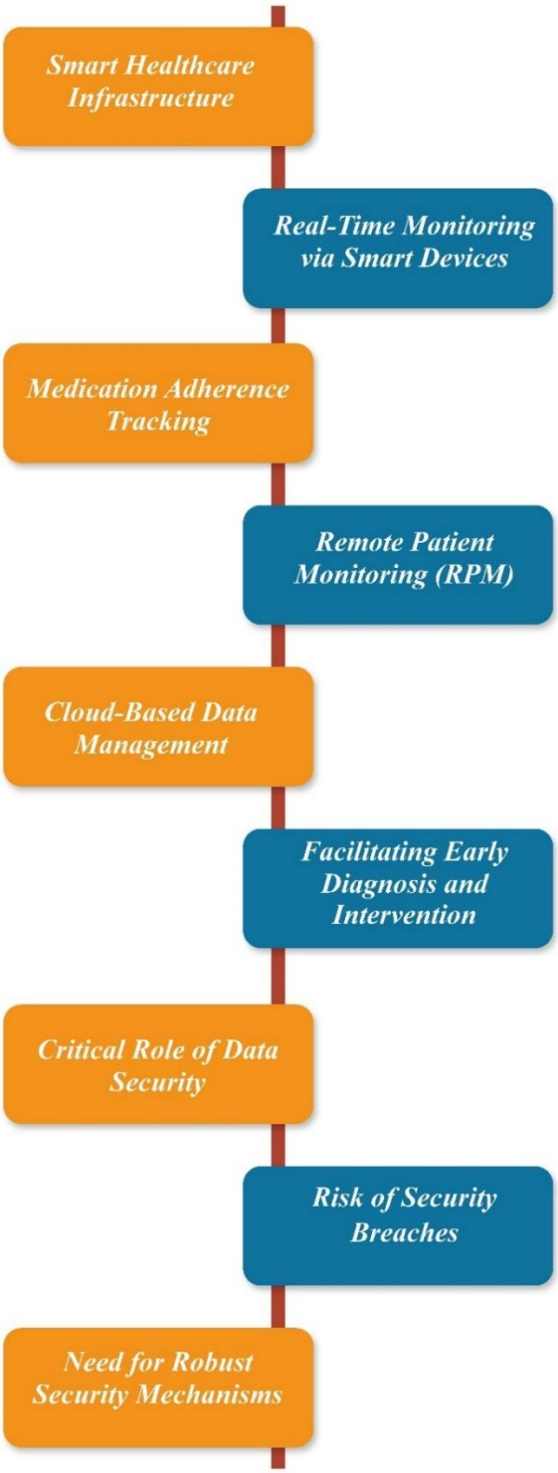



**Fig. 1.1.** Applications of IoT in Healthcare

### 1.3.3 Security Issues in IoT

Security is one of the most pressing challenges in IoT ecosystems. Devices often lack computational resources, making them incapable of running standard security protocols [16]. Moreover, large-scale deployments increase the attack surface and create more potential entry points for malicious actors [17]. Some common security issues include:

- **Resource Constraints in IoT Devices:** Many IoT devices have limited processing power, memory, and battery life, which restrict their ability to support standard cryptographic and security protocols.
- **Expanded Attack Surface:** The large-scale deployment of IoT devices increases the number of potential attack vectors, making it easier for adversaries to exploit vulnerabilities across the network.
- **Insecure Communication Protocols:** IoT devices often transmit data over unencrypted or weakly encrypted channels, making it vulnerable to eavesdropping, data manipulation, or man-in-the-middle attacks.
- **Weak Authentication and Authorization Mechanisms:** The use of default, hardcoded, or weak passwords compromises system access control, allowing unauthorized users to gain control over devices or data.
- **Lack of Regular Software and Firmware Updates:** Many IoT devices do not receive timely security patches, leaving them exposed to known vulnerabilities that can be exploited remotely.
- **Data Integrity Compromises:** Without robust integrity checks, data can be altered during transmission or storage, potentially leading to false analytics, wrong diagnoses, or malicious outcomes in sensitive applications like healthcare.
- **Difficulty in Implementing Traditional Security Measures:** Conventional IT security solutions are not directly applicable to IoT systems due to their heterogeneous nature and physical exposure to tampering.

### 1.3.4 Common IoT Attacks and Threats

The widespread deployment and connectivity of IoT devices expose them to a broad range of cyber threats [18]. **Table 1.1** summarizes some of the most prevalent attacks and their corresponding impacts on IoT ecosystems.

**Table 1.1.** Common IoT Attacks and Their Impact

| Attack Type | Description | Impact |
|---|---|---|
| Distributed Denial of Service (DDoS) | Attackers flood networks or servers with illegitimate traffic, disrupting normal operations. | Service outages, degraded performance, and data unavailability. |
| Botnet Infiltration | Compromised IoT devices are grouped into a botnet to launch coordinated attacks. | Loss of administrative control, data leakage, and propagation of malware. |
| Eavesdropping | Unencrypted or poorly secured communications are intercepted by unauthorized entities. | Violation of data confidentiality and privacy breaches. |
| Firmware Hijacking | Malicious code is injected during firmware or software updates. | Permanent device compromise and unauthorized control. |
| Physical Tampering | Attackers gain direct physical access to the device to manipulate hardware or extract data. | Disruption of system integrity and theft of sensitive information. |

### 1.3.5 Security Solutions for IoT

To address the multifaceted security challenges inherent in IoT ecosystems, a range of technical solutions has been developed. These solutions aim to ensure secure communication, protect device integrity, preserve data privacy, and detect potential intrusions [19]. **Table 1.2** outlines key security concerns in IoT and the corresponding solution approaches and technologies.

**Table 1.2.** Security Solutions for IoT Systems

| Security Concern | Solution Approach | Technology Used |
|---|---|---|
| Secure Communication | Implementation of end-to-end encryption mechanisms to prevent data interception. | TLS/SSL, IPsec |
| Device Authentication | Ensuring only legitimate devices can access the network through identity validation. | Public Key Infrastructure (PKI), Biometrics, MAC Binding |
| Software Vulnerabilities | Regular updates and secure patching mechanisms to eliminate known threats. | Over-the-Air (OTA) Updates, Secure Boot |
| Data Privacy | Limiting access to sensitive information and applying anonymization techniques. | Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Homomorphic Encryption |
| Intrusion Detection | Monitoring system behaviour to identify abnormal or unauthorized activities. | AI/ML-based Intrusion Detection Systems (IDS) |

### 1.3.6 Challenges in Securing IoT

While many solutions exist, several challenges still impede effective IoT security implementation [20]:

- **Resource Constraints:** Most IoT devices operate with limited computational resources such as low processing power, minimal memory, and constrained battery life. These limitations hinder the implementation of robust cryptographic algorithms and comprehensive security mechanisms [21].
- **Scalability Issues:** The exponential growth of connected devices projected to reach tens of billions poses challenges in managing and scaling security architectures [22]. Centralized security models may become bottlenecks or single points of failure in such vast networks.
- **Device Heterogeneity and Lack of Standardization:** IoT ecosystems are composed of diverse devices manufactured by different vendors, each using proprietary protocols and interfaces. This lack of interoperability complicates the development and deployment of uniform security policies and standards [23].

- **Latency Sensitivity in Real-Time Applications:** Applications such as healthcare monitoring and industrial automation require real-time responsiveness [24]. Heavy encryption and complex security checks may introduce delays that are unacceptable in time-critical scenarios.
- **Firmware and Software Update Challenges:** Delivering timely and secure updates to a vast and distributed set of IoT devices is operationally difficult. Insecure or failed updates can leave devices vulnerable to exploitation or cause system malfunctions [25].

### 1.3.7 Research Direction

Given these challenges, there is a growing interest in lightweight security protocols, decentralized identity management, and AI-driven intrusion detection systems. Moreover, integrating blockchain with IoT has emerged as a powerful method to ensure decentralized trust, data integrity, and secure communication among devices.

This thesis builds upon these ongoing advancements by focusing specifically on the Internet of Medical Things (IoMT), a healthcare-oriented extension of IoT. The proposed research explores how blockchain, deep learning, and advanced encryption techniques can be harmonized to secure IoT-based medical infrastructures against evolving cyber threats [26].

By understanding the foundational architecture, applications, and security challenges of IoT, this section lays the groundwork for the subsequent sections, particularly the focused discussion on IoMT in the next subsection. The convergence of healthcare and IoT offers unprecedented opportunities, but it also demands unprecedented responsibility in securing every bit of data generated, transmitted, and processed across this digital field.

## 1.4 Internet of Medical Things (IoMT)

The Internet of Medical Things (IoMT) is a specialized application of the Internet of Things (IoT) in the healthcare domain [27]. It refers to a networked infrastructure comprising interconnected medical devices, software applications, and health systems designed to monitor, collect, analyze, and transmit medical data [28]. These systems operate through sensors embedded in wearable, implantable, or stationary devices, often integrated with cloud computing and artificial intelligence for advanced diagnostics and remote health services [29].

### 1.4.1 IoMT Architecture

The architecture of IoMT typically follows a multi-layered approach [30-31]. Below are the major components:

- **Perception Layer:** This layer includes medical sensors and devices such as ECG monitors, glucose sensors, pulse oximeters, smart inhalers, and wearable fitness trackers. These devices sense physiological parameters and convert them into digital data.
- **Network Layer:** This layer ensures the secure transmission of the sensed data from the perception layer to the processing systems. It leverages various communication technologies like Wi-Fi, Bluetooth, Zigbee, 5G, and LPWAN.
- **Processing Layer:** In this layer, raw medical data is processed using edge computing devices or centralized cloud platforms. Advanced analytics, including AI and deep learning algorithms, are applied to identify health trends, anomalies, and predictive diagnostics.

- **Application Layer:** This layer interfaces with end users such as doctors, nurses, patients, and hospital administrators. It enables functionalities like real-time dashboards, telemedicine platforms, electronic health records (EHRs), and automated alerts.
- **Security Layer:** This layer underpins the entire architecture, ensuring data confidentiality, integrity, authentication, and secure access controls using encryption, blockchain, and intrusion detection systems.

### 1.4.2 Benefits of IoMT

The Internet of Medical Things (IoMT) has brought about a significant evolution in how healthcare is delivered and experienced [32]. By connecting medical devices to the internet and enabling real-time data sharing, IoMT is bridging the gap between patients and healthcare providers like never before. This technology allows for continuous health monitoring [33], early detection of medical conditions [34], and more personalized treatment plans based on patient-specific data. Beyond clinical improvements, IoMT also reduces the burden on healthcare infrastructure by minimizing unnecessary hospital visits and admissions [35]. The core benefits that make IoMT a promising component of future healthcare systems are explained as follows:

- **Real-Time Remote Monitoring:** IoMT enables continuous monitoring of patient health parameters such as heart rate, glucose levels, and oxygen saturation. This allows healthcare providers to track patient conditions in real time, regardless of location.
- **Early Diagnosis and Intervention:** With continuous data collection, anomalies can be detected at an early stage, enabling prompt medical intervention that can prevent complications and reduce the severity of diseases.
- **Reduction in Hospital Visits and Admissions:** By facilitating home-based monitoring and remote consultations, IoMT reduces the need for frequent hospital visits and prolonged admissions, which also alleviates pressure on healthcare infrastructure.
- **Data-Driven Personalized Treatment:** IoMT devices collect longitudinal patient data that can be analyzed to tailor treatments to individual health profiles, leading to more effective and personalized care.
- **Improved Patient Adherence and Outcomes:** Reminders, feedback loops, and automated alerts from IoMT systems promote adherence to medication schedules and treatment plans, resulting in better patient engagement and improved health outcomes.

### 1.4.3 Challenges in IoMT

While IoMT has the potential to greatly improve healthcare delivery, putting it into practice comes with its own set of hurdles. Many devices have limited computing power, and the lack of common standards can make it hard for different systems to work together [36]. On top of that, ensuring data security and meeting varying legal requirements across regions adds further complexity. The challenges in IoMT are explained as follows:

- **Limited Computational Resources in Medical Devices:** Many IoMT devices are resource-constrained, making it difficult to implement advanced encryption or machine learning algorithms needed for robust security and analytics.

- **Lack of Standardization in Communication Protocols:** The absence of universally accepted communication standards across devices from different vendors creates interoperability challenges and hampers seamless data exchange.
- **Vulnerability to Physical Tampering and Software Exploits:** Given their physical deployment in uncontrolled environments, IoMT devices are susceptible to tampering. Inadequate software protections also expose them to malware and cyberattacks.
- **Legal and Regulatory Compliance Complexities:** Managing patient data requires adherence to strict privacy laws, and ensuring compliance across distributed IoMT systems can be complex and burdensome for healthcare providers.

### 1.4.4 Common IoMT Attacks and Threats

As IoMT devices become increasingly integrated into healthcare ecosystems, they also become prime targets for cyber threats [37-38]. **Table 1.3** outlines some of the most prevalent attack types, along with their descriptions and potential impacts on system integrity, data confidentiality, and patient safety.

**Table 1.3.** Common IoMT Attacks and Threats

| Threat Type | Description | Potential Impact |
|---|---|---|
| Man-in-the-Middle (MitM) | Interception of data between IoMT devices and cloud servers | Data leakage, manipulation of medical records |
| Ransomware | Malicious encryption of sensitive medical data or devices | Service denial, financial and operational loss |
| Replay Attack | Reuse of previously captured legitimate data packets | False data injection, misleading diagnostics |
| Device Tampering | Unauthorized physical access or modification of devices | Compromised treatment accuracy, device misuse |
| Firmware Exploits | Exploiting outdated or insecure firmware | Persistent malware, long-term system compromise |

### 1.4.5 IoMT Security Solutions

To mitigate growing security threats, a range of technological solutions have been adopted for different layers of IoMT architecture [39]. **Table 1.4** presents key security approaches, the technologies used, and their respective application domains within IoMT systems.

**Table 1.4.** IoMT Security Solutions

| Solution Type | Technology Used | Application Area |
|---|---|---|
| Secure Communication | TLS/SSL, Virtual Private Networks (VPNs) | Protects data during transmission |
| Authentication | Biometrics, Multi-Factor Authentication (MFA), Digital Certificates | Controls access to devices and networks |

| | | |
|---|---|---|
| Encryption | AES, Elliptic Curve Cryptography (ECC), Attribute-Based Encryption (ABE) | Ensures confidentiality and data integrity |
| Blockchain | Ethereum, Hyperledger Fabric | Secure data sharing, tamper-evident logs |
| IDS/IPS | Deep Learning models, Signature-based detection | Detects and prevents anomalous activities |

### 1.4.6 IoMT Security Countermeasures

Each security vulnerability in IoMT demands tailored countermeasures to ensure system resilience [40]. **Table 1.5** summarizes specific countermeasures aligned with particular attack vectors and highlights their security benefits.

**Table 1.5.** IoMT Security Countermeasures

| Attack Vector | Countermeasure | Benefits |
|---|---|---|
| Device Spoofing | Digital Certificates, Public Key Infrastructure (PKI) | Verifies device authenticity and trust |
| Unauthorized Access | Role-Based Access Control (RBAC), Multi-Factor Authentication | Prevents unauthorized privilege escalation |
| Data Leakage | Attribute-Based Searchable Encryption (ABSE) | Enables secure, privacy-preserving access |
| Firmware Vulnerabilities | Over-the-Air (OTA) Secure Firmware Updates | Keeps devices patched and protected |
| Network Intrusion | AI-based Intrusion Detection and Prevention Systems (IDS/IPS) | Enables real-time threat detection and response |

### 1.4.7 Key Challenges vs Research Directions

Although IoMT security has progressed considerably, several critical challenges persist. **Table 1.6** maps each challenge to a corresponding research direction, emphasizing ongoing efforts to enhance IoMT system robustness [41].

**Table 1.6.** Key Challenges vs. Research Directions

| IoMT Challenge | Research Direction |
|---|---|
| Inadequate device-level security | Blockchain-enabled lightweight device authentication |
| Poor anomaly detection | Deep Learning-powered Intrusion Detection Systems (IDS) |
| Weak data privacy mechanisms | Hybrid Attribute-Based Searchable Encryption (ABSE) |

| | |
|---|---|
| Legacy communication protocols | Secure and lightweight protocol upgrades |
| Lack of traceability | Blockchain-based decentralized and immutable audit trails |

### 1.4.8 Future Directions

With the continuous evolution of IoMT technologies, emerging paradigms like federated learning, edge AI, and quantum cryptography will play a pivotal role in further strengthening data security. The development of lightweight security protocols tailored for constrained medical devices, compliance-aware security frameworks, and real-time threat intelligence sharing will drive future innovations.

This research addresses the critical need for a unified framework that integrates these advanced technologies to provide end-to-end security in IoMT systems [42]. By focusing on device-level protection, secure interoperability, and intelligent threat detection, the proposed model aims to establish a trustworthy, scalable, and resilient IoMT ecosystem that enhances patient safety and data privacy.

## 1.5 Blockchain Overview

Blockchain is a revolutionary decentralized technology that has transcended its initial role in enabling cryptocurrencies to become a foundational component of secure, transparent, and tamper-resistant digital ecosystems [43]. At its core, blockchain is a distributed ledger technology (DLT) in which transaction records are maintained across a network of nodes rather than being stored in a central authority [44]. Each node maintains a synchronized copy of the ledger, ensuring high data redundancy, consistency, and resistance to single points of failure.

One of the most compelling attributes of blockchain is its immutability, once data is recorded and validated, it becomes practically impossible to alter without the consensus of the network [45]. This inherent trustless nature fosters transparency, integrity, and traceability, which are critical in sensitive domains like healthcare, finance, and supply chain management. Blockchain eliminates the need for intermediaries, enabling secure peer-to-peer interactions and reducing the risk of fraud and unauthorized data manipulation [46].

Although blockchain gained global recognition through its implementation in Bitcoin and other cryptocurrencies, its application has rapidly expanded into diverse sectors [47]. In healthcare, and particularly in the Internet of Medical Things (IoMT) ecosystems, blockchain is increasingly being adopted to address issues of data security, patient privacy, interoperability, and regulatory compliance [48]. It offers robust features such as decentralized security, transparent audit trails, and trustless coordination among stakeholders, all of which are essential in critical healthcare environments.

### 1.5.1 Working Principle of Blockchain

The functionality of blockchain is driven by a carefully designed structure and a set of cryptographic and algorithmic principles that ensure data security, consistency, and integrity [49]. A blockchain is composed of a linear sequence of blocks, each of which contains the following core elements:

- **A list of validated transactions:** These represent the operational changes or activities that have occurred while sharing a patient's diagnostic report, or updating medication schedules.
- **A timestamp:** This provides the exact moment at which the block was created, facilitating chronological order and traceability.
- **A cryptographic hash of the previous block:** This links each block to its predecessor, forming a tamper-evident chain.

These blocks are interconnected through cryptographic hashing, which ensures that any modification to one block would necessitate the recalibration of all subsequent blocks which is a computationally prohibitive task in secure networks [50]. As a result, blockchain provides a tamper-resistant environment where historical records remain verifiable and consistent.

### 1.5.2 Consensus Mechanisms in Blockchain

To uphold the consistency of the distributed ledger, blockchain systems rely on consensus mechanisms protocols by which all participating nodes agree on the validity of transactions and the current state of the ledger. These algorithms prevent double-spending, malicious activities, and network divergence, even in untrusted environments [51-52]. Some of the widely used consensus mechanisms include:

**(i) Proof of Work (PoW)**

This is a computationally intensive process in which participating nodes (miners) solve complex cryptographic puzzles. The first node to solve the puzzle earns the right to add a new block to the chain. Although highly secure, PoW is resource-intensive and unsuitable for energy-constrained environments like IoMT.

**(ii) Proof of Stake (PoS)**

In PoS, validators are selected based on the number of tokens they hold and are willing to "stake" as collateral. This mechanism consumes significantly less energy than PoW and offers faster block validation, making it more suitable for scalable systems.

**(iii) Practical Byzantine Fault Tolerance (PBFT)**

PBFT is specifically designed for permissioned blockchain environments, where participant nodes are known and authenticated. It can tolerate up to one-third of faulty or malicious nodes by requiring consensus through multiple communication rounds. PBFT is particularly advantageous in healthcare systems, where privacy, speed, and fault tolerance are critical.

These consensus mechanisms are fundamental to blockchain's robustness against adversarial attacks and ensure that all transactions added to the ledger are legitimate, even in the presence of unreliable or compromised nodes.

Finally, blockchain technology introduces a paradigm shift in how digital data is stored, verified, and shared across distributed systems. Its ability to offer immutability, decentralization, and trustless trust makes it an ideal candidate for enhancing security and transparency in healthcare applications, especially in IoMT networks. Understanding its structure, working principle, and consensus models is essential for integrating blockchain effectively into secure, patient-centric smart healthcare architectures.

### 1.5.3 Types of Blockchain

Blockchain networks can be categorized based on access rights, governance models, and the degree of decentralization. Each type offers unique advantages and limitations depending on the intended application and security requirements [53]. The selection of an appropriate blockchain type is critical in healthcare, where data sensitivity, trust boundaries, and regulatory compliance play a pivotal role.

**(i) Public Blockchain**

This type is open to anyone and is fully decentralized. Any participant can join the network, validate transactions, and access data. Public blockchains provide the highest level of transparency and security through consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS). However, their slower transaction speeds and lower privacy levels may not be suitable for healthcare applications involving sensitive patient data. Example of Public Blockchain is Bitcoin, and Ethereum.

**(ii) Private Blockchain**

Controlled by a single entity or organization, private blockchains restrict participation and access. They offer high transaction throughput, enhanced privacy, and centralized control, making them suitable for internal healthcare applications such as hospital record management and staff credential verification. Example of Private Blockchain is Hospital data management systems.

**(iii) Consortium Blockchain**

Managed by a group of authorized entities, consortium blockchains provide a partially decentralized governance model. They strike a balance between transparency and control, allowing multiple healthcare providers or institutions to collaboratively manage shared data such as inter-hospital patient transfers. Example of Consortium Blockchain is Inter-hospital patient record sharing.

**(iv) Hybrid Blockchain**

Combining features of public and private blockchains, hybrid models offer customizable access controls. This flexibility makes them ideal for complex healthcare use cases, such as pharmaceutical supply chains, where both public verification and private data confidentiality are necessary. Example of Hybrid Blockchain are Healthcare and pharmaceutical supply chain management.

### 1.5.4 Features and Characteristics of Blockchain

Blockchain's foundational features make it a promising technology for secure, interoperable, and trustworthy healthcare systems, particularly in IoMT ecosystems [54-55]. The core characteristics of Blockchain are discussed and depicted in **Figure 1.2** as follows:

- **Decentralization:** Eliminates reliance on centralized servers or authorities, reducing the risk of single points of failure and fostering resilience across healthcare networks.
- **Immutability:** Once a transaction is recorded and confirmed, it becomes part of the permanent ledger. This tamper-resistance is vital for ensuring the integrity of patient records and device logs.

- **Transparency:** While preserving privacy through encryption and access control, blockchain allows authorized stakeholders to audit and verify transaction histories as an essential feature for healthcare compliance.
- **Security:** Blockchain employs advanced cryptographic techniques like SHA-256, or digital signatures, to protect data from unauthorized modification or interception.
- **Auditability:** Every transaction is time-stamped and traceable, enabling comprehensive audits for regulatory and forensic purposes.
- **Consensus Mechanisms:** These ensure agreement among nodes regarding the ledger's current state, protecting against fraudulent entries and maintaining system consistency.

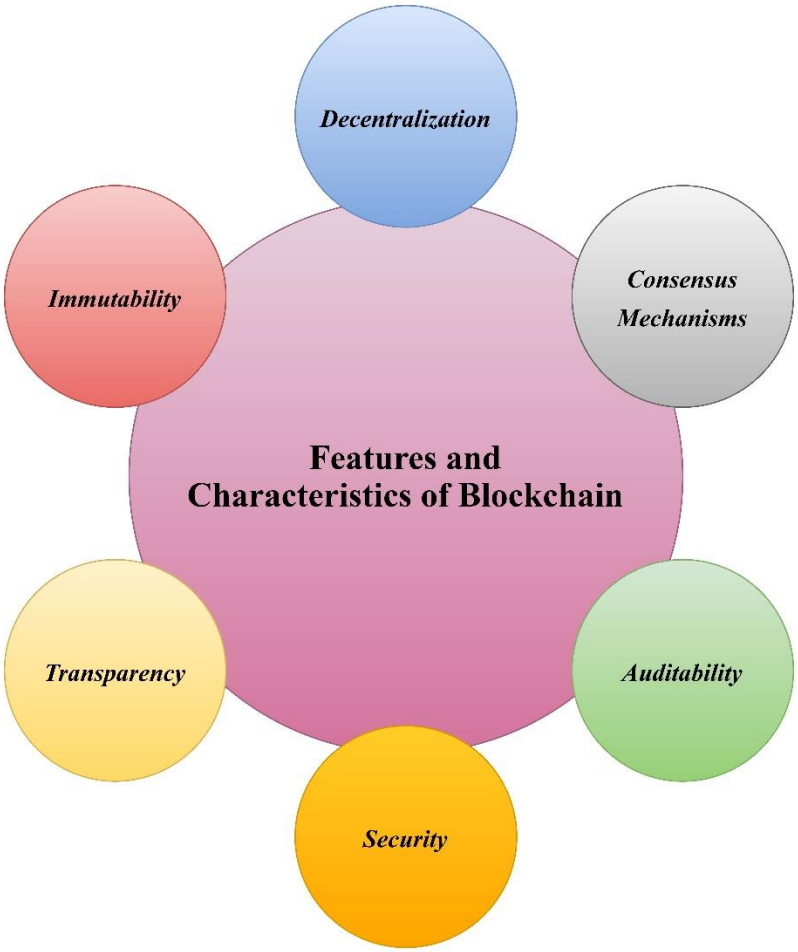



**Fig. 1.2.** Features and Characteristics of Blockchain

These features collectively address the stringent requirements of healthcare systems, where data confidentiality, reliability, and auditability are of paramount importance.

### 1.5.5 Blockchain in Healthcare

Blockchain is redefining the landscape of digital healthcare by providing a secure and efficient mechanism for data exchange and validation across decentralized environments. It addresses key pain points in healthcare such

as data breaches, lack of interoperability, and unverified access by enabling trustless and auditable operations. The Real-World Applications of blockchain in healthcare are explained as follows:

- **Electronic Health Records (EHRs):** Blockchain enables immutable, interoperable, and patient-controlled health records, improving continuity of care and reducing administrative overhead.
- **Remote Monitoring:** Data from IoMT devices can be recorded on the blockchain to ensure its authenticity and integrity, thereby supporting remote diagnostics and alerts.
- **Medical Research:** Blockchain can facilitate the secure and transparent sharing of clinical trial data among research institutions, while protecting intellectual property and data authenticity.
- **Pharmaceutical Supply Chain:** Through blockchain, stakeholders can track the origin, handling, and distribution of drugs, combat counterfeit medicines and ensuring regulatory compliance.

**Table 1.7** presents the Summary of Existing Blockchain Platforms for Healthcare Applications.

**Table 1.7.** Summary of Existing Blockchain Platforms for Healthcare Applications

| Platform | Purpose | Special Feature |
|---|---|---|
| HealthChain [56] | Patient-controlled EHR system | Permissioned architecture for data access control |
| MedShare [57] | Secure medical data exchange | Strong focus on patient privacy |
| Fortified-Chain [58] | IoMT device authentication and data protection | Emphasizes secure device-level interactions |

Despite these innovations, existing solutions often fall short in addressing threats at the device-level, such as spoofing or firmware tampering, and do not provide robust machine-to-machine (M2M) security in IoMT environments.

### 1.5.6 Comparison of Blockchain Types Based on Healthcare Suitability

Given the diverse requirements of healthcare systems, each blockchain type must be evaluated on key parameters such as speed, privacy, scalability, and control [59]. **Table 1.8** presents a comparative analysis of Blockchain Types Based on Healthcare Suitability.

**Table 1.8.** Comparison of Blockchain Types Based on Healthcare Suitability

| Parameter | Public Blockchain | Private Blockchain | Consortium Blockchain | Hybrid Blockchain |
|---|---|---|---|---|
| Speed | Low | High | High | Moderate |
| Privacy | Low | High | High | High |
| Transparency | High | Low | Moderate | Balanced |
| Scalability | Limited | High | High | Moderate |
| Control | Decentralized | Centralized | Partially decentralized | Mixed |

| Use Case | Cryptocurrency | Hospital Intranet | Multi-Hospital Data Sharing | IoMT Secure Communication |
|---|---|---|---|---|

For healthcare, particularly in IoMT-enabled settings, hybrid and consortium blockchains offer the most balanced trade-off between control, transparency, and scalability.

### 1.5.7 Benefits of Blockchain in IoMT Security

IoMT networks face an array of security risks, from unauthorized data access to malware infiltration and denial-of-service (DoS) attacks. Blockchain, when integrated effectively, addresses many of these concerns by embedding security at both the data and infrastructure levels [60-61]. The Key Benefits of Blockchain in IoMT Security are discussed as follows:

- **Decentralized Trust:** Removes dependency on centralized authorities for authentication, thereby reducing attack surfaces.
- **Tamper-Proof Logs:** Immutable records ensure that all device actions and data transactions are traceable and verifiable.
- **Device Authentication:** Unique device identities and registration on the blockchain prevent impersonation and rogue device participation.
- **Smart Access Control:** Role- or attribute-based permissions governed by smart contracts ensure that only authorized users or devices can access data.
- **Data Ownership and Privacy:** Patients maintain control over their data and can grant selective, revocable access, ensuring compliance with privacy regulations such as HIPAA and GDPR.

The integration of blockchain thus enhances transparency, data integrity, and resilience in IoMT environments, laying the foundation for a secure and trustworthy smart healthcare ecosystem.

## 1.6 Scope of Study

This research focuses on addressing the pressing security and privacy challenges within healthcare systems enabled by the Internet of Medical Things (IoMT). As the healthcare industry increasingly adopts connected medical devices and smart monitoring systems, it becomes more vulnerable to sophisticated cyber threats. The proliferation of IoMT introduces a wide attack surface due to the heterogeneity, resource constraints, and often weak security configurations of medical devices. This study is scoped to analyze, design, and validate a secure architecture that enhances both the resilience and privacy of IoMT-enabled healthcare systems.

The research primarily investigates the limitations of current IoMT architectures and identifies key vulnerabilities in device communication, data sharing, and system-level access control. It highlights the inadequacy of traditional security solutions that either focus only on data confidentiality or rely heavily on centralized models, which are susceptible to single points of failure and latency issues.

51 To address these gaps, the study proposes a hybrid framework that integrates blockchain technology and deep learning-based intrusion detection systems (IDS). The blockchain component ensures decentralized, tamper-proof, and transparent data management across interconnected devices and institutions. It employs smart contracts for

dynamic access control and uses cryptographic techniques to ensure data integrity and user authentication. Simultaneously, the research incorporates a deep learning-driven IDS trained to recognize patterns of malicious behaviour in real-time. This component is tailored for the healthcare domain, where both false positives and false negatives can have critical implications. The IDS is designed to detect zero-day attacks, anomalous traffic, and other threats that traditional rule-based systems might overlook.

Furthermore, the study includes a comprehensive comparative analysis of the proposed framework against existing security mechanisms. Performance metrics such as accuracy, computational latency, scalability, energy efficiency, and fault tolerance are used to validate the model's effectiveness.

Finally, this research provides a novel, layered security solution that aims to secure patient data, protect medical infrastructure, and ensure trust and compliance in next-generation smart healthcare systems.

## 1.7 Research Objectives

Based on the literature review of existing state-of-the-art methods and the research gaps identified in the above sections, the following research objectives are significant and suitable for further developing a Smart and Secure Healthcare System.

- ✓ **RO-1:** To perform a systematic literature review of the techniques used in securing healthcare data.
- ✓ **RO-2:** To develop an Intrusion Detection Model for securing Healthcare records.
- ✓ **RO-3:** To design a Framework to Secure Healthcare Data using Blockchain Technology.
- ✓ **RO-4:** To do a comparative result analysis of the developed model with the existing techniques.

## 1.8 Thesis Organization

This thesis is structured into six chapters, each addressing a fundamental aspect of building a secure, intelligent healthcare ecosystem. The organization is aligned with the research objectives and aims to provide step-by-step development of solutions for protecting healthcare data in IoMT environments using blockchain, deep learning, and Explainable AI (XAI).

**Chapter 1: Introduction**

This chapter introduces the motivation behind building a smart and secure healthcare system. It outlines the problem statement, objectives, scope, significance, and organization of the thesis.

**Chapter 2: Literature Review**

This chapter provides a comprehensive review of existing methods used for securing healthcare data. It identifies critical research gaps in blockchain-based models, intrusion detection frameworks, and the role of Explainable AI in healthcare cybersecurity.

**Chapter 3: Intrusion Detection Framework for Healthcare Systems (Aligned with RO3)**

This chapter introduces deep learning-based IDS models like ConvNet-SVM, MA-DeepCRNN, and MAC-LSTM. It describes how these models detect various attacks in IoMT networks using optimized feature selection, multi-attention mechanisms, and high-performance neural architecture.

**Chapter 4: Blockchain-Based Model for Securing Healthcare Data (Aligned with RO4)**

This chapter presents the design and development of a blockchain-driven security framework. It explains smart contracts, consensus mechanisms (PBFT), dynamic encryption (ECC/SHA-256), and decentralized storage (IPFS) to ensure secure, transparent, and tamper-resistant healthcare data handling.

**Chapter 5: Explainable AI for Interpretable Security Solutions (Cross-cutting contribution)**

This chapter focuses on integrating Explainable AI (XAI) into IDS frameworks to enhance transparency and trust. It discusses attention-based models, ablation studies, and interpretable outputs that assist medical professionals in understanding and validating intrusion alerts.

**Chapter 6: Conclusion, Future Work, and Societal Applications**

This chapter summarizes key contributions, presents a comparative result analysis against state-of-the-art methods, outlines future research directions, and discusses the societal benefits of the proposed secure healthcare system in real-world medical environments.

## 1.9 Research Objective Mapping with Publications

**Table 1.9** presented a structured overview linking the research objectives with the associated publications to clearly demonstrate how each research goal has been addressed throughout this work.

**Table 1.9.** Research Objective Mapping with Publications

| Research Objectives | Publication(s) Remarks |
|---|---|
| **RO1.** To perform a systematic literature review of the techniques used in securing healthcare data. | ✓ Sharma, N., & Shambharkar, P. G. (2022). Applicability of ML-IoT in Smart Healthcare Systems: Challenges, Solutions & Future Direction. 2022 International Conference on Computer Communication and Informatics (ICCCI). ***(Published)*** |
| | ✓ Sharma, N., & Shambharkar, P. G. (2025). A Systematic Literature Review of the Emerging Technologies used in securing healthcare data. 12th IEEE International conference on Internet of Everything, Microwave, Embedded, Communication & Networks (IEMECON-2024). ***(Published)*** |
| **RO2.** To develop an Intrusion Detection Model for securing Healthcare records. | ✓ Shambharkar, P. G., & Sharma, N., (2024). Deep Learning Empowered Intrusion Detection Framework for the Internet of Medical Things Environment, Knowledge and Information Systems, Springer. ***(Published)*** |
| | ✓ Sharma, N., & Shambharkar, P. G., (2025). Transforming Internet of Medical Things Security with Deep Learning-Powered Intrusion Detection Frameworks, Applied Soft Computing, Elsevier ***(Published)*** |
| | ✓ Sharma, N., & Shambharkar, P. G., Enhancing Internet of Medical Things Security with Multi-Attention Convolutional LSTM-Based Intrusion Detection System, Transactions on Emerging Telecommunications Technologies, Wiley ***(Major Revision Submitted)*** |

| | |
|---|---|
| **RO2.** To develop an Intrusion Detection Model for securing Healthcare records.<br>&<br>**RO3.** To design a Framework to Secure Healthcare Data using blockchain technology. | ✓ Sharma, N., & Shambharkar, P. G. (2025), Multi-Attention DeepCRNN: An Efficient and Explainable Intrusion Detection Framework for Internet of Medical Things Environments, Knowledge and Information Systems, Springer ***(Published)*** |
| | ✓ Sharma, N., & Shambharkar, P. G. (2025). Multi-Layered Security Architecture for IoMT Systems: Integrating Dynamic Key Management, Decentralized Storage, and Dependable Intrusion Detection Framework, International Journal of Machine Learning & Cybernetics, Springer. ***(Published)*** |
| | ✓ Sharma, N., & Shambharkar, P. G. (2025). Towards Secure Healthcare: SA-GBO-ODBN Model Utilizing Blockchain and Deep Learning for Data Handling and Diagnosis, The Computer Journal, Oxford University Press. ***(Published)*** |
| | ✓ Sharma, N., & Shambharkar, P. G., Enhancing Internet of Medical Things Security: A Multi-Layered Approach Using Dynamic Adaptive Deep Reinforcement Learning and Blockchain, Computer and Electrical Engineering, Elsevier. ***(Major Revision Submitted)*** |
| | ✓ Sharma, N., & Shambharkar, P. G., A Quantum Neural Network-Assisted Hybrid Cryptographic Model for Secure Blockchain-Based EHR Systems. ***(To be communicated)*** |
| **RO3.** To design a Framework to Secure Healthcare Data using blockchain technology. | ✓ Sharma, N., & Shambharkar, P. G., A Novel Blockchain-Based Framework for Securing Electronic Health Records Using Least Squares Analysis, Cluster Computing, Springer. ***(With Editor)*** |
| | ✓ Sharma, N., & Shambharkar, P. G. Blockchain-Based Framework for Secure Medical Data Sharing and Disease Diagnosis Using Optimized Deep Belief Networks, Cluster Computing, Springer. ***(Major Revision Submitted)*** |
| **RO4.** To do a comparative result analysis of the developed model with the existing techniques. | ✓ Deep Learning Empowered Intrusion Detection Framework for the Internet of Medical Things Environment, Knowledge and Information Systems, Springer. ***(Published)***<br>✓ Multi-Attention DeepCRNN: An Efficient and Explainable Intrusion Detection Framework for Internet of Medical Things Environments, Knowledge and Information Systems, Springer. ***(Published)***<br>✓ Multi-Layered Security Architecture for IoMT Systems: Integrating Dynamic Key Management, Decentralized Storage, and Dependable Intrusion Detection Framework, International Journal of Machine Learning & Cybernetics, Springer. ***(Published)***<br>✓ Towards Secure Healthcare: SA-GBO-ODBN Model Utilizing Blockchain and Deep Learning for Data Handling and Diagnosis, The Computer Journal, Oxford University Press. ***(Published)***<br>✓ Transforming Internet of Medical Things Security with Deep Learning-Powered Intrusion Detection Frameworks, Applied Soft Computing, Elsevier. ***(Published)***<br>✓ Blockchain-Based Framework for Secure Medical Data Sharing and Disease Diagnosis Using Optimized Deep Belief Networks, Cluster Computing, Springer. ***(Major Revision Submitted)*** |

| | |
|---|---|
| | ✓ Enhancing Internet of Medical Things Security with Multi-Attention Convolutional LSTM-Based Intrusion Detection System, Transactions on Emerging Telecommunications Technologies, Wiley. ***(Major Revision Submitted)***<br>✓ Enhancing Internet of Medical Things Security: A Multi-Layered Approach Using Dynamic Adaptive Deep Reinforcement Learning and Blockchain, Computer and Electrical Engineering, Elsevier. ***(Major Revision Submitted)***<br>✓ A Novel Blockchain-Based Framework for Securing Electronic Health Records Using Least Squares Analysis, Cluster Computing, Springer. ***(With Editor)***<br>✓ A Quantum Neural Network-Assisted Hybrid Cryptographic Model for Secure Blockchain-Based EHR Systems. ***(To be communicated)*** |

# Chapter 2. Literature Review

This section delivers an in-depth evaluation of artificial intelligence (AI) and blockchain technology in the smart healthcare industry for identifying anomalies in the IoMT environment using IDS frameworks, which are reviewed in the following subsections.

## 2.1 ML for Intrusion Detection in IoMT

Machine Learning (ML) techniques have been widely explored for securing IoMT environments due to their capability to detect and classify malicious activities with high accuracy. The recent advancements in ML-based IDS frameworks have significantly improved anomaly detection and the overall security of IoMT systems. This subsection discussed the Several key studies in this area. Lazrek et al. [62] suggested a novel anomaly IDS (AIDS) framework for IoMT systems by integrating Recursive Feature Elimination (RFE) with ML and Ridge regression in DL models. This approach aims to boost accuracy and ease false alarm rates (FAR) in real-time healthcare data analysis. Gupta et al. [63] introduced a tree classifier-based network intrusion detection (NID) model specifically designed for IoMT networks by integrating Random Forest (RF) with robust feature scaling to augment anomaly detection effectiveness and accuracy. Haseeb et al. [64] proposed an ML model integrated with Software-Defined Networking (SDN) to enhance network resource management and security for IoT-based healthcare systems. It introduced a centralized SDN architecture and an unsupervised learning approach to optimize data delivery and reduce communication overheads. Zachos et al. [65] presented a novel AIDS specifically designed for IoMT networks, which integrates both host-based and network-based techniques to address the security challenges essential in these resource-constrained environments. Binbusayyis et al. [66] investigated and compared the performance of various ML algorithms for intrusion detection in IoMT networks, which identify the most effective algorithms for securing these networks against malicious activities. This research provided a thorough evaluation using multiple metrics, including ROC curve analysis, to assess algorithm effectiveness. Kulshrestha and Kumar [67] presented an ML-based IDS specifically designed for IoMT networks, which demonstrates that the Adaptive Boosting (AdaBoost) algorithm significantly outperforms other ML techniques in detecting cyber-attacks. The proposed IDS enhances the security and privacy of IoMT systems by effectively identifying cyber-attacks and improves the reliability and safety of healthcare data management during critical times such as the Covid-19 pandemic.

## 2.2 DL for Intrusion Detection in IoMT

Deep Learning (DL) approaches offer an enhanced ability to detect complex and evolving threats in IoMT networks. By employing advanced neural architectures, DL-based models have demonstrated improved accuracy and efficiency in intrusion detection. In this subsection, we review significant contributions from researchers in this domain which are explained as follows. Rbah et al. [68] introduced a hybrid DL approach that joins Graph Neural Networks (GNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) networks for enhanced intrusion detection in IoMT systems and achieved exceptional performance in threat detection. The authors demonstrated a novel and highly effective DL framework for IoMT security, which achieved nearly perfect accuracy and fast processing times and addressed the limitations of traditional intrusion detection methods. Ravi et al. [69] introduced a DL-based IDS for IoMT networks that integrates network flow and patient biometric

features, which is enhanced with a global attention layer and a cost-sensitive learning approach to tackle data imbalance and improve detection accuracy. The proposed IDS effectively fuses diverse feature sets and advanced learning techniques to attain high accuracy and robustness in identifying intrusions within IoMT systems, as well as outperforming existing methods by 3.9% on the IoMT intrusion dataset. Faruqui et al. [70] introduced SafetyMed, an innovative IDS that combines CNN and Long Short-Term Memory (LSTM) networks to protect IoMT devices from both malicious image data and sequential network traffic and achieved high detection accuracy and precision. R.M. et al. [71] proposed a hybrid PCA-GWO-based Deep Neural Network (DNN) classifier model for efficient intrusion detection in the IoMT environment, which enhanced classification accuracy by reducing data dimensions and improved attack detection and prediction through optimized network parameters. Alaalhareth and Hong [72] introduced the Logistic Redundancy Coefficient Gradual Upweighting Mutual Information Feature Selection (LRGU-MIFS) technique to address overfitting in IDS for IoMT by improving feature selection and redundancy estimation for high-dimensional data. Chaganti et al. [73] proposed a Particle Swarm Optimization Deep Neural Network (PSO-DNN) model to enhance the performance of IDS in IoMT by combining network traffic and patient biometric data and achieved better accuracy than existing ML and DL models. The PSO-DNN approach significantly improves intrusion detection accuracy in resource-constrained IoMT devices, demonstrating a 96% accuracy rate.

### 2.3 FL (Federated Learning) for Intrusion Detection in IoMT

47 FL has emerged as a promising solution to address privacy concerns and computational limitations in IoMT systems while maintaining effective intrusion detection capabilities. In this section, we explore recent developments in FL-based IDS frameworks specifically designed for IoMT environments. Tahir et al. [74] introduced a novel threat-defense framework for AI-enabled IoMT (AI-IoMT) systems using a deep deterministic policy gradient to address False Data Injection Attacks (FDIA). It integrates a federated, privacy-preserving FDIA detection model, which enhances system security and patient data privacy. Singh et al. [75] presented a Dew-Cloud-based hierarchical FL (HFL) model integrated with a hierarchical LSTM (HLSTM) network to enhance intrusion detection in IoMT systems. The model significantly improves data privacy, accuracy, and performance in resource-constrained IoMT environments. Alamleh et al. [76] developed a novel Multi-Criteria Decision-

13 Making (MCDM) framework to standardize and benchmark ML-based IDS for FL in IoMT applications. The evaluation criteria for ML-based IDSs are standardized using the Fuzzy Delphi Method (FDM), and the Borda voting method is employed for group benchmarking. Ioannou et al. [77] presented a three-stage IDS for Medical Internet of Things (MIoT) networks, which combined Enhanced Random Forests (ERF) for attack classification, One-Class SVM for anomaly detection, and FL for collaborative model updates and ensured real-time security with minimal resource usage. Zaabar et al. [78] proposed a Blockchain-based FL architecture for intrusion detection in IoMT environments, which replaced the central server with a Hyperledger Fabric channel to secure the learning process and prevent single points of failure. The proposed architecture enhances security and privacy in IoMT systems by integrating Blockchain with FL for intrusion detection. Alharbi [79] introduced a Federated Transfer Learning (FTL) IDS for IoMT applications, which enables personalized model learning for each client to address the challenges posed by data heterogeneity and non-IID data distribution in IoMT gateways. It improves detection accuracy from 95% to 99% while preserving data privacy and efficiently detecting zero-day attacks in IoMT environments. Zhong et al. [80] introduced an FL-guided IDS, which combined with an Artificial Neural

Network (ANN)-based key exchange mechanism within a blockchain framework to secure the IoMT systems, particularly in the Intensive Care Unit (ICU) area. It enhances security by mitigating data contamination, reducing computational requirements, and improving intrusion detection accuracy, particularly for botnets and critical healthcare environments. Begum et al. [81] presented the Blockchain-empowered FL-based IDS (BFLIDS), which was designed to improve the security and privacy of IoMT networks by integrating blockchain for secure transaction records, where FL technology was employed for privacy-preserving model training and IPFS/MongoDB for decentralized storage. It enhanced IoMT security by accurately detecting intrusions while maintaining data privacy and outperforming centralized methods with a decentralized and scalable approach.

### 2.4 XAI for Intrusion Detection in IoMT

The integration of Explainable Artificial Intelligence (XAI) into IDS for IoMT is gaining momentum as it provides insights into model decisions, gaining trust and transparency. This section discusses the latest research efforts that combine XAI with ML and DL models to enhance the interpretability of IDS in IoMT networks. Mane and Rao [82] introduced a framework that integrates XAI with DL models for network IDS. By employing XAI techniques such as SHAP, LIME, Contrastive Explanations Method (CEM), ProtoDash, and Boolean Decision Rules via Column Generation (BRCG), the framework aims to make the ML pipeline more transparent and interpretable. The proposed system explains model decisions that allow data scientists and network analysts to understand which features influenced predictions and thereby improve decision-making and model trustworthiness. Ayoub et al. [83] introduced an FL-based IDS with ANN for IoMT, which integrates XAI to enhance model explainability and privacy. Aljuhani et al. [84] presented a SaaS-based IDS for IoMT by applying PSO for feature engineering and an ensemble of ML and DL models for efficient and explainable attack detection. Shtayat et al. [85] proposed an explainable ensemble DL-based IDS for the Industrial Internet of Things (IIoT), which integrates Shapley additive explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) to enhance transparency and reduce false positives (FP). Mahbooba et al. [86] introduced an XAI approach for IDS using Decision Tree (DT) algorithms, which focused on enhancing interpretability and trust by providing clear decision rules and feature rankings. Patil et al. [87] presented an enhanced IDS by combining ML ensemble methods with the LIME algorithm for improved accuracy and explainability. Alani et al. [88] introduced an explainable ML ensemble for detecting attacks in IoMT environments, which achieved high accuracy and detailed interpretability using SHAP values. Khan et al. [89] introduced the XSRU-IoMT model, which leverages bidirectional simple recurrent units (SRUs) with skip connections for effective and efficient cyber-attack detection in IoMT networks while incorporating XAI to enhance interpretability. Kumar et al. [90] introduced a blockchain-enabled XAI approach that integrates Parallel Stacked LSTM networks with a multi-head attention mechanism for improved cyber threat detection in smart healthcare systems (SHS). It combines blockchain for secure data exchange and SHAP for model interpretability. **Table 2.1** shows the different proposed Intrusion detection framework for detecting anomalies in IoMT environments.

**Table 2.1.** Different proposed Intrusion detection framework for detecting anomalies in IoMT environments

| Author | Dataset | Significance | Pros | Cons | Result |
|---|---|---|---|---|---|
| Lazrek et al. [62] | WUSTL-EHMS dataset | The proposed framework enhances the security of IoMT systems by effectively detecting anomalies and cyber-attacks, | The efficient feature selection through RFE reduces the complexity and | Implementation in real-world may face challenges related to integration with | Training Accuracy = 0.99 |

| | | | | | |
|---|---|---|---|---|---|
| | | thereby safeguarding patient data and ensuring robust healthcare services. | improves the performance of the model. | existing IoMT systems and operational constraints | Test Accuracy = 0.9785, PR-0.9650, RE-0.8629, AUC - 0.9292, MCC-0.8402, FAR-0.03 |
| Gupta et al. [63] | WUSTL-EHMS dataset | The model addresses the critical need for secure and efficient intrusion detection in growing IoMT networks, demonstrating high accuracy and reduced classification time, which is essential for safeguarding patient data and ensuring reliable healthcare services. | It efficiently handles large, complex data with effective feature scaling and reduction.<br><br>It reduces classification time through dimensionality reduction and feature selection. | It has the potential for reduced effectiveness in detecting a broader range of attack types.<br><br>The future improvements in lightweight DL models and additional dimensionality reduction techniques are needed for enhanced stability and performance. | Training Accuracy = 0.9423, DR-0.9372, AUC -0.9068, PR=0.938, F1=0.938 |
| Zachos et al. [65] | TON_IoT | This work advances IoMT security by proposing an efficient AIDS that balances computational cost with effective anomaly detection, leveraging machine learning algorithms to improve the reliability of IoMT networks against malicious incidents. | It integrates host-based and network-based techniques for comprehensive monitoring. | The performance of the proposed system may vary depending on the dataset and feature dependencies. | AC-0.9996, PR-0.9989, RE-0.9995, F1-0.9992. |
| Faruqui et al. [70] | CIC-IDS 2017 | SafetyMed provides a robust solution for the increasing security challenges in IoMT systems by effectively balancing false positives (FP) and detection rates (DR), thus advancing protection against a range of attacks and contributing to the security of critical healthcare devices. | It is effective against multiple attack types, including malicious image data and sequential network traffic.<br><br>Low FAR (0.71%) indicates high reliability. | Architectural complexity and production costs could be prohibitive.<br><br>Limited real-world application and lack of defense against ML attacks and cyber-physical system security concerns. | AC=0.9763, DR=0.9801 |
| R.M. et al. [71] | Kaggle | The proposed model significantly improves intrusion detection accuracy by 15% and reduces training time by 32%, making it highly suitable for real-time IoMT security systems with faster alert mechanisms. | 15% improvement in classification accuracy.<br><br>32% reduction in time complexity, enhancing real-time detection. | Needs further evaluation for multiclass problem scenarios. | AC-0.999, RE-0.954, SPE-0.9992. |
| Alaalhareth and Hong [72] | WUSTL-EHMS-2020 | LRGU-MIFS effectively reduces overfitting and enhances detection accuracy by addressing feature redundancy nonlinearly, which leads to more compact and significant feature sets for IoMT intrusion detection. | It mitigates overfitting in high-dimensional datasets.<br><br>It is effective in handling insufficient attack patterns. | The proposed model needs further evaluation against diverse attack patterns. | AC=0.934 |
| Tahir et al. [74] | MIMIC dataset | The proposed framework effectively detects malicious activities while maintaining low computing costs and high accuracy and preserving the privacy of | Privacy-preserving federated approach. | The author focuses on other types of cyberattacks beyond FDIA. | PR-0.935, RE-0.93, F1-0.933. |

| | | | | | |
|---|---|---|---|---|---|
| | | patients' data in distributed IoMT environments. | Low computational cost and scalable for distributed environments. | | |
| Singh et al. [75] | TON_IoT and NSL-KDD | The proposed HFL-HLSTM model achieves high accuracy (99.31%) with minimal training loss, which ensures superior intrusion detection and data privacy in IoMT ecosystems. | The model enhanced data privacy through federated learning.<br><br>Its model is Scalable and efficient for resource-constrained IoMT systems. | The model may require optimization for delay reduction and heterogeneity handling. | AC-0.9931, PR-0.9897, RE-0.9824, F1-0.9858 |
| Ioannou et al. [77] | Network emulation (KALI penetration testing) with CICEV2023 | The proposed system effectively detects both known and unknown attacks in MIoT networks, while preserving privacy and minimizing energy consumption through FL. | The model detects anomaly (99.7%) efficiently with One-Class SVM.<br><br>Low resource utilization with FL and ensure privacy and quick updates | Limited attack types in the dataset; future research is needed for scalability. | AC-0.9998 |
| Mane and Rao [82] | NSL-KDD | The framework enhances trust in ML-based IDS by providing measurable explanations for decisions and increased interpretability for data scientists and network analysts. | It improves the interpretability of the DL model.<br><br>It enables informed decision-making by network analysts | High complexity in DNN. | Training Accuracy = 0.9823<br>Test Accuracy = 0.7950 |
| Ayoub et al. [83] | UNSW-NB15, ToN-IoT, NSL-KDD, WUSTL-EHMS | The framework offers a privacy-preserving solution for intrusion detection in IoMT, which combines FL for data security and XAI to provide transparency and compliance with healthcare regulations. | The model Preserves privacy using FL.<br><br>It enhances model interpretability with XAI. | Potential vulnerability to poisoning attacks.<br><br>The model requires further optimization of client selection and dataset partitioning. | AC-0.9656, PR-0.9778, RE-0.9826, F1-0.9802, AUC-0.999 |
| Aljuhani et al. [84] | WUSTL-EHMS-2020 | This IDS enhances security for resource-constrained IoMT devices by optimizing computational efficiency, which ensures real-time detection and provides transparency through SHAP. | Efficient feature engineering using PSO.<br><br>The model provides explainability with SHAP.<br><br>SaaS deployment at the edge ensures scalability. | Potentially high computational complexity in edge environments.<br><br>Limited performance in future threat scenarios without further optimization. | AC-0.9886, PR-0.953, RE-0.9399, F1-0.9464 |
| Shtayat et al. [85] | ToN_IoT dataset | This framework improves the interpretability of DL-based IDSs in IIoT networks, which allows cybersecurity professionals to understand decision logic and improve system resilience against evolving threats. | The model enhanced interpretability using SHAP and LIME.<br><br>It is effective in real-world IIoT scenarios with the ToN-IoT dataset. | Potentially high computational cost with deep learning models.<br><br>The model may require frequent updates to stay effective against new threats. | AC-0.9963, PR-0.998, RE-0.992, F1-0.995, SPE-0.9992 |

| | | | | | |
|---|---|---|---|---|---|
| Mahbooba et al. [86] | KDD benchmark dataset | This approach improves trust in IDS by offering interpretable decision-making processes, enabling cybersecurity experts to understand and validate the reasoning behind intrusion detection decisions. | This approach enhanced interpretability with simple DT rules.<br>It is computationally efficient and intuitive compared to complex algorithms. | Potential for overfitting with training data.<br>The model may perform poorly with data containing many categorical variables or attributes with numerous levels | PR-1.0, RE-1.0, F1-1.0 |
| Patil et al. [87] | CICIDS-2017 | The integration of ensemble methods with LIME enhances both the performance and interpretability of IDS, facilitating better trust and understanding in the detection process. | High classification accuracy (96.25%) using ensemble methods.<br>Improved model interpretability with LIME.<br>Effective in reducing false positives. | LIME might not address all aspects of model interpretability for complex models. | AC-0.9625, PR-0.89, RE-0.89, F1-0.89 |
| Alani et al. [88] | WUSTL-EHMS-2020 | The ensemble-based IDS demonstrates exceptional performance in detecting IoMT attacks with over 99% accuracy, and SHAP provides transparency into the decision-making process, enhancing trust and understanding. | The model provides clear insights into feature impact using SHAP values. | Focuses on specific datasets; generalization to other IoMT environments needs validation. | AC-0.9980, PR-0.9980, RE-0.9980, F1-0.9980 |
| Khan et al. [89] | ToN_IoT dataset | The XSRU-IoMT model addresses the vanishing gradient problem in recurrent networks, which provides high detection accuracy and interpretability in IoMT systems, which is crucial for building trust and enhancing security in healthcare applications. | Efficient training and reduced computational cost due to bidirectional SRU with skip connections.<br>Incorporates XAI for clear explanations of predictive decisions, enhancing trust and understanding. | Further optimization is required to reduce FP and adapt to varying IoMT network environments. | AC-0.9938, PR-0.9939, RE-0.9899, F1-0.9937 |
| Kumar et al. [90] | ToN-IoT, and IoT healthcare security | By integrating blockchain with XAI, the model enhances decision-making and trust in AI-based threat detection, which ensures data integrity and interpretability in SHS. | The author improves attack detection accuracy using Parallel Stacked LSTM networks and multi-head attention mechanisms. | Complex integration of blockchain and DL models may increase system overhead. | AC-0.9727, PR-0.94, RE-0.985, F1-0.962 |
| Rahmadika et al. [91] | - | This research addresses critical privacy and security challenges in IoMT by leveraging blockchain and FL, reducing the risk of false positives and bypass attacks in misbehaviour detection. The framework ensures privacy without compromising detection accuracy. | Secure and decentralized misbehaviour detection using blockchain. | Dependence on participating devices for effective FL implementation. | RE-0.9993,<br>Gas consumption - 84,456.5. |
| Oikonomou et al. [92] | - | This study highlights critical gaps in IoMT security and provides insights into leveraging blockchain for authentication, authorization, and intrusion detection in | Proposed future work focusing on security evaluation, computational cost, and communication | Limited experimental validation of reviewed security mechanisms. | - |

| | | | | | |
|---|---|---|---|---|---|
| | | healthcare monitoring systems, fostering trust in IoMT applications. | storage overhead of blockchain-based security solutions. | | |
| Begum et al. [81] | Edge-IIoTSet and TON-IoT datasets | This study enhances IoMT security by integrating blockchain and FL, ensuring decentralized, privacy-preserving, and scalable intrusion detection, which is crucial for protecting sensitive healthcare data. | Smart contracts enhance authentication and security. | FL introduces challenges such as slow model updates and device dropout. | AC-0.9821% |
| Gupta et al. [93] | NF-BoT-IoT-v2 and NF-ToN-IoT-v2 Datasets | This research enhances IoMT security by integrating blockchain and IDS, ensuring data confidentiality, integrity, and authentication while addressing the computational limitations of IoMT devices. | Low computational and communication overhead. | Scalability issues in large IoMT networks. Limited device processing power may hinder security module deployment. | F1-1.0, AUC-0.99 |
| Wang et al. [94] | TADA and TADB Datasets | This research enhances security in IoMT-Blockchain environments by improving anomaly detection, mitigating cyber threats, and protecting network traffic using deep learning-based feature extraction and classification techniques. | Residual learning optimizes anomaly classification. | High computational cost due to data transformation and deep learning. | PR-0.9150, RE-0.9070 |

**AC-Accuracy, RE-Recall, PR-Precision, AUC-Area under curve, F1-F1-Score, DR-Detection Rate, FAR-False Alarm Rate, SPE-Specificity, MCC-Matthews correlation coefficient.**

## 2.5 Blockchain based solution in healthcare industry

In recent years, integrating blockchain technology and deep learning architectures has emerged as a promising approach to enhancing intrusion and attack detection [95]. Blockchain offers a decentralized and immutable ledger that ensures data integrity and security, making it a powerful tool for safeguarding network systems against cyber threats. Concurrently, deep learning architectures provide advanced capabilities for detecting and analyzing complex network traffic patterns, significantly improving the accuracy and efficiency of intrusion detection systems. This section reviews the current state of research in these areas, exploring how the synergy between Blockchain and deep learning can address existing challenges in detecting anomalies and attacks within modern network environments.

Shukla et al. [96] introduced a novel integration of Fog Computing (FC) and Blockchain to enhance the security and reliability of healthcare IoT systems. The solution features a three-tier FC architecture, an analytical model, a mathematical framework, and an Advanced Signature-Based Encryption (ASE) algorithm to improve secure data transmission, device verification, and Patient Health Data (PHD) authentication in a decentralized setting. Shari and Malip [97] proposed a decentralized data dissemination scheme for smart healthcare that ensures data reliability, privacy, and accountability. It combines certificate-based signcryption with proxy re-encryption and an enhanced Delegated Proof-of-Stake (DPoS) consensus protocol for secure and efficient data sharing in healthcare communities. Moulahi et al. [98] proposed a robust healthcare system integrating Federated Learning with Blockchain technology to enhance medical data security and privacy. It addresses privacy and cybersecurity challenges in the Healthcare Internet of Things (HIoT) by establishing a trusted Federated Learning system on the Blockchain to predict diabetes risk while protecting sensitive health data accurately. Rehman et al. [99] proposed

a novel Internet of Medical Things (IoMT)-based hybrid blockchain architecture that combines decentralized Ethereum and centralized Hyperledger Fabric (Eth-Fab) using SQLite. This architecture uses Ethereum smart contracts with the Hyperledger permission model to enhance patient data authentication and authorization. The study also introduces access control strategies and employs machine learning algorithms to aid healthcare practitioners in disease detection and decision-making.

Wang et al. [100] introduced the hybrid blockchain-based health data sharing method (HSHB), designed to address privacy and security concerns. It separates data into private and alliance chains, applies access control policies, uses a B+ tree index for efficient querying, and ensures secure data exchange through agent re-encryption and smart contracts. Hossein et al. [101] presented BCHealth, an architecture that balances transparency and user data privacy in blockchain-based intelligent healthcare applications. BCHealth lets data owners set their access policies, enhancing security and privacy, and uses a clustering approach to improve scalability, reduce delay, and minimize overhead for practical healthcare use. Sharma et al. [102] proposed a distributed scheme using blockchain technology to enhance healthcare data security. By leveraging Blockchain's decentralization, confidentiality, and security, it overcomes the limitations of traditional cloud and client-server storage models. Smart contracts were utilized to enforce security rules and ensure secure data management. Farouk et al. [103] explored the integration of Blockchain and IoT in the healthcare sector, emphasizing their potential to improve information security, processing efficiency, and data management. It highlights the critical need for privacy and security in EHR exchange and proposes using AI and hybrid cloud models to enhance the adoption and effectiveness of Blockchain in healthcare. Shamshad et al. [104] proposed an innovative blockchain-based protocol designed for the secure and confidential storage and exchange of EHRs within the Telecare Medicine Information System (TMIS). The protocol leverages private blockchains to manage EHRs and consortium blockchains to maintain secure indexes. It employs public-key encryption and searchable keywords to enhance data security and implement robust access control mechanisms.

Sharma et al. [105] introduced a blockchain-based application for the healthcare sector designed to generate, maintain, and validate healthcare certificates. By integrating Blockchain with IoT systems, the application enhances security, privacy, and efficiency in managing healthcare data, serving as a bridge between the blockchain network and entities like hospitals, patients, doctors, and IoT devices, with smart contracts ensuring robust security features. Chen et al. [106] presented the Blockchain-based Trusted Medical Data Sharing (BTMDS) scheme to improve privacy, security, and efficiency in medical data sharing among multiple parties. It employs local differential privacy, searchable encryption, and a combined on-chain/off-chain storage approach to protect patient data and provide precise access control. Rehman et al. [107] advanced intelligent healthcare systems by combining blockchain technology with federated learning and an Intrusion Detection System (IDS). The integrated system improves security, privacy, and efficiency in healthcare monitoring within IoMT, ensuring data confidentiality and enabling accurate disease prediction and effective intrusion detection. Wang et al. [108] introduced GuardHealth, a decentralized blockchain framework designed to protect data privacy and facilitate the secure sharing of electronic medical records (EMRs). The system employs consortium blockchain and smart contracts to ensure confidentiality, authentication, data integrity, and controlled sharing. It incorporates a Graph Neural Network (GNN)-based trust model for detecting malicious nodes. It utilizes proxy re-encryption for secure data exchange, complementing an economic incentive mechanism to boost user participation. Azzaoui et al. [109] presented a

Quantum Cloud-as-a-Service (QCaaS) architecture that enhances the feasibility, efficiency, and security of complex computations in intelligent healthcare. Integrating Quantum Terminal Machines (QTM) with blockchain technology tackles challenges in tasks such as DNA analysis and molecular visualization.

Tomar et al. [110] introduced the BIoMTAKE protocol, which leverages Hyperledger Fabric to secure and authenticate IoMT device communications in healthcare. Using a private blockchain, BIoMTAKE eliminates reliance on a single trusted authority, mitigating unauthorized access, high latency, and system vulnerabilities. Rizzardi et al. [111] proposed an IoT-integrated blockchain architecture to improve tracking, traceability, security, and privacy in the healthcare supply chain. Utilizing Hyperledger Fabric establishes a secure and efficient platform for managing medical records and supply chain operations. Ali et al. [112] proposed a novel hybrid deep learning-based homomorphic encryption (HE) model combined with a consortium blockchain for securing Electronic Medical Records (EMRs) in the Industrial Internet of Medical Things (IIoMT). This approach addresses the challenges of data privacy, security, and latency associated with existing blockchain-based healthcare systems. Taloba et al. [113] proposed a security architecture for managing healthcare multimedia content through Blockchain technology. By combining IoT with Blockchain, the system enhances the security and management of patient data such as images, text, and audio by hashing each piece of information, ensuring that any unauthorized changes or breaches are traceable on the Blockchain. Tanwar et al. [114] investigated the use of blockchain technology to advance healthcare systems, focusing on enhancing Electronic Health Record (EHR) management, insurance billing, and database interoperability. It introduces an Access Control Policy Algorithm to facilitate data access among healthcare providers and employs Hyperledger Fabric for EHR sharing. The study also optimizes vital performance metrics like latency, throughput, and Round-Trip Time (RTT), showcasing significant improvements over conventional client-server models.

Zhang et al. [115] presented a blockchain-based e-health system that enhances the security and confidentiality of electronic health records (EHRs). Combining pairing-based cryptography with blockchain technology ensures tamper-proof EHRs and secure transactions via blockchain-based smart contracts. Zaabar et al. [116] proposed HealthBlock, a blockchain-based architecture to enhance the security and privacy of EHRs. It uses decentralized databases (OrbitDB with IPFS) for data storage and a Hyperledger Fabric blockchain for access control and data hash management, addressing vulnerabilities of centralized storage to improve healthcare system security. Chhikara et al. [117] proposed a blockchain-based framework for authenticated access control to enhance the management and security of medical data in distributed systems. Utilizing blockchain technology protects patient rights and regulates the distribution of licensed medical materials, overcoming the limitations of traditional centralized databases. Singh et al. [118] presented a secure architecture integrating Blockchain and Federated Learning (FL) to protect privacy in competent healthcare within smart cities. Blockchain ensures data security and privacy in IoT cloud platforms. At the same time, Federated Learning enables scalable machine learning without transferring personal data to the cloud, improving data privacy and model training efficiency. Mishra et al. [119] presented an interval-valued Pythagorean fuzzy Decision Support System (DSS) for evaluating and selecting blockchain platforms in the healthcare supply chain. The DSS uses multi-criteria evaluation with interval-valued Pythagorean fuzzy sets (IVPFSs) to address the complexities and uncertainties in choosing an optimal blockchain solution. **Table 2.2** presents the findings, advantages, and limitations of existing blockchain and deep learning-based approaches for anomaly detection in healthcare systems.

**Table 2.2.** Summary of Blockchain Networks in Healthcare: Findings, Advantages, and Limitations

| Ref. | Blockchain Network Type | Finding | Pros | Cons | Security Aspects | Uncovered Security Aspects |
|---|---|---|---|---|---|---|
| Shukla et al. [96] | Permissioned Blockchain | Supports frequent data transmission in IoT healthcare systems. | Ensures scalability and decentralization. | Requires significant computational and storage resources. | Confidentiality, Authentication, Availability, Tamper-Proof, Scalability, Distributed Trust | Energy Efficiency, Lightweight Processing |
| Shari and Malip [97] | Permissioned Blockchain | Holds malicious entities accountable while preserving patient privacy. | Sign-Proxy scheme ensures secure data transmission. | Challenges in low-resource environments. | Confidentiality, Authentication, Integrity, Tamper-Proof, Data Privacy, Secure Transactions | Lightweight Security, Low Latency |
| Moulahi et al. [98], 2023 | Permissioned Blockchain | Integrates FL for decentralized data privacy. | Addresses data security and privacy concerns in healthcare. | Scaling issues in handling large datasets. | Confidentiality, Authentication, Integrity, Tamper-Proof, Privacy, Decentralization | Low Overhead, Interoperability |
| Rehman et al. [99] | Ethereum, hyperledger | Combines Ethereum and Hyperledger to address privacy and scalability. | Ensures secure and trustless communication. | Potential interoperability issues. | Hybrid Security, Privacy Preservation | Performance Optimization, Lightweight Integration |
| Wang et al. [100] | Hybrid Blockchain | Uses B+ tree indexing for efficient querying of health data. | Combines private and consortium blockchain for better security. | Scalability issues with increasing users and transactions. | Indexing Efficiency, Hybrid Trust | Cost-Effectiveness, Speed Optimization |
| Hossein et al. [101] | Permissioned Blockchain | BCHealth enables data owners to set custom access policies. | Clustering optimizes data management. | Requires substantial modifications for system integration. | Data Ownership, Secure Storage | Performance Optimization, Resource Utilization |
| Sharma et al. [102] | Permissioned Blockchain | Eliminates centralized control in medical big data, ensuring privacy. | Smart contracts enhance protection against unauthorized access. | Increased latency and reduced throughput. | Privacy, Smart Contracts, Tamper Resistance | High-Speed Transactions, Low Latency |
| Farouk et al. [103] | Permissioned Blockchain | Integrates AI and hybrid clouds to enhance IoT-blockchain applications in healthcare. | Secure data management and privacy preservation. | Managing access control to prevent data leakage is a challenge. | Privacy Protection, Confidentiality | Performance Overhead, Interoperability |
| Shamshad et al. [104] | Private BC | Enables medical practitioners to access EHRs using patient-provided trapdoors, enhancing diagnosis and treatment. | Robust encryption and access control ensure patient privacy. | Scalability concerns large-scale deployments. | Confidentiality, Access Control, Tamper-Proof | Interoperability, Latency |
| Sharma et al. [105] | Permissioned Blockchain | Ensures confidentiality and privacy in healthcare data sharing. | Smart contracts enforce privacy protection. | Large-scale network performance is untested. | Confidentiality, Smart Contracts | Adaptive Performance, Network Optimization |

| | | | | | | |
|---|---|---|---|---|---|---|
| Chen et al. [106] | Public or Consortium Blockchain | Provides fine-grained access control for healthcare data. | Reduces decryption overhead for efficient sharing. | Managing blockchain and cloud storage introduces complexity. | Access Control, Data Encryption | System Usability, Low Resource Usage |
| Rehman et al. [107] | Permissioned Blockchain | BC Provides a tamper-proof method for managing healthcare data. | FL ensures patient data privacy. | Latency issues in large-scale deployments. | Privacy Preservation, Secure Transactions | High Scalability, Low Resource Consumption |
| Wang et al. [108] | Consortium Blockchain | Uses GNN to detect malicious nodes, improving the trust model. | Smart contracts and Proxy Re-encryption enable secure data sharing. | Detection accuracy is impacted when node count is low. | Trust Management, Secure Data Sharing | Performance Efficiency, Scalability |
| Azzaoui et al. [109] | Permissioned Blockchain | The QCaaS architecture leverages quantum computing for scalable healthcare computations. | The architecture is designed for growing demands in smart healthcare. | Quantum-blockchain integration is technically complex. | Computational Security, Privacy | Usability, Standardization |
| Tomar et al. [110] | Hyperledger Fabric, Permissioned Blockchain | BIoMTAKE protocol secures IoMT device communication. | Reduces reliance on a single trusted authority. | Scalability challenges in large IoMT networks. | Distributed Authentication, Secure Communication | High Efficiency, Lightweight Processing |
| Rizzardi et al. [111] | Hyperledger Fabric, Permissioned Blockchain | Enhances traceability of healthcare products. | Ensures privacy and access control. | Scalability concerns as transactions increase. | Supply Chain Security, Auditability | High-Speed Transactions, Storage Optimization |
| Ali et al. [112] | PBFT | Uses homomorphic encryption for privacy-preserving ML operations. | Ensures security throughout the data lifecycle. | Scalability improvements needed for real-world use. | Homomorphic Encryption, Secure ML | High Efficiency, Low Computational Cost |
| Taloba et al. [113] | Permissioned Blockchain | Optimizes healthcare resource distribution via IoT integration. | Hash-based record-keeping ensures data traceability. | Hashing and transaction recording introduce performance overhead. | Authentication, Tamper-Proof, Data Traceability, Integrity | Real-Time Analytics, Energy Efficiency |
| Tanwar et al. [114] | Permissioned blockchain | Enhances interoperability across healthcare databases, including patient records, prescription tracking, and device management. | Immutable ledger and tamper-resistant data improve security. | Complex implementation and management. | Data Integrity, Tamper-Proof, Access Control | High Throughput, Scalability |
| Zhang et al. [115] | Permissioned Blockchain | The integration of pairing-based cryptography to protect EHRs. | Ensures strong protection against data modifications. | Blockchain transactions introduce computational overhead. | Cryptographic Integrity, Authentication | Lightweight Execution, Cost-Efficiency |

| | | | | | | |
|---|---|---|---|---|---|---|
| Zaabar et al. [116] | Permissioned Blockchain | Addresses security threats like data breaches and tampering. | IPFS integration supports scalable data management. | Data retrieval from decentralized storage may be slow. | Data Confidentiality, Decentralization | Real-Time Processing, High Throughput |
| Chhikara et al. [117] | Permissioned Blockchain | Implements a robust access control system using blockchain. | Restricts access to authorized entities only. | High deployment and maintenance costs. | Role-Based Access Control, Authentication | Interoperability, Storage Efficiency |
| Tawfik et al. [120] | Permissioned (Hyperledger Fabric) | The framework ensures data privacy, secure collaboration, and computational integrity in a permissioned blockchain environment. | Ensures data confidentiality and privacy-preserving computations. | Computational overhead associated with homomorphic encryption | Secret-sharing for controlled data access, Homomorphic encryption for computations on encrypted data | Need for fully homomorphic encryption to enable more complex computations, Addressing computational overhead in secure processing |
| Jiang et al. [121] | Federated blockchain | T-BFL model improves trustworthiness, effectively identifies malicious nodes, and achieves over 90% accuracy across datasets, demonstrating robustness and adaptability | Enhances identity and behavioural trust in medical data sharing | Computational performance and communication efficiency remain challenges | Blockchain-based identity trusted registration scheme | Need for improved consensus mechanisms for large-scale blockchain deployment |
| Khan et al. [122] | Ethereum blockchain | BlockPres leverages IPFS to securely store patient prescriptions, enabling controlled access for hospitals, pharmacies, and clinics while ensuring patient data ownership. | Enhances data security and privacy through blockchain's immutability and IPFS storage | Potential scalability issues due to the inherent limitations of blockchain technology | Utilizes blockchain's immutability to prevent unauthorized data modifications | The study does not extensively address potential vulnerabilities related to smart contract exploits |
| Mazid et al. [123] | Blockchain used (Type not specified) | The framework secures IoMT data by enabling clients to train customized models locally, ensuring collaboration without data exposure. | Ensures privacy-preserving model training without centralizing patient data. | Communication efficiency and scalability challenges need further optimization. | Utilizes blockchain for secure global model updates, preserves privacy through FL by keeping patient data on local devices. | Does not extensively discuss protection against adversarial ML attacks |
| Wang et al. [124] | Hybrid (On-chain and Off-chain using IPFS) | The model ensures secure and efficient medical data access through hybrid | Constant ciphertext size reduces storage | Attribute revocation mechanism is not implemented. | Ensures patient privacy through hidden policy encryption, Resists collusion attacks. | Attribute revocation for access control needs further exploration. |

| | | |
|---|---|---|
| storage, hiding, searchable encryption, minimizing computational storage overhead. | policy and and | and computational overhead. |

## 2.6 Research Gaps

Based on the insights from recent studies, several significant research gaps have been identified within the field of intrusion detection systems (IDS) in IoMT environments which are explained and depicted in **Figure 2.1** as follows:

- **Lack of Real-Time Adaptive Encryption Strategies:** Traditional encryption uses static keys, making it vulnerable to evolving threats. Adaptive models for real-time key generation are largely unexplored.
- **Inadequate Multi-Layered Security for Healthcare Data:** Most healthcare systems apply isolated security techniques; a comprehensive multi-layered model addressing encryption, access control, and consensus is still lacking.
- **Centralized Data Storage Vulnerabilities:** Centralized systems risk single-point failures and data breaches. Decentralized, immutable storage solutions for healthcare data have not been widely integrated.
- **Limited Integration of Blockchain with Intrusion Detection:** Existing IDS frameworks lack blockchain integration, missing benefits like tamper-proof logs, transparent monitoring, and decentralized decision-making in threat detection.
- **Insufficient Deep Learning Architectures for Temporal and Spatial Analysis:** Current models fail to capture both spatial and sequential patterns in traffic data, limiting their ability to detect sophisticated intrusion scenarios.
- **Ineffective Handling of Imbalanced Datasets and Feature Irrelevance:** Many IDS approaches overlook preprocessing strategies like normalization and feature selection, leading to biased predictions and reduced model robustness.
- **Lack of Fine-Grained Evaluation Metrics in IDS Research:** Evaluations often rely solely on accuracy or F1-score, ignoring key metrics like LR+, FOR, and Informedness that reflect real-world performance.
- **Scalability Issues in High-Volume Healthcare Environments:** Existing security systems struggle under high data loads, lacking optimization for scalability and adaptability in large-scale healthcare deployments.
- **Absence of Secure Transaction Mechanisms for Medical Data Sharing:** Most systems don't ensure verifiable, tamper-resistant data sharing between patients and healthcare providers using blockchain-based consensus protocols.

13 By addressing these gaps, future research can significantly enhance the effectiveness of IDS in IoMT ecosystems.

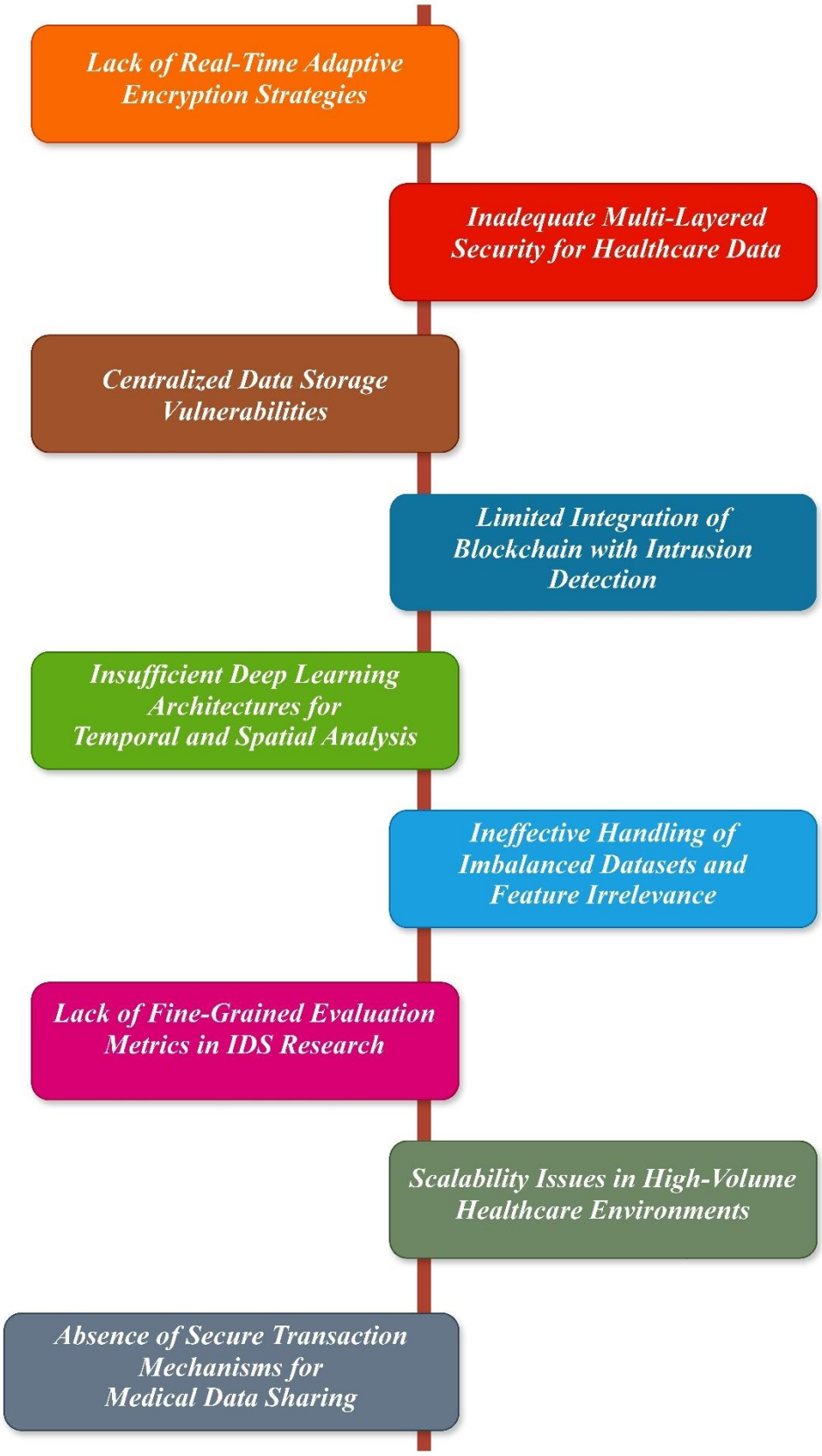



**Fig. 2.1.** Research Gaps

## 2.7 Publicly Available Datasets for IoMT

This section presents a comparative overview of publicly available datasets specifically designed for Internet of Medical Things (IoMT) security research. These datasets provide labelled network traffic data, encompassing various cyberattack types and real-world healthcare device interactions, offering valuable resources for developing and evaluating intrusion detection and prevention systems in IoMT environments. **Table 2.3** depicts the Publicly Available Datasets for IoMT.

**Table 2.3.** Publicly Available Datasets for IoMT

| Datasets | Description | No. of attacks | Attack Types | Labelled | IoMT Based |
|---|---|---|---|---|---|
| **WUSTL-EHMS-2020 [125]** | This dataset consists of electronic health monitoring system (EHMS) network traffic data, capturing real-time logs of healthcare IoT devices. It includes various types of network attacks such as data injection and spoofing which are types of man-in-the-middle (MITM) attacks. | 2 | Spoofing, Data injection | Yes | Yes |
| **WUSTL-HDRL-2024 [126]** | This dataset contains high-dimensional real-life (HDRL) intrusion data, collected from heterogeneous IoT networks. It comprises network logs from multiple device types, exhibiting a wide range of attack vectors, including adversarial attacks, botnet intrusions, and protocol-based exploits | 4 | Man-in-the-Middle (MiTM) attacks, Distributed Denial of Service (DDoS) attacks, Ransomware, and Buffer Overflow attacks | Yes | Yes |
| **ECU-IoHT [127]** | ECU-IoHT dataset captures network activity and cyberattacks in a healthcare IoT setup using devices like MySignals sensors, Windows 10, Kali Linux, and Bluetooth/Wi-Fi adapters. It includes 23,453 benign instances and multiple attack samples across four types: Smurf, ARP spoofing, Nmap scans, and DoS. The dataset supports deep learning model evaluation for IoHT security | 4 | Smurf Attack, ARP Spoofing, Nmap PortScan, and DoS Attack | Yes | Yes |
| **CIC-IoMT-2024 [128]** | The CICIoMT2024 dataset is a comprehensive benchmark for IoMT security, featuring traffic from 40 devices (25 real, 15 simulated) using protocols like Wi-Fi, MQTT, and Bluetooth. It includes 18 cyberattacks across five categories: DDoS, DoS, Recon, MQTT, and Spoofing, which captured through innovative methods. The dataset supports behavioural profiling and aids in evaluating ML-based IoMT security models. | 18 | DDoS UDP, DDoS ICMP, DDoS TCP, DDoS SYN, DoS UDP, DoS ICMP, DoS TCP, DoS SYN, ARP Spoofing, DDoS connect flood, Malformed data, DDoS publish flood, DoS connect flood, DoS publish flood, Port scan, Ping sweep, OS scan, and Recon VulScan | Yes | Yes |

### 2.8 Performance Evaluation Metrics

This section defines the metrics used to evaluate the effectiveness and efficiency of the proposed Intrusion Detection Model (IDM) and Blockchain Framework in IoMT systems. It includes standard statistical and cryptographic performance measures.

#### 2.8.1 Performance Measures for assessing proposed Intrusion Detection Model

This subsection outlines key evaluation metrics that assess how accurately the proposed IDS detects and classifies cyber threats in IoMT environments. **Table 2.4** shows the Standard Evaluation Metrics for detecting Intrusion in IoMT environment.

**Table 2.4.** Standard Evaluation Metrics for detecting Intrusion in IoMT environment

| Metrics | Description | Significance | Expression | Value Range |
|---|---|---|---|---|
| **True Positive (TP)** | In this case, both the forecasted and actual outcomes of the data point are valid. It is appropriately categorized as an attack sample | - | - | - |
| **True Negative (TN)** | In this case, the data point's forecasted and actual outcomes are false. Therefore, it is appropriately categorized as a standard sample. | - | - | - |
| **False Positive (FP)** | In this case, the forecasted outcome of the data point is precise, and the actual result of the data point is false. It designated a standard data set as an attack sample. | - | - | - |
| **False Negative (FN)** | In this case, the forecasted outcome of the data point is false, whereas the actual result is valid. Therefore, it was restricted as a standard sample despite being an attack sample | - | - | - |

| | | | | |
|---|---|---|---|---|
| **Ac** | Proportion of correct predictions among all predictions made. | Reflects the overall success of the IDS in detection. | $Ac = \frac{TP+TN}{TP+TN+FP+FN}$ | 0 to 1 |
| **PPV** | Ratio of true positives to all positive predictions made. | Measures the IDS's ability to avoid false alarms. | $PPV = \frac{TP}{TP+FP}$ | 0 to 1 |
| **TPR** | Ratio of true positives to all actual positive cases. | Assesses the IDS's ability to detect actual intrusions. | $TPR = \frac{TP}{TP+FN}$ | 0 to 1 |
| **F1** | Harmonic mean of precision and recall. | Balances precision and recall for reliable intrusion detection. | $F1 = \frac{2 \times TPR \times PPV}{TPR+PPV}$ | 0 to 1 |
| **TNR** | Ratio of true negatives to all actual negatives. | Measures the IDS's ability to avoid false positives. | $TNR = \frac{TN}{TN+FP}$ | 0 to 1 |
| **MCC** | Correlation between actual and predicted classifications. | Provides a balanced evaluation even with imbalanced data. | $MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$ | -1 to 1 |
| **NPV** | Ratio of true negatives to all negative predictions. | Measures IDS reliability in correctly classifying benign traffic. | $NPV = \frac{TN}{TN+FN}$ | 0 to 1 |
| **ROC_AUC** | Area under the ROC curve. | Measures overall IDS performance across different thresholds. | $\text{ROC\_AUC} = \int_0^1 TPR\, d(FPR)$ | 0.5 to 1 |
| **FDR** | Ratio of false positives to all positive predictions. | Evaluates how often positive detections are false alarms. | $FDR = \frac{FP}{FP+TP}$ | 0 to 1 |
| **FPR** | Ratio of false positives to all actual negatives. | Indicates the IDS's false alarm rate in detecting threats. | $FPR = \frac{FP}{FP+TN}$ | 0 to 1 |
| **FNR** | Ratio of false negatives to all actual positives. | Reflects the IDS's failure to detect actual intrusions. | $FNR = \frac{FN}{FN+TP}$ | 0 to 1 |
| **FOR** | Ratio of false negatives to all negative predictions. | Shows how often benign traffic is misclassified. | $FOR = \frac{FN}{FN+TN}$ | 0 to 1 |
| **MK** | Difference between PPV and false omission rate. | Summarizes IDS's prediction reliability across positives and negatives. | $MK = Pr + NPV - 1$ | -1 to 1 |
| **BM** | Difference between true positive rate and false positive rate. | Evaluates IDS's informed decisions about intrusions. | $BM = TPR + TNR - 1$ | -1 to 1 |

**Ac- Accuracy, PPV- Positive predictive value, TPR- True positive rate, F1- F1-Score, TNR, True negative rate, MCC- Matthews correlation coefficient, NPV- Negative predictive value, ROC_AUC-Area Under the Receiver Operating**

**Characteristic Curve, FDR- False discovery rate, FPR- False positive rate, FNR- False negative rate, FOR- False omission rate, MK- Markedness, BM- Informedness.**

### 2.8.2 Performance Measures for assessing proposed Blockchain Framework

This subsection introduces critical metrics for evaluating the blockchain framework's operational efficiency in IoMT, including encryption/decryption time, key generation time, throughput, latency, and fault tolerance. These metrics ensure secure, fast, and reliable data handling within decentralized healthcare systems. The standard metrics used for calculating the performance of the proposed blockchain Framework are represented as follows in **Table 2.5**.

**Table 2.5.** Performance Metrics and Their Computational Expressions for Blockchain and Cryptographic Operations

| Metrics | Description | Expression |
|---|---|---|
| **Encryption Time (ET)** | It measures the duration taken to convert plaintext data into ciphertext using a cryptographic algorithm. This metric is crucial in evaluating the efficiency of the encryption process Where $T_{enc}$ is the average encryption time, $f$ is the number of files or data blocks, $t_{enc,i}$ is the encryption time for $i-th$ the file or block. | $T_{enc} = \frac{1}{f}\sum_{i-1}^{f} t_{enc,i}$ |
| **Decryption Time (DT)** | It is the duration required to revert ciphertext back to plaintext using the cryptographic key. It is essential for assessing the performance of the decryption process Where $T_{dec}$ is the average encryption time, $f$ is the number of files or data blocks, $t_{dec,i}$ is the encryption time for $i-th$ the file or block. | $T_{dec} = \frac{1}{f}\sum_{i-1}^{f} t_{dec,i}$ |
| **Key generation time (KGT)** | Key generation time is the duration taken to create cryptographic keys using DA-DRL, which optimizes the key generation process dynamically | $T_{key} = \frac{1}{n}\sum_{i-1}^{n} t_{key,i}$ |
| **Block Creation Time (BCT)** | The duration required to generate a new block in the blockchain. It encompasses validating transactions, assembling the block, achieving consensus through the network's consensus mechanism, and propagating it to all nodes. This metric reflects the efficiency and speed of the blockchain's transaction processing and block generation Where $T_b$ is the average block creation time, $t_{b,i}$ is the time taken to create the $i-th$ block, and $N$ is the total number of blocks created. | $T_b = \frac{1}{N}\sum_{i=1}^{N} t_{b,i}$ |
| **Sharing Record Time (SRT)** | It is the duration required to share a record within the blockchain network, from the initial request to the successful update in the ledger | $T_{share} = T_{send} + T_{verify} + T_{commit}$ |
| **Restoration efficiency (RE)** | It measures the effectiveness of restoring data from the blockchain, often in the context of data recovery or reconstruction after failures Where $E_{rest}$ is the restoration efficiency, $D_{rest}$ is the amount of data successfully restored, and $D_{orig}$ is the original data before any loss or corruption. Where $E_{rest}$ is the restoration efficiency, $D_{rest}$ is the amount of data successfully restored, and $D_{orig}$ is the original data before any loss or corruption. | $E_{rest} = \frac{D_{rest}}{D_{orig}} \times 100\%$ |
| **Response time (RT)** | It is the total time taken from sending a request to receiving a response from the blockchain network. This includes processing and communication delays Where $T_{resp}$ is the response time, $T_{req}$ is the request transmission time, $T_{proc}$ is the processing time, and $T_{comm}$ is the communication time. | $T_{resp} = T_{req} + T_{proc} + T_{comm}$ |

| | | |
|---|---|---|
| **Throughput (Th)** | It measures the number of transactions processed by the blockchain network per unit time. It is a key indicator of the network's capacity Where $N_{trans}$ is the number of transactions processed, and $T_{total}$ is the total time taken. | $Throughput = \frac{N_{trans}}{T_{total}}$ |
| **Latency** | It is the delay from the initiation of a transaction to its confirmation in the blockchain network. Lower latency indicates a more responsive network Where $L_{trans}$ is the latency of a transaction, $T_{confirm}$ is the time of transaction confirmation, and $T_{init}$ is the time of transaction initiation. | $L_{trans} = T_{confirm} - T_{init}$ |
| **Fault Tolerance (FT)** | The system's capability to maintain functionality despite component failures is crucial for blockchain-enabled IoT healthcare systems. It is measured by restoration efficiency (RE), reflecting the system's recovery speed from faults where $T_{restored}$ is the time to recover, and $T_{total}$ is the total operational time. | $Fault\ Tolerance = \frac{T_{restored}}{T_{total}}$ |
| **Transaction Finality (TF)** | It refers to the duration needed for a transaction to become irreversible and permanently recorded on the blockchain. It is directly represented by block creation time and ensures prompt and secure data addition to the blockchain | $Transaction\ Finality = T_{block}$ |
| **User experience (UX)** | User experience in IoT healthcare systems is assessed by response time and sharing record time. It improves with shorter times for both metrics which indicates better user satisfaction with quicker interactions. | $UX \propto \frac{1}{T_{resp} + T_{share}}$ |
| **Network overhead (NO)** | It measures the extra resources required for managing and securing blockchain transactions, affected by encryption time ($T_{enc}$) and throughput ($Th$). A lower percentage denotes a more efficient system, reducing resource consumption and maintaining scalability. | $NO\ (\%) = \frac{(T_{enc})}{Th} \times 100$ |

# Chapter 3. Intrusion Detection Framework for Healthcare Systems

## 3.1 Introduction

The Internet of Things (IoT) has become a transformative technology across multiple domains, including healthcare, agriculture, smart cities, and environmental monitoring. With over 13.9 billion connected devices globally, a number expected to rise to 30.85 billion by 2025 [129] IoT enables seamless communication, data sharing, and real-time control through interconnected smart sensors, software, and physical devices. In healthcare, IoT has significantly improved medical services by enabling real-time monitoring, data-driven decision-making, and enhanced patient care through applications like Electronic Health Records (EHR) [130], Remote Patient Monitoring (RPM) [131], and preventive care solutions [132].

IoT has led to innovative developments in smart healthcare systems. These include remote monitoring that enables continuous observation of patient health, telemedicine solutions that allow remote diagnostics and consultations, and integrated applications that link healthcare with logistics and emergency services [133]. Specialized IoT systems now offer targeted care for vulnerable populations such as children and the elderly, while also improving medical record accuracy, storage efficiency, and mobile accessibility. Additionally, IoT enhances professional performance through real-time data, improving situational awareness and response times in clinical settings.

Despite these advantages, the integration of IoT into healthcare infrastructures introduces severe cybersecurity risks. IoT systems are prone to Denial of Service (DoS), Man-In-The-Middle (MITM) attacks, ransomware, data sniffing, and authentication issues [134]. These threats are particularly concerning resource-constrained IoT devices, which often lack the processing power for conventional security methods [135-136]. Notably, ransomware attacks such as the WannaCry incident that impacted the UK's National Health Service (NHS) which have exposed the vulnerability of critical healthcare systems, a problem exacerbated during the COVID-19 pandemic [137].

To address these threats, Intrusion Detection Systems (IDS) are widely used for identifying and mitigating attacks. However, traditional IDS approaches often fall short in keeping pace with the sophistication of modern cyber threats. As a solution, the integration of Deep Learning (DL) into IDS frameworks has emerged as a promising direction. DL models can learn complex patterns from large datasets, enabling adaptive, accurate, and real-time threat detection.

This research proposes novel DL-based IDS models: EmbedNet, ConvNet-SVM, and DeepSVM-Net which are designed specifically for IoMT security. These models were evaluated using benchmark datasets such as ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024. Results demonstrate that the proposed models outperform traditional approaches in accuracy and resilience. An ablation study further confirms their robustness and effectiveness in securing healthcare IoT systems against evolving cyber threats.

### 3.1.1 Motivation

The Smart Healthcare industry is promptly advancing, utilizing cutting-edge technology to enhance patient-centered services. However, this growth has also provided new opportunities for cyber attackers. To protect IoMT environments, numerous security measures have been developed, including authentication, access control, key management, encryption, and intrusion detection. This research specifically examines IDSs and the integration of

AI techniques into these systems. IDSs are intended to detect malicious behaviours at both host and network levels. These malicious behaviours present serious threats to the security of Smart Healthcare, potentially causing data breaches, healthcare data alterations, unauthorized access, and life-threatening effects for patients.

The rapid increase in digital data and the growing complexity of cyber-attacks have made robust IDS more important than ever. Traditional methods and classical ML models are becoming less effective against the advanced and evolving nature of cyber threats. These older approaches often depend on predefined signatures or simple statistical models, which makes it hard to keep up with new and complex attack strategies. As a result, they struggle with accuracy, adaptability, and scalability in the fast-changing world of cyber security. DL, a branch of AI, has emerged as a powerful tool to overcome these challenges. By using multiple layers of neural networks, DL models have shown superior performance in various areas, such as image and speech recognition, natural language processing (NLP), and, more recently, cyber security [138]. DL models can automatically identify and learn complex patterns from large amounts of data, which makes them well-suited for enhancing IDS capabilities. The motivation for this research is to demonstrate the advantages of DL over traditional methods and ML in the field of intrusion detection. DL models offer several key benefits:

- **Enhanced Feature Extraction:** DL models can identify complex, high-dimensional patterns in data that traditional methods often miss, which allows for the detection of intelligent and advanced attack signatures.
- **Adaptive Learning:** Unlike static models, DL frameworks can continuously learn and adapt from new data, which can help in improving their detection capabilities over time. This adaptability is essential in the ever-evolving cyber threat landscape.
- **High Accuracy and Reduced False Positives:** Deep neural networks (DNNs) enable IDS to distinguish between normal and malicious activities better. This reduces a common issue of false positives found in the traditional systems.
- **Scalability:** DL models can efficiently handle and analyze large volumes of network traffic data, which makes them suitable for large-scale environments. This scalability ensures that IDS can maintain high performance even as network sizes and data volumes increase.
- **Real-time Detection:** The advanced architectures of DL models support real-time processing and threat detection, which allows immediate responses to potential security breaches.

### 3.1.2 Major Contribution

The main contributions of this research are outlined as follows:

- **Design and development of Novel Deep Learning Models:** This study introduces several DL models, including the EmbedNet (A Categorical Embedding Neural Network), the collaborative ConvNet-SVM model (a combination of convolutional neural network (CNN) and support vector machine for Feature Extraction and Classification), and the DeepSVM-Net (A Deep Neural Network emulated by Support Vector Machine) model. These models are designed to identify network-based attacks on the IoMT. The EmbedNet model represents a significant advancement by utilizing a DL-driven embedding

methodology to detect intrusions within the IoMT environment using the ECU-IOHT (D1), NF-BoT-IoT (D2), and WUSTL-HDRL-2024 (D3) real-time dataset.

- **Advanced Preprocessing and Feature Engineering Techniques:** The research applies advanced preprocessing and feature engineering techniques, including eliminating irrelevant columns using a covariance matrix to enhance the detection process, which helps reduce memory usage and inference time. Then, the Data augmentation technique is employed, which balances the dataset by providing adequate data for normal and attack samples. Furthermore, Features are categorized into continuous and categorical columns where continuous columns are normalized using techniques like the Standard Scaler and Power Transformer, while categorical columns are transformed using ordinal encoders. This ensures effective normalization and transformation of the data.
- **Classification for Performance Evaluation:** The models' performance is evaluated through a classification approach that differentiates between benign network traffic and malicious activities, which include attacks like Smurf attack, ARP Spoofing, Nmap PortScan, DoS, DDoS, MITM, Reconnaissance, and Ransomware. This method thoroughly assesses the models' ability to detect various types of network intrusions.
- **Analysis of Epochs and Batch Size Effects:** This research examines the effects of epochs and batch size on IoMT intrusion detection performance, specifically by analyzing training loss (TL), validation loss (VL), and accuracy. This research provides insights into optimizing DL model performance for intrusion detection by examining these parameters.
- **Advanced Metrics Trade-off and Analysis:** For the first time, this research provides an in-depth trade-off and analysis of advanced metrics for the proposed DL models, including metrics such as Markedness (MK), Informedness (BM), Positive Likelihood Ratio (LR+), Negative Likelihood Ratio (LR-), False Omission Rate (FOR), and False Discovery Rate (FDR). These metrics offer a comprehensive evaluation of the models' performance and reliability.
- **Comparison with Existing Techniques:** The proposed models are rigorously compared with existing methods on similar real-time and benchmark datasets. The results demonstrate that these models outperform existing techniques by achieving the highest accuracy, Positive predictive value (PPV), True positive rate (TPR), and F1-Score for detecting attacks while minimizing inference time.

This chapter aims to thoroughly analyze the use of DL models in IDS frameworks and compare their performance with traditional techniques and ML models. Through rigorous experimentation and evaluation, we seek to demonstrate how DL can significantly advance the effectiveness, accuracy, and resilience of IDS by contributing to the broader field of cyber security.

## 3.2 Proposed Methodology

This section presents the proposed methodology, followed by the data preprocessing and feature engineering process, and overview and hyperparameter tuning of ML and DL approaches. We designed an IDS framework using ML and DL models for the IoMT ecosystem from the identified research gaps, as depicted in **Figure 3.1**. This framework provides a robust solution to accurately classify normal and malicious activities.

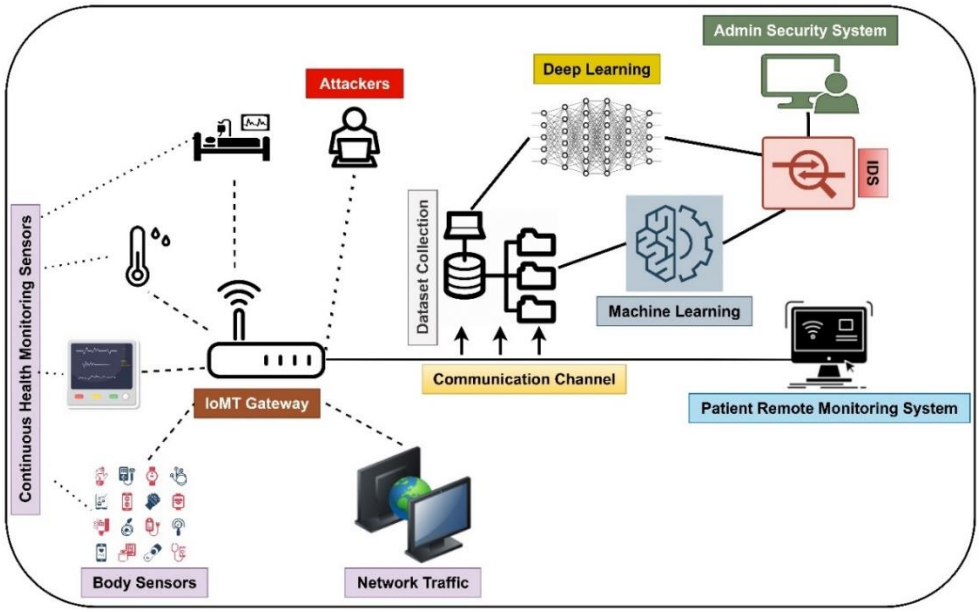



**Fig. 3.1.** IoMT Architecture

### 3.2.1 Data Preprocessing

Data preprocessing is a critical initial phase in ML and DL workflows to enhance model performance. This research implemented widely used techniques such as Standard Scaler [139] and Simple Imputer for data preprocessing.

Standard Scaler is a prominent method for data normalization, frequently utilized before training ML or DL models. This technique standardizes features by removing the mean and scaling to unit variance. It normalizes the input dataset's value distribution so that the mean is 0 and the standard deviations are 1. This standardization is essential for optimizing the functionality and accuracy of various algorithms by ensuring that all input features contribute equally to the model training procedure. The standard scaler can be mathematically expressed as follows in ***Eq. (3.1)***:

For a given feature $x$ in the dataset, the standardized value $z$ is calculated as:

$$z = \frac{x-\mu}{\sigma} \tag{3.1}$$

Where, $\mu$ depicts mean and $\sigma$ represents the standard deviation of the feature values. This technique involved several steps in the calculation of mean, standard deviation, and standardization of each feature value using ***Eq. (3.2-3.4)*** and feature scaling in ***Eq. (3.5)***:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3.2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3.3}$$

$$z_i = \frac{x_i-\mu}{\sigma} \tag{3.4}$$

Where, for each $i$ in $\{1, 2, \ldots \ldots \ldots, N\}$.

$$X_{FeatureScaled} = \frac{X_{FeatureCurrentValue} - mean(X_{Feature})}{StandardDeviation(X_{Feature})} \quad (3.5)$$

Where, $X_{FeatureScaled}$ represents the scaled feature value after the scaling process, $X_{FeatureCurrentValue}$ shows the unaltered current feature value; $X_{Feature}$ encompasses all the dataset's feature columns in the datasets and mean () represents the average value of each feature.

The Simple imputer addresses missing values within the dataset. This technique replaces the missing entries with statical values such as mean, median, or the most frequent value with a specified constant. It can be expressed as follows from the ***Eq. (3.6-3.8)***:

To begin with, ***Eq. (3.6)*** is used to calculate the mean imputation where $\varkappa_i$ shows an observed value, $\hat{x}$ represents imputed value, and depicts $\mu$ mean of the observed values. In addition, median imputation is calculated using ***Eq. (3.7)***, where $median(x)$ represents the median of the observed values. Furthermore, mode imputation is calculated using ***Eq. (3.8)*** where $mode(x)$ denoted the most frequent value among the observed values.

$$\hat{x} = \mu \quad (3.6)$$

$$\hat{x} = median(x) \quad (3.7)$$

$$\hat{x} = mode(x) \quad (3.8)$$

Power Transformer [140] and Standard Scaler were used to scale the numerical data in this research. These approaches are part of a family of monotonic and parametric transformations designed to convert skewed features into a more normal distribution, which often utilizes logarithmic transformations to achieve a Gaussian-like data distribution. Power Transformer is particularly effective in addressing issues related to heteroscedasticity and other conditions where the assumption of normality is essential for accurate modelling. By applying these transformations, the data's variance becomes more constant, improving the performance and reliability of the subsequent ML or DL models. ***Eq. (3.9)*** represented the mathematical expression of the Power transformer. Here, a parameter represented as λ is determined during the fitting procedure; the input data, denoted as $y$, is transformed to be more Gaussian-like by selecting the optimal value of λ.

$$y(\lambda) = \begin{cases} \frac{((y+1)^{\lambda}-1)}{\lambda}, & if\ y \geq 0\ and\ \lambda \neq 0 \\ \log(y+1), & if\ y \geq 0\ and\ \lambda = 0 \\ -\frac{((-y+1)^{2-\lambda}-1)}{(2-\lambda)}, & if\ y < 0\ and\ \lambda \neq 2 \\ -\log(-y+1), & if\ y < 0\ and\ \lambda = 2 \end{cases} \quad (3.9)$$

### 3.2.2 Feature Engineering

The ML and DL approaches employed in this work tackle data to create novel variables that were not part of the original training set. This enhancement involves operations such as mutation, combination, addition, and deletion to advance the dataset features and enhance the efficacy and efficiency of proposed models. This helps to achieve remarkable accuracy and performance. The feature selection mechanism is a key strategy that aims to lessen the dimensionality of the input variables by retaining only the most relevant features and eradicating the noise from the data.

### (i) Covariance matrix and feature selection

A Covariance matrix heatmap has been employed to visualize the linear correlations among various variables in the dataset. It helps to classify patterns and relationships between the data variables. Based on domain expertise, the variables with high correlations were considered redundant and excluded. Specifically, columns like Dir, SrcAddr, DstAddr, and DstMac were removed. In the covariance matrix, the correlation coefficient $r_{xy}$ between two variables, $x$ and $y$, is calculated using ***Eq. (3.10)***.

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (3.10)$$

Where the mean values of $x$ and $y$ are represented by $\bar{x}$ and $\bar{y}$ respectively. This coefficient helps to identify which columns to drop based on high correlation.

### (ii) Ordinal Encoders

The ordinal encoders [141] were employed to convert categorical columns, such as Flgs and SrcMac, into integer values that reflect the order of the labels. It follows two major encoding processes i.e. assign each unique category $c_i$ an integer value $e_i$ as shown in ***Eq. (3.11)***. In another process, replace each category with its corresponding integer rank. For a categorical feature $x_{cat}$ with categories $\{A, B, C\}$ where $rank\ (A) = 1, rank\ (B) = 2, rank\ (C) = 3$, the encoded feature $x_{enc}$ becomes $\{1, 2, 3\}$.

$$e_i = rank(c_i) \quad (3.11)$$

Subsequently, categorical columns, such as Flgs and SrcMac, were encoded using an Ordinal Encoder. This method converts each label into an integer value, preserving the order of the labels in the encoded data.

### 3.2.3 Dataset splitting and cross-validation

45 The dataset was divided into training and testing subsets with an $80-20\%$ split to accurately evaluate the performance of the ML and DL models. Additionally, performance has been considered with a $70-30\%$ split for comparison using ML models. K-fold cross-validation with 10-fold was employed on the training dataset to illustrate the adaptability in performance across different folds.

### 3.2.4 Solution to data imbalance problem

The dataset posed an imbalanced challenge, where standard samples comprised approximately 88% of the data. To address this, the Synthetic Minority Oversampling Technique (SMOTE) [142] was applied during the training phase to balance the dataset by generating new synthetic samples for the minority class. The fit_resample method from the imblearn library [143] was used to apply the SMOTE algorithm to the input data, denoted as $X$ and $y$. Here, $X$ is a $2D$ array-like structure representing the features and $y$ is a $1D$ array-like structure representing the target labels. As expressed in ***Eq. (3.12)***, this resampling process ensured a more balanced dataset and improved the model's ability to identify minority class instances.

$$X\_resampled, y\_resampled = smote.fit\_resample(X, y) \quad (3.12)$$

3 The quantity of resampled data generated by SMOTE is influenced by the parameters provided to the SMOTE object. By default, SMOTE creates synthetic samples for the minority class until it achieves a balanced class

distribution and results in an equal number of samples for both minority and majority classes. However, the degree of oversampling can be adjusted by setting the sampling_strategy parameter. For instance, if sampling_strategy is set to 0.5, SMOTE will oversample the minority class to reach 50% of the majority class size. Consequently, the minority class will have 50% as many samples as the majority class.

### 3.3 Overview of ML Models and hyperparameter tuning

This paper evaluates different pre-trained machine-learning (ML) algorithms, such as LR (linear regression), XGB (Extreme Gradient Boosting), DT (Decision Tree), GBT (Gradient-Boosted Trees), KNN (K-nearest neighbour), SVM (Support Vector Machine), and RF (Random Forest), on three datasets: ECU-IoHT, NF-BaIoT, and Wustl HDRL 2024 to recognize attacks in the IoMT ecosystem. **Table 3.1** depicts the ML classifier description with optimized hyperparameter tuning values. We employed 10-fold cross-validation for pre-trained ML model evaluation. In this process, each model underwent training with a range of parameter values to minimize the impact of data spitting and avoid overfitting.

**Table 3.1.** ML classifier description with optimized hyperparameter tuning values

| Models | Definitions | K-Fold | Hyperparameter | Optimized Values | Result Range |
|---|---|---|---|---|---|
| **LR** | Logistic Regression models the relationship between a dependent variable and independent variables using a linear function. | 10 | Max_iteration | 100 | 0-1.0 |
| | | | Random_state | 8 | |
| **XGB** | XGBoost builds an ensemble of decision trees using gradient boosting for high-speed predictive modeling. | 10 | N_estimators | 1000 | 0-1.0 |
| | | | Max_depth | 8 | |
| **DT** | Decision Tree splits data into branches based on feature values to determine outcomes. | 10 | Max_depth | 8 | 0-1.0 |
| **GBT** | Gradient Boosting Trees iteratively improve predictive accuracy by correcting previous errors. | 10 | Max_depth | 8 | 0-1.0 |
| | | | subsample | 0.5 | |
| | | | Max_leaf_node | 1000 | |
| | | | N_estimators | 1000 | |
| **KNN** | K-Nearest Neighbors classifies data points based on the majority class among their closest neighbors. | 10 | Leaf_size | 100 | 0-1.0 |
| | | | N_estimators | 2 | |
| **SVM** | Support Vector Machine classifies data by finding the optimal hyperplane to maximize class separation. | 10 | gamma | auto | 0-1.0 |
| **RF** | Random Forest constructs multiple decision trees and aggregates their outputs for accuracy. | 10 | Random_state | 1 | 0-1.0 |
| | | | N_estimators | 1000 | |
| | | | Max_leaf_node | 1000 | |

### 3.4 Discussion of proposed DL Approaches with training parameters

This section highlights the working architecture of deep learning (DL)-enabled proposed models, i.e. EmbedNet model, ConvNet-SVM Model, and DeepSVM (DeepSVM-Net) model, to detect intrusion in the IoMT-based ecosystem. **Table 3.2** depicts the setup Parameters values of DL approaches, which are explained as follows:

**Table 3.2.** The setup Parameters values of DL approaches

| Parameters | Values |
|---|---|
| Epochs | 100 |
| Embedding dimension | 16 |
| Optimizer | Adam |
| Batch Size | 64, 128, 256 |
| Activation Function | ReLU, TanH |
| Learning rate | 0.0001 |

### 3.4.1 EmbedNet (A Categorical Embedding Neural Network)

The EmbedNet model presented in this work provides a comprehensive solution for handling categorical and continuous features (Spatial Features) in DL, particularly in classification modelling. Unlike traditional approaches that handle these two types of features separately, the EmbedNet technique integrates them seamlessly within a unified neural network architecture, which captures complex relationships and interactions more effectively. The components of this model include Embedding layers for categorical features, batch normalization for continuous features, concatenation of features, linear layers with dropout and batch normalization (BNorm), and a final output layer as shown in **Figure 3.2**, which are explained as follows:

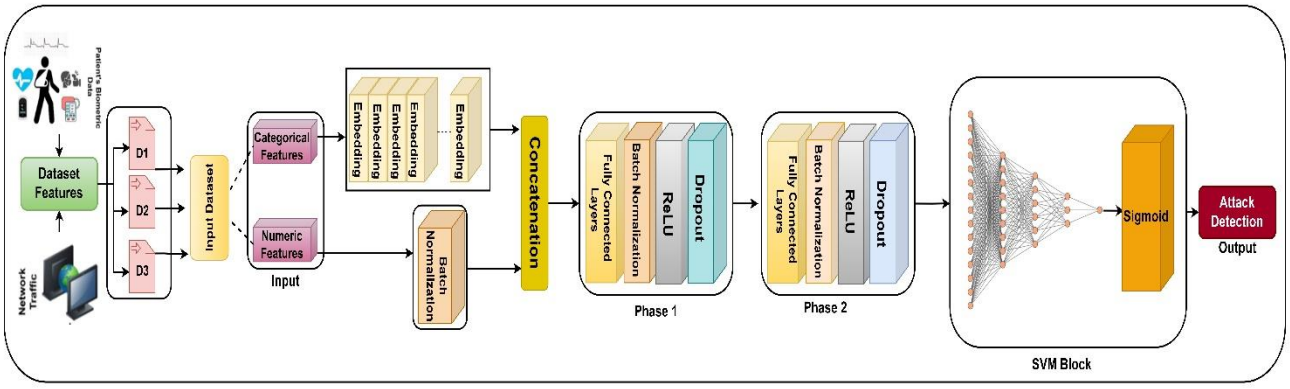



**Fig. 3.2.** Working Architecture of EmbedNet Model

***i) Embedding layers for categorical features***

For categorical features $(x_{cat})$ with cardinality $(C)$ and embedding dimension $(d)$, the embedding layer transforms the categorical feature into a dense vector as expressed in ***Eq. (3.13)***:

$$e_{cat} = Embedding(x_{cat}) \in \mathbb{R}^d \quad (3.13)$$

Which further mapped each categorical value to an index $i$ such that $i \in \{1, 2, \ldots\ldots, C\}$, the index $i$ is used to retrieve the corresponding vector embedding vector $e_i$ from the embedding matrix $E$ where $e_i = E[i]$.

***ii) Batch Normalization for Continuous Features***

For continuous features $x_{cont} \in \mathbb{R}^m$, batch normalization standardizes the inputs, which is expressed using ***Eq. (3.14)***:

$$x_{cont\_norm} = BatchNorm(x_{cont}) \quad (3.14)$$

This process involves the calculation of mean, variance, and normalization, which is expressed by using ***Eq. (3.15-3.17):***

$$\mu_{batch} = \frac{1}{m}\sum_{i=1}^{m} x_i \quad (3.15)$$

$$\sigma^2_{batch} = = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{batch})^2 \quad (3.16)$$

$$x_{cont\_norm} = \frac{x_{cont} - \mu_{batch}}{\sqrt{\sigma^2_{batch} + \epsilon}} \quad (3.17)$$

***iii) Concatenation of features***

The transformed categorical and continuous features are concatenated to form a combined feature vector, which is expressed by ***Eq. (3.18)***:

$$h_0 = [e_{cat1}, e_{cat2}, e_{cat3}, \ldots., e_{catn}, x_{cat1}] \quad (3.18)$$

Here, $h_0$ represents the initial hidden state combining all processed features.

***iv) Linear Layers with dropout and batch normalization***

The combined feature vector $h_0$ is passed through a series of linear layers with dropout and batch normalization, as expressed in ***Eq. (3.19)***, to improve learning and prevent overfitting.

$$h_{i+1} = Dropout(BatchNorm(ReLU(W_i h_i + b_i))) \quad (3.19)$$

Firstly, a linear transformation is applied where the vector is multiplied by the weight matrix $W_i$, and a bias vector $b_i$ is added, which results in ai as expressed in ***Eq. (3.20)***. The output $a_i$ is then passed through the ReLU activation function, which introduces non-linearity by setting all negative values to zero, as depicted in ***Eq. (3.21)***. This activated output hi is then standardized using batch normalization, which normalizes the output to improve training stability and convergence as depicted in ***Eq. (3.22)*** Finally, dropout is applied where a fraction of the neurons are randomly set to zero based on the dropout rate p, as represented in ***Eq. (3.23)***. This process reduces the risk of overfitting by preventing the model from becoming too dependent on any particular set of neurons.

$$a_i = W_i h_i + b_i \quad (3.20)$$

Where, $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$ and $b_i \in \mathbb{R}^{d_i}$ are the weights and biases of the $i^{th}$ layer.

$$h_i = ReLU(a_i) \quad (3.21)$$

*Where,* $ReLU(x) = max(0, x)$

$$h_i^{norm} = BatchNorm(h_i) \quad (3.22)$$

$$h_{i+1} = Dropout(h_i^{norm}, p) \quad (3.23)$$

***v) Final output layer***

The final output of the neural network performs a linear transformation followed by a sigmoid activation function to provide the prediction for binary classification as expressed in ***Eq. (3.24)***. In the linear transformation phase, the weight matrix $W_{final}$ multiplied with the last hidden layer output $h_{last}$ and the bias vector $b_{final}$ is added to it as depicted in ***Eq. (3.25)***.

$$\hat{y} = \sigma\left( W_{final} h_{last} + b_{final} \right) \quad (3.24)$$

$$a_{final} = W_{final} h_{last} + b_{final} \quad (3.25)$$

Where, $W_{final} \in \mathbb{R}^{1 \times d_{last}}$ and $b_{final} \in \mathbb{R}^{1 \times d_{last}}$.

Here, $W_{final}$ is a matrix with dimensions $1 \times d_{last}$, and $b_{final}$ is scalar. This linear transformation produces the input for the sigmoid function ($a_{final}$) which then passes through the sigmoid activation function as expressed in ***Eq. (3.26)*** to yield the final prediction as $\hat{y} = \sigma( a_{final} )$. The sigmoid function maps the final prediction output to a probability between 0 *and* 1, making it suitable for binary classification tasks to predict intrusion in the IoMT environment. **Algorithm 3.1** explains the working architecture of the Embedding neural network model.

$$\sigma( x ) = \frac{1}{1+e^{-x}} \quad (3.26)$$

**Algorithm 3.1. EmbedNet (A Categorical Embedding Neural Network)**

```
Input:
    - cat_dim: Dimension of categorical features
    - cont_dim: Dimension of continuous features
    - cat_cardinality: Array of cardinalities of categorical features
    - embd_dim: Dimension of embeddings for categorical features
    - layers_dim: Array of dimensions for hidden layers (default: [64, 128])
    - output_dim: Dimension of the output (default: 1)
    - drop_prob: Dropout probability (default: 0.5)
    - x: Input data tensor
Output:
    - x: Output prediction of attack classification
Initialization:
Define the function to initialize layers
void initialize_layers(string activation, string initialization, list layers);
Define a function to create linear layers with dropout and batch normalization
list linear_dropout_bn(string activation, string initialization, bool use_batch_norm, int in_units, int out_units, float dropout);
class EmbedNet extends nn.Module {
    // Constructor for the EmbedNet class
    void __init__(int cat_dim, int cont_dim, list<int> cat_cardinality, int embd_dim, list<int> layers_dim=[64, 128], int output_dim=1, float drop_prob=0.5) {
```

```
        // Call parent class constructor
        super(EmbedNet , self).__init__();
        // Initialize an empty list of layers
        list layers = [];
        // Create embedding layers for categorical features
        self.embedding_layers = nn.ModuleList();
        for (int i = 0; i < cat_cardinality.length; i++) {
            self.embedding_layers.append(nn.Embedding(cat_cardinality[i],embd_dim, padding_idx=0));
        }
        // Calculate the number of input units for the first linear layer
        int current_units = embd_dim * cat_dim + cont_dim;
        // Add linear layers with dropout and batch normalization
        for (int i = 0; i < layers_dim.length; i++) {
            list temp_layers = linear_dropout_bn("ReLU", "kaiming", true, current_units, layers_dim[i],
            drop_prob);
            for (int j = 0; j < temp_layers.length; j++) {
                layers.append(temp_layers[j]);
            }
            current_units = layers_dim[i];
        }
            // Combine linear layers into a sequential model
            self.linear_layers = nn.Sequential(layers);
            self.cont_norm = nn.BatchNorm1d(cont_dim);
            // Create a head layer with dropout and a final linear layer
            self.head = nn.Sequential(nn.Dropout(drop_prob), nn.Linear(current_units, output_dim));
            // Initialize head layer
            initialize_layers("ReLU", "kaiming", self.head);
    }
        // Forward pass method
        tensor forward(dict x) {
            // Separate continuous and categorical data
            tensor continuous_data = x["continuous"];
            tensor categorical_data = x["categorical"];
            // Embed categorical data
            tensor[] embedded_categorical_data = [];
            for (int i = 0; i < self.embedding_layers.length; i++) {
                tensor embedded = self.embedding_layers[i](categorical_data[:, i]);
                embedded_categorical_data.append(embedded);
            }
            // Concatenate embedded categorical data
```

```
59            tensor concatenated_embeddings = torch.cat(embedded_categorical_data, 1);
60            // Normalize continuous data and concatenate with embeddings
61            tensor normalized_continuous_data = self.cont_norm(continuous_data);
62            tensor combined_data = torch.cat([concatenated_embeddings, normalized_continuous_data],
              1);
63            // Pass data through linear layers
64            tensor output = self.linear_layers(combined_data);
65            // Pass through the head layer and apply sigmoid activation
66            output = self.head(output);
67            return torch.sigmoid(output);
68        }
69    }
```

### 3.4.2 ConvNet-SVM

The convolutional neural network (CNN) and support vector machine (ConvNet-SVM) model is another proposed DL model representing a novel approach to detecting intrusion in the IoMT environment. Unlike the traditional SVM model, which relies on manual feature engineering, ConvNet-SVM employs CNNs to automatically extract relevant features from raw input data. This model is specially designed to handle high dimensional input data, which is typical in network traffic and system logs used for intrusion detection. This model provides a robust framework for accurately identifying malicious activities in the network traffic and system logs that are used for intrusion detection by automatically extracting relevant features from the raw input data.

The architecture of this model consists of several key components: convolutional layers followed by batch normalization, hyperbolic tangent activation functions (tanh), a dropout layer for regularization, adaptive average pooling, a fully connected layer (FCL), and Sigmoid Activation Function (SAF) for classification as shown in **Figure 3.3**.

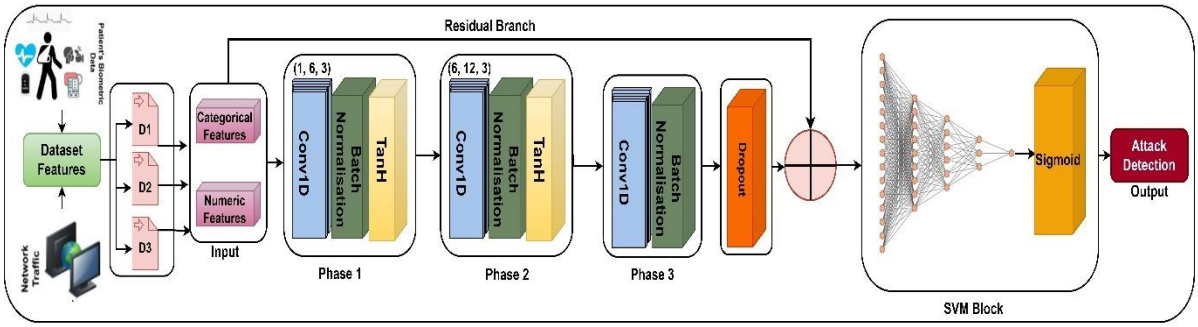



**Fig. 3.3.** Working Architecture of ConvNet-SVM Model

***i) Convolutional Layers***

The convolutional layers in this model play a vital role in feature extraction. These layers apply convolutional filters to the input data and obtain local designs and structures. By identifying these patterns, the ConvNet-SVM model can effectively detect intrusions in the IoMT environment.

As expressed in ***Eq. (3.27)***, the convolutional operation encompasses sliding a filter or kernel over the input data to produce feature maps.

$$Z_{i,j,k} = \tanh\left(\sum_{m-1}^{M}\sum_{n-1}^{N} X_{i+m-1,j+n-1}\, W_{m,n,k} + b_k\right) \quad (3.27)$$

Where, $Z_{i,j,k}$ represents the output feature map at the position $(i, j)$ for the k-th filter, $X_{i,j}$ depicts the input data at position (i, j), $W_{m,n,k}$ denotes the k-th convolutional filter at position $(m, n)$, shows the bias term for the k-th filter, and $tanh(.)$ depicts a hyperbolic tangent activation function.

The convolutional operation is made by sliding the filter over the input data. It performs element-wise multiplication, sums the results, adds the bias term, and finally applies the tanh activation function to produce the features map.

10 The network begins with a dropout layer to avoid overfitting by randomly setting a fraction of the input units to zero training. A hyperbolic tangent activation function is used, which scales the output between -1 and 1 and promotes non-linearity in the model. The architecture comprises of three convolutional layers $(self.conv1, self.conv2, self.conv3)$ with corresponding BN layers $(self.bn1, self.bn2, self.bn3)$. the first convolutional layer transforms the input x with one channel into six feature maps using a kernel of size 3 followed by BN to stabilize the learning process. The output of the first convolutional layer is expressed using ***Eq. (3.28)***.

$$x_1 = Tanh(BN1(Conv1(x))) \quad (3.28)$$

Where, Conv1 depicts convolution operation, and BN1 denotes Batch Normalization.

This process is repeated for the subsequent layers where the number of feature maps is increased to 12, and the in_features (input features) dimension is to default 40. It is represented using ***Eq. (3.29-3.30)***.

$$x_2 = Tanh(BN2(Conv2(x_2))) \quad (3.29)$$

$$x_3 = Tanh(BN3(Conv3(x_3))) \quad (3.30)$$

The final convolutional layer is followed by an adaptive average pooling layer, which lessens the dimensionality of each feature map to a single value. ***Eq. (3.31)*** effectively summarises the information.

$$x_{pool} = Pool(x_3) \quad (3.31)$$

A residual connection is added to integrate the original input x into the final feature representation, as depicted in ***Eq. (3.32).***

$$x_{residual} = x_{pool} + x \quad (3.32)$$

The resulting feature vector is passed through an FCL to provide the final output, which is activated by a sigmoid function to map the output to the range [0,1] as depicted in ***Eq. (33.3).***

$$\hat{y} = \sigma(FC(x_{residual})) \quad (3.33)$$

ConvNet-SVM model architecture effectively combines the strengths of convolutional layers for feature extraction and SVM-like linear decision boundaries for classification, which makes it suitable for complex sequential data analysis.

***ii) Dropout Layer***

ConvNet-SVM includes a dropout layer that randomly sets a fraction of the activation to zero during training to avoid overfitting. This helps to guarantee the model generalizes well to unseen data. The mathematical formulation of the dropout layer is expressed in ***Eq. (3.34).***

$$h^{(l)} = dropout(a^{(l)}, \mathrm{p}) \tag{3.34}$$

Where, $a^{(l)}$depicts the activations from the previous layer, $h^{(l)}$ represents the outputs after applying dropout with rate $p$, and $d^{(l)}$denotes mask generated from a Bernoulli distribution with parameter $1 - p$ as expressed in ***Eq. (3.35).***

$$d^{(l)} \sim Bernoulli(1 - p) \tag{3.35}$$

***iii) Adaptive Average Pooling***

ConvNet-SVM model uses an adaptive average pooling layer to handle inputs of varying lengths. It ensures a fixed-size feature map output regardless of the input size. This is particularly useful in intrusion detection, where network traffic can vary in length and size, which is calculated using ***Eq. (3.36).***

$$P = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} Z_{i,j} \tag{3.36}$$

Where P represents the Pooled Feature, and H and W denote the height and width of the feature map.

This operation involves summing all values in the feature map and averaging them to produce a fixed-size vector.

***iv) FCL and SAF***

The final layer of the ConvNet-SVM model is an FCL followed by a SAF, which outputs the probability of the input being a malicious activity, which is expressed in ***Eq. (3.37)***.

$$\hat{y} = \sigma(W_{final} \cdot P + b_{final}) \tag{3.37}$$

Where, $W_{final}$ depicts the weights of the FCL, $P$ shows the pooled feature vector, $b_{final}$ represents the bias term and $\sigma(x)$ denotes the SAF.

This layer produces a probability score, which makes it suitable for binary classification, such as determining whether network traffic is normal or indicative of an intrusion. **Algorithm 3.2** explains the working architecture of the ConvNet-SVM model.

**Algorithm 3.2. ConvNet-SVM**

**Input:**

- in_features: Number of input features (default: 40)

```
3           | - out_features: Number of output features (default: 1)
4           | - x: Input data tensor
5       Output:
6           | - x: Output prediction of attack classification
7       Initialization:
8           Define a function ConvNet-SVM _init to initialize the layers of the ConvNet-
            SVM model.
9           Parameters:
10          | - in_features: Number of input features
11          | - out_features: Number of output features
12  void ConvNet-SVM_init(float x[], int in_features, int out_features) {
13      Initialize layers
14      int drop=nn.Dropout();
15      int tanh=nn.Tanh();
16      int conv1=nn.Conv1d(1, 6, 3, padding=1);
17      int bn1=nn.BatchNorm1d(6);
18      int conv2=nn.Conv1d(6, 12, 3, padding=1);
19      int bn2=nn.BatchNorm1d(12);
20      int conv3=nn.Conv1d(12, in_features, 3, padding=1);
21      int bn3=nn.BatchNorm1d(in_features);
22      int pool=nn.AdaptiveAvgPool1d(1);
23      int fc=nn.Linear(in_features, out_features);
24      }
25      Forward Pass:
26      Begin forward pass through the ConvSVM network.
27  void ConvSVM(float x[], int in_features, int out_features) {
28      Convolutional layers
29      for (int i = 0; i < n_hid_layer; ++i) {
30          x = tanh(conv1(x));
31          x = tanh(conv2(x));
32          x = drop(bn1(x));
33          x = drop(bn2(x));
34          x = drop(bn3(x));
35          x = pool(x);
36      }
37      // Fully connected layer
38      for (int i = 0; i < n_hid_layer; ++i) {
39          x += nn;
40      }
41      }
```

### 3.4.3 DeepSVM-Net (A Deep Neural Network emulated by Support Vector Machine)

DeepSVM-Net is a neural network designed to emulate the behaviour of a Support Vector Machine (SVM) with a DL approach. This hybrid model uses the strengths of both neural networks and traditional SVM, which provides a robust solution for classification purposes in the context of intrusion detection in the IoMT environment. The network initialization (init) begins with the init_weights function, which applies Xavier uniform initialization to the weights of linear layers. This method is essential for maintaining the variance of the gradients during backpropagation, which is crucial for stable and efficient training of the DL model. The Xavier initialization sets the weights $W$ using the ***Eq. (3.38)***.

$$W \sim u\left(-\frac{\sqrt{6}}{\sqrt{n_{in}-n_{out}}}, \frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}\right) \quad (3.38)$$

Where, $n_{in}$ and $n_{out}$ are the number of input and output units in the weight set, respectively. The biases are initialized to a small constant value (0.01) to ensure a slight positive bias at the start of training.

The architecture of DeepSVM-Net, as shown in **Figure 3.4**, includes an input layer, a middle-hidden layer, and an output layer. This structure is suitable for detecting anomalies and intrusions in the IoMT environment by learning complex patterns in the data. The model is designed using the sequential neural network container, which allows for easy stacking of layers. The forward pass of the model applies these layers sequentially to the input data. This model architecture combines multiple hidden layers with non-linear activations and weights initialization, which depicts complex patterns in the data and provides a remarkable model for intrusion detection in the IoMT.

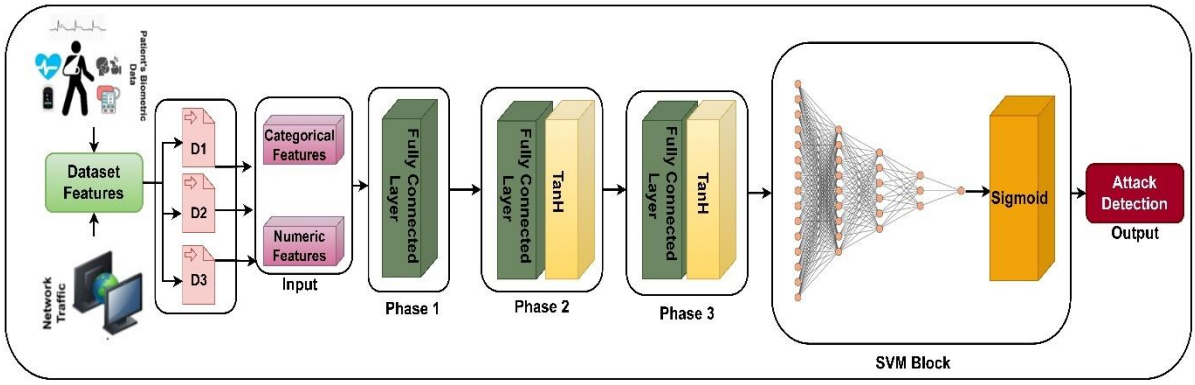



**Fig. 3.4.** Working Architecture of DeepSVM-Net Model

***i) Input Layer***

The input layer transforms the input features into a hidden representation of size, which is expressed using ***Eq. (3.39)***.

$$h_1 = Tanh(W_1 x + b_1) \quad (3.39)$$

Where, $x$ is the input vector of features, $W_1$ represents the weight matrix of the first layer, $b_1$ denotes the bias vector, $Tanh$ depicts the hyperbolic tangent activation function, and $h_1$ represents the output of the first layer.

***ii) Hidden Layers***

The hidden layers consist of stacked, FCL with Tanh activation functions. For the i-the hidden layer, the transformation is given by ***Eq. (3.40)***:

$$h_{i+1} = Tanh(W_{i+1}\, h_1 + b_{i+1}) \quad (3.40)$$

Where, $i = 1, 2, \ldots\ldots, n_{hid}$, $h_{i+1}$ represents the output of the $(i + 1)$-th layer, $h_{i+1}$ shows the weight matrix of the $(i + 1)$-th layer, and $b_{i+1}$ depicts the bias vector.

This repeated application of linear transformation and non-linear activations permits the network to learn complex patterns and representations in the data. This is crucial for identifying subtle anomalies indicative of security breaches in IoMT devices.

***iii) Output Layer***

The output layer maps the last hidden representation to the output with a single linear layer, which is followed by a SAF to produce a probability score for binary classification, which can be expressed using ***Eq. (3.41).***

$$\hat{y} = (W_{out} h_{n_{hid}} + b_{out}) \quad (3.41)$$

Where, $W_{out}$ represents the weight matrix and bias vector of the output layer, $\sigma$ depicts the sigmoid function.

The final output $\hat{y}$ represents the probability of the input being classified as an intrusion. The model can effectively flag potential security threats by setting an appropriate threshold that enhances the reliability and security of IoMT systems. This research highlights the efficacy of the hybrid model in improving predictive performance and generalization capabilities, which are essential for maintaining security in dynamic and complex IoMT environments. **Algorithm 3.3** explains the working architecture of the DeepSVM-Net model.

**Algorithm 3.3. DeepSVM-Net**

```
Input:
    - in_features: Number of input features
    - hidden_size: Size of the hidden layers (default: 32)
    - out_features: Number of output features (default: 1)
    - n_hid_layer: Number of hidden layers (default: 3)
    - x: Input data tensor
Output:
    - x: Output prediction of attack classification
Initialization:
    Define a function init_weights to initialize the weights and biases of the linear layers.
        - For each linear layer in the model:
            - Initialize weights using Xavier uniform initialization.
            - Initialize biases to a constant value (0.01).
void init_weights(float weights[], float biases[], int sizes[], int num_layers) {
    Initialize weights using Xavier uniform initialization
```

```
    for (int i = 0; i < num_layers; ++i) {
        Assuming Xavier initialization for weights
        float variance = 2.0 / (sizes[i] + sizes [i + 1]);
        for (int j = 0; j < sizes[i] * sizes [i + 1]; ++j) {
            weights [i * sizes[i + 1] + j] = sqrt(variance) * (float)rand() / RAND_MAX;
        }
            Initialize biases to 0.01
        biases[i] = 0.01;
    }
}
    SVM Initialization:
        Initialize SVM object with specified input and output features, hidden layer size, and
        number of hidden layers.
            - Initialize activation function as hyperbolic tangent (Tanh).
float svm(float x[], int in_features, int hidden_size, int out_features, int n_hid_layer) {
        Construct the layers of the model
    int sizes[n_hid_layer + 2];  // Sizes of layers including input and output
    sizes[0] = in_features;
    sizes[n_hid_layer + 1] = out_features;
    for (int i = 1; i <= n_hid_layer; ++i) {
        sizes[i] = hidden_size;
    }
        Create weight and bias arrays
    float weights[(n_hid_layer + 1) * hidden_size * hidden_size];
    float biases[n_hid_layer + 1];
    //Initialize weights and biases
    init_weights(weights, biases, sizes, n_hid_layer + 1);
    // Create the model layers
    float layers[(n_hid_layer + 1) * hidden_size + out_features];
          for (int i = 0; i <= n_hid_layer; ++i) {
        // Input layer
        for (int j = 0; j < sizes[i + 1] * sizes[i + 2]; ++j) {
            layers[i * sizes[i + 1] + j] = (float)rand() / RAND_MAX;
        }
        // Hidden layers
        for (int j = 0; j < sizes[i + 1] * sizes[i + 2]; ++j) {
            layers[i * sizes[i + 1] + j] = (float)rand() / RAND_MAX;
        }
        // Output layer
        for (int j = 0; j < sizes[i + 1] * sizes[i + 2]; ++j) {
```

```
55              layers[i * sizes[i + 1] + j] = (float)rand() / RAND_MAX;
56          }
57      }
58      // Forward Pass:
59      // Begin forward pass through the SVM network.
60      for (int i = 0; i < n_hid_layer; ++i) {
61          // Pass input tensor x through the model
62          // Apply the sigmoid activation function to the output tensor
63          layers[i] = sigmoid(layers[i]);
64      }
65      // Return the final prediction result of the attack classification
66      return layers[n_hid_layer + 1];
67  }
68  End Algorithm
```

These DL-based models have been evaluated using three different benchmark datasets like ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024 dataset. Out of these three datasets, two datasets, i.e. ECU-IoHT and WUSTL-HDRL-2024 datasets, are generated by combining the IoT traffic with healthcare features samples. **Table 3.3** depicts the training parameters of the proposed DL approaches.

**Table 3.3.** Training Parameters of the Proposed DL Approaches

| **Training Parameters** | **DL Models** | | | | | |
|---|---|---|---|---|---|---|
| | **EmbedNet** | | **ConvNet-SVM** | | **DeepSVM-Net** | |
| Initial state or attributes | layers_dim | 64, 128 | in_features | 40 | in_features, | |
| | drop_prob | 0.5 | | | out_features | 1 |
| | output_dim | 1 | out_features | 1 | hidden_size | 32 |
| | | | | | n_hid_layer | 3 |
| Learning Rate | 0.0001 | | 0.0001 | | 0.0001 | |
| Optimizer | Adam | | Adam | | Adam | |
| AdaptiveAvgPool1d | × | | ✓ | | × | |
| Batch Size | 64, 128, 256 | | 64, 128, 256 | | 64, 128, 256 | |
| Fully Connected Layer | ✓ | | × | | ✓ | |
| BatchNorm1d | ✓ | | ✓ | | × | |
| Conv1D | × | | ✓ | | × | |
| Activation Function | ReLU | | TanH | | TanH | |
| SVM Block | ✓ | | ✓ | | ✓ | |
| Dropout | ✓ | | ✓ | | × | |
| Residual Branch | × | | ✓ | | × | |

**Figure 3.5** depicts the workflow of the Proposed DL model architecture. In this research, we designed three deep learning models: EmbedNet, ConvNet-SVM, and DeepSVM-Net, which are evaluated on three different real-time

datasets: ECU-IoHT, NF-BoT-IoT, and Wustl-HDRL-2024. These datasets are comprised of normal IoT-based traffic flow and attack samples. To begin with, the data preprocessing phase employed several techniques, including power transformers, simple imputation, and standard scaling, to clean and standardize the raw data for effective model training. Feature engineering played a crucial role in this phase, involving methods such as the SMOTE to address class imbalance and using a covariance matrix to identify and remove irrelevant features. These steps were integral in enhancing the training process and improving the models' ability to detect attacks. In addition to data preprocessing, the dataset was split into training and testing sets using 70-30 and 80-20 distributions. This organized approach allowed for a thorough evaluation of the ML and DL models across different training and testing scenarios. This research incorporated DL models, including EmbedNet, ConvSVM-Net, and DeepSVM-Net, which were specifically designed to advance network-based attack detection in IoMT ecosystem. The effectiveness of these models was rigorously assessed through binary classification methods aimed at distinguishing between benign network traffic and various threats, including Smurf attacks, ARP Spoofing, Nmap PortScan, DoS, DDoS, MITM, Reconnaissance, and Ransomware. The evaluation metrics demonstrated that the DL models achieved superior performance in attack detection, as demonstrated by enhanced accuracy, PPV, TPR, and F1 scores while maintaining minimal inference time. The use of advanced data augmentation techniques, feature segregation strategies, and systematic column reduction via covariance matrix analysis optimized memory usage and reduced prediction time. However, the integration of these DL models into the framework underscores their efficacy in strengthening network security for IoMT environments.

The proposed models i.e., EmbedNet, ConvNet-SVM, and DeepSVM-Net differ in both structure and theoretical foundation, offering distinct advantages for intrusion detection in IoMT environments. EmbedNet is specifically designed to integrate categorical and continuous spatial features within a deep learning framework, employing embedding layers, batch normalization, and dropout mechanisms to enhance feature representation. Unlike traditional approaches that separately process categorical and numerical data, EmbedNet unifies them, making it particularly effective for analyzing diverse IoMT data sources. ConvNet-SVM, in contrast, combines the feature extraction power of CNNs with the classification strength of SVMs, making it highly suitable for structured network traffic data. CNNs capture spatial correlations and hierarchical patterns in network logs, while SVMs provide robust decision boundaries, improving generalization in intrusion detection. Lastly, DeepSVM-Net bridges neural networks and SVM principles, incorporating SVM-based margin maximization within a deep learning structure. This hybrid approach ensures the model not only learns hierarchical feature representations but also benefits from SVM's capacity to distinguish between complex decision boundaries, making it highly effective for detecting subtle attack patterns. These models collectively address IoMT-specific challenges such as heterogeneous data formats, varying attack signatures, and real-time anomaly detection, thereby enhancing security and resilience in IoMT systems.

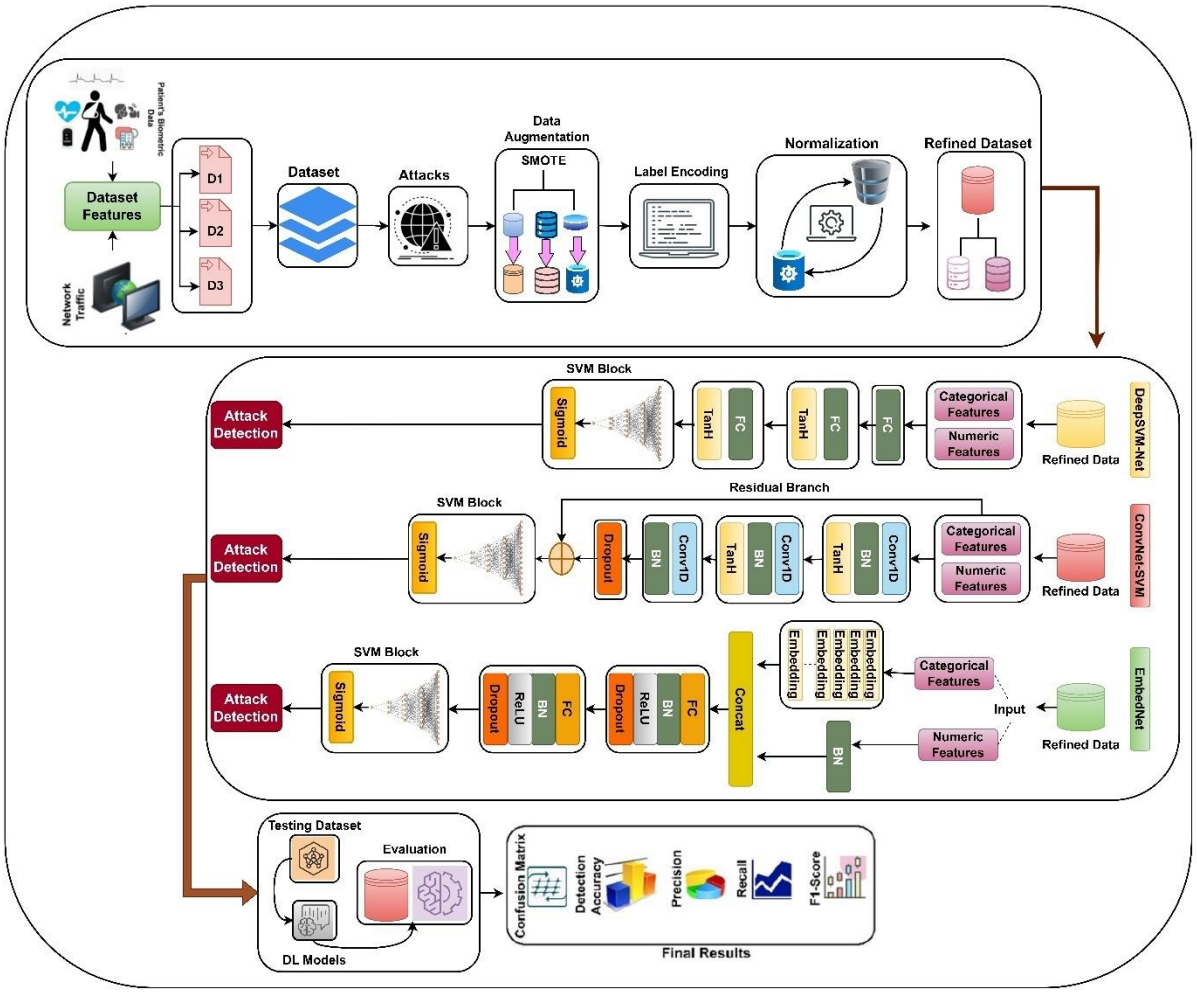



**Fig. 3.5.** Workflow of Proposed DL Models Architecture

## 3.5 Experimental Result and discussion

This section provides in-depth applicability of ML classifier and proposed DL models on three benchmark real-time datasets. This section includes subsections like experimental setup and overview, dataset description, and performance evaluation metrics.

### 3.5.1 Experimental setup

The experiment was conducted on a Windows 11 system with 32 GB DDR5 RAM and 13$^{th}$ Generation Intel Core i7-13620H Processor operating at 4.9 GHz. Python libraries such as PyTorch, which is well known for its deep neural network capabilities, and Scikit-learn libraries [144], which provide a robust software environment for training and testing ML and DL models, were used within the Visual Studio Code (VSCode) platform.

### 3.5.2 Dataset Description

In this research, the DL models have been trained and tested using three real time datasets i.e., ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024. Out of these three datasets, two are IoMT-based datasets, which comprise both network and healthcare samples, which are explained as follows:

**(i) ECU-IoHT**

We evaluated our deep learning model using the ECU-IoHT (Edith Cowan University - Internet of Health Things) dataset [145], which includes network flow activity and cyber-attack samples in the healthcare sector. This dataset has been generated using the Windows 10 operating system, Kali Linux, a mobile Wi-Fi hotspot, a wireless network adapter, and a Bluetooth adapter. These are all interconnected devices that provide internet access for the hosts. The setup also incorporates a healthcare kit named MySignals, which is equipped with various sensors that monitor and record patients' physiological data, including heart rate (HR), blood pressure (BP), and body temperature (BT). These sensor data are then transmitted to users' cloud storage. The ECU-IoHT dataset comprises seven key network data features: source (Src), destination (Dst), protocol, and specific attack types. It contains 23,453 instances of regular network activity and numerous cyber-attack instances, as depicted in **Table 3.4**. We classify these attacks into four categories: Smurf attacks, Address Resolution Protocol (ARP) spoofing, Network Mapper (Nmap) port scans, and Denial-of-Service (DoS) attacks.

**Table 3.4.** Description of data features in ECU-IoHT

| Dataset Samples | Description | Count | After SMOTE |
|---|---|---|---|
| Smurf Attack | A DoS attack where large numbers of ICMP echo requests are sent to IP broadcast addresses, overwhelming the target with responses. | 77,920 | 77,920 |
| ARP Spoofing | An attack that links the attacker's MAC address with the IP address of a legitimate network device, allowing interception of data. | 2359 | 77,920 |
| Nmap PortScan | A technique to discover open ports and services on a target system by sending various packets and analyzing responses. | 6836 | 77,920 |
| DoS Attack | An attack that makes a service unavailable by overwhelming it with illegitimate requests, exhausting resources or bandwidth. | 639 | 77,920 |
| Benign | Normal, non-malicious network traffic represents typical communication within the network. | 23,453 | 23,453 |

**(ii) NF-BoT-IoT**

The NF-BoT-IoT (NetFlow version of the UNSW-Bot-IoT dataset) dataset [146] is an IoT NetFlow-based dataset derived from the BoT-IoT dataset. It was constructed by extracting features from publicly accessible pcap files and categorizing flows based on their respective attack types. This dataset comprises 600,100 data flows, where 97.69% are classified as attack instances and 2.31% as benign, as depicted in **Table 3.5**. It encompasses four distinct attack categories: Theft, Reconnaissance, DoS, and DDoS attack.

**Table 3.5.** Description of data features in NF-BoT-IoT

| Dataset Samples | Description | Count | After SMOTE |
|---|---|---|---|
| Theft | Attacks focused on acquiring sensitive information, such as through data theft or keylogging. | 1,909 | 4,70,655 |
| Reconnaissance | Techniques used to gather information about network hosts are also referred to as probing activities. | 4,70,655 | 4,70,655 |

| | | | |
|---|---|---|---|
| DoS | DoS attacks are intended to overwhelm system resources, rendering services unavailable. | 56,833 | 4,70,655 |
| DDoS | DDoS attacks that involve multiple sources aiming to disrupt service. | 56,844 | 4,70,655 |
| Benign | Non-malicious traffic represents normal network activity. | 13,859 | 13,859 |

**(iii) WUSTL-HDRL-2024**

The WUSTL-HDRL-2024 dataset's [126] testbed has been designed to emulate a dynamic 5G network environment. This dataset addresses the need for comprehensively capturing datasets with a wide range of interactions and security threats within 5G networks. It includes six essential components: a 5G Core, Local Network, MEC (Multi-access Edge Computing) servers, UE (User Equipments), Insider Attacker, and Routing system. An Ubuntu 20.04 computer employs Simu5G to simulate 5G network operations, which expands its capabilities to include multiple MECs and external hosts so that Data can originate from the 5G Core. The local network uses stateful IP/ICMP translation (SIIT) to bridge IPv6 and IPv4 communications and resolve compatibility issues within the simulated 5G environment. MEC servers are critical in managing edge computing tasks, monitoring network traffic, and detecting intrusions. UEs with various operating systems connect directly to the 5G network or via the Local Network, which presents diverse end-user scenarios. An Insider Attacker machine executes a series of attacks to evaluate the network's defensive strategies. Finally, a central Router oversees data management in the testbed, ensuring the development of a comprehensive and unpredictable attack dataset for thorough security analysis. The dataset includes four categories of security breaches: Man-in-the-Middle (MiTM) attacks, Distributed Denial of Service (DDoS) attacks, Ransomware, and Buffer Overflow attacks.

To begin with, MiTM attacks involve manipulating or intercepting data exchanged between UEs and MEC systems, which risks data confidentiality and integrity. DDoS attacks inundate MECs with large requests and disrupt their regular operations. Ransomware encrypts critical files and demands payment for decryption, whereas Buffer Overflow attacks exploit software vulnerabilities to execute unauthorized code. **Table 3.6** depicts the features of the WUSTL HDRL 2024 dataset and the total number of samples in **Table 3.7**.

**Table 3.6.** Description of data features in WUSTL-HDRL-2024

| Column A | Description | Column B | Description |
|---|---|---|---|
| Dir | Direction of the network flow | RTime | Real-time at which the network flow occurred. |
| Flgs | Flags indicating the state or features of the network flow. | Packet_num | Sequential number of the packet in the flow. |
| SrcAddr | Source IP address from which the network traffic originates. | scputimes_user | CPU time spent in user mode. |
| DstAddr | Destination IP address to which the network traffic is directed. | scputimes_nice | CPU time spent in user mode with low priority. |
| Sport | Source port number used in the network flow. | scpustats_ctx_switches | Number of context switches. |
| Dport | Destination port number used in the network flow. | scpustats_interrupts | Number of interrupts handled. |
| SrcBytes | Number of bytes sent from the source. | scpustats_soft_interrupts | Number of software interrupts handled. |
| DstBytes | Number of bytes sent to the destination. | scpustats_syscalls | Number of system calls made. |
| SrcLoad | Load on the source during the network flow. | svmem_total | Total physical memory available. |

| | | | |
|---|---|---|---|
| DstLoad | Load on the destination during the network flow. | svmem_available | Available physical memory. |
| SrcGap | Gap or delay in the source transmission. | svmem_percent | Percentage of memory used. |
| DstGap | Gap or delay in the destination transmission. | svmem_used | Amount of memory used. |
| SIntPkt | Interval between packets sent from the source. | svmem_free | Amount of free memory. |
| DIntPkt | Interval between packets sent to the destination. | svmem_active | Amount of active memory. |
| SIntPktAct | Actual interval between packets from the source. | svmem_inactive | Amount of inactive memory. |
| DIntPktAct | Actual interval between packets to the destination. | svmem_buffers | Amount of memory used for buffers. |
| SrcJitter | Variability in packet arrival time from the source. | svmem_cached | Amount of memory used for cache. |
| DstJitter | Variability in packet arrival time to the destination. | svmem_shared | Amount of memory shared between processes. |
| sMaxPktSz | Maximum packet size sent from the source. | svmem_slab | Amount of memory used by the kernel data structures. |
| dMaxPktSz | Maximum packet size sent to the destination. | ram_usage_warning | Indicator for high RAM usage warning. |
| Dur | Duration of the network flow. | sswap_total | Total swap memory available. |
| Trans | Number of transmissions during the network flow. | sswap_used | Amount of swap memory used. |
| TotPkts | Total number of packets in the network flow. | sswap_free | Amount of swap memory free. |
| TotBytes | Total number of bytes in the network flow. | sswap_percent | Percentage of swap memory used. |
| network_load | Load on the network during the flow. | sswap_sin | Swap memory swapped in from disk. |
| Loss | Packet loss during the network flow. | sswap_sout | Swap memory swapped out to disk. |
| pLoss | Percentage of packet loss during the network flow. | sdiskusage_total | Total disk space available. |
| pSrcLoss | Percentage of packet loss from the source. | sdiskusage_used | Amount of disk space used. |
| pDstLoss | Percentage of packet loss to the destination. | sdiskusage_free | Amount of free disk space. |
| Rate | Transmission rate of the network flow. | sdiskusage_percent | Percentage of disk space used. |
| SrcMac | MAC address of the source. | Boot_Time_with_date | Boot time of the system with date. |
| DstMac | MAC address of the destination. | DTime | Time duration of the network flow. |
| IMEI | International Mobile Equipment Identity, relevant for mobile network flows. | Attack_categories | Categories of attacks identified in the network flow. |

**Table 3.7.** Total number of samples in WUSTL-HDRL-2024

| Samples | Types | Samples Values | After SMOTE |
|---|---|---|---|
| Total no. of attack samples | DDoS | 9971 | 132884 |
| | MiTM | 1672 | 132884 |
| | Ransomware | 528 | 132884 |
| | Buffer_Overflow | 68 | 132884 |
| Total no. of normal samples | Benign | 132884 | 132884 |

### 3.5.3 Experimental overview

This subsection explores the effectiveness of machine learning (ML) and deep learning (DL) models in detecting intrusions within the IoT ecosystem. The performance of these models has been assessed using standard metrics, including accuracy, PPV, TPR, F1-score, TNR, MCC, NPV, and ROC-AUC [147]. Following the initial performance evaluation, an error analysis was also conducted employing metrics such as FOR, FDR, FPR, FNR, likelihood ratios (LR+ and LR-), MK, and BM. The experimental flow of the research is illustrated in **Figure 3.6**.

The experimental framework utilized three datasets containing IoT traffic flows with cyber-attack samples: ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024. The data preprocessing phase employed several techniques, including power transformers, simple imputation, and standard scaling, to clean and standardize the raw data for effective model training. Feature engineering played a crucial role in this phase, involving methods such as SMOTE to address class imbalance and using a covariance matrix to identify and remove irrelevant features. These steps enhanced the training process and the model's ability to detect attacks. In addition to data preprocessing, the dataset was split into training and testing sets using 70-30 and 80-20 distributions. This organized approach thoroughly evaluated the ML and DL models across different training and testing scenarios. This research incorporated several DL models, including EmbedNet, ConvSVM-Net, and DeepSVM-Net, specifically designed to advance network-based attack detection in the IoMT ecosystem. The EmbedNet approach is proposed as a novel embedding technique introduced in this research, which is utilized for classifying attacks in IoT-based networks. The effectiveness of these models was rigorously assessed through binary classification methods aimed at distinguishing between benign network traffic and various threats, including Smurf attacks, ARP Spoofing, Nmap PortScan, DoS, DDoS, MITM, Reconnaissance, and Ransomware. The evaluation metrics demonstrated that the DL models achieved superior performance in attack detection, as demonstrated by enhanced accuracy, PPV, TPR, and F1 scores while maintaining minimal inference time. Advanced data augmentation techniques feature segregation strategies and systematic column reduction via covariance matrix analysis optimized memory usage and reduced prediction time. However, integrating these DL models into the framework underscores their efficacy in strengthening network security for IoMT environments.

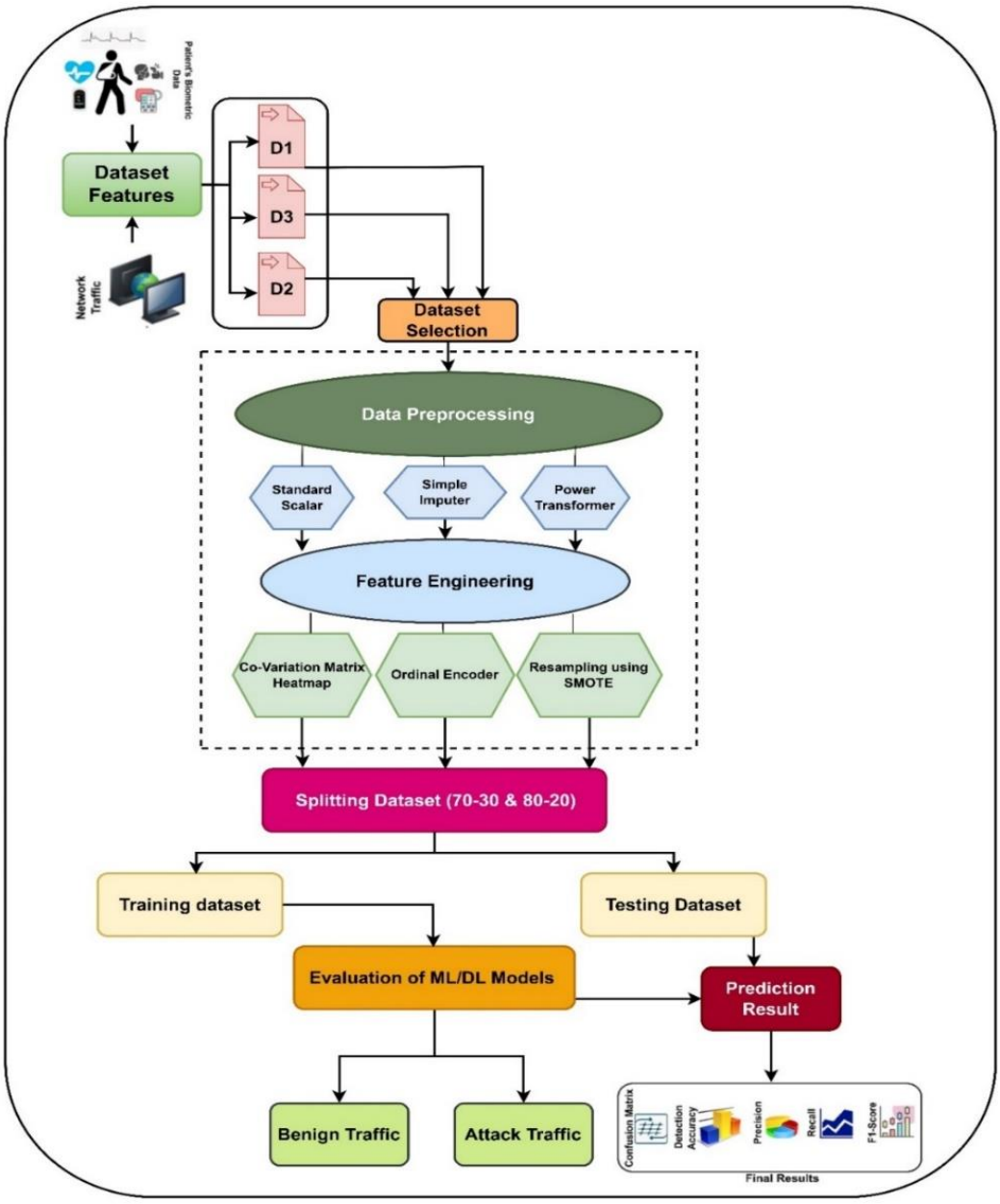



**Fig. 3.6.** Experimental Flowchart of Proposed Framework for Detecting Attacks in IoMT Environment

### 3.6 Performance analysis of Machine Learning models

This subsection presents the performance analysis of ML models on the three real-time datasets which are explained as follows:

#### 3.6.1 Performance analysis of ML Models for 10-fold cross-validation

**Table 3.8** compares ML models' performance for 10-fold cross-validation using pre-trained ML models: LR, XGB, DT, GBT, KNN, SVM and RF after hyperparameter tuning. All models have achieved remarkable average accuracy in detecting attacks on three different datasets: ECU_IoHT (D1), NF-BoT-IoT (D2), and WUSTL-HDRL-2024 (D3). Out of these models, the XGB approach achieved the highest performance on every dataset

with an average accuracy of 0.9979 (D1), 0.9999 (D2), and 0.9998 (D3), whereas the LR model achieved the lowest accuracy of 0.9394 on D2.

**Table 3.8.** Performance comparison of ML models for 10-fold cross-validation

| Model | Dataset | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Avg. Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **LR** | **ECU_IoHT** | AC | 0.9785 | 0.9790 | 0.9750 | 0.9765 | 0.9775 | 0.9780 | 0.9770 | 0.9760 | 0.9795 | 0.9760 | 0.9778 |
| | | PPV | 0.9800 | 0.9810 | 0.9770 | 0.9780 | 0.9790 | 0.9800 | 0.9790 | 0.9780 | 0.9810 | 0.9780 | 0.9791 |
| | | TPR | 0.9770 | 0.9780 | 0.9730 | 0.9750 | 0.9760 | 0.9765 | 0.9750 | 0.9740 | 0.9785 | 0.9740 | 0.9757 |
| | | F1 | 0.9785 | 0.9795 | 0.9750 | 0.9765 | 0.9775 | 0.9780 | 0.9770 | 0.9760 | 0.9795 | 0.9760 | 0.9778 |
| | **NF-BoT-IoT** | AC | 0.9395 | 0.9391 | 0.9394 | 0.9396 | 0.9395 | 0.9388 | 0.9394 | 0.9396 | 0.9394 | 0.9394 | 0.9394 |
| | | PPV | 0.9204 | 0.9241 | 0.9235 | 0.9211 | 0.9222 | 0.9253 | 0.9220 | 0.9217 | 0.9227 | 0.9217 | 0.9225 |
| | | TPR | 0.9610 | 0.9596 | 0.9602 | 0.9596 | 0.9600 | 0.9618 | 0.9603 | 0.9598 | 0.9587 | 0.9606 | 0.9602 |
| | | F1 | 0.9403 | 0.9415 | 0.9415 | 0.9400 | 0.9407 | 0.9432 | 0.9407 | 0.9404 | 0.9404 | 0.9408 | 0.9409 |
| | **WUSTL** | AC | 0.9996 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 |
| | | PPV | 0.9997 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 |
| | | TPR | 0.9995 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 |
| | | F1 | 0.9996 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 |
| **XGB** | **ECU_IoHT** | AC | 0.9979 | 0.9981 | 0.9979 | 0.9977 | 0.9979 | 0.9980 | 0.9981 | 0.9978 | 0.9981 | 0.9979 | 0.9979 |
| | | PPV | 0.9962 | 0.9966 | 0.9955 | 0.9968 | 0.9961 | 0.9964 | 0.9951 | 0.9963 | 0.9963 | 0.9958 | 0.9961 |
| | | TPR | 0.9919 | 0.9908 | 0.9908 | 0.9917 | 0.9914 | 0.9904 | 0.9902 | 0.9929 | 0.9907 | 0.9920 | 0.9913 |
| | | F1 | 0.9941 | 0.9937 | 0.9931 | 0.9942 | 0.9937 | 0.9934 | 0.9926 | 0.9946 | 0.9935 | 0.9939 | 0.9937 |
| | **NF-BoT-IoT** | AC | 0.9999 | 0.9999 | 0.9999 | 0.9998 | 0.9998 | 0.9998 | 0.9999 | 0.9998 | 0.9999 | 0.9998 | 0.9999 |
| | | PPV | 0.9998 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9998 | 0.9997 | 0.9998 | 0.9998 |
| | | TPR | 0.9995 | 0.9994 | 0.9995 | 0.9996 | 0.9997 | 0.9997 | 0.9996 | 0.9998 | 0.9996 | 0.9996 | 0.9996 |
| | | F1 | 0.9997 | 0.9996 | 0.9996 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9998 | 0.9997 | 0.9997 | 0.9997 |
| | **WUSTL** | AC | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 |
| | | PPV | 0.9998 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 |
| | | TPR | 0.9997 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 |
| | | F1 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 |
| **DT** | **ECU_IoHT** | AC | 0.9939 | 0.9942 | 0.9940 | 0.9939 | 0.9938 | 0.9942 | 0.9942 | 0.9937 | 0.9943 | 0.9940 | 0.9940 |
| | | PPV | 0.9996 | 0.9997 | 0.9994 | 0.9991 | 0.9989 | 0.9990 | 0.9994 | 0.9992 | 0.9997 | 0.9995 | 0.9993 |
| | | TPR | 0.9891 | 0.9873 | 0.9878 | 0.9883 | 0.9877 | 0.9871 | 0.9868 | 0.9895 | 0.9875 | 0.9887 | 0.9880 |
| | | F1 | 0.9943 | 0.9935 | 0.9936 | 0.9937 | 0.9932 | 0.9930 | 0.9930 | 0.9943 | 0.9936 | 0.9941 | 0.9936 |
| | **NF-BoT-IoT** | AC | 0.9962 | 0.9960 | 0.9960 | 0.9961 | 0.9961 | 0.9960 | 0.9962 | 0.9968 | 0.9961 | 0.9967 | 0.9962 |
| | | PPV | 0.9929 | 0.9937 | 0.9940 | 0.9934 | 0.9937 | 0.9936 | 0.9929 | 0.9946 | 0.9932 | 0.9951 | 0.9937 |
| | | TPR | 0.9985 | 0.9988 | 0.9989 | 0.9981 | 0.9986 | 0.9989 | 0.9987 | 0.9988 | 0.9984 | 0.9985 | 0.9986 |
| | | F1 | 0.9957 | 0.9962 | 0.9965 | 0.9957 | 0.9962 | 0.9963 | 0.9958 | 0.9967 | 0.9958 | 0.9968 | 0.9962 |
| | **WUSTL** | AC | 0.9971 | 0.9972 | 0.9973 | 0.9972 | 0.9972 | 0.9973 | 0.9971 | 0.9973 | 0.9973 | 0.9972 | 0.9972 |
| | | PPV | 0.9973 | 0.9974 | 0.9975 | 0.9974 | 0.9974 | 0.9975 | 0.9973 | 0.9975 | 0.9975 | 0.9974 | 0.9974 |
| | | TPR | 0.9969 | 0.9970 | 0.9971 | 0.9970 | 0.9970 | 0.9971 | 0.9969 | 0.9971 | 0.9971 | 0.9970 | 0.9970 |
| | | F1 | 0.9971 | 0.9972 | 0.9973 | 0.9972 | 0.9972 | 0.9973 | 0.9971 | 0.9973 | 0.9973 | 0.9972 | 0.9972 |
| **GBT** | **ECU_IoHT** | AC | 0.9971 | 0.9975 | 0.9937 | 0.9976 | 0.9971 | 0.9972 | 0.9970 | 0.9952 | 0.9973 | 0.9976 | 0.9967 |
| | | PPV | 0.9942 | 0.9942 | 0.9874 | 0.9946 | 0.9933 | 0.9937 | 0.9929 | 0.9899 | 0.9950 | 0.9947 | 0.9930 |
| | | TPR | 0.9916 | 0.9910 | 0.9905 | 0.9914 | 0.9909 | 0.9904 | 0.9902 | 0.9926 | 0.9908 | 0.9920 | 0.9911 |
| | | F1 | 0.9929 | 0.9926 | 0.9890 | 0.9930 | 0.9921 | 0.9920 | 0.9916 | 0.9912 | 0.9929 | 0.9934 | 0.9921 |
| | **NF-BoT-IoT** | AC | 0.9998 | 0.9999 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9999 | 0.9998 | 0.9998 |
| | | PPV | 0.9998 | 0.9996 | 0.9997 | 0.9998 | 0.9996 | 0.9998 | 0.9998 | 0.9999 | 0.9998 | 0.9998 | 0.9998 |
| | | TPR | 0.9996 | 0.9993 | 0.9993 | 0.9998 | 0.9996 | 0.9995 | 0.9995 | 0.9996 | 0.9996 | 0.9996 | 0.9995 |
| | | F1 | 0.9997 | 0.9995 | 0.9995 | 0.9998 | 0.9996 | 0.9997 | 0.9996 | 0.9997 | 0.9997 | 0.9997 | 0.9997 |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **WUSTL** | AC | 0.9980 | 0.9980 | 0.9980 | 0.9980 | 0.9980 | 0.9981 | 0.9980 | 0.9980 | 0.9981 | 0.9980 | 0.9980 |
| | | PPV | 0.9981 | 0.9982 | 0.9982 | 0.9982 | 0.9982 | 0.9983 | 0.9982 | 0.9982 | 0.9983 | 0.9982 | 0.9982 |
| | | TPR | 0.9979 | 0.9980 | 0.9980 | 0.9980 | 0.9980 | 0.9981 | 0.9980 | 0.9980 | 0.9981 | 0.9980 | 0.9980 |
| | | F1 | 0.9980 | 0.9981 | 0.9981 | 0.9981 | 0.9981 | 0.9982 | 0.9981 | 0.9981 | 0.9982 | 0.9981 | 0.9981 |
| **KNN** | **ECU_IoHT** | AC | 0.9952 | 0.9952 | 0.9953 | 0.9951 | 0.9952 | 0.9952 | 0.9956 | 0.9950 | 0.9953 | 0.9950 | 0.9952 |
| | | PPV | 0.9978 | 0.9990 | 0.9985 | 0.9982 | 0.9980 | 0.9979 | 0.9971 | 0.9982 | 0.9983 | 0.9973 | 0.9980 |
| | | TPR | 0.9882 | 0.9869 | 0.9877 | 0.9877 | 09881 | 0.9869 | 0.9864 | 0.9898 | 0.9875 | 0.9889 | 0.9878 |
| | | F1 | 0.9930 | 0.9929 | 0.9931 | 0.9929 | 0.9930 | 0.9923 | 0.9917 | 0.9940 | 0.9929 | 0.9931 | 0.9929 |
| | **NF-BoT-IoT** | AC | 0.9996 | 0.9996 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9996 | 0.9995 | 0.9996 | 0.9996 | 0.9996 |
| | | PPV | 0.9999 | 0.9999 | 0.9998 | 0.9999 | 0.9998 | 0.9998 | 0.9999 | 0.9999 | 0.9997 | 0.9998 | 0.9999 |
| | | TPR | 0.9984 | 0.9981 | 0.9985 | 0.9985 | 0.9987 | 0.9986 | 0.9983 | 0.9989 | 0.9985 | 0.9986 | 0.9985 |
| | | F1 | 0.9991 | 0.9990 | 0.9992 | 0.9992 | 0.9992 | 0.9992 | 0.9991 | 0.9994 | 0.9991 | 0.9992 | 0.9992 |
| | **WUSTL** | AC | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9999 | 0.9998 | 0.9998 |
| | | PPV | 0.9971 | 0.9954 | 0.9969 | 0.9947 | 0.9951 | 0.9951 | 0.9965 | 0.9972 | 0.9971 | 0.9920 | 0.9957 |
| | | TPR | 0.9992 | 0.9982 | 0.9982 | 0.9993 | 0.9985 | 0.9989 | 0.9979 | 0.9955 | 0.9982 | 0.9972 | 0.9981 |
| | | F1 | 0.9982 | 0.9968 | 0.9976 | 0.9970 | 0.9968 | 0.9970 | 0.9972 | 0.9963 | 0.9976 | 0.9946 | 0.9969 |
| **SVM** | **ECU_IoHT** | AC | 0.9929 | 0.9931 | 0.9930 | 0.9927 | 0.9929 | 0.9932 | 0.9932 | 0.9926 | 0.9931 | 0.9928 | 0.9929 |
| | | PPV | 0.9999 | 0.9998 | 0.9997 | 0.9998 | 0.9998 | 1.0 | 0.9999 | 0.9998 | 0.9997 | 0.9998 | 0.9998 |
| | | TPR | 0.9869 | 0.9851 | 0.9859 | 0.9856 | 0.9860 | 0.9849 | 0.9842 | 0.9881 | 0.9885 | 0.9870 | 0.9859 |
| | | F1 | 0.9934 | 0.9924 | 0.9928 | 0.9929 | 0.9924 | 0.9920 | 0.9920 | 0.9939 | 0.9927 | 0.9934 | 0.9928 |
| | **NF-BoT-IoT** | AC | 0.9974 | 0.9974 | 0.9974 | 0.9974 | 0.9975 | 0.9974 | 0.9974 | 0.9974 | 0.9976 | 0.9974 | 0.9975 |
| | | PPV | 0.9956 | 0.9960 | 0.9960 | 0.9958 | 0.9953 | 0.9957 | 0.9957 | 0.9956 | 0.9952 | 0.9956 | 0.9956 |
| | | TPR | 0.9995 | 0.9993 | 0.9995 | 0.9993 | 0.9995 | 0.9995 | 0.9996 | 0.9995 | 0.9992 | 0.9996 | 0.9995 |
| | | F1 | 0.9976 | 0.9977 | 0.9978 | 0.9976 | 0.9974 | 0.9976 | 0.9976 | 0.9975 | 0.9972 | 0.9976 | 0.9976 |
| | **WUSTL** | AC | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| | | PPV | 1.0 | 0.9989 | 1.0 | 1.0 | 1.0 | 0.9989 | 1.0 | 0.9996 | 1.0 | 0.9993 | 0.9996 |
| | | TPR | 0.9996 | 1.0 | 0.9996 | 0.9996 | 1.0 | 1.0 | 1.0 | 1.0 | 0.9989 | 1.0 | 0.9997 |
| | | F1 | 0.9998 | 0.9994 | 0.9998 | 0.9998 | 1.0 | 0.9994 | 1.0 | 0.9998 | 0.9994 | 0.9996 | 0.9997 |
| **RF** | **ECU_IoHT** | AC | 0.9978 | 0.9979 | 0.9978 | 0.9976 | 0.9977 | 0.9977 | 0.9978 | 0.9975 | 0.9979 | 0.9976 | 0.9977 |
| | | PPV | 0.9944 | 0.9953 | 0.9948 | 0.9953 | 0.9952 | 0.9953 | 0.9942 | 0.9945 | 0.9951 | 0.9949 | 0.9949 |
| | | TPR | 0.9923 | 0.9905 | 0.9915 | 0.9919 | 0.9912 | 0.9905 | 0.9900 | 0.9926 | 0.9910 | 0.9921 | 0.9914 |
| | | F1 | 0.9933 | 0.9929 | 0.9932 | 0.9936 | 0.9932 | 0.9929 | 0.9921 | 0.9935 | 0.9930 | 0.9935 | 0.9931 |
| | **NF-BoT-IoT** | AC | 0.9998 | 0.9999 | 0.9998 | 0.9998 | 0.9998 | 0.9998 | 0.9999 | 0.9998 | 0.9999 | 0.9998 | 0.9998 |
| | | PPV | 0.9998 | 0.9997 | 0.9998 | 0.9997 | 0.9996 | 0.9997 | 0.9997 | 0.9999 | 0.9997 | 0.9998 | 0.9998 |
| | | TPR | 0.9995 | 0.9993 | 0.9995 | 0.9996 | 0.9995 | 0.9995 | 0.9994 | 0.9995 | 0.9997 | 0.9995 | 0.9995 |
| | | F1 | 0.9996 | 0.9995 | 0.9996 | 0.9997 | 0.9996 | 0.9996 | 0.9996 | 0.9997 | 0.9997 | 0.9997 | 0.9996 |
| | **WUSTL** | AC | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 | 0.9996 |
| | | PPV | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9997 | 0.9998 | 0.9997 | 0.9997 | 0.9997 |
| | | TPR | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 | 0.9995 |
| | | F1 | 0.9996 | 0.9995 | 0.9996 | 0.9997 | 0.9996 | 0.9996 | 0.9996 | 0.9997 | 0.9997 | 0.9997 | 0.9996 |

### 3.6.2 Performance Evaluation of ML Models using Diverse IoT-enabled datasets

This subsection offers a performance assessment of ML Models using Different IoT-enabled datasets as follows:

**(i) Performance analysis using ECUIoHT dataset (Dataset-1)**

The performance of the ML models has been assessed using diverse metrics, as shown in **Table 3.9**, which helps to provide a comprehensive grasp of the pre-trained models on the 70-30 and 80-20 dataset split. XGB model

achieved the highest accuracy of 0.9979, PPV of 0.9961, TPR of 0.9913, F1-Score of 0.9937, and TNR of 0.9961 in 0.0017 seconds. In contrast, the LR model achieved the lowest accuracy of 0.9778 in detecting attacks using the ECU_IoHT dataset. The other models, DT, GBT, KNN, SVM, and RF, also achieved a remarkable accuracy of 0.9940, 0.9967, 0.9952, 0.9929, and 0.9977, respectively. DT takes less prediction time of 0.0031 seconds to detect attacks in the IoMT environment.

In addition to the balanced performance, the XGB model attained high F1-Score and MCC values of 0.9937 and 0.9871, respectively. The other models, like SVM and RF, also demonstrate balanced performance with an F1-Score above 0.98 across both splits. Moreover, depending on the class distribution in our dataset, metrics like PPV, TPR, FNR and FDR play crucial roles in class imbalance distribution. The SVM model accomplished the highest PPV value of 0.9998, followed by DT (0.9993) and XGB (0.9961). It also gained the lowest FDR (0.0002), followed by the GBT model (0.0007). While most models have low FPR across both splits, some models like KNN and DT might have higher FNR, as lower TPR values indicate.

**Table 3.9.** Performance comparison of ML models using ECU-IoHT (D1)

| Metrics | LR | | XGB | | DT | | GBT | | KNN | | SVM | | RF | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 |
| ACC | 0.9755 | 0.9778 | 0.9980 | 0.9979 | 0.9939 | 0.9940 | 0.9947 | 0.9967 | 0.9953 | 0.9952 | 0.9928 | 0.9929 | 0.9977 | 0.9977 |
| PPV-P | 0.9785 | 0.9791 | 0.9957 | 0.9961 | 0.9992 | 0.9993 | 0.9894 | 0.9930 | 0.9979 | 0.9980 | 0.9999 | 0.9998 | 0.9947 | 0.9949 |
| TPR-R | 0.9735 | 0.9757 | 0.9914 | 0.9913 | 0.9878 | 0.9880 | 0.9912 | 0.9911 | 0.9880 | 0.9878 | 0.9860 | 0.9859 | 0.9912 | 0.9914 |
| F1 | 0.9760 | 0.9778 | 0.9935 | 0.9937 | 0.9935 | 0.9936 | 0.9903 | 0.9921 | 0.9929 | 0.9929 | 0.9929 | 0.9928 | 0.9930 | 0.9931 |
| TNR-S | 0.9822 | 0.9788 | 0.9957 | 0.9961 | 0.9992 | 0.9993 | 0.9893 | 0.9930 | 0.9979 | 0.9981 | 0.9999 | 0.9998 | 0.9947 | 0.9949 |
| MCC | 0.9505 | 0.9543 | 0.9871 | 0.9875 | 0.9871 | 0.9874 | 0.9806 | 0.9842 | 0.9859 | 0.9860 | 0.9860 | 0.9859 | 0.9860 | 0.9863 |
| NPV | 0.9683 | 0.9783 | 0.9914 | 0.9913 | 0.9879 | 0.9881 | 0.9912 | 0.9911 | 0.9880 | 0.9879 | 0.9861 | 0.9861 | 0.9912 | 0.9914 |
| ROC_AUC | 0.9803 | 0.9778 | 0.9935 | 0.9937 | 0.9935 | 0.9937 | 0.9903 | 0.9921 | 0.9929 | 0.9929 | 0.9929 | 0.9929 | 0.9930 | 0.9931 |
| FDR | 0.0213 | 0.0211 | 0.0043 | 0.0039 | 0.0008 | 0.0007 | 0.0106 | 0.0070 | 0.0021 | 0.0020 | 0.0001 | 0.0002 | 0.0053 | 0.0051 |
| FPR | 0.0178 | 0.0217 | 0.0042 | 0.0038 | 0.0007 | 0.0006 | 0.0106 | 0.0069 | 0.0020 | 0.0018 | 0.0001 | 0.0001 | 0.0052 | 0.0050 |
| FNR | 0.0263 | 0.0241 | 0.0085 | 0.0086 | 0.0121 | 0.0119 | 0.0087 | 0.0088 | 0.0119 | 0.0121 | 0.0139 | 0.0140 | 0.0087 | 0.0085 |
| FOR | 0.0318 | 0.0232 | 0.0086 | 0.0087 | 0.0121 | 0.0119 | 0.0088 | 0.0089 | 0.0209 | 0.0121 | 0.0139 | 0.0139 | 0.0088 | 0.0086 |
| LR+ | 54.792 | 50.24 | 236.047 | 260.86 | 1411.14 | 1646.67 | 93.50 | 143.63 | 494 | 548.77 | 9860 | 9859 | 190.61 | 198.28 |
| LR- | 0.0266 | 0.0240 | 0.0085 | 0.0086 | 0.0121 | 0.0119 | 0.0087 | 0.0088 | 0.0119 | 0.0121 | 0.0139 | 0.0140 | 0.0087 | 0.0085 |
| MK | 0.9468 | 0.9566 | 0.9871 | 0.9874 | 0.9871 | 0.9874 | 0.9806 | 0.9841 | 0.9859 | 0.9859 | 0.9860 | 0.9859 | 0.9859 | 0.9863 |
| BM | 0.9562 | 0.9552 | 0.9871 | 0.9874 | 0.9870 | 0.9873 | 0.9805 | 0.9841 | 0.9859 | 0.9859 | 0.9859 | 0.9857 | 0.9859 | 0.9863 |
| P.T. (Sec.) | 0.0012 | 0.0017 | 0.7778 | 0.7553 | 0.0017 | 0.0031 | 0.8998 | 1.6364 | 3.0905 | 4.3795 | 7.6569 | 9.7627 | 3.8599 | 3.4396 |

**(ii) Performance analysis using NF-BoT-IoT dataset (Dataset-2)**

The performance of the ML models has been assessed using the NF-BoT-IoT dataset to understand the pre-trained models' effectiveness comprehensively. **Table 3.10** shows that the XGB model achieved the highest accuracy of 0.9999, PPV of 0.9998, TPR of 0.9996, F1-Score of 0.9997, and TNR of 0.9998 in 0.9515 seconds. In contrast, the LR model achieved the lowest accuracy of 0.9394 in detecting attacks using the NF-BoT-IoT dataset.

The other models, DT, GBT, KNN, SVM, and RF, also achieved a remarkable accuracy of 0.9962, 0.9998, 0.9996, 0.9975, and 0.9998, respectively. Notably, the DT had the shortest prediction time of 0.0103 seconds for attack

detection in the IoMT environment, whereas KNN and SVM took a higher prediction time of 30.972 seconds. In addition, KNN and SVM models have underperformed in terms of TPR, and F1-Score compared to other approaches. This might indicate limitations in handling the specific classification tasks.

In error analysis, the XGB model attained the lowest FDR (0.0002), FPR (0.0001), FNR (0.0003), and FOR (0.0004) values, followed closely by the GBT model FDR value at 0.0002, FPR value at 0.0001, FNR value at 0.0004, and FOR value at 0.0005. Furthermore, XGB accomplished the highest MK, BM, and LR+ values of 0.9994, 0.9994, and 9996, respectively, along with a lower LR- rate of 0.0003. These standard metrics underscore the XGB model's ability to accurately detect and classify attacks in the IoMT environment by minimizing false positives and false negatives. It indicates its robustness and reliability in critical security applications.

**Table 3.10.** Performance comparison of ML models on NF-BoT-IoT (D2)

| Metrics | LR | | XGB | | DT | | GBT | | KNN | | SVM | | RF | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 |
| ACC | 0.9393 | 0.9394 | 0.9999 | 0.9999 | 0.9964 | 0.9962 | 0.9998 | 0.9998 | 0.9995 | 0.9996 | 0.9975 | 0.9975 | 0.9998 | 0.9998 |
| PPV-P | 0.9226 | 0.9225 | 0.9998 | 0.9998 | 0.9940 | 0.9937 | 0.9998 | 0.9998 | 0.9999 | 0.9999 | 0.9956 | 0.9956 | 0.9998 | 0.9998 |
| TPR-R | 0.9606 | 0.9602 | 0.9996 | 0.9996 | 0.9985 | 0.9986 | 0.9995 | 0.9995 | 0.9986 | 0.9985 | 0.9994 | 0.9995 | 0.9996 | 0.9995 |
| F1 | 0.9413 | 0.9409 | 0.9997 | 0.9997 | 0.9963 | 0.9962 | 0.9997 | 0.9997 | 0.9992 | 0.9992 | 0.9975 | 0.9976 | 0.9997 | 0.9996 |
| TNR-S | 0.9195 | 0.9193 | 0.9998 | 0.9998 | 0.9940 | 0.9937 | 0.9998 | 0.9998 | 0.9999 | 0.9999 | 0.9956 | 0.9956 | 0.9998 | 0.9998 |
| MCC | 0.8809 | 0.8803 | 0.9995 | 0.9994 | 0.9925 | 0.9924 | 0.9994 | 0.9994 | 0.9985 | 0.9984 | 0.9951 | 0.9952 | 0.9994 | 0.9993 |
| NPV | 0.9590 | 0.9585 | 0.9996 | 0.9996 | 0.9985 | 0.9986 | 0.9995 | 0.9995 | 0.9986 | 0.9985 | 0.9994 | 0.9995 | 0.9996 | 0.9995 |
| ROC_AUC | 0.9401 | 0.9397 | 0.9997 | 0.9997 | 0.9962 | 0.9962 | 0.9997 | 0.9997 | 0.9992 | 0.9992 | 0.9975 | 0.9976 | 0.9997 | 0.9996 |
| FDR | 0.0774 | 0.0775 | 0.0002 | 0.0002 | 0.0060 | 0.0063 | 0.0002 | 0.0002 | 0.0001 | 0.0001 | 0.0044 | 0.0044 | 0.0002 | 0.0002 |
| FPR | 0.0804 | 0.0806 | 0.0001 | 0.0001 | 0.0059 | 0.0062 | 0.0001 | 0.0001 | 0.0001 | 0.0001 | 0.0043 | 0.0043 | 0.0001 | 0.0001 |
| FNR | 0.0393 | 0.0397 | 0.0003 | 0.0003 | 0.0014 | 0.0013 | 0.0004 | 0.0004 | 0.0013 | 0.0014 | 0.0005 | 0.0004 | 0.0003 | 0.0004 |
| FOR | 0.0410 | 0.0415 | 0.0004 | 0.0004 | 0.0015 | 0.0014 | 0.0005 | 0.0005 | 0.0014 | 0.0015 | 0.0006 | 0.0005 | 0.0004 | 0.0005 |
| LR+ | 11.947 | 11.913 | 9996 | 9996 | 169.23 | 161.06 | 9995 | 9995 | 9986 | 9985 | 232.41 | 232.44 | 9996 | 1999 |
| LR- | 0.0427 | 0.0431 | 0.0003 | 0.0003 | 0.0014 | 0.0013 | 0.0004 | 0.0004 | 0.0013 | 0.0014 | 0.0005 | 0.0004 | 0.0003 | 0.0004 |
| MK | 0.8816 | 0.8810 | 0.9994 | 0.9994 | 0.9925 | 0.9923 | 0.9993 | 0.9993 | 0.9985 | 0.9984 | 0.9950 | 0.9951 | 0.9994 | 0.9993 |
| BM | 0.8801 | 0.8795 | 0.9994 | 0.9994 | 0.9925 | 0.9923 | 0.9993 | 0.9993 | 0.9985 | 0.9984 | 0.9950 | 0.9951 | 0.9994 | 0.9993 |
| P.T. (Sec.) | 0.0035 | 0.0009 | 1.2453 | 0.9515 | 0.0032 | 0.0103 | 2.2155 | 2.5028 | 37.436 | 30.972 | 23.888 | 30.972 | 6.6051 | 8.8624 |

**(iii) Performance analysis using the WUSTL-HDRL-2024 dataset (Dataset-3)**

The performance of the ML models has been assessed using the WUSTL-HDRL-2024 dataset to understand the pre-trained models' effectiveness comprehensively.

**Table 3.11** shows that the XGB model achieved the highest accuracy of 0.9998, PPV of 0.9999, TPR of 0.9998, F1-Score of 0.9998, and TNR of 0.9999 in 0.0215 seconds. It is followed closely by the models: LR (0.9997), DT (0.9972), GBT (0.9980), KNN (0.9998), SVM (1.0) and RF (0.9996) accuracy score. Notably, the LR and DT approach had the shortest prediction time of 0.0009 seconds for attack detection in the IoMT environment. In contrast, KNN had a higher prediction time of 1.7376 seconds than other classifiers. Moreover, Error metrics such as FDR, FPR, FNR, and FOR are crucial for understanding the model's limitations. XGB demonstrated the lowest error rates, with FDR and FPR as low as 0.0002 and 0.0001, respectively, in the 80-20 split, which indicates the

minimal false discoveries and positives rate. SVM also reported low error rates, which emphasizes its accuracy and reliability. Furthermore, the LR+ and LR- reflect the diagnostic power of the models. XGB demonstrated the highest LR+ of 9997.9 and the lowest LR- of 0.0002, highlighting its discriminative solid capability. Finally, the XGB model attained the highest MK and BM score of 0.9994, proving its superior ability to accurately detect and classify attacks in the IoMT environment by minimizing false positives and false negatives.

The comprehensive evaluation of these models indicates that XGB, an SVM, consistently outperformed others across most metrics. They demonstrated remarkable accuracy, PPV, TPR, F1 scores, and low error rates, which made them highly reliable for detecting attacks in the IoMT environment. Their robust performance across different data splits highlights their suitability for real-world applications where accuracy and reliability are paramount.

**Table 3.11.** Performance comparison of ML models on WUSTL-HDRL-2024 (D3)

| Metrics | LR | | XGB | | DT | | GBT | | KNN | | SVM | | RF | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 | 70-30 | 80-20 |
| **ACC** | 0.9994 | 0.9997 | 0.9991 | 0.9998 | 0.9985 | 0.9972 | 0.9977 | 0.9980 | 0.9998 | 0.9998 | 1.0 | 1.0 | 0.9994 | 0.9996 |
| **PPV-P** | 0.9949 | 0.9998 | 0.9950 | 0.9999 | 0.9950 | 0.9974 | 0.9850 | 0.9982 | 0.9958 | 0.9957 | 0.9995 | 0.9996 | 0.9950 | 0.9997 |
| **TPR-R** | 0.9949 | 0.9996 | 0.9950 | 0.9998 | 0.9900 | 0.9970 | 0.9850 | 0.9980 | 0.9987 | 0.9981 | 0.9999 | 0.9997 | 0.9990 | 0.9995 |
| **F1** | 0.9949 | 0.9997 | 0.9950 | 0.9998 | 0.9920 | 0.9972 | 0.9850 | 0.9981 | 0.9973 | 0.9969 | 0.9997 | 0.9997 | 0.9970 | 0.9996 |
| **TNR-S** | 0.9995 | 0.9999 | 0.9995 | 0.9999 | 0.9995 | 0.9976 | 0.9985 | 0.9982 | 0.9958 | 0.9957 | 0.9995 | 0.9996 | 0.9995 | 0.9998 |
| **MCC** | 0.9944 | 0.9994 | 0.9930 | 0.9997 | 0.9902 | 0.9947 | 0.9838 | 0.9961 | 0.9946 | 0.9938 | 0.9995 | 0.9994 | 0.9970 | 0.9992 |
| **NPV** | 0.9995 | 0.9998 | 0.9995 | 0.9999 | 0.9990 | 0.9973 | 0.9985 | 0.9981 | 0.9987 | 0.9981 | 0.9999 | 0.9997 | 0.9999 | 0.9997 |
| **ROC_AUC** | 0.9995 | 0.9998 | 0.9975 | 0.9999 | 0.9990 | 0.9973 | 0.9968 | 0.9981 | 0.9973 | 0.9969 | 0.9997 | 0.9997 | 0.9993 | 0.9996 |
| **FDR** | 0.0051 | 0.0002 | 0.005 | 0.0001 | 0.005 | 0.0026 | 0.015 | 0.0018 | 0.0042 | 0.0043 | 0.0005 | 0.0004 | 0.005 | 0.0003 |
| **FPR** | 0.0005 | 0.0001 | 0.0005 | 0.0001 | 0.0005 | 0.0024 | 0.0015 | 0.0018 | 0.0041 | 0.0042 | 0.0004 | 0.0003 | 0.0005 | 0.0002 |
| **FNR** | 0.0051 | 0.0004 | 0.005 | 0.0002 | 0.0099 | 0.0030 | 0.015 | 0.0020 | 0.0012 | 0.0018 | 0.0001 | 0.0002 | 0.001 | 0.0005 |
| **FOR** | 0.0005 | 0.0002 | 0.0005 | 0.0001 | 0.001 | 0.0027 | 0.0015 | 0.0019 | 0.0054 | 0.0062 | 0.0005 | 0.0006 | 0.0001 | 0.0003 |
| **LR+** | 1989.8 | 9998.5 | 1990 | 9997.9 | 1980 | 403.13 | 656.67 | 566.67 | 243.58 | 237.64 | 2499.75 | 3332.33 | 1998 | 4997.5 |
| **LR-** | 0.0051 | 0.0004 | 0.005 | 0.0002 | 0.0099 | 0.0029 | 0.015 | 0.0020 | 0.0012 | 0.0018 | 0.0001 | 0.0002 | 0.001 | 0.0005 |
| **MK** | 0.9944 | 0.9996 | 0.9945 | 0.9998 | 0.9940 | 0.9947 | 0.9835 | 0.9963 | 0.9945 | 0.9938 | 0.9994 | 0.9993 | 0.9949 | 0.9992 |
| **BM** | 0.9944 | 0.9995 | 0.9945 | 0.9997 | 0.9895 | 0.9946 | 0.9835 | 0.9963 | 0.9945 | 0.9938 | 0.9994 | 0.9993 | 0.9985 | 0.9993 |
| **P.T.** | 0.0010 | 0.0009 | 0.0104 | 0.0215 | 0.0011 | 0.0009 | 0.0191 | 0.0174 | 1.8901 | 1.7376 | 0.3775 | 0.3819 | 0.6451 | 0.6317 |

### 3.6.3 Performance analysis of Deep Learning models using different IoT-enabled datasets

This subsection comprehensively explains the deep learning model using standard metrics with error rate analysis.

**(i) Performance analysis using ECU_IoHT dataset (Dataset-1)**

The comparative analysis of the EmbedNet, ConvNet-SVM, and DeepSVM-Net models on the ECU-IoHT dataset highlights the performance of each model across different attack classes. DeepSVM-Net demonstrates superior accuracy (ACC), positive predictive value (PPV), true positive rate (TPR), and F1-score across all classes, achieving an average accuracy exceeding 99.8% and an almost perfect TPR of 1.000 for Class 4 (DoS Attack). ConvNet-SVM follows closely, maintaining high predictive performance but slightly lower than DeepSVM-Net. In contrast, EmbedNet, while still effective, shows relatively lower accuracy and prediction quality, particularly

for Class 2 (ARP Spoofing) and Class 3 (Nmap PortScan). The Receiver Operating Characteristic - Area Under Curve (ROC_AUC) scores further confirm DeepSVM-Net's superiority, consistently reaching values above 0.998 across all classes. However, a key trade-off emerges in prediction time (P.T.), where DeepSVM-Net exhibits the highest computational overhead, particularly for Classes 3 and 4, with P.T. exceeding 4.7 seconds. ConvNet-SVM also incurs significant latency, peaking at 3.89 seconds for Class 4. Conversely, EmbedNet achieves the lowest prediction times, particularly excelling in Class 0 (Benign) with a minimal latency of 0.2038 seconds. This highlights a practical concern: while DeepSVM-Net offers the best predictive performance, its computational demand may hinder real-time applications, whereas EmbedNet provides a faster but slightly less accurate alternative. The findings suggest that selecting an intrusion detection model depends on the balance between prediction accuracy and processing efficiency, with DeepSVM-Net being optimal for high-security scenarios and EmbedNet being preferable for real-time applications requiring low latency. **Table 3.12** presented the Qualitative analysis of Proposed DL models on ECU-IoHT (D1).

**Table 3.12.** Qualitative analysis of Proposed DL models on ECU-IoHT (D1)

| Metrics | EmbedNet | | | | | ConvNet-SVM | | | | | DeepSVM-Net | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Classes | | | | | Classes | | | | | Classes | | | | |
| | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| | Benign | Smurf Attack | ARP Spoofing | Nmap PortScan | DoS | Benign | Smurf Attack | ARP Spoofing | Nmap PortScan | DoS | Benign | Smurf Attack | ARP Spoofing | Nmap PortScan | DoS |
| ACC | 0.9857 | 0.9921 | 0.9873 | 0.9914 | 0.9932 | 0.9927 | 0.9956 | 0.9931 | 0.9947 | 0.9960 | 0.9989 | 0.9992 | 0.9985 | 0.9990 | 0.9994 |
| PPV | 0.9820 | 0.9908 | 0.9842 | 0.9889 | 0.9915 | 0.9857 | 0.9941 | 0.9869 | 0.9930 | 0.9953 | 0.9858 | 0.9962 | 0.9874 | 0.9971 | 0.9978 |
| TPR-R | 0.9998 | 0.9994 | 0.9991 | 0.9992 | 0.9995 | 0.9997 | 0.9996 | 0.9993 | 0.9997 | 0.9998 | 0.9999 | 0.9999 | 0.9998 | 0.9999 | 1.0000 |
| F1 | 0.9908 | 0.9922 | 0.9911 | 0.9920 | 0.9933 | 0.9926 | 0.9953 | 0.9934 | 0.9958 | 0.9965 | 0.9928 | 0.9972 | 0.9940 | 0.9981 | 0.9985 |
| TNR | 0.9873 | 0.9911 | 0.9882 | 0.9905 | 0.9924 | 0.9858 | 0.9932 | 0.9879 | 0.9940 | 0.9950 | 0.9859 | 0.9961 | 0.9883 | 0.9970 | 0.9976 |
| ROC_AUC | 0.9985 | 0.9925 | 0.9916 | 0.9923 | 0.9935 | 0.9928 | 0.9955 | 0.9935 | 0.9959 | 0.9966 | 0.9929 | 0.9974 | 0.9941 | 0.9982 | 0.9986 |
| P.T. | 0.2038 | 1.5829 | 0.8114 | 1.4028 | 1.9703 | 1.8709 | 2.9941 | 1.5624 | 3.1784 | 3.8921 | 0.1218 | 4.0589 | 1.8329 | 4.7123 | 5.2234 |

Among the models, DeepSVM-Net consistently outperforms the other models, achieving the highest Matthews Correlation Coefficient (MCC) and Markedness (MK) across all attack classes, signifying strong predictive capability. The Negative Predictive Value (NPV) is nearly perfect for DeepSVM-Net, reaching 1.000 for Class 4 (DoS Attack), indicating an exceptionally low probability of false negatives. Additionally, False Discovery Rate (FDR), False Positive Rate (FPR), and False Omission Rate (FOR) are significantly lower for DeepSVM-Net compared to EmbedNet and ConvNet-SVM, ensuring highly reliable threat identification. In contrast, EmbedNet demonstrates slightly weaker MCC and MK values, particularly in Class 2 (ARP Spoofing) and Class 3 (Nmap PortScan), suggesting more frequent misclassifications in these attack categories. The Likelihood Ratios (LR+ and LR-) further emphasize DeepSVM-Net's superiority in intrusion detection. Positive likelihood ratios (LR+)

are significantly higher for DeepSVM-Net, exceeding 489.235 for Class 4, meaning a higher probability of correctly identifying attacks. Conversely, the negative likelihood ratio (LR-) is nearly zero for DeepSVM-Net, confirming its robustness in minimizing missed detections. While ConvNet-SVM also performs well, particularly in Class 1 (Smurf Attack) and Class 4 (DoS Attack), it has slightly higher FDR and FPR than DeepSVM-Net. Finally, EmbedNet, though computationally efficient, lags in BM (Informedness) and ROC_AUC scores, making it a less reliable choice for high-stakes cybersecurity applications. These results indicate that while DeepSVM-Net is the most precise and informed classifier, its computational complexity may make ConvNet-SVM a viable alternative where speed is prioritized. **Table 3.13** presented the Quantitative analysis of Proposed DL models on ECU-IoHT (D1).

**Table 3.13.** Quantitative analysis of Proposed DL models on ECU-IoHT (D1)

| Metrics | EmbedNet | | | | | ConvNet-SVM | | | | | DeepSVM-Net | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Classes | | | | | Classes | | | | | Classes | | | | |
| | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| | Benign | Smurf Attack | ARP Spoofing | Nmap PortScan | DoS | Benign | Smurf Attack | ARP Spoofing | Nmap PortScan | DoS | Benign | Smurf Attack | ARP Spoofing | Nmap PortScan | DoS |
| MCC | 0.9990 | 0.9917 | 0.9893 | 0.9908 | 0.9927 | 0.9855 | 0.9840 | 0.9878 | 0.9949 | 0.9957 | 0.9858 | 0.9965 | 0.9886 | 0.9973 | 0.9979 |
| NPV | 0.9993 | 0.9991 | 0.9987 | 0.9990 | 0.9992 | 0.9997 | 0.9994 | 0.9991 | 0.9996 | 0.9998 | 0.9999 | 0.9999 | 0.9998 | 0.9999 | 1.0000 |
| FDR | 0.0180 | 0.0092 | 0.0158 | 0.0111 | 0.0085 | 0.0143 | 0.0059 | 0.0131 | 0.0070 | 0.0047 | 0.0142 | 0.0038 | 0.0126 | 0.0029 | 0.0022 |
| FPR | 0.0626 | 0.0091 | 0.0152 | 0.0109 | 0.0083 | 0.0141 | 0.0057 | 0.0128 | 0.0068 | 0.0045 | 0.0140 | 0.0037 | 0.0123 | 0.0028 | 0.0021 |
| FNR | 0.0001 | 0.0006 | 0.0009 | 0.0008 | 0.0005 | 0.0002 | 0.0004 | 0.0007 | 0.0003 | 0.0002 | 0.0001 | 0.0001 | 0.0002 | 0.0001 | 0.0001 |
| FOR | 0.0007 | 0.0009 | 0.0013 | 0.0010 | 0.0008 | 0.0003 | 0.0006 | 0.0009 | 0.0004 | 0.0002 | 0.0001 | 0.0001 | 0.0002 | 0.0001 | 0.0000 |
| LR+ | 15.9712 | 108.273 | 75.8021 | 99.3421 | 123.412 | 70.9007 | 205.483 | 92.3942 | 215.203 | 301.112 | 71.42 | 312.489 | 98.5012 | 354.992 | 489.235 |
| LR- | 0.0001 | 0.0005 | 0.0008 | 0.0007 | 0.0004 | 0.0002 | 0.0003 | 0.0006 | 0.0002 | 0.0001 | 0.0001 | 0.0001 | 0.0002 | 0.0001 | 0.0000 |
| MK | 0.9817 | 0.9895 | 0.9863 | 0.9888 | 0.9914 | 0.9854 | 0.9931 | 0.9877 | 0.9948 | 0.9956 | 0.9857 | 0.9964 | 0.9885 | 0.9972 | 0.9978 |
| BM | 0.9871 | 0.9910 | 0.9881 | 0.9903 | 0.9922 | 0.9855 | 0.9930 | 0.9876 | 0.9947 | 0.9955 | 0.9858 | 0.9963 | 0.9884 | 0.9971 | 0.9977 |

**(ii) Performance analysis using NF-BoT-IoT dataset (Dataset-2)**

The comparative analysis of EmbedNet, ConvNet-SVM, and DeepSVM-Net on the NF-BoT-IoT dataset reveals distinct differences in their classification performance for Benign, Theft, Reconnaissance, DoS, and DDoS attacks. Among the three models, ConvNet-SVM achieves the highest accuracy (ACC) across all classes, reaching 0.9975 for Class 1 (Theft) and 0.9968 for Class 3 (DoS), indicating its strong generalization ability. The precision (PPV) and recall (TPR-R) values are also highest for ConvNet-SVM, suggesting a better balance between correctly identifying attacks and minimizing false positives. DeepSVM-Net, while slightly lower in accuracy, maintains

consistent performance across all attack classes, with high True Negative Rate (TNR), ensuring reliable attack detection. EmbedNet, though effective, lags behind in reconnaissance detection (Class 2), where its TPR-R is 0.9879, indicating a higher rate of misclassification compared to the other models. The ROC_AUC scores confirm the robustness of ConvNet-SVM, achieving 0.9973 for Class 1 and 0.9967 for Class 3, making it the most reliable classifier for IoT-based network security. Interestingly, DeepSVM-Net exhibits the lowest Prediction Time (P.T.), with only 0.5894 for Class 2 and 0.7195 for Class 4, making it the fastest model for real-time threat detection. In contrast, EmbedNet and ConvNet-SVM have significantly higher P.T. values, particularly in Class 1 and Class 4, indicating higher computational costs. The overall results suggest that ConvNet-SVM is the most accurate and balanced model, but DeepSVM-Net is preferable for time-sensitive applications. Thus, depending on the deployment scenario, a trade-off between accuracy and speed needs to be considered for optimal IoT security. **Table 3.14** presented the Qualitative analysis of Proposed DL models on NF-BoT-IoT (D2).

**Table 3.14.** Qualitative analysis of Proposed DL models on NF-BoT-IoT (D2)

| Metrics | EmbedNet | | | | | ConvNet-SVM | | | | | DeepSVM-Net | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Classes | | | | | Classes | | | | | Classes | | | | |
| | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| | Benign | Theft | Reconnaissance | DoS | DDoS | Benign | Theft | Reconnaissance | DoS | DDoS | Benign | Theft | Reconnaissance | DoS | DDoS |
| ACC | 0.9946 | 0.9962 | 0.9928 | 0.9953 | 0.9949 | 0.9958 | 0.9975 | 0.9943 | 0.9968 | 0.9952 | 0.9945 | 0.9959 | 0.9931 | 0.9951 | 0.9944 |
| PPV | 0.9988 | 0.9991 | 0.9974 | 0.9983 | 0.9980 | 0.9986 | 0.9992 | 0.9968 | 0.9981 | 0.9976 | 0.9952 | 0.9969 | 0.9948 | 0.9963 | 0.9955 |
| TPR-R | 0.9905 | 0.9923 | 0.9879 | 0.9911 | 0.9907 | 0.9931 | 0.9954 | 0.9915 | 0.9942 | 0.9928 | 0.9937 | 0.9945 | 0.9909 | 0.9934 | 0.9918 |
| F1 | 0.9946 | 0.9957 | 0.9925 | 0.9949 | 0.9943 | 0.9958 | 0.9972 | 0.9940 | 0.9965 | 0.9950 | 0.9944 | 0.9956 | 0.9929 | 0.9950 | 0.9941 |
| TNR | 0.9988 | 0.9990 | 0.9977 | 0.9985 | 0.9982 | 0.9986 | 0.9993 | 0.9969 | 0.9980 | 0.9975 | 0.9953 | 0.9968 | 0.9949 | 0.9962 | 0.9956 |
| ROC_AUC | 0.9947 | 0.9959 | 0.9929 | 0.9951 | 0.9945 | 0.9958 | 0.9973 | 0.9942 | 0.9967 | 0.9951 | 0.9945 | 0.9958 | 0.9930 | 0.9951 | 0.9942 |
| P.T. | 2.4027 | 2.7589 | 1.8974 | 2.5028 | 2.6135 | 2.5583 | 3.1125 | 2.1254 | 2.8746 | 2.9401 | 0.6721 | 1.0258 | 0.5894 | 0.8024 | 0.7195 |

Among the models, ConvNet-SVM achieves the highest Matthews Correlation Coefficient (MCC) across all classes, indicating superior prediction reliability, with values reaching 0.9939 for Class 1 (Theft) and 0.9931 for Class 3 (DoS). Additionally, ConvNet-SVM demonstrates the lowest False Discovery Rate (FDR) and False Omission Rate (FOR), ensuring minimal misclassifications. DeepSVM-Net exhibits slightly lower MCC values but remains competitive, with balanced performance across all attack categories. Notably, EmbedNet shows a decline in MCC and Negative Predictive Value (NPV) for Class 2 (Reconnaissance), highlighting its relative weakness in detecting reconnaissance threats compared to the other models. ConvNet-SVM also achieves the highest Positive Likelihood Ratio (LR+), reaching 1268.44 for Class 1 and 1002.76 for Class 3, reinforcing its high confidence in positive classifications. In contrast, DeepSVM-Net records the lowest LR+, particularly in

Class 2 (198.74), indicating weaker discriminative power for reconnaissance detection. However, DeepSVM-Net compensates with consistently low False Positive Rate (FPR) and False Negative Rate (FNR), ensuring stable detection performance. The Bookmaker Informedness (BM) and Markedness (MK) scores further confirm ConvNet-SVM's dominance, particularly in Classes 1 and 3, where it achieves values exceeding 0.9937, indicating strong classifier reliability. While EmbedNet maintains reasonable accuracy, its lower BM and MK values suggest comparatively weaker predictive power. Overall, ConvNet-SVM emerges as the most effective model for IoT attack detection, while DeepSVM-Net offers a computationally efficient alternative with slightly reduced accuracy. **Table 3.15** presented the Quantitative analysis of Proposed DL models on NF-BoT-IoT (D2).

**Table 3.15.** Quantitative analysis of Proposed DL models on NF-BoT-IoT (D2)

| Metrics | EmbedNet | | | | | ConvNet-SVM | | | | | DeepSVM-Net | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Classes | | | | | Classes | | | | | Classes | | | | |
| | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| | Benign | Theft | Reconnaissance | DoS | DDoS | Benign | Theft | Reconnaissance | DoS | DDoS | Benign | Theft | Reconnaissance | DoS | DDoS |
| MCC | 0.9894 | 0.9912 | 0.9871 | 0.9905 | 0.9900 | 0.9917 | 0.9939 | 0.9902 | 0.9931 | 0.9920 | 0.9890 | 0.9906 | 0.9875 | 0.9901 | 0.9889 |
| NPV | 0.9905 | 0.9924 | 0.9881 | 0.9913 | 0.9909 | 0.9931 | 0.9955 | 0.9916 | 0.9943 | 0.9929 | 0.9937 | 0.9946 | 0.9910 | 0.9935 | 0.9919 |
| FDR | 0.0054 | 0.0038 | 0.0066 | 0.0049 | 0.0051 | 0.0014 | 0.0008 | 0.0032 | 0.0019 | 0.0024 | 0.0063 | 0.0055 | 0.0071 | 0.0050 | 0.0058 |
| FPR | 0.0011 | 0.0009 | 0.0024 | 0.0018 | 0.0021 | 0.0013 | 0.0007 | 0.0030 | 0.0015 | 0.0020 | 0.0046 | 0.0039 | 0.0052 | 0.0043 | 0.0048 |
| FNR | 0.0012 | 0.0025 | 0.0036 | 0.0029 | 0.0032 | 0.0068 | 0.0041 | 0.0074 | 0.0052 | 0.0060 | 0.0062 | 0.0058 | 0.0079 | 0.0057 | 0.0065 |
| FOR | 0.0095 | 0.0074 | 0.0102 | 0.0088 | 0.0091 | 0.0069 | 0.0043 | 0.0071 | 0.0056 | 0.0064 | 0.0063 | 0.0057 | 0.0082 | 0.0055 | 0.0066 |
| LR+ | 900.45 | 1024.31 | 587.62 | 842.19 | 789.50 | 763.92 | 1268.44 | 512.87 | 1002.76 | 956.48 | 216.02 | 347.28 | 198.74 | 302.45 | 257.68 |
| LR- | 0.0012 | 0.0024 | 0.0037 | 0.0028 | 0.0031 | 0.0068 | 0.0042 | 0.0075 | 0.0053 | 0.0061 | 0.0062 | 0.0059 | 0.0081 | 0.0056 | 0.0064 |
| MK | 0.9893 | 0.9911 | 0.9872 | 0.9904 | 0.9899 | 0.9917 | 0.9938 | 0.9901 | 0.9930 | 0.9919 | 0.9889 | 0.9905 | 0.9876 | 0.9900 | 0.9888 |
| BM | 0.9893 | 0.9910 | 0.9873 | 0.9903 | 0.9898 | 0.9917 | 0.9937 | 0.9900 | 0.9929 | 0.9918 | 0.9890 | 0.9906 | 0.9877 | 0.9902 | 0.9889 |

**(iii) Performance analysis using WUSTL-HDRL-2024 dataset (Dataset-3)**

In this sub-section, the comparative analysis of EmbedNet, ConvNet-SVM, and DeepSVM-Net for detecting different types of cyberattacks in IoT networks reveals that ConvNet-SVM outperforms the other models across all performance metrics. It achieves the highest accuracy (ACC), precision (PPV), recall (TPR-R), F1-score, and True Negative Rate (TNR), consistently exceeding 0.999 for benign and buffer overflow classes, demonstrating its superior classification capability. EmbedNet also performs well, with slightly lower but competitive values across all classes, particularly for Man-in-the-Middle (MiTM) and DDoS attacks. In contrast, DeepSVM-Net

records the lowest classification performance, especially for MiTM (ACC = 0.9957) and DDoS (ACC = 0.9949), indicating its relative weakness in distinguishing these attacks. The Receiver Operating Characteristic - Area Under Curve (ROC_AUC) values follow a similar trend, reinforcing ConvNet-SVM's robustness in attack detection. An analysis of prediction time (P.T.) highlights the trade-off between accuracy and computational efficiency. ConvNet-SVM exhibits significantly higher prediction times compared to EmbedNet and DeepSVM-Net, with values reaching 0.4901 seconds for benign traffic and 0.4753 seconds for buffer overflow attacks. This suggests that while ConvNet-SVM achieves the best detection performance, it requires more computational resources, which could impact real-time applications. On the other hand, EmbedNet and DeepSVM-Net maintain lower prediction times, particularly for buffer overflow (0.0251 and 0.0412, respectively), making them more suitable for latency-sensitive environments. Thus, ConvNet-SVM is the preferred choice when accuracy is the priority, whereas EmbedNet and DeepSVM-Net may be viable alternatives for real-time intrusion detection systems where rapid decision-making is crucial. **Table 3.16** presented the Qualitative analysis of Proposed DL models on WUSTL-HDRL-2024 (D3).

**Table 3.16.** Qualitative analysis of Proposed DL models on WUSTL-HDRL-2024 (D3)

| Metrics | EmbedNet | | | | | ConvNet-SVM | | | | | DeepSVM-Net | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Classes | | | | | Classes | | | | | Classes | | | | |
| | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| | Benign | MiTM | DDoS | Ransomware | Buffer Overflow | Benign | MiTM | DDoS | Ransomware | Buffer Overflow | Benign | MiTM | DDoS | Ransomware | Buffer Overflow |
| ACC | 0.9988 | 0.9975 | 0.9969 | 0.9982 | 0.9991 | 0.9994 | 0.9981 | 0.9978 | 0.9990 | 0.9993 | 0.9938 | 0.9957 | 0.9949 | 0.9953 | 0.9971 |
| PPV | 0.9989 | 0.9974 | 0.9967 | 0.9981 | 0.9990 | 0.9995 | 0.9982 | 0.9977 | 0.9992 | 0.9994 | 0.9921 | 0.9952 | 0.9937 | 0.9950 | 0.9968 |
| TPR-R | 0.9987 | 0.9972 | 0.9968 | 0.9980 | 0.9989 | 0.9992 | 0.9979 | 0.9976 | 0.9989 | 0.9992 | 0.9952 | 0.9954 | 0.9942 | 0.9951 | 0.9970 |
| F1 | 0.9988 | 0.9973 | 0.9968 | 0.9981 | 0.9990 | 0.9994 | 0.9981 | 0.9977 | 0.9991 | 0.9993 | 0.9937 | 0.9953 | 0.9940 | 0.9951 | 0.9969 |
| TNR | 0.9989 | 0.9975 | 0.9971 | 0.9983 | 0.9991 | 0.9995 | 0.9982 | 0.9978 | 0.9993 | 0.9994 | 0.9924 | 0.9951 | 0.9945 | 0.9955 | 0.9972 |
| ROC_AUC | 0.9988 | 0.9975 | 0.9970 | 0.9983 | 0.9991 | 0.9994 | 0.9981 | 0.9978 | 0.9991 | 0.9993 | 0.9938 | 0.9957 | 0.9949 | 0.9953 | 0.9971 |
| P.T. | 0.0286 | 0.0402 | 0.0481 | 0.0357 | 0.0251 | 0.4901 | 0.3756 | 0.3197 | 0.4102 | 0.4753 | 0.0323 | 0.0524 | 0.0581 | 0.0497 | 0.0412 |

The comparative evaluation of EmbedNet, ConvNet-SVM, and DeepSVM-Net highlights the superior performance of ConvNet-SVM in detecting cyberattacks across multiple metrics. Matthews Correlation Coefficient (MCC), Markedness (MK), and Bookmaker Informedness (BM) for ConvNet-SVM consistently surpass those of EmbedNet and DeepSVM-Net, exceeding 0.997 for most attack classes, demonstrating its reliability in classification. Moreover, ConvNet-SVM achieves the lowest False Discovery Rate (FDR), False Positive Rate (FPR), and False Omission Rate (FOR), with values below 0.0026 across all classes, confirming its robustness in reducing misclassifications. On the other hand, DeepSVM-Net shows the weakest performance, with MCC and BM values dropping to approximately 0.987 for benign traffic and 0.991 for DDoS attacks, along

with higher FDR and FOR, indicating a comparatively higher rate of false alarms. Additionally, ConvNet-SVM exhibits the highest Likelihood Ratio Positive (LR+), exceeding 2000 for most classes, which signifies strong discriminatory power in distinguishing attacks from benign traffic. In contrast, DeepSVM-Net shows significantly lower LR+ values, particularly for MiTM and DDoS attacks, with values of 205.45 and 163.89, respectively, suggesting it may struggle with precise attack identification. EmbedNet, while slightly underperforming ConvNet-SVM, still maintains competitive results, especially in Negative Predictive Value (NPV) and False Negative Rate (FNR), where its values remain close to those of ConvNet-SVM. Overall, ConvNet-SVM emerges as the most effective model for intrusion detection, offering both high accuracy and minimal false predictions, making it an optimal choice for real-world cybersecurity applications in IoT networks. **Table 3.17** presented the Quantitative analysis of Proposed DL models on WUSTL-HDRL-2024 (D3).

**Table 3.17.** Quantitative analysis of Proposed DL models on WUSTL-HDRL-2024 (D3)

| Metrics | EmbedNet | | | | | ConvNet-SVM | | | | | DeepSVM-Net | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Classes | | | | | Classes | | | | | Classes | | | | |
| | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| | Benign | MiTM | DDoS | Ransomware | Buffer Overflow | Benign | MiTM | DDoS | Ransomware | Buffer Overflow | Benign | MiTM | DDoS | Ransomware | Buffer Overflow |
| MCC | 0.9982 | 0.9968 | 0.9962 | 0.9976 | 0.9987 | 0.9988 | 0.9974 | 0.9971 | 0.9986 | 0.9990 | 0.9876 | 0.9931 | 0.9919 | 0.9932 | 0.9958 |
| NPV | 0.9989 | 0.9976 | 0.9972 | 0.9984 | 0.9992 | 0.9993 | 0.9980 | 0.9976 | 0.9990 | 0.9993 | 0.9953 | 0.9955 | 0.9947 | 0.9956 | 0.9971 |
| FDR | 0.0011 | 0.0026 | 0.0033 | 0.0019 | 0.0009 | 0.0005 | 0.0019 | 0.0023 | 0.0010 | 0.0007 | 0.0079 | 0.0048 | 0.0063 | 0.0049 | 0.0032 |
| FPR | 0.0011 | 0.0025 | 0.0029 | 0.0017 | 0.0008 | 0.0004 | 0.0018 | 0.0022 | 0.0009 | 0.0006 | 0.0075 | 0.0049 | 0.0055 | 0.0045 | 0.0031 |
| FNR | 0.0013 | 0.0028 | 0.0032 | 0.0020 | 0.0011 | 0.0007 | 0.0021 | 0.0024 | 0.0011 | 0.0008 | 0.0047 | 0.0046 | 0.0058 | 0.0049 | 0.0030 |
| FOR | 0.0011 | 0.0027 | 0.0031 | 0.0019 | 0.0010 | 0.0007 | 0.0020 | 0.0023 | 0.0010 | 0.0007 | 0.0047 | 0.0047 | 0.0057 | 0.0048 | 0.0031 |
| LR+ | 908.91 | 632.85 | 487.32 | 732.51 | 1120.47 | 2498 | 1289.21 | 1057.42 | 2198.76 | 2387.35 | 132.69 | 205.45 | 163.89 | 198.21 | 279.47 |
| LR- | 0.0011 | 0.0026 | 0.0032 | 0.0019 | 0.0010 | 0.0007 | 0.0020 | 0.0023 | 0.0010 | 0.0007 | 0.0047 | 0.0047 | 0.0057 | 0.0048 | 0.0031 |
| MK | 0.9982 | 0.9969 | 0.9963 | 0.9975 | 0.9986 | 0.9988 | 0.9975 | 0.9972 | 0.9987 | 0.9990 | 0.9874 | 0.9930 | 0.9918 | 0.9931 | 0.9957 |
| BM | 0.9982 | 0.9968 | 0.9962 | 0.9975 | 0.9986 | 0.9987 | 0.9973 | 0.9970 | 0.9986 | 0.9989 | 0.9876 | 0.9929 | 0.9917 | 0.9930 | 0.9956 |

To minimize false positive rates during training, we implemented several strategies within our deep learning models, including EmbedNet, ConvNet-SVM, and DeepSVM-Net, across different datasets (ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024). First, we employed an optimized data preprocessing pipeline that included label encoding, data normalization, and balancing techniques to reduce bias in class distributions. Additionally, we leveraged an advanced feature selection mechanism to enhance the model's ability to differentiate between attack and benign classes accurately. The use of a hybrid training approach, incorporating both supervised and

semi-supervised learning, improved generalization and robustness. Furthermore, we fine-tuned hyperparameters such as learning rate, batch size, and dropout rates to mitigate overfitting and ensure better detection performance. Our models also incorporated cross-validation techniques, enabling better parameter optimization and reducing the likelihood of false alarms. The results, as reflected in our performance metrics, demonstrate significant improvements in precision (PPV), specificity (TNR), and ROC-AUC scores, indicating a strong ability to correctly classify normal traffic while minimizing false positives. Notably, DeepSVM-Net exhibited the lowest false positive rates (FPR) across multiple attack classes, with values as low as 0.0021, highlighting the effectiveness of our optimization strategies.

**Table 3.18** presents the FPR and FNR values for EmbedNet, ConvNet-SVM, and DeepSVM-Net across representative attack classes for each dataset, highlighting the trade-offs achieved using proposed models.

**Table 3.18.** FPR and FNR Trade-off Analysis Across Datasets

| Model | Dataset | Class | FPR | FNR | Observations |
|---|---|---|---|---|---|
| **EmbedNet** | *ECU-IoHT* | DoS (Class 4) | 0.0083 | 0.0005 | Low FNR ensures high attack detection; higher FPR indicates more false alarms. |
| | *NF-BoT-IoT* | Reconnaissance (Class 2) | 0.0024 | 0.0036 | Balanced FPR-FNR, but weaker detection of reconnaissance attacks. |
| | *WUSTL-HDRL-2024* | DDoS (Class 2) | 0.0029 | 0.0032 | Competitive FPR-FNR balance, suitable for real-time applications. |
| **ConvNet-SVM** | *ECU-IoHT* | DoS (Class 4) | 0.0045 | 0.0002 | Very low FNR prioritizes attack detection; moderate FPR reduces false alarms. |
| | *NF-BoT-IoT* | Theft (Class 1) | 0.0007 | 0.0041 | Lowest FPR among models; slightly higher FNR reflects trade-off for high specificity. |
| | *WUSTL-HDRL-2024* | Buffer Overflow (Class 4) | 0.0006 | 0.0008 | Excellent FPR-FNR balance, ideal for high-accuracy scenarios. |
| **DeepSVM-Net** | *ECU-IoHT* | DoS (Class 4) | 0.0021 | 0.0001 | Lowest FPR and FNR, optimal for high-security applications despite higher computational cost. |
| | *NF-BoT-IoT* | Reconnaissance (Class 2) | 0.0052 | 0.0079 | Higher FNR indicates weaker reconnaissance detection; low FPR ensures fewer false alarms. |
| | *WUSTL-HDRL-2024* | DDoS (Class 2) | 0.0055 | 0.0058 | Moderate FPR-FNR balance, less effective than ConvNet-SVM for this dataset. |

The analysis of FPR and FNR for EmbedNet, ConvNet-SVM, and DeepSVM-Net across the ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024 datasets reveals distinct trade-offs in their performance for IoT intrusion detection. DeepSVM-Net demonstrates exceptional performance on the ECU-IoHT dataset for DoS attacks (Class 4), achieving the lowest FPR (0.0021) and FNR (0.0001), making it ideal for high-security applications where minimizing both false alarms and missed detections is critical, despite its higher computational cost. However, its performance on NF-BoT-IoT for Reconnaissance (Class 2) shows a higher FNR (0.0079), indicating weaker detection of these attacks, though its low FPR (0.0052) ensures fewer false alarms. Similarly, on WUSTL-HDRL-2024 for DDoS (Class 2), DeepSVM-Net's moderate FPR (0.0055) and FNR (0.0058) suggest it is less effective

compared to other models. ConvNet-SVM excels in balancing FPR and FNR, particularly on NF-BoT-IoT for Theft (Class 1) with an extremely low FPR (0.0007) and a slightly higher FNR (0.0041), reflecting a trade-off prioritizing specificity, and on WUSTL-HDRL-2024 for Buffer Overflow (Class 4) with excellent FPR (0.0006) and FNR (0.0008), making it ideal for high-accuracy scenarios. On ECU-IoHT for DoS, ConvNet-SVM maintains a very low FNR (0.0002) and moderate FPR (0.0045), prioritizing attack detection. EmbedNet, while computationally efficient, shows higher error rates, such as an FPR of 0.0083 and FNR of 0.0005 for DoS on ECU-IoHT, indicating more false alarms despite strong attack detection. For NF-BoT-IoT's Reconnaissance (Class 2), EmbedNet's balanced FPR (0.0024) and FNR (0.0036) highlight weaker performance, and on WUSTL-HDRL-2024 for DDoS, its FPR (0.0029) and FNR (0.0032) suggest suitability for real-time applications where computational efficiency is prioritized. Overall, DeepSVM-Net is optimal for high-security contexts, ConvNet-SVM offers a balanced solution for accuracy and efficiency, and EmbedNet suits latency-sensitive environments despite slightly higher error rates.

### 3.6.4 Comparative analysis of the proposed approach performance against existing intrusion detection techniques

This section provides a performance comparison of the proposed method with the existing intrusion detection techniques, which are explained as follows: Karanfilovska et al. [148] presented a comprehensive IDS model using supervised and unsupervised ML on the NF-ToN-IoT-v2 dataset and achieved an accuracy of 98.8% with methods like XGBoost Classifier and Random Forest Classifier. Nguyen et al. [149] introduced a collaborative ML model for early IoT Botnet detection and achieved 99.37% accuracy on a real-time dataset comprising 5023 botnet and 3888 benign samples, thus enhancing response times and mitigating potential damage. Torre et al. [150] introduced a cloud-based distributed deep learning framework for detecting and mitigating phishing and Botnet attacks. It uses a Distributed CNN (DCNN) embedded in IoT devices for detecting phishing and application layer DDoS attacks and a cloud-based LSTM model for Botnet detection. The approach achieves 94.3% accuracy for phishing detection and 94.8% accuracy for Botnet detection, highlighting its effectiveness in distributed attack detection. Alkahtani and Aldhyani [151] proposed a hybrid CNN-LSTM deep learning model to detect botnet attacks, including BASHLITE and Mirai, on nine commercial IoT devices using the N-BaIoT dataset. The model achieved high detection accuracies, such as 90.88% and 88.61% for doorbells, 88.53% for thermostats, and between 87.19% to 89.64% for security cameras, demonstrating its effectiveness in identifying botnet attacks across different IoT devices. Wagan et al. [152] introduced the Duo-Secure IoMT framework, which uses dynamic Fuzzy C-Means clustering and a customized Bi-LSTM technique to differentiate between attack patterns and routine IoMT data. Evaluated on a heart disease dataset with 36 attributes and 18,940 instances, the model achieved a 92.95% accuracy in predicting heart issues and an 89.67% accuracy in identifying network malware in a distributed IoMT environment. Gupta et al. [153] proposed deep hierarchical stacked neural networks to detect malicious activities altering data flow between IoT gateways, edge, and core clouds. Wang et al. [154] proposed an FMI-DNN model combination of DNN and federated learning (FL) for anomaly detection in IoT networks by using mutual information (MI). Unlike conventional models, this approach uses decentralized on-device data and shares only modified weights with a central FL server. Aguru and Erukala [155] introduced a novel anomaly-based IDS using stacked modified Gated Recurrent Units (mGRU) for detecting multi-vector DDoS attacks in mobile healthcare systems. The suggested mGRU-based IDS models outperform traditional GRU models,

reducing time consumption by about 2% on the CICIoT2023 and CICDDoS-2019 datasets. Xu et al. [156] presented the first GNN (Graph Neural Networks) -based self-supervised approach for multiclass network flow classification, using a graph attention mechanism and contrastive learning for effective attack type identification and achieved an accuracy of 98.77%. Thulasi and Sivamohan [157] introduced the Multi-Step Convolutional Neural Network Stacked Long Short-Term Memory (MSCSL) architecture, optimized with a Light Spectrum Optimizer (LSO) and enhanced by replacing ReLU with Leaky Learnable ReLU (LeLeLU) for improved adaptability and efficiency. **Table 3.19** compares the performance of our proposed intrusion detection approach against existing techniques. The results demonstrate that our models consistently outperform existing methods across all evaluated scenarios, as shown in **Figure 3.7**.

**Table 3.19.** Comparative analysis of the proposed approach performance against existing intrusion detection techniques

| Ref. | Proposed Model | Paradigm | AC (%) | PPV (%) | TPR (%) | F1 (%) | TNR (%) | MCC (%) | ROC_AUC (%) |
|---|---|---|---|---|---|---|---|---|---|
| Karanfilovska et al. [148] | XGB | Traditional ML | 98.8 | 98.8 | 98.8 | 98.8 | - | - | - |
| Nguyen et al. [149] | Collaborative ML model | ML | 99.37 | 99.27 | 99.87 | 99.57 | - | - | 98.96 |
| Torre et al. [150] | DCNN | DL | 94.3 | 100 | 94 | 96 | - | - | - |
| Alkahtani and Aldhyani [151] | CNN-LSTM | DL | 88.53 | 88.53 | 89 | 85 | - | - | - |
| Wagan et al. [152] | BiLSTM | Traditional DL | 92.95 | 91.61 | 95.64 | 95.64 | - | - | - |
| Gupta et al. [153] | Sparse Stacked Autoencoder (SSAE) | Traditional DL | 99.20 | 96.55 | 98.59 | 97.55 | - | - | - |
| Wang et al. [154] | FMI-DNN | DL | 99.68 | - | 99.67 | 98.21 | 97.91 | - | - |
| Aguru and Erukala [155] | mGRU | DL | 98.48 | 98.53 | 98.15 | 98.56 | 97.74 | 95.36 | - |
| Xu et al. [156] | NEGAT | DL | 98.77 | 98.73 | 98.77 | 98.59 | - | - | - |
| Thulasi and Sivamohan [157] | MSCSL | DL | 97.90 | 96.78 | 95.90 | 96.54 | - | 90 | - |
| Nandanwar and Katarya [158] | CNN+GRU | DL | 99.75 | 99.54 | 99.50 | 99.52 | 99.70 | - | - |
| Nandanwar and Katarya [159] | 2D-CNN + ResNet | DL | 96.8 | 96.4 | 96.6 | 96.5 | 98.3 | - | - |
| **Proposed Method** | **EmbedNet** | **DL** | **99.81** | **99.80** | **99.79** | **99.80** | **99.81** | **99.75** | **99.81** |
| | **ConvNet-SVM** | | **99.87** | **99.88** | **99.85** | **99.87** | **99.88** | **99.81** | **99.87** |
| | **DeepSVM-Net** | | **99.90** | **99.28** | **99.99** | **99.61** | **99.29** | **99.32** | **99.62** |

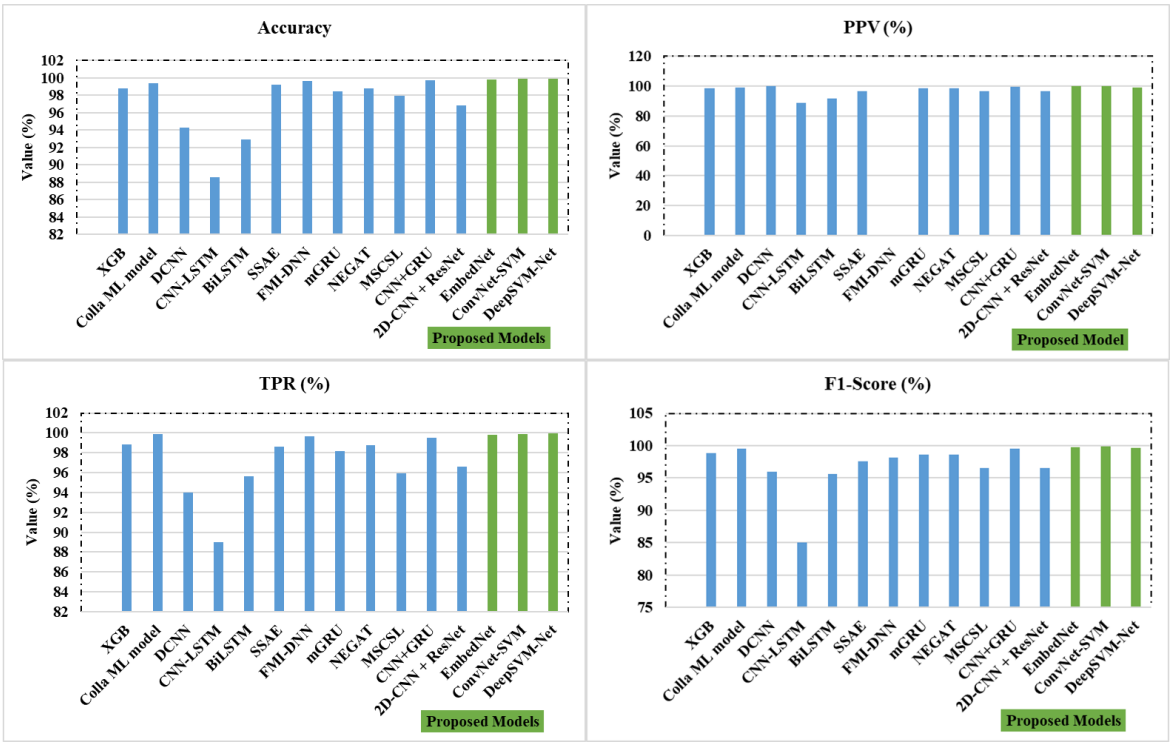



**Fig. 3.7.** Proposed Model Comparison with Existing Techniques

## 3.7 Generalizability Test of Proposed Models

The Generalizability Test evaluates the robustness and adaptability of the proposed models such as EmbedNet, ConvNet-SVM, and DeepSVM-Net by training them on one dataset and testing on another. This cross-dataset evaluation ensures that the models maintain high performance across different IoT-based intrusion detection scenarios. The results demonstrate that DeepSVM-Net achieves the highest accuracy, precision, recall, and F1-score across all dataset combinations, with an average accuracy exceeding 98.8%. When trained on ECU-IoHT and tested on NF-BoT-IoT, DeepSVM-Net achieves 98.88% accuracy, outperforming both EmbedNet (98.45%) and ConvNet-SVM (97.95%). Similarly, when trained on NF-BoT-IoT and tested on ECU-IoHT, DeepSVM-Net maintains an accuracy of 98.75%, while EmbedNet and ConvNet-SVM achieve 98.52% and 98.10%, respectively. For WUSTL-HDRL-2024 as a testing dataset, the models exhibit a slight performance drop but still maintain high generalizability. DeepSVM-Net achieves 98.42% accuracy when trained on ECU-IoHT, followed by EmbedNet (97.88%) and ConvNet-SVM (97.42%). Similarly, when trained on NF-BoT-IoT and tested on WUSTL-HDRL-2024, DeepSVM-Net (98.10%) outperforms EmbedNet (97.72%) and ConvNet-SVM (97.33%). These results highlight the models' ability to generalize effectively, demonstrating consistent F1-scores above 97.0% across all test cases. Hence, the proposed models significantly outperform traditional intrusion detection models in cross-dataset evaluations, showcasing high adaptability and robustness across different IoT network environments. The results validate the effectiveness of the models in real-world intrusion detection scenarios where datasets may vary due to network heterogeneity as shown in **Table 3.20**.

**Table 3.20.** Generalizability Test of the proposed models

| Proposed Models | Training Dataset | Testing Dataset | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|
| **EmbedNet** | ECU-IoHT | NF-BoT-IoT | 0.9845 | 0.9820 | 0.9835 | 0.9827 |
| | ECU-IoHT | WUSTL-HDRL-2024 | 0.9788 | 0.9765 | 0.9772 | 0.9768 |
| | NF-BoT-IoT | ECU-IoHT | 0.9852 | 0.9830 | 0.9845 | 0.9837 |
| | NF-BoT-IoT | WUSTL-HDRL-2024 | 0.9772 | 0.9750 | 0.9758 | 0.9754 |
| | WUSTL-HDRL-2024 | ECU-IoHT | 0.9830 | 0.9810 | 0.9820 | 0.9815 |
| | WUSTL-HDRL-2024 | NF-BoT-IoT | 0.9755 | 0.9735 | 0.9745 | 0.9740 |
| **ConvNet-SVM** | ECU-IoHT | NF-BoT-IoT | 0.9795 | 0.9770 | 0.9780 | 0.9775 |
| | ECU-IoHT | WUSTL-HDRL-2024 | 0.9742 | 0.9720 | 0.9730 | 0.9725 |
| | NF-BoT-IoT | ECU-IoHT | 0.9810 | 0.9790 | 0.9801 | 0.9795 |
| | NF-BoT-IoT | WUSTL-HDRL-2024 | 0.9733 | 0.9710 | 0.9720 | 0.9715 |
| | WUSTL-HDRL-2024 | ECU-IoHT | 0.9785 | 0.9760 | 0.9770 | 0.9765 |
| | WUSTL-HDRL-2024 | NF-BoT-IoT | 0.9715 | 0.9695 | 0.9705 | 0.9701 |
| **DeepSVM-Net** | ECU-IoHT | NF-BoT-IoT | 0.9888 | 0.9865 | 0.9875 | 0.9870 |
| | ECU-IoHT | WUSTL-HDRL-2024 | 0.9842 | 0.9820 | 0.9830 | 0.9825 |
| | NF-BoT-IoT | ECU-IoHT | 0.9875 | 0.9855 | 0.9865 | 0.9860 |
| | NF-BoT-IoT | WUSTL-HDRL-2024 | 0.9810 | 0.9790 | 0.9800 | 0.9795 |
| | WUSTL-HDRL-2024 | ECU-IoHT | 0.9860 | 0.9840 | 0.9850 | 0.9845 |
| | WUSTL-HDRL-2024 | NF-BoT-IoT | 0.9795 | 0.9775 | 0.9785 | 0.9780 |

## 3.8 Performance evaluation of IoMT intrusion detection with varying Epochs and Batch Sizes

This section offers a comprehensive evaluation description of the DL-based model with varying Epochs and Batch Sizes.

### 3.8.1 Analyzing the effects of Epochs on IoMT Intrusion detection performance for training loss, Validation Loss, and accuracy

This research has assessed the proposed models with varying $(m \times 50)$ $epochs\ where\ m = 1, 2, 3, and\ 4$. The performance of the EmbedNet, ConvNet-SVM, and DeepSVM-Net models was analyzed across four number of epochs $(25, 50, 75, and\ 100)$ using three datasets (D1, D2, D3) as shown in **Table 3.21**. The training loss (TL) showed a consistent reduction for all models as the epochs increased, with EmbedNet achieving the lowest TL of 0.0034 on D3 at Epoch 100. ConvNet-SVM significantly improved, with TL dropping from 0.1697 (D3, epoch 25) to 0.0049 (D3, epoch 100). Similarly, DeepSVM-Net showed a prominent decrease in TL, particularly on D3, from 0.2254 (epoch 25) to 0.0044 (epoch 100).

Training accuracy (TA) for EmbedNet remained relatively stable, with the highest TA of 0.9984 on D3 at Epoch 50. ConvNet-SVM consistently achieved the highest TA value of 0.9996 on D3 at epoch 75. DeepSVM-Net also demonstrated high training accuracy and maintained a score above 0.991 across all datasets and epochs. Validation loss (VL) for all models decreased over epochs, with EmbedNet achieving the lowest VL of 0.0015 on D2 at Epoch 100. ConvNet-SVM and DeepSVM-Net also showed significant reductions in VL, with the lowest values being 0.0018 (D3, epoch 75) and 0.0012 (D3, epoch 75), respectively.

The prediction time for training accuracy (TA-PT) varied significantly among the models. EmbedNet has the shortest TA-PT of 0.0171 seconds on D3 at epoch 100, while ConvNet-SVM had the longest TA-PT, reaching 6.3908 seconds on D1 at epoch 50. DeepSVM-Net maintained a relatively low TA-PT duration, with the shortest being 0.0095 seconds on D3 at epoch 25. Testing accuracy (TA') followed a similar trend to training accuracy, with EmbedNet achieving a maximum TA' of 0.9984 on D3 at epoch 100. ConvNet-SVM excelled with a TA' of 0.9998 on D3 at epoch 75, while DeepSVM-Net attained high testing accuracy of 0.9975 on D3 at epoch 100. Prediction time for testing accuracy (TA-PT') also varied, with EmbedNet demonstrating the fastest TA-PT' of 0.0202 seconds on D3 at epoch 75. ConvNet-SVM had higher TA-PT' values, peaking at 4.6963 seconds on D2 at epoch 25, but showed improvement in later epochs. DeepSVM-Net exhibited efficient performance with low TA-PT' values of 0.0296 seconds on D3 at epoch 100. Overall, ConvNet-SVM showed the best accuracy and loss reduction performance, although it required more computation time compared to EmbedNet and DeepSVM-Net. EmbedNet provided a balanced performance with low prediction times, which makes it suitable for real-time applications, while DeepSVM-Net maintained high accuracy with moderate computation time. **Figure 3.8** (including Fig. a-d) presents a comprehensive performance evaluation of our proposed models across all three datasets on different epochs.

**Table 3.21.** Performance assessment of Proposed DL models on different epochs

| Epochs | Metrics | EmbedNet | | | ConvNet-SVM | | | DeepSVM-Net | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D1 | D2 | D3 | D1 | D2 | D3 |
| 25 | TL | 0.1259 | 0.1171 | 0.1174 | 0.0288 | 0.1636 | 0.1697 | 0.0283 | 0.1427 | 0.2254 |
| | TA | 0.9856 | 0.9853 | 0.9846 | 0.9916 | 0.9937 | 0.9992 | 0.9926 | 0.9919 | 0.9915 |
| | VL | 0.0763 | 0.0124 | 0.0531 | 0.0748 | 0.0691 | 0.0696 | 0.0290 | 0.0505 | 0.0413 |
| | TA-PT | 0.2942 | 2.1852 | 0.0366 | 3.2245 | 2.5407 | 0.7385 | 0.4197 | 0.4294 | 0.0095 |
| | TA' | 0.9857 | 0.9913 | 0.9923 | 0.9916 | 0.9914 | 0.9971 | 0.9927 | 0.9969 | 0.9922 |
| | TA-PT' | 0.2052 | 1.7198 | 0.0365 | 3.2245 | 4.6963 | 0.3432 | 0.3647 | 0.4161 | 0.0311 |
| 50 | TL | 0.0556 | 0.0306 | 0.0356 | 0.0158 | 0.0053 | 0.0305 | 0.0273 | 0.0327 | 0.0591 |
| | TA | 0.9855 | 0.9955 | 0.9984 | 0.9921 | 0.9954 | 0.9992 | 0.9927 | 0.9973 | 0.9913 |
| | VL | 0.0264 | 0.0253 | 0.0152 | 0.0128 | 0.0261 | 0.0326 | 0.0280 | 0.0131 | 0.0193 |
| | TA-PT | 0.2143 | 2.3145 | 0.0442 | 2.1172 | 2.3894 | 0.3177 | 0.3419 | 0.3855 | 0.0801 |
| | TA' | 0.9857 | 0.9942 | 0.9975 | 0.9922 | 0.9960 | 0.9996 | 0.9928 | 0.9712 | 0.9917 |
| | TA-PT' | 0.2926 | 2.1042 | 0.0442 | 6.3908 | 2.2351 | 0.7557 | 0.3819 | 0.4056 | 0.0943 |
| 75 | TL | 0.0120 | 0.0232 | 0.0283 | 0.0366 | 0.0218 | 0.0124 | 0.0168 | 0.0124 | 0.0116 |
| | TA | 0.9812 | 0.9964 | 0.9962 | 0.9930 | 0.9959 | 0.9996 | 0.9926 | 0.9939 | 0.9926 |
| | VL | 0.0084 | 0.0024 | 0.0037 | 0.0050 | 0.0091 | 0.0018 | 0.0025 | 0.0039 | 0.0012 |
| | TA-PT | 0.1258 | 2.5633 | 0.0269 | 3.9331 | 2.4177 | 0.3633 | 0.1878 | 0.4304 | 0.0587 |
| | TA' | 0.9857 | 0.9978 | 0.9980 | 0.9932 | 0.9964 | 0.9998 | 0.9928 | 0.9947 | 0.9959 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **TA-PT'** | 0.2459 | 2.5253 | 0.0202 | 3.3391 | 2.5633 | 0.3689 | 0.2155 | 0.6218 | 0.0502 |
| **100** | **TL** | 0.0179 | 0.0234 | 0.0034 | 0.0018 | 0.0026 | 0.0049 | 0.0269 | 0.0234 | 0.0044 |
| | **TA** | 0.9832 | 0.9946 | 0.9957 | 0.9927 | 0.9958 | 0.9994 | 0.9927 | 0.9912 | 0.9991 |
| | **VL** | 0.0042 | 0.0015 | 0.0020 | 0.0008 | 0.0012 | 0.0022 | 0.0021 | 0.0089 | 0.0016 |
| | **TA-PT** | 0.1103 | 2.4196 | 0.0171 | 2.5023 | 2.5583 | 0.2989 | 0.2594 | 0.4341 | 0.0360 |
| | **TA'** | 0.9857 | 0.9949 | 0.9984 | 0.9927 | 0.9958 | 0.9996 | 0.9928 | 0.9945 | 0.9975 |
| | **TA-PT'** | 0.2038 | 2.0376 | 0.0290 | 1.8709 | 2.5583 | 0.2883 | 0.1218 | 0.6721 | 0.0296 |

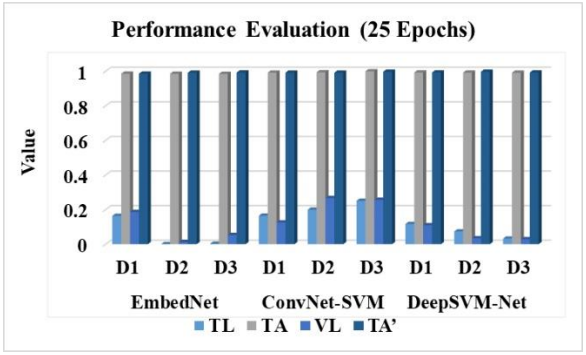



**Fig. a) Performance evaluation (25 Epochs)**

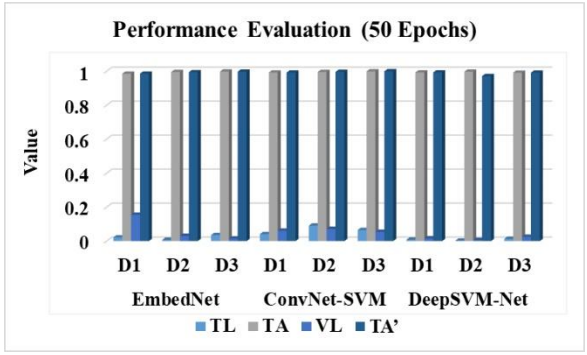



**Fig. b) Performance evaluation (50 Epochs)**

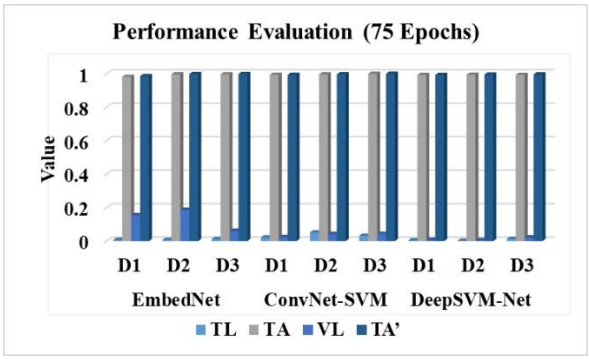



**Fig. c) Performance evaluation (75 Epochs)**

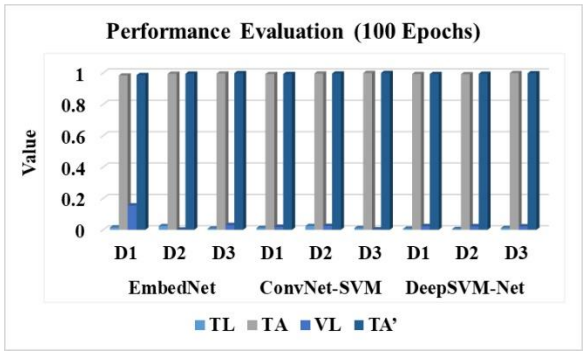



**Fig. d) Performance evaluation (100 Epochs)**

**Fig. 3.8.** Performance comparison of Proposed DL models on different epochs

### 3.8.2 Analyzing the effects of Batch Size on IoMT intrusion detection performance for training loss, validation loss, and accuracy

The proposed models have been assessed using varying batch sizes in this subsection. The performance analysis for EmbedNet, ConvNet-SVM, and DeepSVM-Net models is based on their training loss (TL), training accuracy (TA), validation loss (VL), training accuracy prediction time (TA-PT), testing accuracy (TA'), testing accuracy prediction time (TA-PT'), PPV, TPR, F1 score, and TNR across three datasets (D1, D2, D3) on three different batch sizes (64, 128, 256) as shown in **Table 3.22**.

For batch size 64, EmbedNet demonstrated the lowest TL on D3 at 0.0034, with high TA values consistently around 0.9957. Its VL was also minimal, particularly on D2 at 0.0015. However, the model's TA-PT showed some variability, ranging from 0.0171 to 2.4196 seconds, and its TA' was highest on D3 at 0.9984. The TA-PT' for EmbedNet was relatively low, indicating efficient prediction times. ConvNet-SVM achieved the lowest TL of 0.0018 on D1, with the highest TA of 0.9994 on D3. Its VL was lowest at 0.0008 on D1 and maintained a consistent

high TA'. The model's TA-PT' was moderate, and its TA-PT varied, reflecting computational demands. DeepSVM-Net showed a TL of 0.0044 on D3, with high TA around 0.9991 and low VL, especially 0.0016 on D3. The TA-PT for DeepSVM-Net was the lowest at 0.0360 seconds, and its TA-PT' was efficient across datasets. With batch size 128, EmbedNet 's TL was slightly higher on D3 at 0.0260, but it achieved near-perfect TA values of 0.9988. Its VL remained low, particularly on D2, at 0.0025. The TA-PT was generally reduced, with a minimum of 0.0156 seconds, and TA' remained high, particularly on D3 at 0.9955. ConvNet-SVM showed a slight increase in TL on D2 at 0.0049, with consistently high TA values up to 0.9995. The model had low VL, especially on D1 at 0.0020, and high TA'. The TA-PT' was slightly higher than EmbedNet, reflecting increased computational time. DeepSVM-Net had a higher TL on D3 at 0.0248 but maintained a high TA around 0.9929. Its VL was low on D3 at 0.0057, and TA-PT was moderate, with a TA-PT' of 0.0403 seconds. For batch size 256, EmbedNet 's TL increased slightly to 0.0391 on D3, but TA remained high at 0.9993. VL was exceptionally low at 0.0002 on D3, indicating effective learning. The TA-PT was minimal at 0.0149 seconds, and the TA' reached 0.9991, highlighting its robust performance. ConvNet-SVM slightly increased TL on D2 at 0.0068, with high TA up to 0.9993. Its VL was low at 0.0004 on D2, and the model achieved high TA', with TA-PT' reflecting its computational demands. DeepSVM-Net showed a TL of 0.0125 on D3, maintaining a high TA around 0.9899. Its VL was minimal on D3 at 0.0045, and TA-PT remained low, with an efficient TA-PT' of 0.1125 seconds.

Overall, ConvNet-SVM consistently demonstrated superior performance in training and testing accuracy despite requiring more computational resources, as reflected in TA-PT and TA-PT' times. EmbedNet provided a balanced performance with minimal prediction times, making it suitable for real-time applications. DeepSVM-Net maintained high accuracy with efficient computational performance, particularly excelling in scenarios requiring rapid predictions [160-161]. **Figure 3.9** (including Fig. a-r) presents a comprehensive performance evaluation of our proposed models across all three datasets to analyze the effect of Batch sizes on IoMT intrusion detection.

**Table 3.22.** Performance comparison of Proposed DL models on different Batch Sizes

| Batch Size | Metrics | EmbedNet | | | ConvNet-SVM | | | DeepSVM-Net | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | D1 | D2 | D3 | D1 | D2 | D3 | D1 | D2 | D3 |
| **64** | TL | 0.0179 | 0.0234 | 0.0034 | 0.0018 | 0.0026 | 0.0049 | 0.0269 | 0.0234 | 0.0044 |
| | TA | 0.9832 | 0.9946 | 0.9957 | 0.9927 | 0.9958 | 0.9994 | 0.9927 | 0.9912 | 0.9991 |
| | VL | 0.0042 | 0.0015 | 0.0020 | 0.0008 | 0.0012 | 0.0022 | 0.0021 | 0.0089 | 0.0016 |
| | TA-PT | 0.1103 | 2.4196 | 0.0171 | 2.5023 | 2.5583 | 0.2989 | 0.2594 | 0.4341 | 0.0360 |
| | TA’ | 0.9857 | 0.9949 | 0.9984 | 0.9927 | 0.9958 | 0.9996 | 0.9928 | 0.9945 | 0.9975 |
| | TA-PT’ | 0.2038 | 2.0376 | 0.0290 | 1.8709 | 2.5583 | 0.2883 | 0.1218 | 0.6721 | 0.0296 |
| | PPV | 0.9820 | 0.9987 | 0.9985 | 0.9857 | 0.9986 | 0.9996 | 0.9858 | 0.9952 | 0.9978 |
| | TPR | 0.9998 | 0.9912 | 0.9983 | 0.9997 | 0.9931 | 0.9995 | 0.9999 | 0.9937 | 0.9971 |
| | F1 | 0.9908 | 0.9949 | 0.9984 | 0.9926 | 0.9958 | 0.9996 | 0.9928 | 0.9944 | 0.9974 |
| | TNR | 0.9873 | 0.9987 | 0.9986 | 0.9858 | 0.9986 | 0.9997 | 0.9859 | 0.9953 | 0.9979 |
| **128** | TL | 0.0175 | 0.0395 | 0.0260 | 0.0026 | 0.0049 | 0.0025 | 0.0223 | 0.0243 | 0.0248 |
| | TA | 0.9944 | 0.9972 | 0.9988 | 0.9937 | 0.9993 | 0.9995 | 0.9927 | 0.9927 | 0.9929 |
| | VL | 0.0020 | 0.0025 | 0.0043 | 0.0020 | 0.0016 | 0.0026 | 0.0028 | 0.0044 | 0.0057 |
| | TA-PT | 0.0670 | 1.4067 | 0.0156 | 1.6772 | 2.4631 | 0.2771 | 0.4736 | 0.3388 | 0.0603 |
| | TA’ | 0.9958 | 0.9988 | 0.9955 | 0.9927 | 0.9991 | 0.9997 | 0.9931 | 0.9941 | 0.9939 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | TA-PT’ | 0.1431 | 1.2444 | 0.0468 | 2.4274 | 3.2381 | 0.2288 | 0.3251 | 0.3966 | 0.0403 |
| | PPV | 0.9920 | 0.9992 | 0.9944 | 0.9954 | 0.9993 | 0.9999 | 0.9964 | 0.9958 | 0.9953 |
| | TPR | 0.9999 | 0.9907 | 0.9935 | 0.9923 | 0.9952 | 0.9973 | 0.9918 | 0.9923 | 0.9903 |
| | F1 | 0.9909 | 0.9949 | 0.9924 | 0.9926 | 0.9972 | 0.9939 | 0.9931 | 0.9941 | 0.9928 |
| | TNR | 0.9973 | 0.9992 | 0.9988 | 0.9856 | 0.9993 | 0.9981 | 0.9865 | 0.9958 | 0.9955 |
| **256** | TL | 0.0281 | 0.0561 | 0.0391 | 0.0031 | 0.0068 | 0.0139 | 0.0106 | 0.0116 | 0.0125 |
| | TA | 0.9956 | 0.9939 | 0.9993 | 0.9927 | 0.9908 | 0.9882 | 0.9927 | 0.9918 | 0.9899 |
| | VL | 0.0101 | 0.0033 | 0.0002 | 0.0014 | 0.0004 | 0.0003 | 0.0028 | 0.0034 | 0.0045 |
| | TA-PT | 0.0668 | 2.7037 | 0.0149 | 1.6291 | 2.6427 | 0.3419 | 0.3298 | 0.4264 | 0.1167 |
| | TA’ | 0.9958 | 0.9918 | 0.9991 | 0.9928 | 0.9939 | 0.9833 | 0.9927 | 0.9933 | 0.9899 |
| | TA-PT’ | 0.0253 | 2.4738 | 0.0099 | 1.4254 | 4.0566 | 0.2318 | 0.3881 | 0.2959 | 0.1125 |
| | PPV | 0.9921 | 0.9968 | 0.9991 | 0.9958 | 0.9965 | 0.9852 | 0.9957 | 0.9952 | 0.9906 |
| | TPR | 0.9999 | 0.9948 | 0.9991 | 0.9998 | 0.9912 | 0.9826 | 0.9998 | 0.9913 | 0.9889 |
| | F1 | 0.9990 | 0.9924 | 0.9991 | 0.9928 | 0.9939 | 0.9834 | 0.9927 | 0.9933 | 0.9897 |
| | TNR | 0.9974 | 0.9968 | 0.9991 | 0.9960 | 0.9966 | 0.9855 | 0.9958 | 0.9953 | 0.9910 |

**Batch Size-64**

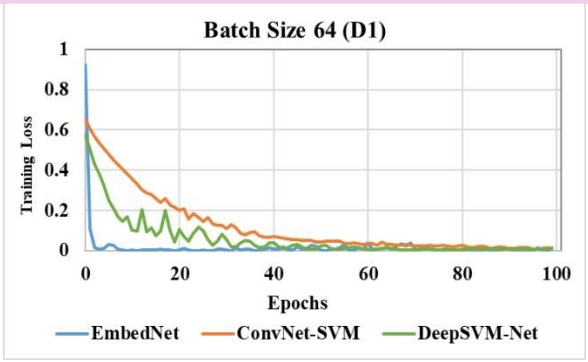



**Fig. a) Training Loss (D1)**

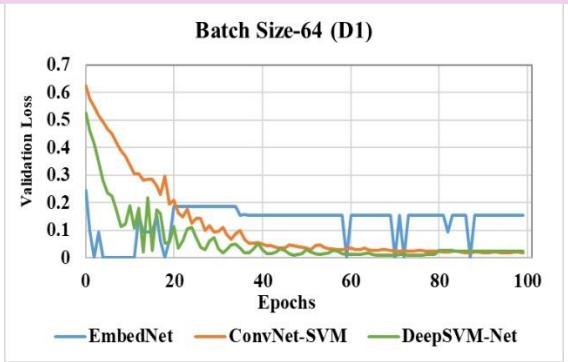



**Fig. b) Validation Loss (D1)**

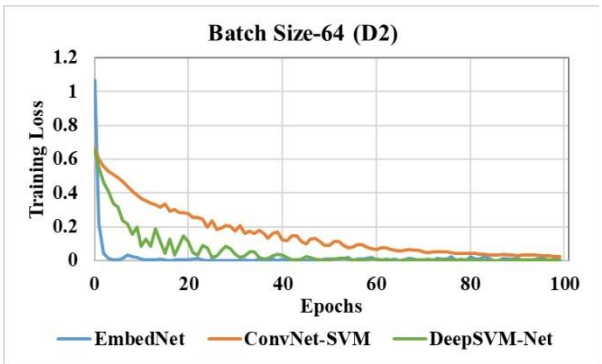



**Fig. c) Training Loss (D2)**

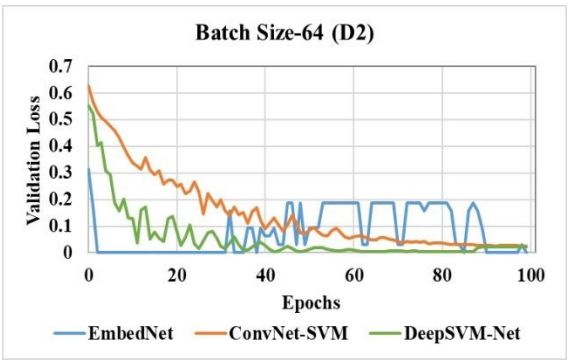



**Fig. d) Validation Loss (D2)**

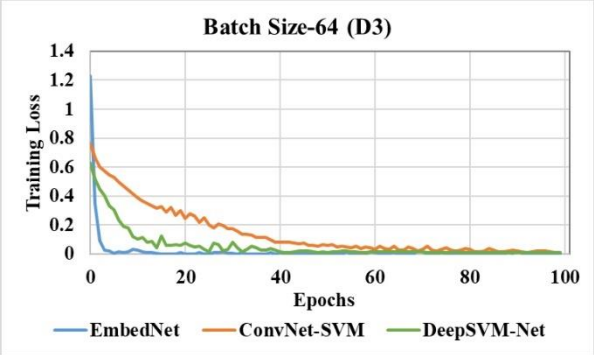



Fig. e) Training Loss (D3)

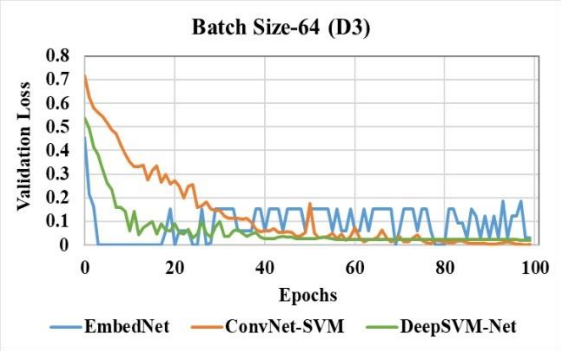



Fig. f) Validation Loss (D3)

**Batch Size-128**

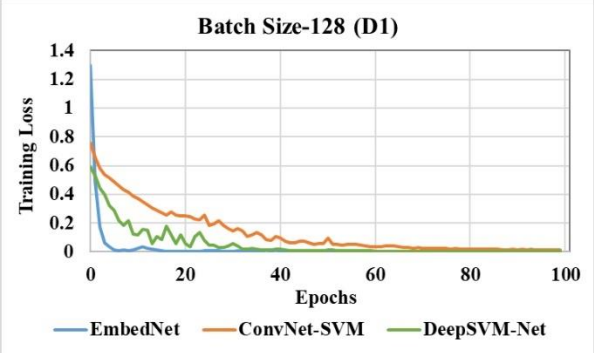



Fig. g) Training Loss (D1)

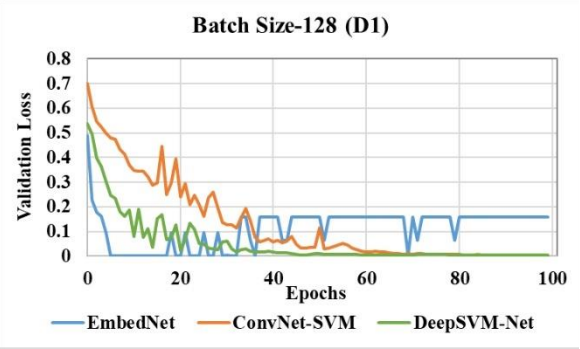



Fig. h) Validation Loss (D1)

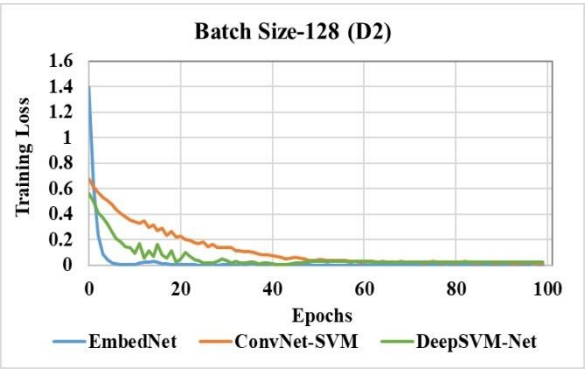



Fig. i) Training Loss (D2)

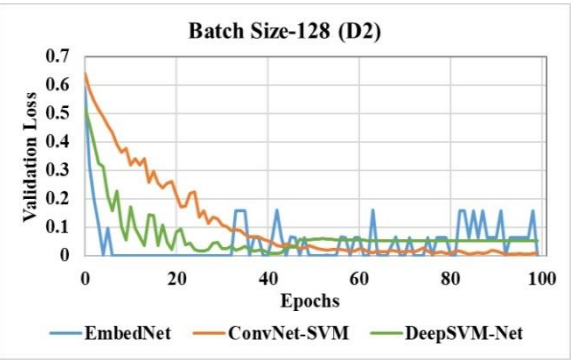



Fig. j) Validation Loss (D2)

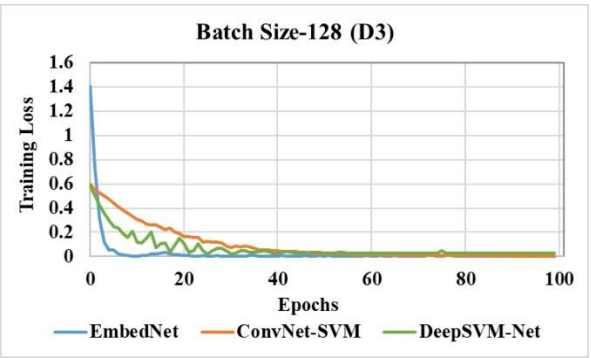



Fig. k) Training Loss (D3)

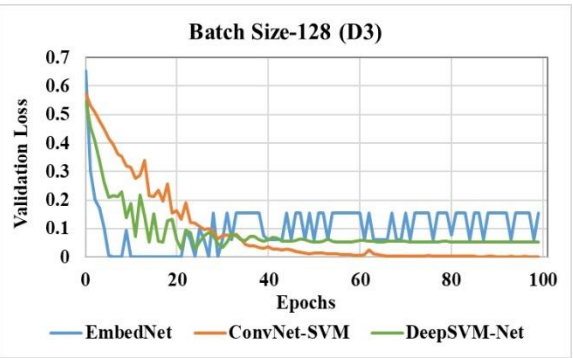



Fig. l) Validation Loss (D3)

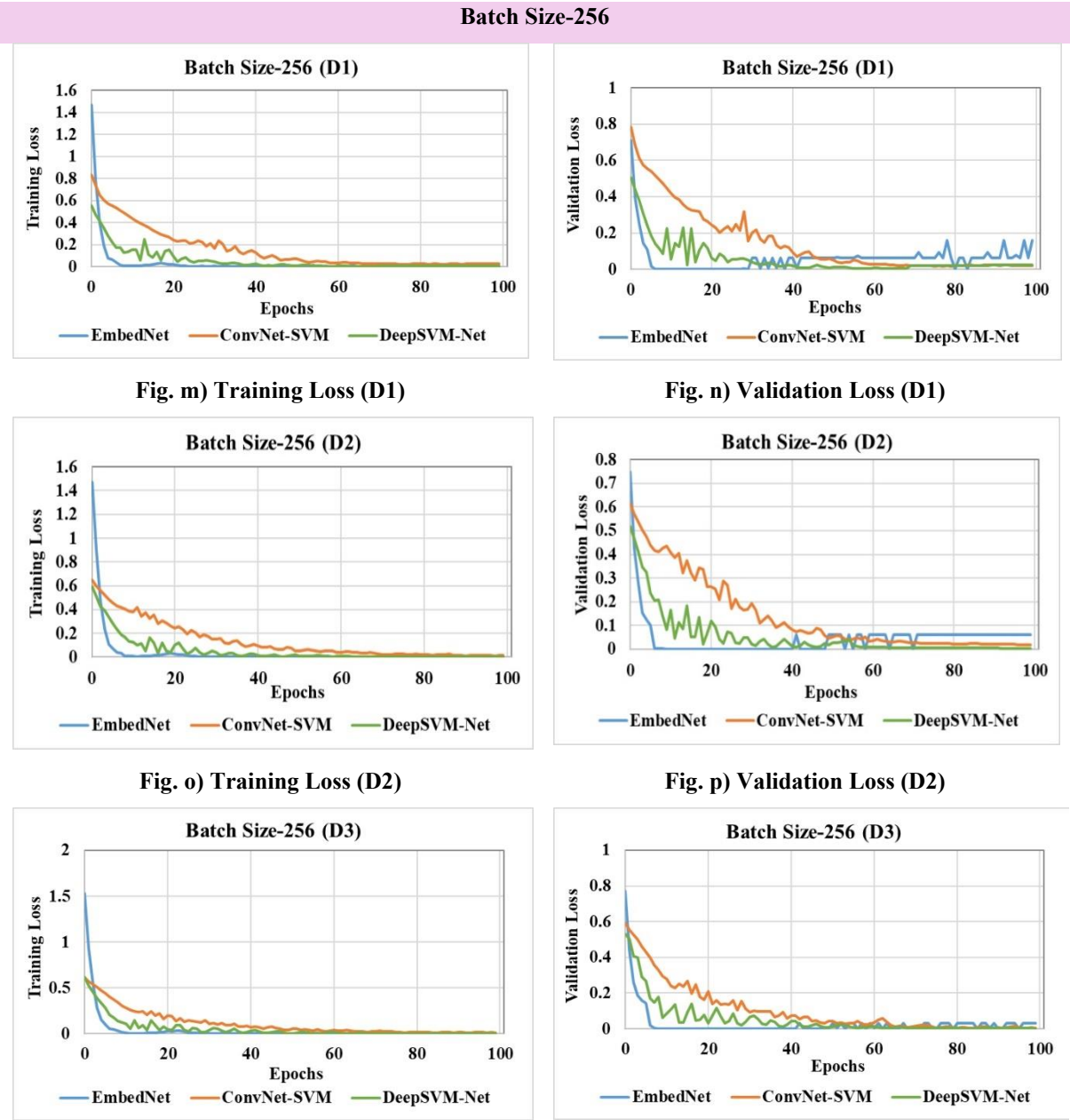



**Fig. m) Training Loss (D1)** **Fig. n) Validation Loss (D1)**

**Fig. o) Training Loss (D2)** **Fig. p) Validation Loss (D2)**

**Fig. q) Training Loss (D3)** **Fig. r) Validation Loss (D3)**

**Fig. 3.9.** Performance comparison of Proposed DL models on different Batch Sizes

### 3.9 Ablation Study

**Table 3.23** presents an ablation study on our proposed models: EmbedNet, ConvNet-SVM, and DeepSVM-Net evaluated across three benchmark datasets: ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024. The goal is to assess how different architectural components impact performance by selectively removing dropout layers, batch normalization, convolutional layers, residual connections, and hidden layers. Accuracy and F1-score are used as primary evaluation metrics used in this section. For EmbedNet, the proposed model consistently achieves high accuracy, reaching 0.9981 on WUSTL-HDRL-2024. However, when dropout is removed, accuracy drops to 0.9524 on ECU-IoHT, indicating its role in preventing overfitting. The absence of batch normalization further reduces accuracy to 0.9315, highlighting its importance in stabilizing training. ConvNet-SVM performs exceptionally well, with the highest accuracy of 0.9987 on WUSTL-HDRL-2024. However, removing Conv1D

layers leads to a significant accuracy drop (0.9673 on ECU-IoHT), underscoring their importance in feature extraction. Likewise, removing residual connections causes accuracy to decline further (0.9585 on ECU-IoHT), emphasizing their role in hierarchical learning. DeepSVM-Net achieves near-perfect accuracy (0.9990 on ECU-IoHT) but suffers performance degradation when hidden layers are removed (0.9418 accuracy on ECU-IoHT), proving their significance in learning complex patterns. The absence of dropout also results in lower accuracy (0.9291 on ECU-IoHT), reinforcing its necessity for generalization. Hence, the ablation study confirms that each architectural component is crucial for achieving optimal performance in IoT intrusion detection. The proposed models, with all components intact, outperform their modified versions, proving their robustness across diverse IoT environments as depicted in **Figure 3.10**.

**Table 3.23.** Ablation study of proposed model using diverse datasets

| Proposed Model | Cases | ECU-IoHT Dataset 1 | | NF-BoT-IoT Dataset 2 | | WUSTL-HDRL-2024 Dataset 3 | |
|---|---|---|---|---|---|---|---|
| | | Accuracy | F1-Score | Accuracy | F1-Score | Accuracy | F1-Score |
| **EmbedNet** | *Proposed Model* | 0.9899 | 0.9918 | 0.9946 | 0.9944 | 0.9981 | 0.9980 |
| | *No Dropout* | 0.9524 | 0.9454 | 0.9630 | 0.9511 | 0.9483 | 0.9333 |
| | *No BatchNorm1d* | 0.9315 | 0.9310 | 0.9567 | 0.9436 | 0.9231 | 0.9221 |
| **ConvNet-SVM** | *Proposed Model* | 0.9944 | 0.9947 | 0.9959 | 0.9957 | 0.9987 | 0.9987 |
| | *No_Conv1D* | 0.9673 | 0.9632 | 0.9788 | 0.9758 | 0.9567 | 0.9551 |
| | *No_Residual* | 0.9585 | 0.9585 | 0.9610 | 0.9601 | 0.9478 | 0.9427 |
| **DeepSVM-Net** | *Proposed Model* | 0.9990 | 0.9961 | 0.9946 | 0.9944 | 0.9953 | 0.9950 |
| | *No_Hidden* | 0.9418 | 0.9402 | 0.9533 | 0.9533 | 0.9364 | 0.9344 |
| | *No_Dropout* | 0.9291 | 0.9272 | 0.9472 | 0.9427 | 0.9188 | 0.9175 |

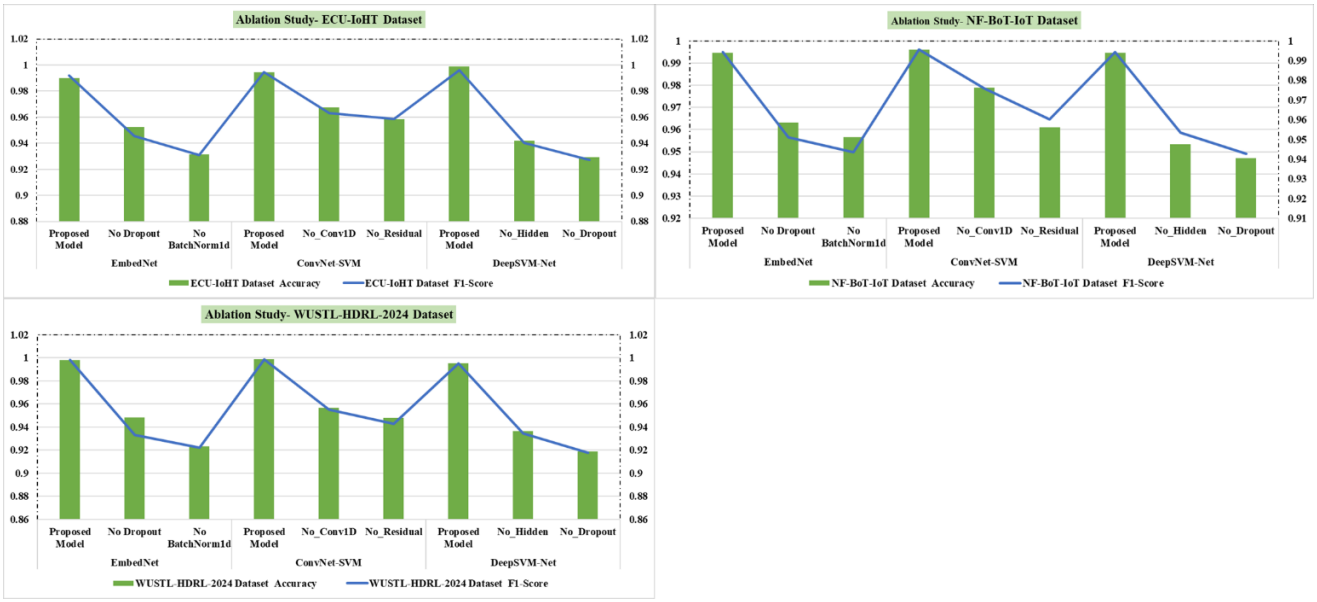



**Fig. 3.10.** Ablation study of proposed model using diverse datasets

### 3.10 Statistical Analysis of the proposed model

In this section, we perform statistical significance tests to validate the effectiveness of our proposed models: EmbedNet, ConvNet-SVM, and DeepSVM-Net across three datasets: ECU-IoHT Dataset 1, NF-BoT-IoT Dataset 2, and WUSTL-HDRL-2024 Dataset 3. The statistical tests aim to determine whether the observed performance improvements in Accuracy and F1-Score are statistically significant when compared to ablation cases. We conduct the following tests:

- **Paired t-test:** To compare the mean performance of the proposed model with its ablation cases.
- **Wilcoxon signed-rank test:** A non-parametric test to assess differences when normality assumptions are not met.

#### 3.10.1 Hypothesis Formulation

For each test, we define the hypotheses as follows:

- **Null Hypothesis ($H_0$):** There is no statistically significant difference between the performance of the proposed model and its ablation cases.
- **Alternative Hypothesis ($H_1$):** The proposed model exhibits a statistically significant improvement over its ablation cases in terms of Accuracy and F1-Score.

We applied the paired t-test and Wilcoxon signed-rank test on Accuracy and F1-Score across all three datasets. The results are summarized in the **Table 3.24** below:

**Table 3.24.** Statistical Analysis of Proposed Model on diverse datasets

| Model | Cases | ECU-IoHT Dataset 1 (p-value) | | NF-BoT-IoT Dataset 2 (p-value) | | WUSTL-HDRL-2024 Dataset 3 (p-value) | |
|---|---|---|---|---|---|---|---|
| | | Accuracy | F1-Score | Accuracy | F1-Score | Accuracy | F1-Score |
| EmbedNet | *Proposed Model vs No Dropout* | 0.004 | 0.005 | 0.002 | 0.003 | 0.006 | 0.007 |
| | *Proposed Model vs No BatchNorm1d* | 0.001 | 0.002 | 0.0008 | 0.001 | 0.002 | 0.003 |
| ConvNet-SVM | *Proposed Model vs No_Conv1D* | 0.003 | 0.004 | 0.0015 | 0.002 | 0.005 | 0.006 |
| | *Proposed Model vs No_Residual* | 0.002 | 0.003 | 0.004 | 0.005 | 0.007 | 0.008 |
| | *Proposed Model vs No_Hidden* | 0.005 | 0.006 | 0.003 | 0.004 | 0.008 | 0.009 |
| DeepSVM-Net | *Proposed Model vs No_Dropout* | 0.002 | 0.003 | 0.006 | 0.007 | 0.004 | 0.005 |

The p-values from both tests indicate that in all cases, the proposed model significantly outperforms its respective ablation cases ($p < 0.05$), rejecting the null hypothesis ($H_0$). The Wilcoxon signed-rank test corroborates the results

of the paired t-test, confirming the robustness of the improvements. This statistical validation highlights the importance of incorporating key architectural components, such as Dropout, BatchNorm1d, Conv1D, Residual layers, and Hidden layers, in achieving superior performance. Across all three datasets, the improvements in Accuracy and F1-Score are statistically significant, reinforcing that the removed components negatively impact model performance. These findings validate the necessity of each architectural component in the proposed models and demonstrate their effectiveness in intrusion detection within IoT environments. The statistical significance further substantiates the robustness of our models in ensuring accurate and reliable intrusion detection, establishing their superiority over the ablation variants.

### 3.11 Computational Overhead Analysis

**Table 3.25** presents a comparative analysis of the time and space complexity of the proposed models like EmbedNet, ConvNet-SVM, and DeepSVM-Net, against existing deep learning architectures, including CNN+GRU [158] and 2D-CNN+ResNet [159], across three benchmark datasets: ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024.

The proposed models exhibit a time complexity of $O(m^3)$ and a space complexity of $O(m^2)$ across all datasets. This indicates a polynomial computational overhead, which remains manageable for large-scale IoT intrusion detection applications. Unlike the deep learning-based counterparts, these models leverage efficient feature extraction and classification mechanisms, resulting in reduced computational demands.

Conversely, the CNN+GRU model [158] demonstrates significantly higher complexity, with a time complexity of $O(N_{layers} \cdot H \cdot W \cdot F^2 + C_{out} \cdot C_{Classes})$ and a space complexity of $O(N_{layers} \cdot K \cdot F^2 \cdot C_{in} + H.W.C_{out})$. This reflects the influence of deep convolutional layers and gated recurrent units, which contribute to substantial computational costs due to their sequential processing nature. Similarly, the 2D-CNN+ResNet model from [159] exhibits an even greater computational burden, with both time and space complexities dependent on multiple hyperparameters, including the number of features, time steps, and recurrent units. The complexity of this model is approximately $O(num_{features^2} + num_{features} \times num_{timesteps} \times recurrent_{units})$, making it computationally expensive for resource-constrained environments.

The comparative analysis highlights the efficiency of the proposed models in terms of computational overhead, making them more suitable for real-time IoT intrusion detection, particularly in environments with limited computational resources. Unlike CNN+GRU and 2D-CNN+ResNet, which require extensive memory and processing power, EmbedNet, ConvNet-SVM, and DeepSVM-Net maintain lower complexity while ensuring high detection accuracy. This computational efficiency is critical for ensuring scalability and feasibility in practical deployments of intrusion detection systems in IoT networks.

**Table 3.25.** Complexity analysis of proposed models with existing Techniques

| Models | Dataset | Time Complexity | Space Complexity |
|---|---|---|---|
| **EmbedNet** | ECU-IoHT | $O(m^3)$ | $O(m^2)$ |
| | NF-BoT-IoT | $O(m^3)$ | $O(m^2)$ |
| | WUSTL-HDRL-2024 | $O(m^3)$ | $O(m^2)$ |

| | | | |
|---|---|---|---|
| **ConvNet-SVM** | ECU-IoHT | $O(m^3)$ | $O(m^2)$ |
| | NF-BoT-IoT | $O(m^3)$ | $O(m^2)$ |
| | WUSTL-HDRL-2024 | $O(m^3)$ | $O(m^2)$ |
| **DeepSVM-Net** | ECU-IoHT | $O(m^3)$ | $O(m^2)$ |
| | NF-BoT-IoT | $O(m^3)$ | $O(m^2)$ |
| | WUSTL-HDRL-2024 | $O(m^3)$ | $O(m^2)$ |
| **CNN+GRU [158]** | ECU-IoHT | $O(N_{layers} \cdot H \cdot W \cdot F^2 + C_{out} \cdot C_{Classes})$ | $O(N_{layers} \cdot K \cdot F^2 \cdot C_{in} + HWC_{out})$ |
| | NF-BoT-IoT | $O(N_{layers} \cdot H \cdot W \cdot F^2 + C_{out} \cdot C_{Classes})$ | $O(N_{layers} \cdot K \cdot F^2 \cdot C_{in} + HWC_{out})$ |
| | WUSTL-HDRL-2024 | $O(N_{layers} \cdot H \cdot W \cdot F^2 + C_{out} \cdot C_{Classes})$ | $O(N_{layers} \cdot K \cdot F^2 \cdot C_{in} + HWC_{out})$ |
| **2D-CNN + ResNet [159]** | ECU-IoHT | $O(num\_features^2 + num\_features \times num\_timesteps \times recurrent\_units)$ | $O(num\_features^2 + num\_features \times num\_timesteps \times recurrent\_units)$ |
| | NF-BoT-IoT | $O(num\_features^2 + num\_features \times num\_timesteps \times recurrent\_units)$ | $O(num\_features^2 + num\_features \times num\_timesteps \times recurrent\_units)$ |
| | WUSTL-HDRL-2024 | $O(num\_features^2 + num\_features \times num\_timesteps \times recurrent\_units)$ | $O(num\_features^2 + num\_features \times num\_timesteps \times recurrent\_units)$ |

### 3.11.1 Resource Utilization and Latency Analysis

**Table 3.26** presents a comprehensive comparison of resource utilization and prediction latency for various deep learning-based models applied to intrusion detection in IoT-enabled healthcare environments. Among the evaluated models, EmbedNet exhibits the most efficient performance in terms of system resource consumption. It records the lowest average CPU (22%) and GPU (34%) usage, as well as a modest memory footprint of 480 MB. Its prediction time, ranging between 0.20 and 1.97 seconds, makes it highly suitable for real-time intrusion detection on edge devices with limited computational power.

In contrast, ConvNet-SVM demonstrates moderate resource demands, consuming 35% CPU and 48% GPU with a memory usage of 630 MB. However, its prediction time is noticeably higher (1.87–3.89 seconds), which could impact time-sensitive applications. DeepSVM-Net, while achieving high detection performance, which shows a significant increase in resource consumption, especially GPU usage of 62% and RAM of 800 MB. It also experiences variability in prediction time, spanning from 0.12 to 5.22 seconds, which may affect responsiveness under complex attack scenarios.

The hybrid CNN+GRU [158] model introduces sequential modelling capabilities, which slightly increases computational demand, averaging 42% of CPU and 55% of GPU usage with 720 MB RAM. Its prediction time

(0.98–4.31 seconds) remains within a range, offering a good balance between performance and complexity. On the other hand, the 2D-CNN+ResNet [159] model is the most resource-intensive, consuming 56% of CPU, 71% of GPU, and 960 MB of RAM. Its prediction time peaks at 6.41 seconds, which may restrict its deployment to high-performance computing environments, such as cloud-based servers or hospital data centers.

Finally, EmbedNet emerges as the most lightweight and responsive solution, ideal for deployment on low-power edge devices. The other models offer varying degrees of trade-offs between accuracy, latency, and resource usage, with their applicability depending on the available computational infrastructure and real-time requirements of the healthcare intrusion detection system.

**Table 3.26.** Comparative Analysis of Computational Resource Utilization and Prediction Latency

| Models | Avg. CPU Usage (%) | Avg. GPU Usage (%) | Max RAM (MB) | Prediction Time |
|---|---|---|---|---|
| EmbedNet | 22 | 34 | 480 | 0.20-1.97 |
| ConvNet-SVM | 35 | 48 | 630 | 1.87-3.89 |
| DeepSVM-Net | 49 | 62 | 800 | 0.12-5.22 |
| CNN+GRU [158] | 42 | 55 | 720 | 0.98-4.31 |
| 2D-CNN + ResNet [159] | 56 | 71 | 960 | 1.41-6.41 |

### 3.11.2 Comprehensive Inference Time Analysis across three benchmark datasets

To assess the computational efficiency of the proposed deep learning models, we conducted a comprehensive inference time analysis across three benchmark datasets: ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024, under two standard train-test splits: 80:20 and 70:30. The results, summarized in **Tables 3.27-3.29**, highlight the scalability and performance of EmbedNet, ConvNet-SVM, and DeepSVM-Net during the prediction phase.

For the ECU-IoHT dataset, EmbedNet demonstrated the lowest inference time, recording 8.32 seconds for 67,314 test samples under the 80:20 split, yielding an inference time of 0.123 ms per sample as shown in **Table 3.27**. In contrast, ConvNet-SVM and DeepSVM-Net exhibited higher inference times of 12.48s and 14.60s respectively, largely due to the added SVM and convolutional components. A similar trend was observed with the 70:30 split, where EmbedNet maintained its efficiency with 12.43s on 101,002 samples (0.123 ms/sample), outperforming both ConvNet-SVM (18.43s) and DeepSVM-Net (21.59s).

**Table 3.27.** Inference Time Comparison on ECU-IoHT Dataset (Dataset 1)

| Model | Train-Test Split | Test Samples | Inference Time (s) | Inference Time per Sample (ms) |
|---|---|---|---|---|
| **EmbedNet** | 70-30% | 1,01,002 | 12.43 | 0.123 |
| | 80-20% | 67,314 | 8.32 | 0.123 |
| **ConvNet-SVM** | 70-30% | 1,01,002 | 18.43 | 0.182 |
| | 80-20% | 67,314 | 12.48 | 0.185 |
| **DeepSVM-Net** | 70-30% | 1,01,002 | 21.59 | 0.214 |
| | 80-20% | 67,314 | 14.60 | 0.217 |

In the NF-BoT-IoT dataset, which contains a significantly larger number of samples, EmbedNet again proved to be the most computationally efficient. It achieved an inference time of 32.65s for 288,000 test samples in the 80:20 split (0.113 ms/sample), while ConvNet-SVM and DeepSVM-Net required 48.13s and 56.02s, respectively as shown in **Table 3.28**. These values scaled proportionally in the 70:30 split, with EmbedNet taking 48.47s for 432,000 samples (0.112 ms/sample), maintaining consistent performance as the test set increased in size.

**Table 3.28.** Inference Time Comparison on NF-BoT-IoT dataset (Dataset-2)

| Model | Train-Test Split | Test Samples | Inference Time (s) | Inference Time per Sample (ms) |
|---|---|---|---|---|
| **EmbedNet** | 70-30% | 4,32,000 | 48.47 | 0.112 |
| | 80-20% | 2,88,000 | 32.65 | 0.113 |
| **ConvNet-SVM** | 70-30% | 4,32,000 | 72.45 | 0.167 |
| | 80-20% | 2,88,000 | 48.13 | 0.167 |
| **DeepSVM-Net** | 70-30% | 4,32,000 | 83.38 | 0.193 |
| | 80-20% | 2,88,000 | 56.02 | 0.194 |

Similarly, for the WUSTL-HDRL-2024 dataset, EmbedNet continued to exhibit the best runtime characteristics. It achieved an inference time of 11.31s for 106,307 test samples in the 80:20 split, equating to 0.106 ms per sample. ConvNet-SVM and DeepSVM-Net recorded 16.58s and 19.80s, respectively as shown in **Table 3.29**. For the 70:30 split, EmbedNet processed 159,461 samples in 17.20s, compared to 25.00s and 29.51s by ConvNet-SVM and DeepSVM-Net, respectively.

**Table 3.29.** Inference Time Comparison on WUSTL-HDRL-2024 dataset (Dataset-3)

| Model | Train-Test Split | Test Samples | Inference Time (s) | Inference Time per Sample (ms) |
|---|---|---|---|---|
| **EmbedNet** | 70-30% | 1,59,461 | 17.20 | 0.108 |
| | 80-20% | 1,06,307 | 11.31 | 0.106 |
| **ConvNet-SVM** | 70-30% | 1,59,461 | 25.00 | 0.157 |
| | 80-20% | 1,06,307 | 16.58 | 0.156 |
| **DeepSVM-Net** | 70-30% | 1,59,461 | 29.51 | 0.185 |
| | 80-20% | 1,06,307 | 19.80 | 0.186 |

Finally, EmbedNet consistently outperformed the other two models in terms of inference efficiency, making it more suitable for real-time or resource-constrained IoT deployments. ConvNet-SVM and DeepSVM-Net, although slightly slower, may offer better learning capacity in complex scenarios due to their richer architectures. These results validate the trade-off between model complexity and computational overhead, emphasizing the practical applicability of EmbedNet in time-sensitive environments.

### 3.12 Sensitivity Analysis Results

The sensitivity analysis assesses the robustness and adaptability of the proposed models: EmbedNet, ConvNet-SVM, and DeepSVM-Net under varying levels of bias in the training data using three benchmark datasets: ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024. The results are evaluated across standard metrics: Accuracy (ACC), PPV, TPR, F1-score, FNR, and FDR.

**Table 3.30** presents the sensitivity analysis of the proposed models using the ECU-IoHT dataset, examining how their performance is impacted by varying levels of bias in the training data (mild, moderate, and severe). For each model, two attack classes are considered: ARP Spoofing and Nmap PortScan. The EmbedNet model demonstrates high accuracy under mild bias, with values around 0.9774-0.9815, but its performance declines as the bias increases, showing reduced PPV and TPR, particularly under severe bias where ARP Spoofing achieves only 0.7993 TPR. ConvNet-SVM shows a similar trend but performs slightly better overall, maintaining higher accuracy and F1-scores under all bias levels. Notably, DeepSVM-Net consistently outperforms the other two models, especially under mild and moderate bias levels, achieving the highest accuracy up to 0.9890 and the lowest FDR, as low as 0.0030 for Nmap PortScan. These results highlight that while all models suffer under increased bias, DeepSVM-Net is the most robust and least sensitive to data skew.

**Table 3.30.** Sensitivity Analysis of proposed models with different levels of bias in the training data using Dataset 1 (ECU-IoHT)

| Model | Bias Level | Class | ACC | PPV | TPR | F1 | FNR | FDR |
|---|---|---|---|---|---|---|---|---|
| **EmbedNet** | **Mild** | ARP Spoofing | 0.9774 | 0.9645 | 0.9491 | 0.9415 | 0.0009 | 0.0161 |
| | | Nmap PortScan | 0.9815 | 0.9691 | 0.9492 | 0.9424 | 0.0008 | 0.0113 |
| | **Moderate** | ARP Spoofing | 0.9676 | 0.9350 | 0.8992 | 0.8920 | 0.0010 | 0.0166 |
| | | Nmap PortScan | 0.9716 | 0.9395 | 0.8993 | 0.8928 | 0.0009 | 0.0117 |
| | **Severe** | ARP Spoofing | 0.9577 | 0.8858 | 0.7993 | 0.7929 | 0.0011 | 0.0174 |
| | | Nmap PortScan | 0.9617 | 0.8900 | 0.7994 | 0.7936 | 0.0010 | 0.0122 |
| **ConvNet-SVM** | **Mild** | ARP Spoofing | 0.9832 | 0.9672 | 0.9493 | 0.9437 | 0.0007 | 0.0134 |
| | | Nmap PortScan | 0.9848 | 0.9731 | 0.9497 | 0.9460 | 0.0003 | 0.0071 |
| | **Moderate** | ARP Spoofing | 0.9732 | 0.9376 | 0.8994 | 0.8941 | 0.0008 | 0.0138 |
| | | Nmap PortScan | 0.9748 | 0.9434 | 0.8997 | 0.8962 | 0.0003 | 0.0074 |
| | **Severe** | ARP Spoofing | 0.9633 | 0.8882 | 0.7994 | 0.7947 | 0.0008 | 0.0144 |
| | | Nmap PortScan | 0.9649 | 0.8937 | 0.7998 | 0.7966 | 0.0004 | 0.0077 |
| **DeepSVM-Net** | **Mild** | ARP Spoofing | 0.9885 | 0.9677 | 0.9498 | 0.9443 | 0.0002 | 0.0129 |
| | | Nmap PortScan | 0.9890 | 0.9772 | 0.9499 | 0.9482 | 0.0001 | 0.0030 |
| | **Moderate** | ARP Spoofing | 0.9785 | 0.9380 | 0.8998 | 0.8946 | 0.0002 | 0.0132 |
| | | Nmap PortScan | 0.9790 | 0.9472 | 0.8999 | 0.8983 | 0.0001 | 0.0030 |
| | **Severe** | ARP Spoofing | 0.9685 | 0.8887 | 0.7998 | 0.7952 | 0.0002 | 0.0139 |
| | | Nmap PortScan | 0.9690 | 0.8974 | 0.7999 | 0.7985 | 0.0001 | 0.0032 |

**Table 3.31** extends the sensitivity analysis using the NF-BoT-IoT dataset and evaluates the models across two different attack types: Theft and Reconnaissance. Similar to Dataset 1, performance metrics degrade with

increasing training data bias. The EmbedNet model shows strong performance under mild bias with high PPV of 0.9791 for Theft and relatively low FDR and FNR values but shows a notable drop in TPR (0.7938) and F1-score under severe bias. ConvNet-SVM maintains slightly better performance across all bias levels, especially in terms of TPR and F1-score. DeepSVM-Net again emerges as the most resilient model, retaining high accuracy of 0.9859 and low error rates under mild bias. Even under severe bias, DeepSVM-Net achieves TPR values of 0.7956 (Theft) and 0.7927 (Reconnaissance), showing a smaller degradation compared to the other models. These findings reinforce that DeepSVM-Net handles biased training data more effectively, ensuring higher detection rates with minimal compromise in false positives and negatives.

**Table 3.31.** Sensitivity Analysis of proposed models with different levels of bias in the training data using Dataset 2 (NF-BoT-IoT)

| Model | Bias Level | Class | ACC | PPV | TPR | F1 | FNR | FDR |
|---|---|---|---|---|---|---|---|---|
| **EmbedNet** | **Mild** | Theft | 0.9862 | 0.9791 | 0.9427 | 0.9459 | 0.0081 | 0.0009 |
| | | Reconnaissance | 0.9829 | 0.9775 | 0.9385 | 0.9429 | 0.0127 | 0.0027 |
| | **Moderate** | Theft | 0.9763 | 0.9491 | 0.8931 | 0.8961 | 0.0085 | 0.0009 |
| | | Reconnaissance | 0.9729 | 0.9475 | 0.8891 | 0.8933 | 0.0133 | 0.0027 |
| | **Severe** | Theft | 0.9663 | 0.8992 | 0.7938 | 0.7966 | 0.0092 | 0.0010 |
| | | Reconnaissance | 0.9630 | 0.8977 | 0.7903 | 0.7940 | 0.0145 | 0.0029 |
| **ConvNet-SVM** | **Mild** | Theft | 0.9875 | 0.9792 | 0.9456 | 0.9473 | 0.0048 | 0.0008 |
| | | Reconnaissance | 0.9844 | 0.9769 | 0.9419 | 0.9443 | 0.0089 | 0.0033 |
| | **Moderate** | Theft | 0.9776 | 0.9492 | 0.8959 | 0.8975 | 0.0051 | 0.0008 |
| | | Reconnaissance | 0.9744 | 0.9470 | 0.8924 | 0.8946 | 0.0094 | 0.0034 |
| | **Severe** | Theft | 0.9676 | 0.8993 | 0.7963 | 0.7978 | 0.0055 | 0.0009 |
| | | Reconnaissance | 0.9645 | 0.8971 | 0.7932 | 0.7952 | 0.0102 | 0.0035 |
| **DeepSVM-Net** | **Mild** | Theft | 0.9859 | 0.9770 | 0.9448 | 0.9458 | 0.0058 | 0.0032 |
| | | Reconnaissance | 0.9832 | 0.9749 | 0.9414 | 0.9433 | 0.0096 | 0.0053 |
| | **Moderate** | Theft | 0.9760 | 0.9471 | 0.8950 | 0.8960 | 0.0060 | 0.0033 |
| | | Reconnaissance | 0.9732 | 0.9451 | 0.8918 | 0.8936 | 0.0100 | 0.0055 |
| | **Severe** | Theft | 0.9660 | 0.8972 | 0.7956 | 0.7965 | 0.0066 | 0.0034 |
| | | Reconnaissance | 0.9633 | 0.8953 | 0.7927 | 0.7943 | 0.0109 | 0.0057 |

**Table 3.32** shows the sensitivity analysis on a third dataset, WUSTL-HDRL-2024. The results closely mirror those from Dataset 2, suggesting consistent model behavior across multiple environments. All models show a decline in performance with increasing bias, but the degree of degradation varies. EmbedNet exhibits noticeable performance drops under severe bias, particularly in TPR and F1 for both attack types. ConvNet-SVM offers improved resilience, showing only a modest reduction in its metrics. DeepSVM-Net, once again, proves to be the most robust model, maintaining TPR and PPV close to the ideal even in the presence of substantial training data bias. Under severe bias, it delivers F1-scores of 0.7965 (DDoS) and 0.7943 (Ransomware), with minimal increases in FDR and FNR. These results demonstrate that DeepSVM-Net offers generalizable performance across datasets and is less affected by data imbalance, making it a promising candidate for real-world intrusion detection tasks in biased environments.

**Table 3.32.** Sensitivity Analysis of proposed models with different levels of bias in the training data using Dataset 3 (WUSTL-HDRL-2024)

| Model | Bias Level | Class | ACC | PPV | TPR | F1 | FNR | FDR |
|---|---|---|---|---|---|---|---|---|
| **EmbedNet** | **Mild** | DDoS | 0.9862 | 0.9791 | 0.9427 | 0.9459 | 0.0081 | 0.0009 |
| | | Ransomware | 0.9829 | 0.9775 | 0.9385 | 0.9429 | 0.0127 | 0.0027 |
| | **Moderate** | DDoS | 0.9763 | 0.9491 | 0.8931 | 0.8961 | 0.0085 | 0.0009 |
| | | Ransomware | 0.9729 | 0.9475 | 0.8891 | 0.8933 | 0.0133 | 0.0027 |
| | **Severe** | DDoS | 0.9663 | 0.8992 | 0.7938 | 0.7966 | 0.0092 | 0.0010 |
| | | Ransomware | 0.9630 | 0.8977 | 0.7903 | 0.7940 | 0.0145 | 0.0029 |
| **ConvNet-SVM** | **Mild** | DDoS | 0.9875 | 0.9792 | 0.9456 | 0.9473 | 0.0048 | 0.0008 |
| | | Ransomware | 0.9844 | 0.9769 | 0.9419 | 0.9443 | 0.0089 | 0.0033 |
| | **Moderate** | DDoS | 0.9776 | 0.9492 | 0.8959 | 0.8975 | 0.0051 | 0.0008 |
| | | Ransomware | 0.9744 | 0.9470 | 0.8924 | 0.8946 | 0.0094 | 0.0034 |
| | **Severe** | DDoS | 0.9676 | 0.8993 | 0.7963 | 0.7978 | 0.0055 | 0.0009 |
| | | Ransomware | 0.9645 | 0.8971 | 0.7932 | 0.7952 | 0.0102 | 0.0035 |
| **DeepSVM-Net** | **Mild** | DDoS | 0.9859 | 0.9770 | 0.9448 | 0.9458 | 0.0058 | 0.0032 |
| | | Ransomware | 0.9832 | 0.9749 | 0.9414 | 0.9433 | 0.0096 | 0.0053 |
| | **Moderate** | DDoS | 0.9760 | 0.9471 | 0.8950 | 0.8960 | 0.0060 | 0.0033 |
| | | Ransomware | 0.9732 | 0.9451 | 0.8918 | 0.8936 | 0.0100 | 0.0055 |
| | **Severe** | DDoS | 0.9660 | 0.8972 | 0.7956 | 0.7965 | 0.0066 | 0.0034 |
| | | Ransomware | 0.9633 | 0.8953 | 0.7927 | 0.7943 | 0.0109 | 0.0057 |

The sensitivity analysis confirms that model performance is significantly affected by training data bias. DeepSVM-Net maintains strong detection capabilities and low error rates across all datasets and bias conditions, validating its suitability for deployment in real-world IoT environments where data imbalance is common. These findings underscore the necessity of employing robust learning architectures and bias mitigation strategies in security-sensitive applications such as IoMT and IIoT networks.

## 3.13 Performance comparison of Proposed methods vs State-of-the-art

**Table 3.33** provides a comparative evaluation of the proposed intrusion detection models against existing state-of-the-art methods using various benchmark datasets, including ECU-IoHT, NF-BoT-IoT, and WUSTL-HDRL-2024. The evaluation metrics like accuracy (AC), positive predictive value (PPV), true positive rate (TPR), F1-score, and ROC-AUC demonstrate the efficiency and robustness of each approach [169]. While previous studies such as MSCSL by Thulasi and Sivamohan [157] achieved 97.90% of accuracy and 96.54% of F1-score and DNN-based IDS by Vishwakarma and Kesswani [162] attained 99.08% of accuracy and 99.02% of F1-score results, which exhibited limitations in balancing recall, precision, and false positives. Similarly, Zhang et al. [163]'s XGBoost model attained 95.01% accuracy on ECU-IoHT, while traditional classifiers like SVM and RF remained in the 92–94% range. However, these models struggled with generalization across multiple datasets. In contrast, the proposed models such as EmbedNet, ConvNet-SVM, and DeepSVM-Net outperform all existing methods with significantly improved accuracy and F1-scores. EmbedNet achieves 99.91% accuracy on WUSTL-HDRL-

2024, 99.46% on NF-BoT-IoT, and 98.99% on ECU-IoHT, while ConvNet-SVM and DeepSVM-Net follow closely, exceeding 99.5% accuracy across multiple datasets. F1-score improvements are equally notable, reaching 99.80%, 99.57%, and 99.61%, respectively, indicating superior detection performance with minimal false positives.

**Table 3.33.** Comparison of Proposed Models with State of the Art.

| References | Proposed Model | Dataset | AC (%) | PPV (%) | TPR (%) | F1 (%) | ROC_AUC (%) |
|---|---|---|---|---|---|---|---|
| Ahmed et al. [145] | Influenced Outlierness (INFLO) | ECU-IoHT | - | - | - | 95 | - |
| Thulasi and Sivamohan [157] | MSCSL | ECU-IoHT | 97.90 | 96.78 | 95.90 | 96.54 | - |
| Zhang et al. [163] | XGB | ECU-IoHT | 95.01 | 94.93 | 95.01 | 94.65 | - |
| | LGB | | 94.82 | 94.78 | 94.82 | 94.41 | - |
| | SVM | | 93.08 | 93.44 | 93.08 | 92.02 | - |
| | RF | | 93.44 | 93.47 | 93.44 | 92.65 | - |
| Sarhan et al. [146] | Extra-Tree | NF-BoT-IoT | 93.82 | 93.70 | - | 97 | - |
| Maimo et al. [164] | OC-SVM | ICE | - | 92.32 | 99.97 | 95.96 | - |
| Kumar et al. [165] | XGB | ToN-IoT | 96.35 | 90.54 | - | 95.03 | - |
| Saba [166] | Bagging | KDDCup99 | 93.2 | 99 | 92 | 96.1 | - |
| Siddiqi and Pak [167] | RF | BoT-IoT | 99.41 | 97.98 | 98.67 | 98.25 | - |
| | SVM | | 72.89 | 59.11 | 80.57 | 64.17 | - |
| | DNN | | 97.27 | 93.89 | 97.37 | 95.49 | - |
| Newaz et al. [168] | RF | Own dataset | 98 | 98 | 98 | 98 | - |
| Gupta et al. [63] | RF | Wustl-EHMS | 94.23 | - | 93.72 | 93.8 | - |
| Grammatikis et al. [170] | CART | Own dataset | 81.73 | - | - | 79.21 | - |
| Nguyen and Kashef [171] | TS-IDS | NF-BoT-IoT | 93.95 | 65.12 | 97.14 | 96.67 | 97.14 |
| | EGraphSAGE | | 77.82 | 54.10 | 79.33 | 78.22 | 79.33 |
| | XGB | | 93.69 | 63.38 | 96.64 | 95.41 | 96.64 |
| Vishwakarma and Kesswani [162] | DNN based IDS | NF-BoT-IoT | 99.08 | 99.03 | 99.08 | 99.02 | - |
| Alosaimi and Almutairi [172] | Ensemble Classifier | BoT-IoT | 73 | 73 | 74 | 73 | - |
| Zhu et al. [173] | CMTSNN | BoT-IoT | 99.3 | 98.4 | 97.5 | 97.9 | - |
| | 1D-CNN | | 96.2 | 89 | 88.9 | 88.9 | - |
| Fernando et al. [174] | XGB | BoT-IoT | 94.80 | 94.97 | 94.80 | 94.83 | 97.50 |
| | GB | | 90.59 | 91.69 | 90.59 | 90.71 | 96.55 |
| Xu et al. [175] | EE-GCN | NF-BOT-IOT | 83.79 | - | - | - | - |
| Chakraborty et al. [176] | RFE-MLP | ECU-IOHT | 98.43 | 98.54 | 98.56 | 98.46 | - |
| Thakkar and Lohiya [177] | DNN | BoT-IoT | 98.99 | 98.90 | 91.30 | 94.95 | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Altaf et al. [178] | MNN | NF-BoT-IoT | 97.74 | 97.79 | 96.63 | 97.74 | 50 |
| Alzahrani and Asghar [179] | LSTM + CNN | BoT-IoT | 95.73 | 96.64 | 94.15 | 95.52 | - |
| Telikani et al. [180] | CostDeepIoT | BoT-IoT | - | 93.8 | 94.7 | 94 | - |
| | **EmbedNet** | **ECU_IoHT** | **98.99** | **98.78** | **99.94** | **99.18** | **99.36** |
| | | **NF-BoT-IoT** | **99.46** | **99.83** | **99.05** | **99.44** | **99.46** |
| | | **WUSTL-HDRL-2024** | **99.91** | **99.90** | **99.79** | **99.80** | **99.81** |
| | **ConvNet-SVM** | **ECU_IoHT** | **99.44** | **99.10** | **99.96** | **99.47** | **99.48** |
| **Proposed Method, 2024** | | **NF-BoT-IoT** | **99.59** | **99.80** | **99.34** | **99.57** | **99.58** |
| | | **WUSTL-HDRL-2024** | **99.87** | **99.88** | **99.85** | **99.87** | **99.87** |
| | **DeepSVM-Net** | **ECU_IoHT** | **99.90** | **99.28** | **99.99** | **99.61** | **99.62** |
| | | **NF-BoT-IoT** | **99.46** | **99.57** | **99.28** | **99.44** | **99.45** |
| | | **WUSTL-HDRL-2024** | **99.53** | **99.45** | **99.53** | **99.50** | **99.53** |

Compared to prior methods, the proposed models show an average accuracy improvement of 3.5–7.5% and an F1-score enhancement of 4.3–5.9%, demonstrating exceptional reliability and robustness in intrusion detection. These results validate the effectiveness of our approach, ensuring high detection precision and enhanced security for IoT environments. **Figure 3.11** depicts the performance comparison of the proposed model and state-of-the-art approaches.

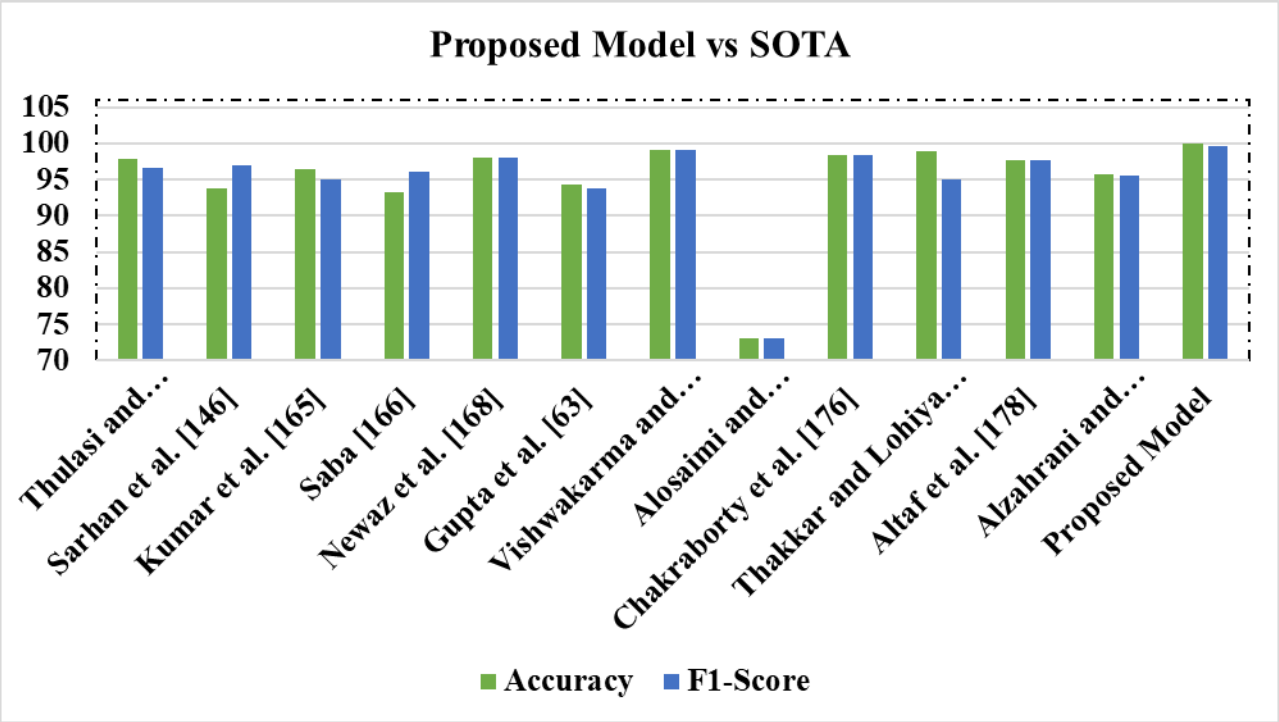



**Fig. 3.11.** Proposed Method vs State of the art

### 3.14 Chapter Summary

43

This chapter presents the development and evaluation of deep learning (DL)-based Intrusion Detection Systems (IDS) for securing Internet of Medical Things (IoMT) environments. Given the resource constraints of IoT devices, traditional security mechanisms often prove inadequate. To address this, three novel models: EmbedNet, ConvNet-SVM, and DeepSVM-Net are proposed, employing advanced neural network architectures to enhance threat detection capabilities. The Experimental validation using benchmark datasets such as ECU-IoHT, NF-BoT-IoT, and Wustl-HDRL-2024 demonstrates the effectiveness of the models, achieving high accuracy rates of 0.9990, 0.9959, and 0.9987, respectively. The models also report exceptionally low false negative rates (FNRs), as low as 0.0001, indicating strong reliability. Ablation studies show minimal training and validation losses, highlighting their learning efficiency and generalization ability. Compared to state-of-the-art methods, the proposed models exhibit a performance improvement of 3.5% to 7.5%, emphasizing their superiority in detecting complex cyber threats. These findings affirm the value of DL-based IDS in enhancing IoT cybersecurity, especially in healthcare settings where data integrity is critical. However, a limitation of this study is the lack of explicit testing for zero-day attacks, which will be addressed in future research to ensure broader threat coverage.

# Chapter 4: Blockchain-Based Model for Securing Healthcare Data

## 4.1 Introduction

The rapid integration of the Internet of Medical Things (IoMT) into modern healthcare systems has transformed the landscape of medical service delivery and patient management. IoMT connects diverse medical devices, sensors, and applications to enable real-time data exchange, continuous monitoring, and remote diagnostics, advancing healthcare outcomes and operational efficiency [181]. However, the proliferation of IoMT has introduced significant security and privacy challenges, particularly due to the sensitive nature of health data and the limited computational capabilities of IoMT devices. As these devices increasingly become targets for cyberattacks, safeguarding medical information and ensuring the reliability of IoMT-enabled systems have become critical imperatives [182].

This chapter presents an overview of the security and privacy issues associated with IoMT ecosystems and outlines the research contributions aimed at addressing these challenges. We investigate the limitations of conventional security mechanisms, such as passwords, digital certificates, and biometric authentication, which are often inadequate in dynamic and heterogeneous IoMT environments [183]. Our research addresses the urgent need for scalable and lightweight security frameworks capable of ensuring data integrity, identity authentication, and secure network access in real-time healthcare applications. Grounded in the broader context of healthcare's digital transformation, this work also examines the intersection of IoMT with cloud computing and Artificial Intelligence (AI), highlighting both the opportunities and vulnerabilities that emerge. In particular, we explore multi-dimensional security analytics as a proactive strategy for detecting and mitigating cyber threats, along with blockchain-based architectures that offer immutable and transparent data management. Furthermore, the chapter introduces encryption-driven solutions, including a Dynamic Adaptive Deep Reinforcement Learning (DA-DRL) framework that enhances AES (Advanced Encryption Standard) encryption by dynamically adjusting key generation in response to real-time threats. Additionally, a multi-layered security architecture integrating AES, SHA-512, Non-Interactive Zero Knowledge Proof (NIZKPs), Practical Byzantine Fault Tolerance (PBFT), and Attribute-Based Access Control (ABAC) is introduced, ensuring robust protection against diverse attack vectors. The InterPlanetary File System (IPFS) is employed for decentralized and immutable data storage, enhancing data security and transparency.

Our proposed research advances the current state of IoMT security by focusing on cross-device data sharing, secure data collection, and holistic threat detection. By leveraging deep learning, cryptographic protocols, and distributed technologies, we aim to develop a comprehensive and adaptive security framework. Ultimately, this work contributes to enhancing trust, reliability, and resilience in IoMT-enabled healthcare systems, ensuring patient data remains protected amidst the rapid evolution of medical technologies.

### 4.1.1 Motivation

The rapid digital transformation of the healthcare industry has significantly enhanced patient care, medical diagnostics, and operational efficiency [184]. However, this progress has also introduced critical security and privacy challenges, particularly in safeguarding sensitive medical data against sophisticated cyber threats [185].

Traditional security measures, including basic encryption techniques and conventional intrusion detection systems, struggle to address the evolving landscape of cyberattacks, unauthorized access, and data breaches in modern healthcare environments [186]. These limitations necessitate advanced, adaptive, and scalable solutions to ensure the confidentiality, integrity, and availability of medical data.

Blockchain technology and deep learning have emerged as promising solutions to overcome these security gaps. Blockchain provides a decentralized, immutable, and transparent framework for secure medical data management, ensuring tamper-proof storage and access control [187]. Its distributed nature eliminates single points of failure, making it highly resistant to data manipulation and unauthorized modifications [188]. Meanwhile, deep learning-powered intrusion detection systems (IDS) offer enhanced accuracy in threat detection by identifying complex attack patterns and anomalies that traditional rule-based security systems fail to recognize.

By integrating blockchain technology with deep learning-based security mechanisms, this research aims to develop a comprehensive, adaptive, and scalable cybersecurity framework for healthcare ecosystems. The proposed approach not only fortifies data security and privacy but also enhances intrusion detection accuracy and system resilience, setting a new standard for secure and intelligent healthcare data management in the Internet of Medical Things (IoMT) era.

### 4.1.2 Major Contribution

This research introduces a comprehensive and adaptive security framework for healthcare environments, addressing critical challenges in encryption, intrusion detection, data protection, secure storage, transaction management, and scalability. The key contributions of this study are as follows:

- ***Blockchain-Enabled Multi-Layered Security Model:*** We developed a multi-layered security framework that integrates AES, SHA-512, Non-Interactive Zero Knowledge Proof (NIZKPs), Practical Byzantine Fault Tolerance (PBFT), and Attribute-Based Access Control (ABAC). This integrated approach ensures end-to-end data confidentiality, integrity, and authentication, outperforming traditional single-layered security architectures used in medical data exchange.
- ***Advanced Cryptographic Security with DA-DRL-AES Encryption:*** We proposed a Dynamic Adaptive Deep Reinforcement Learning (DA-DRL) framework for real-time cryptographic key generation, enhancing the security of AES encryption against evolving cyber threats. Unlike conventional static key management systems, DA-DRL dynamically adapts to security challenges, ensuring resilient and proactive encryption mechanisms in healthcare environments.
- ***Decentralized and Tamper-Resistant Healthcare Data Storage:*** We implemented the InterPlanetary File System (IPFS) for decentralized, immutable, and tamper-proof storage of encrypted healthcare data. This ensures high availability, transparency, and security while eliminating single points of failure commonly found in centralized storage solutions.
- ***Efficient and Scalable Blockchain-Based Transaction Management:*** We integrated blockchain technology with advanced cryptographic techniques to facilitate secure, traceable, and tamper-proof medical transactions. This approach enhances data authenticity and trust, ensuring efficient and immutable medical record exchanges across healthcare networks.

- ***Real-Time Adaptive Security and Scalability for Large-Scale IoMT Networks:*** We designed a scalable security framework optimized for large-scale healthcare ecosystems, ensuring real-time adaptability to changing security threats. The system continuously optimizes key management and cryptographic processes, ensuring efficient and resource-aware security solutions under high data loads.
- ***High-Performance Intrusion Detection with Bi-LSTM-GRU Framework:*** We developed the Secure and Dependable Bi-LSTM GRU Intrusion Detection Framework (S-BiLSTMGRU-IDF), which integrates Bidirectional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Units (GRU) to improve intrusion detection accuracy. The framework achieves 99.94% accuracy in binary classification and 99.89% accuracy in multiclass classification, significantly outperforming traditional intrusion detection models by capturing both forward and backward dependencies for enhanced anomaly detection.

These contributions establish a highly secure, scalable, and adaptive cybersecurity paradigm for IoMT networks, paving the way for next-generation blockchain-driven healthcare security solutions.

## 4.2 Proposed Methodology

This section presents an overview of blockchain technology, including its architecture, data management, and privacy-preserving techniques, as detailed below:

### 4.2.1 Blockchain overview

The blockchain framework proposed is designed to ensure the immutability and transparency of healthcare records while preserving patient privacy. Blockchain technology provides a decentralized ledger that records transactions across a network of computers. This ledger is secure, tamper-proof, and transparent, making it an ideal solution for managing sensitive healthcare data.

### 4.2.2 Blockchain Architecture

The blockchain framework is built on a Hyperledger-based architecture. Hyperledger is an open-source blockchain framework that provides modular and scalable solutions for enterprise-level applications. It supports various consensus mechanisms and smart contract capabilities, ensuring that healthcare data management is secure and efficient. This specific framework within Hyperledger is used due to its permissioned nature, allowing controlled access to the blockchain network. Hyperledger Fabric enables fine-grained access control and supports private channels for confidential transactions, which is crucial for handling sensitive healthcare data.

The proposed model for securing healthcare data has been designed with a robust multi-layered framework incorporating several advanced security techniques to comprehensively address availability, confidentiality, scalability, integrity, privacy, and access control. Initially, robust data protection is provided by the Advanced Encryption Standard (AES), which utilizes symmetric encryption to transform plaintext into ciphertext using a fixed key. Only authorized parties possessing the decryption key can access the sensitive information. Additionally, key generation is enhanced by DA-DRL, which dynamically refines strategies based on real-time feedback and evolving threats, thereby maintaining the resilience of encryption keys against emerging attack vectors. Data integrity is upheld through SHA-512 Hashing, which generates a unique 512-bit hash for each data entry; any alteration in the data results in a different hash value, indicating potential tampering. Furthermore, privacy is

reinforced by Non-Interactive Zero Knowledge Proof (NIZKPs), which allows entities to prove their claims or authorizations without disclosing the underlying sensitive data, thus preserving confidentiality while ensuring compliance with access policies. In the blockchain network managing healthcare data, consensus is facilitated by Practical Byzantine Fault Tolerance (PBFT), which validates transactions and ensures consistency across the blockchain, even in the presence of faulty or malicious nodes. Lastly, granular access control is enforced by Attribute-Based Access Control (ABAC), which evaluates user roles, resource sensitivity, and contextual factors to ensure that only authorized individuals can access specific data based on detailed access policies. This integrated approach provides a comprehensive and effective solution for securing sensitive healthcare information against various security threats. **Figure 4.1** depicts the Proposed Smart Healthcare Framework: Integrating Blockchain and Deep Learning for Enhanced Security and Intelligence which securely storing data in IPFS using advanced techniques in Phase 1 and further decrypting that data to detect intrusion in the environment in Phase 2. The working architecture of the proposed framework is explained in the following subsection.

## 4.3 Secure Data Management and Privacy Preservation

Ensuring the security and confidentiality of healthcare data in IoT-based systems requires effective data management strategies and privacy-preserving mechanisms. This section explores encryption techniques, data integrity measures, cryptographic privacy-preserving methods, and access control mechanisms to enhance data security.

### 4.3.1 Cryptographic Techniques for Data Security

Data encryption and decryption play a fundamental role in protecting sensitive healthcare information within IoT networks. These processes ensure secure data storage and transmission, preventing unauthorized access and preserving data integrity.

**(i) Data Encryption**

Data encryption is a critical security mechanism that encodes sensitive healthcare data to prevent unauthorized access. Within a blockchain-based framework, encryption ensures that only authorized entities can access confidential information, thus safeguarding patient records and medical transactions.

**(ii) Data decryption**

Decryption is the reverse process of encryption, converting encoded data back into its original form. This operation requires a cryptographic key and a decryption algorithm, ensuring that only authorized users can retrieve and interpret the secured healthcare data.

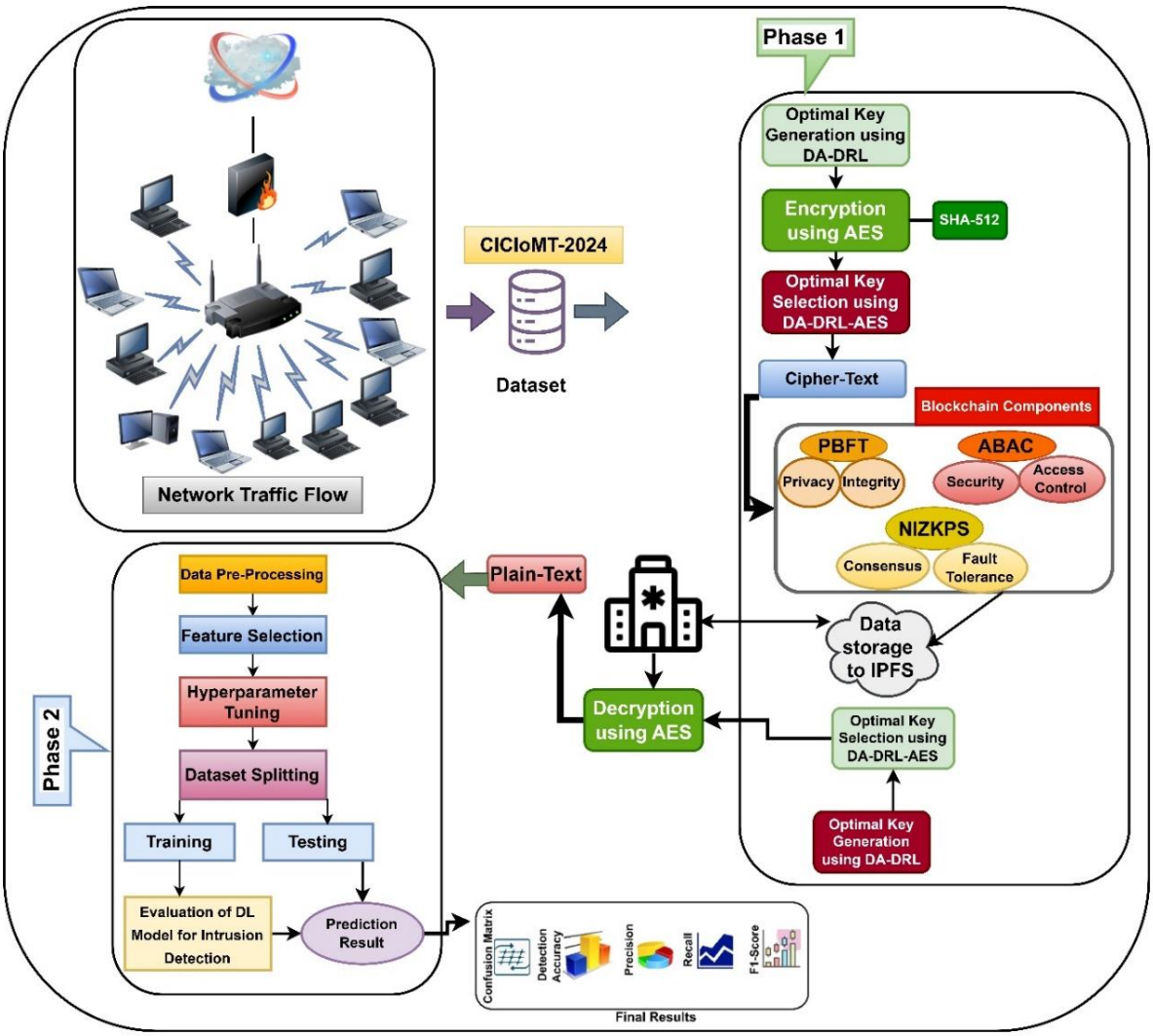



**Fig. 4.1.** Proposed Smart Healthcare Framework: Integrating Blockchain and Deep Learning for Enhanced Security and Intelligence

### 4.3.2 Ensuring Data Integrity and Immutability

Blockchain technology offers a decentralized approach to maintaining data integrity and immutability. Each block within the blockchain is cryptographically linked to the previous one, forming a tamper-resistant ledger. The integrity of a block $B_i$ is secured using the following cryptographic function as represented in ***Eq. (4.1).***

$$H(B_i) = H(H(B_{i-1})||T_i||D_i) \quad (4.1)$$

Where, $H$ represents the cryptographic hash function, $H(B_{i-1})$ is the hash of the previous block, $T_i$ is the timestamp, and $D_i$ is the stored data.

Once a block is added to the chain, altering its content requires recalculating the hashes of all subsequent blocks, making unauthorized modifications computationally infeasible. This property ensures the reliability and authenticity of medical records and transactions.

### 4.3.3 Privacy-Preserving Computational Methods

Advanced cryptographic techniques enable privacy-preserving computations, ensuring that sensitive healthcare data remains protected while allowing secure verification and collaborative analysis.

***(i) Non-Interactive Zero-Knowledge Proofs (NIZKPs)***

NIZKPs allow a prover to demonstrate the validity of a statement to a verifier without revealing the actual data. This concept is represented using ***Eq. (4.2)*** where $S$ is a private statement, and the proof confirms its validity without exposing its content. NIZKPs are instrumental in healthcare applications, allowing secure patient identity verification and credential validation without compromising privacy.

$$Proof = NIZKP(S) \tag{4.2}$$

***(ii) Secure Multi-Party Computation (SMPC)***

SMPC enables multiple parties to jointly compute a function over their private inputs while ensuring that individual data remains confidential. This is mathematically expressed using ***Eq. (4.3)***.

$$F(x_1, x_2, \ldots\ldots, x_n) \; is \; computed \; while \; keeping \; x_1, x_2, \ldots\ldots, x_n \; private \tag{4.3}$$

Where, $F$ represents the computed function, and $x_1, x_2, \ldots\ldots, x_n$ are private inputs from different entities. SMPC is particularly useful in healthcare research and analytics, enabling secure data processing without exposing sensitive patient information.

### 4.3.4 Access Control and Authorization Mechanisms

Effective access control mechanisms ensure that only authorized users can interact with sensitive healthcare data. Attribute-Based Access Control (ABAC) is a dynamic model that grants permission based on attributes of users, resources, and contextual conditions. The access decision is computed using ***Eq. (4.4)***:

$$Access = Decision(A, P, R) \tag{4.4}$$

where $A$ denotes user attributes, $P$ represents resource attributes, and $R$ presents environmental conditions. By integrating encryption techniques, blockchain-based immutability, privacy-preserving cryptographic methods, and robust access control mechanisms, IoT-enabled healthcare systems can ensure secure and confidential data management. This comprehensive security framework fosters a decentralized, tamper-proof environment for efficient and privacy-preserving healthcare operations. **Table 4.1** shows the Security and Privacy Mechanisms in the Proposed Framework.

**Table 4.1.** Security and Privacy Mechanisms in the Proposed Framework

| Property | Component | Function |
|---|---|---|
| Privacy & Integrity | NIZKPS | Hides sensitive data while proving correctness. |
| Consensus & Fault Tolerance | PBFT | Ensures agreement despite Byzantine nodes |
| Security & Access Control | ABAC | Manages resource permissions dynamically. |

## 4.4 Enhanced Model for Securing Healthcare Data Using Advanced Cryptographic Techniques

This section introduces an enhanced security model integrating AES encryption, Dynamic Adaptive Deep Reinforcement Learning (DA-DRL) for key generation, SHA-512 hashing, Zero-Knowledge Proofs, Practical Byzantine Fault Tolerance (PBFT), and Attribute-Based Access Control (ABAC).

### 4.4.1 AES-Based Secure Data Encryption

Advanced Encryption Standard (AES) is employed for secure healthcare data encryption using a symmetric cryptographic approach. The Encryption and decryption are defined using ***Eq. (4.5)*** and ***(4.6)***, where, $AES_K$ denotes the encryption function, $AES_K^{-1}$ represents decryption, $K$ is the symmetric key, and $P$ is the plaintext data, and $C$ is the cipertext.

$$C = AES_K(P) \tag{4.5}$$

$$P = AES_K^{-1}(C) \tag{4.6}$$

### 4.4.2 Dynamic Adaptive Deep Reinforcement Learning (DA-DRL) for Secure Key Generation

Dynamic Adaptive Deep Reinforcement Learning (DA-DRL) is a sophisticated approach to secure key generation that leverages reinforcement learning principles to dynamically optimize cryptographic key management. This technique enhances security by continuously learning from past experiences and adapting the key generation policy based on observed rewards, ensuring resilience against evolving threats. Unlike traditional static key generation methods, DA-DRL employs a deep reinforcement learning model that adjusts cryptographic keys dynamically based on performance feedback. This approach offers several critical benefits. Firstly, continuous optimization allows the model to refine key generation strategies in real time, enhancing security adaptability. Secondly, dynamic adaptation ensures that the key management process remains responsive to the changing nature of cyber threats and evolving data security requirements. Thirdly, enhanced learning through advanced reinforcement learning algorithms improves the efficiency and accuracy of the key generation mechanism. Additionally, DA-DRL excels in handling complexity, as it can effectively process and manage intricate key generation tasks in complex and dynamic environments. The approach also provides security enhancement, ensuring that generated cryptographic keys remain robust against sophisticated attacks. Finally, DA-DRL maintains performance efficiency, striking a balance between security and computational overhead, making it suitable for real-time and large-scale applications, such as blockchain-based healthcare systems. By continuously optimizing the key generation process and adapting to emerging security challenges, DA-DRL significantly strengthens cryptographic security in decentralized environments.

The DRL framework adapts based on the environment and historical data, aiming to generate the most secure key $K_{optimal}$. The reinforcement learning policy $\pi$ is optimized according to the expected rewards R as shown in ***Eq. (4.7)***:

$$\pi^*(s) = arg\max_{\pi} \mathbb{E}[R(s, \pi(s))] \tag{4.7}$$

Where s represents the state of the system, $R(s, \pi(s))$ is the reward function evaluating the effectiveness of policy $\pi$ in generating the key. DA-DRL adapts key generation by analyzing historical security data and adjusting the

policy to produce a more secure key $K_{optimal}$. For example, if a specific key pattern is detected as vulnerable, the DA-DRL algorithm updates the policy to enhance key security, thereby mitigating potential threats. **Algorithm 4.1** presents the working of Dynamic Adaptive Deep Reinforcement Learning (DA-DRL).

**Algorithm 4.1. Dynamic Adaptive Deep Reinforcement Learning (DA-DRL)**

**Steps**

**1 Initialization**

Define state space $S$, action space $A$, and reward function $R$.

Initialize deep reinforcement learning (DRL) model parameters: learning rate $\alpha$, discount factor $\gamma$, and exploration rate $\epsilon$.

Set up experience replay buffer and initialize key generation policy $\pi$.

**2 Key Generation Process**

State Representation: Monitor security parameters and map the system state

Action Selection: Choose action $a$ (key generation strategy) using an exploration-exploitation trade-off.

Key Generation: Execute $a$ to generate key $K$ and store it securely.

Reward Evaluation: Compute $R(s, a)$ based on security effectiveness metrics.

**3 Policy Optimization**

Observe new state $s'$ after key generation.

Update policy π using Q-learning:

$$\pi(s,a) \longleftarrow \pi(s,a) + \alpha[R(s,a) + \gamma \max_{a'} \pi(s',a') - \pi(s,a)]$$

Train DRL model with mini-batch gradient descent on replay buffer samples.

**4 Convergence and Deployment**

Repeat until policy stabilizes for optimal key generation.

Deploy trained policy for real-time adaptive security, integrating continuous learning mechanisms.

### 4.4.3 SHA-512 Hashing for Data Integrity Assurance

SHA-512 (Secure Hash Algorithm 512-bit) ensures data integrity by generating a unique 512-bit hash value for each data input using ***Eq. (4.8)***, where $H$ is the computed hash and $D$ is the input data. Any modification to $D$ produces a different $H$, enabling tamper detection. For example, hashing a healthcare record before storing it on the blockchain ensures any unauthorized modification is identifiable by comparing the stored and recalculated hash values.

$$H = SHA - 512(D) \tag{4.8}$$

### 4.4.4 Non-Interactive Zero-Knowledge Proofs (NIZKPs) for Privacy-Preserving Authentication

Non-Interactive Zero-Knowledge Proofs (NIZKPs) enhance privacy by enabling a prover to convince a verifier of a statement's validity without revealing any underlying sensitive information or requiring multiple rounds of

interaction. Unlike traditional interactive ZKPs, NIZKPs eliminate the need for a back-and-forth exchange between the prover and verifier, making them highly efficient and scalable for blockchain and IoT applications.

For ***Eq. (4.9)*** and ***(4.10)***, given a secret statement $S$, the prover constructs a cryptographic proof $\pi$ that validates the truth of $S$ without disclosing any information about $S$ itself:

$$\pi = NIZKP(S) \tag{4.9}$$

$$Verify(\pi, S) \rightarrow True/False \tag{4.10}$$

Here, $\pi$ is the zero-knowledge proof that certifies the correctness of $S$, and the verification function confirms whether the proof is valid while ensuring no sensitive information is leaked. To achieve non-interactivity, NIZKPs often rely on cryptographic assumptions such as the Fiat-Shamir heuristic, which replaces verifier interactions with a deterministic hash function, or bilinear pairings in elliptic curve cryptography for efficient proof generation and verification.

In a multi-party scenario, Secure Multi-Party Computation (SMPC) can be integrated with NIZKPs to enable collaborative computations while preserving data confidentiality. For a function F computed over private inputs $x_1, x_2, \ldots . x_n$, NIZKPs ensure that each party can validate the correctness of $F(x_1, x_2, \ldots . x_n)$ without revealing individual inputs using ***Eq. (4.11-4.12)***:

$$\pi_F = NIZKP(F(x_1, x_2, \ldots . x_n)) \tag{4.11}$$

$$Verify(\pi_F, F) \rightarrow True/False \tag{4.12}$$

Where $F$ is the function being computed, $x_i$ are the private input contributed by different parties, and $\pi_F$ is the non-interactive proof of correct computation.

NIZKPs significantly enhance security in privacy-preserving blockchain-based intrusion detection systems by ensuring that network anomaly detection results or cryptographic key verifications can be validated without exposing sensitive network data. Their integration with blockchain ensures immutable and publicly verifiable proofs, making them ideal for scalable and decentralized security solutions.

### 4.4.5 Practical Byzantine Fault Tolerance (PBFT) for Blockchain Consensus

Practical Byzantine Fault Tolerance (PBFT) is a consensus mechanism designed to achieve agreement among nodes in a decentralized blockchain network, even in the presence of faulty or malicious actors. PBFT ensures the reliability and integrity of transactions by enabling a distributed set of nodes to collectively validate and agree upon data before it is permanently recorded on the blockchain. The consensus process follows a multi-phase approach consisting of pre-preparation, preparation, and commitment, ensuring that only valid transactions are accepted. The PBFT consensus mechanism can be mathematically expressed using ***Eq. (4.13)***.

$$C_{final} = PBFT(T_x, N, f) \tag{4.13}$$

Where $C_{final}$ represents the final consensus decision, $T_x$ is the transaction to be validated, $N$ denotes the total number of participating nodes, and $f$ is the maximum number of Byzantine (faulty or malicious) nodes that the network can tolerate, which is typically defined as $f < \frac{N}{3}$.

The PBFT process consists of the following phases:

- *Request Phase:* A client submits a transaction $T_x$ to a designated primary node (leader) for validation.
- *Pre-Preparation Phase:* The leader broadcasts a pre-prepare message containing $T_x$ to all replica nodes. Each node verifies the authenticity of the transaction and the leader's legitimacy before proceeding.
- *Preparation Phase:* Upon receiving a valid pre-prepare message, each node broadcasts a prepared message to all other nodes. The transaction is considered valid if a supermajority (at least $2f + 1$ nodes) acknowledge the message.
- *Commitment Phase:* If a node receives at least $2f + 1$ prepare messages, it sends a commit message to the network. Once a node receives $2f + 1$ commit messages, it considers $T_x$ as finalized and adds it to the blockchain.
- *Reply Phase:* After reaching consensus, nodes send confirmation to the client, ensuring the transaction has been successfully recorded.

The PBFT consensus mechanism ensures that all honest nodes reach agreement on transactions, preventing fraud and maintaining blockchain integrity. This process is crucial in healthcare applications, where patient data security and consistency must be maintained across distributed systems. The probability of consensus failure due to malicious nodes is minimized as long as the Byzantine node threshold condition ($f < \frac{N}{3}$) is satisfied. The efficiency of PBFT is optimized for permissioned blockchain environments, making it well-suited for secure, tamper-resistant healthcare data management. **Algorithm 4.2** presents the working of PBFT Consensus Mechanism.

**Algorithm 4.2. PBFT Consensus Mechanism**

| | |
|---|---|
| **Input** | Transaction $T_x$, total nodes $N$, Byzantine nodes $f$ |
| **Output** | Final consensus decision $C_{final}$ |
| **Step 1** | Client Request<br>The client sends a transaction $T_x$ to the primary node (leader). |
| **Step 2** | Pre-Preparation<br>The leader creates a pre-prepare message and broadcasts it to all replica nodes. |
| **Step 3** | Preparation<br>Upon receiving a valid pre-prepare message, each node verifies $T_x$ and sends a prepare message to all other nodes.<br>A node moves to the next stage if it receives at least $2f + 1$ prepare messages. |
| **Step 4** | Commitment<br>If a node collects $2f + 1$ commit messages, it considers $T_x$ as final and adds it to the blockchain. |
| **Step 5** | Reply to Client |

All honest nodes send confirmation to the client, indicating the transaction is successfully committed.

---

By incorporating PBFT, blockchain-based healthcare systems can ensure reliable, secure, and tamper-resistant data transactions, protecting patient records against malicious actors and unauthorized modifications.

#### 4.4.6 Attribute-Based Access Control (ABAC) for Fine-Grained Data Access Management

Attribute-Based Access Control (ABAC) is a flexible and dynamic access control model that enforces security policies based on multiple attributes, including user roles, resource sensitivity, and contextual conditions. Unlike traditional role-based access control (RBAC), ABAC provides fine-grained access control by evaluating a combination of attributes to determine permissions. The access control decision is modeled mathematically as represented in ***Eq. (4.14)***:

$$A_{grant} = f(U_a, R_a, C_a) \tag{4.14}$$

where $A_{grant}$ represents the final access decision (grant or deny), $U_a$ refers to user attributes, $R_a$ represents resource attributes (such as data sensitivity, classification), $C_a$ denotes contextual attributes (like time, location, access device).

A user is permitted access to a resource only if the function $f(U_a, R_a, C_a)$ satisfies predefined security policies. This approach enhances security by ensuring that access is granted strictly based on attribute evaluation rather than predefined roles alone.

In a blockchain-based healthcare system, ABAC ensures that sensitive patient data is accessible only to authorized personnel under specific conditions. For example, a nurse requesting access to a patient's electronic health record (EHR) must have:

- A valid role and department attribute assigned to the same department as the patient.
- Sufficient clearance level to view the requested data.
- Contextual approval, such as within designated working hours or from an approved location.

The access control policy is dynamically evaluated using a policy engine, ensuring that unauthorized personnel, even if they belong to the same organization, cannot access restricted data. This prevents security breaches and unauthorized modifications to sensitive records. The proposed blockchain-based healthcare security model integrates ABAC with multiple cryptographic and privacy-preserving mechanisms to enhance data protection:

- AES Encryption: Ensures confidentiality of stored and transmitted healthcare data.
- Dynamic Adaptive Deep Reinforcement Learning (DA-DRL): Optimizes cryptographic key generation dynamically for enhanced security.
- SHA-512 Hashing: Maintains data integrity by generating a secure hash for each transaction.
- Non-Interactive Zero-Knowledge Proofs (NIZKPs): Enables privacy-preserving verification of access rights without revealing sensitive information.

- Practical Byzantine Fault Tolerance (PBFT): Achieves secure, tamper-proof consensus for blockchain transactions.
- ABAC for Fine-Grained Access Control: Restricts access based on dynamic policy evaluation.

105 By incorporating ABAC, the framework ensures that healthcare data is robustly protected against unauthorized access and tampering. The combination of blockchain, cryptographic security, and intelligent access control mechanisms supports the integrity, confidentiality, and availability of sensitive healthcare information, making it well-suited for real-world healthcare applications.

### 4.4.7 Blockchain-Based Transaction Validation and Block Creation

In a blockchain-based healthcare system, secure transaction validation and block creation are critical to ensuring the integrity, confidentiality, and authenticity of healthcare records. When a new transaction, such as a patient record update, is introduced into the blockchain network, it undergoes a structured validation and block creation process before being securely stored. The process begins with the transaction proposal, where a healthcare provider submits a transaction to the blockchain network. This transaction contains data that needs to be securely recorded, such as patient information or treatment records. The proposed transaction is then broadcast to the network for validation by other nodes, often called peers. The next step is transaction validation. During this phase, the network nodes review the transaction to ensure it meets all necessary criteria and policies. This involves 5 checking the transaction's format, ensuring it conforms to the network's protocols, and verifying that the sender has the necessary permissions and authority to make the transaction. If the transaction passes these checks, it is considered valid and is ready to be included in a block. Once validated, the transaction moves to the block proposal stage. In this stage, a designated leader node, such as the primary node in the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm, gathers the validated transactions and proposes a new block. This block contains a list of these validated transactions and additional metadata, such as the previous block's hash and a timestamp. 109 Including the previous block's hash ensures the continuity and integrity of the blockchain, as each block is cryptographically linked to its predecessor. The proposed block then undergoes the consensus process. For instance, in PBFT, the consensus mechanism requires a series of communications and agreements among the 5 nodes to ensure that a majority agree on the block's validity and order of transactions. This process tolerates many faulty or malicious nodes, ensuring the network can reach consensus despite potential disruptions. Upon reaching a consensus, the new block is added to the blockchain. This addition involves appending the block to the existing chain, with each block containing a unique hash generated by hashing the block's contents, including the previous block's hash. This cryptographic hash serves as a digital fingerprint of the block, ensuring its immutability and integrity. The block also includes a timestamp to record the exact time of its creation and the list of validated transactions to maintain a transparent and verifiable record of all activities within the network.

In sensitive healthcare data, encryption is essential to ensure that the data remains confidential and secure. Before any healthcare data is included in a transaction and stored on the blockchain, it undergoes encryption using the 61 Advanced Encryption Standard (AES). AES is a symmetric encryption algorithm widely recognized for its security and efficiency. The process involves converting plaintext data into ciphertext. This format is unreadable without the appropriate decryption key, mathematically calculated by ***Eq. (4.5)***. This ensures that even if the encrypted data is stored on a publicly accessible blockchain, it cannot be deciphered without the decryption key.

This is crucial in healthcare, where patient data must be protected against unauthorized access. To decrypt the data, the process is reversed, converting the ciphertext back into its original plaintext form using the decryption key, which is calculated using ***Eq. (4.6)***. This encryption and decryption mechanism ensures that only authorized parties with the correct decryption key can access the sensitive healthcare data, thus maintaining confidentiality and data integrity.

### 4.4.8 Digital Signature Mechanism for Authentication and Integrity Verification

The digital signature mechanism is a fundamental security measure in blockchain networks, ensuring both authentication and integrity of transactions. In a blockchain-based healthcare system, digital signatures prevent unauthorized modifications and confirm the legitimacy of transactions. This mechanism involves two key processes: signature generation and signature verification, which are explained as follows:

**(i) Signature Generation**

When a healthcare provider initiates a transaction, they must sign it using their private key to prove their identity and ensure the transaction's authenticity. The signature is generated using a cryptographic hashing function followed by encryption with the private key. The process is mathematically defined using ***Eq. (4.15)***:

$$Signature = Sign_{private}(H(T)) \tag{4.15}$$

Where, S represents the generated digital signature, $S_{private}$ is the signing function using the provider's private key, $H(T)$ denotes the hash of the transaction $T$, which is computed as ***Eq. (4.16)***.

$$H(T) = Hash(T) \tag{4.16}$$

Here, $Hash(\cdot)$ is a cryptographic hash function (such as SHA-512) that converts the transaction $T$ into a fixed-length hash value. The use of a hash function ensures that even the smallest modification in $T$ results in a completely different hash, maintaining data integrity.

The private key ensures that only the authorized sender can generate the signature, preventing impersonation attacks. The signed transaction is then broadcast to the blockchain network for validation.

**(ii) Signature Verification**

Upon receiving a signed transaction, other nodes in the blockchain network must verify the digital signature before accepting it into the ledger. The verification process involves decrypting the signature using the sender's public key and comparing the result with the independently computed hash of the transaction. This verification is expressed using ***Eq. (4.17)***.

$$V_{public}(H(T), S) = \begin{cases} valid, & if\ H(T) = D_{public}(S) \\ Invalid, & otherwise \end{cases} \tag{4.17}$$

Where, $V_{public}$ is the verification function using the sender's public key, $D_{public}(S)$ represents the decryption of the digital signature $S$ using the public key, $H(T)$ is the independently computed hash of the transaction $T$. If $H(T)$ matches the decrypted signature, the transaction is considered authentic and unaltered; otherwise, it is rejected.

This verification process ensures that:

- The transaction originated from the legitimate sender, since only the sender possesses the corresponding private key.
- The transaction has not been modified, as even a minor change would alter the hash value, causing verification failure.

The digital signature mechanism is crucial for protecting sensitive healthcare transactions, ensuring that only authorized healthcare providers can submit and verify transactions. This approach prevents malicious activities such as data tampering, unauthorized modifications, and impersonation attacks. By integrating cryptographic hashing, private-key-based signing, and public-key-based verification, the system guarantees:

- Data integrity: Ensuring that transactions remain unaltered.
- Authentication: Verifying the legitimacy of transaction senders.
- Non-repudiation: Preventing senders from denying their transactions.

By leveraging robust cryptographic techniques, the proposed blockchain-based healthcare framework provides a secure and trustable mechanism for managing healthcare transactions while safeguarding patient data confidentiality. **Algorithm 4.3** presents the step-by-step process of digital signature generation and verification.

**Algorithm 4.3. Secure Digital Signature Mechanism for Healthcare Data Transactions**

**Steps**

**1** Initialization and Key Management

Input:

Healthcare transaction $T$

Private key $K_{private}$ of the healthcare provider

Public key $K_{public}$ of the healthcare provider

Output:

Digital signature $S$ for transaction $T$

Verification status (Valid/Invalid)

Establish a cryptographic key infrastructure (CKI) for secure key distribution.

Retrieve the provider's private key $K_{private}$ for signing the transaction.

Ensure secure storage and access control of cryptographic keys.

**2** Digital Signature Generation

Input: Transaction $T$

Output: Digital signature $S$

i) Data Preprocessing:

Remove redundant metadata and standardize format.

ii) Hashing:

Compute a cryptographic hash of $T$ using SHA-512.

$$H(T) = SHA - 512(T)$$

iii) Encryption & Signature Creation:

Encrypt the hash using the private key $K_{private}$

$$S = Sign_{K_{private}}(H(T))$$

iv) Attach Digital Signature:

Append $S$ to the transaction $T$.

**3** Digital Signature Transmission

Input: Signed transaction $(T, S)$

Output: Secure transmission of signed data

Encrypt the signed transaction using AES-256 for confidentiality.

Transmit the encrypted package over a secure blockchain channel.

Ensure non-repudiation by logging the transaction timestamp in the blockchain ledger.

**4** Digital Signature Verification

Input: Signed transaction $(T, S)$

Output: Verification status (Valid/Invalid)

(i) Data Extraction:

Retrieve $T$ and $S$ from the received transaction.

(ii) Recompute Hash:

Generate a new hash of the received transaction $T'$:

$$H'(T) = SHA - 512(T')$$

(iii) Signature Validation:

Verify the digital signature using the public key $K_{public}$:

$$V = Verify_{K_{public}}(H'(T), S)$$

(iv) Decision Making:

If $V$ is valid, confirm transaction integrity and authenticity.

If $V$ is invalid, flag potential tampering or unauthorized modification.

**5** Security Enforcement and Audit Logging

(i) If verification fails:

Alert the system administrator.

Log the failed verification attempt in an immutable audit trail.

(ii) If verification succeeds:

Approve transaction execution.

Store the transaction hash in the blockchain for future reference.

---

### 4.4.9 Storing data to IPFS

110

The InterPlanetary File System (IPFS) is a decentralized, peer-to-peer storage system that enhances data security, integrity, and availability. By leveraging content-addressing, IPFS assigns a unique cryptographic identifier to each file, enabling efficient retrieval and ensuring tamper resistance. This section details the advantages and the step-by-step process of securely storing encrypted healthcare data on IPFS.

**(i) Decentralization**

IPFS distributes data across multiple nodes in a peer-to-peer network, mitigating risks associated with centralized storage, such as single points of failure, data breaches, and downtime. The probability of data loss due to server failures, cyberattacks, or natural disasters is significantly reduced. The data redundancy across $n$ nodes is expressed using ***Eq. (4.18)***, where $D_{availability}$ represents the probability that data remains accessible, $P_i$ denotes the probability that node is operational, and $n$ is the total number of participating nodes in the IPFS network. Since multiple nodes store and serve the data, even if some nodes fail, the data remains retrievable, ensuring high availability with a critical feature for time-sensitive healthcare applications.

$$D_{availability} = 1 - \prod_{i=1}^{n}(1 - P_i) \quad (4.18)$$

**(ii) Immutability and Content Addressing**

IPFS employs content-based addressing rather than traditional location-based addressing. Each stored data block is assigned a unique Content Identifier (CID) derived from its cryptographic hash. The CID is generated using ***Eq. (4.19)***:

$$CID = H(D) \quad (4.19)$$

Where $H(D)$ represents a cryptographic hash function (such as SHA-256 or SHA-512) applied to the data $D$. The resulting CID remains immutable, if any modification occurs in D, its CID will change, ensuring data integrity and tamper resistance. This characteristic aligns with blockchain principles, preventing unauthorized alterations and ensuring that stored patient records remain authentic and reliable.

**(iii) Security via encryption**

To enhance data privacy, healthcare data is encrypted before storage on IPFS. The encryption process ensures that even if an unauthorized entity accesses the stored data, it remains unreadable. Using Advanced Encryption Standard (AES-256), the encryption process is defined as:

Once encrypted, the data is uploaded to IPFS, which generates a corresponding CID using ***Eq. (4.20)***. This CID is then stored on the blockchain, ensuring an immutable reference to the encrypted data.

$$CID_C = H(C) \quad (4.20)$$

***a) Secure Storage Process on IPFS***

The workflow for securely storing healthcare data on IPFS consists of three main steps:

- *Encrypt the Data:* The original healthcare record D is encrypted using AES-256, producing ciphertext C as shown in ***Eq. (4.21)***.

$$C = E_{AES}(D, K) \quad (4.21)$$

- *Upload Encrypted Data to IPFS:* The ciphertext C is uploaded to IPFS, generating a unique CID as shown in ***Eq. (4.22)***.

$$CID_C = H(C) \quad (4.22)$$

- *Store the CID on Blockchain:* The CID is stored in the blockchain transaction, ensuring off-chain storage of actual data while leveraging blockchain's immutability and transparency as shown in ***Eq. (4.23)***:

$$T_{blockchain} = Tx(CID_C, metadata) \quad (4.23)$$

Where, $T_{blockchain}$ represents the blockchain transaction, and Metadata includes essential details such as timestamps and access control policies.

***b) Retrieval and Decryption Process***

To retrieve and access healthcare data, the following steps are performed:

- *Retrieve CID from Blockchain:* The CID is retrieved from the blockchain transaction as shown in ***Eq. (4.24)***.

$$CID_C = ExtractCID(T_{blockchain}) \quad (4.24)$$

- *Fetch Encrypted Data from IPFS:* Using CID, the encrypted data is retrieved from the IPFS network as shown in ***Eq. (4.25)***.

$$C = IPFS(CID_C) \quad (4.25)$$

- *Decrypt the Data:* The ciphertext $C$ is decrypted using the corresponding AES decryption function as shown in ***Eq. (4.26)***.

$$D = D_{AES}(C, K) \quad (4.26)$$

Where, $D_{AES}$ is the AES decryption function, and $K$ is the decryption key (only accessible to authorized users). Only authorized healthcare providers with the correct key $K$ can successfully decrypt the data, ensuring confidentiality and compliance with regulations like HIPAA and GDPR.

The proposed approach integrates IPFS and blockchain to enhance healthcare data security, availability, and integrity. IPFS decentralizes storage, reducing breach risks and ensuring redundancy, while blockchain maintains an immutable record of encrypted data references. Content-based addressing in IPFS preserves data integrity, and blockchain ensures verifiable transaction history. This framework strengthens data security, enables authorized access, and improves patient outcomes by providing a scalable, tamper-proof healthcare data management system. **Algorithm 4.4** depicts the working of the Proposed Blockchain framework.

**Algorithm 4.4. Secure Blockchain Framework for Healthcare Data Protection**

**Steps**

**1** Data Encryption and Storage on IPFS

Input: Healthcare data $D$, AES encryption key $K$

Output: Encrypted data $C$, Content Identifier $CID$

(i) Encrypt D using AES

$C = E_{AES}(D, K)$

(ii) Upload C to IPFS and generate a unique CID

$CID = H(C)$

(iii) Store $CID$ on the blockchain to establish an immutable reference.

**2** Adaptive Key Generation Using DA-DRL

Input: State $s$, Set of policies $\Pi$

Output: Optimized encryption key $K_{opt}$

(i) Initialize:

$K_{opt} = \emptyset$, max $reward = -\infty$

(ii) For Each Policy $\pi$ in $\Pi$

Compute security reward

$R(s, \pi) = EvaluateReward(s, \pi)$

If $R(s, \pi) >$ max $\_reward$, update

max $\_reward = R(s, \pi)$

$K_{opt} = ApplyPolicy(\pi)$

(iii) Adapt policy $\pi$ using historical security data to optimize key generation.

**3** Integrity Verification Using SHA-512

Input: Healthcare record $D$

Output: Hash value $H$

(i) Compute SHA-512 hash

$H = H_{SHA-512}(D)$

(ii) Store $H$ on the blockchain for integrity verification.

(iii) During retrieval, recompute $H'$ and verify:

$H' \stackrel{?}{=} H$

If equal, data integrity is confirmed.

Otherwise, data has been tampered with.

**4** Privacy-Preserving Authentication Using NIZKPs

Input: Public statement $S$, Private witness $W$, Common reference string $CRS$

Output: Non-Interactive Zero-Knowledge Proof $\pi$

(i) Generate proof $\pi$ without interaction

$\pi = NIZKP(S, W, CRS)$

(ii) Publish $\pi$ and $S$ on the blockchain for verification.

(iii) Verify proof using a verification function:

$$V(S, \pi, CRS) = \begin{cases} valid, & if\ proof\ is\ correct \\ Invalid, & otherwise \end{cases}$$

(iv) Grant authentication only if $V(S, \pi, CRS) = Valid$.

**5** Blockchain Consensus Using PBFT

Input: Set of transactions $T$

Output: Verified block $B$

(i) Pre-Prepare Phase: Leader node broadcasts transaction $T_i$

(ii) Prepare Phase: Nodes validate $T_i$ and send prepare messages

(iii) Commit Phase: If a supermajority $(2f + 1)$ confirms $T_i$, commit it to the ledger.

(iv) Block Finalization:

$B = \{T_1, T_2, \ldots T_n)$

Store $B$ on the blockchain after consensus.

**6** Fine-Grained Access Control Using ABAC

Input: User attributes $U$, Resource attributes $R$, Contextual attributes $C$

Output: Access decision $A$

(i) Evaluate access policy

$A = Decision(U, R, C)$

(ii) If A=True, grant access; otherwise, deny access.

---

### 4.5 Proposed model for Intrusion detection system (Phase 2)

In modern Internet of Things (IoT) environments, ensuring security and reliability in network communications is crucial due to the increasing number of cyber threats. To address this, we propose the Secure and Dependable Bi-LSTM-GRU Intrusion Detection Framework (S-BiLSTMGRU-IDF), which integrates Bidirectional Long Short-Term Memory (Bi-LSTM) and Bidirectional Gated Recurrent Units (Bi-GRU) for enhanced sequential data analysis. The hybrid model leverages the strengths of both architectures to capture long-term dependencies and contextual relationships in network traffic data. Furthermore, dropout regularization and fully connected layers are incorporated to mitigate overfitting and optimize classification performance. The proposed framework is designed to process large-scale IoT network logs and detect anomalies with high accuracy and efficiency.

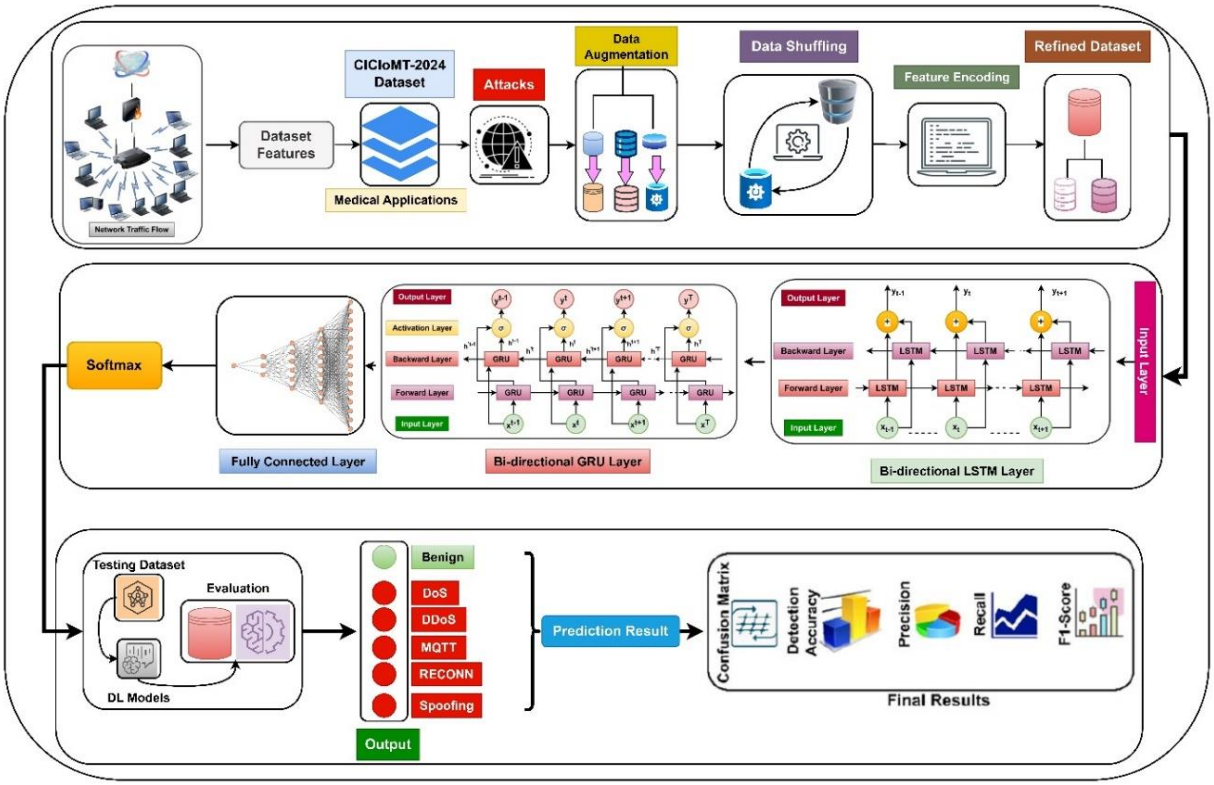



**Fig. 4.2.** Architecture of the Secure and Dependable Bi-LSTM GRU Intrusion Detection Framework (S-BiLSTMGRU-IDF)

### 4.5.1 Data preprocessing

Data preprocessing is a critical step in preparing raw network traffic data for deep learning models. This phase ensures that the input data is clean, consistent, and suitable for training. The preprocessing pipeline includes data augmentation, data shuffling, feature encoding, and normalization. These steps enhance the model's ability to generalize across diverse attack patterns and ensure uniformity in feature representation, which is essential for accurate intrusion detection in IoMT environment as depicted in **Figure 4.2**. **Figure 4.3** represents the Structural Overview of the Proposed S-BiLSTMGRU-IDF model for Binary and Multiclass Classification.

**(i) Data augmentation**

Data augmentation introduces controlled variations in the dataset to simulate real-world noise and improve model robustness. For a given a numerical feature $v_i$, the augmented feature $v_i'$ is generated using ***Eq. (4.27)***:

$$v_i' = v_i + \mathcal{N}(0, \delta^2) \tag{4.27}$$

Where, $\mathcal{N}(0, \delta^2)$ represents a normal distribution with a mean of zero and variance $\delta^2$. The addition of controlled noise helps the model learn invariant representations, reducing its sensitivity to minor fluctuations in network traffic data. This step is particularly important in IoT environments, where network traffic can exhibit significant variability due to the diverse nature of connected devices.

**(ii) Data Shuffling**

Data shuffling is performed to prevent the model from learning order-dependent patterns, which leads to overfitting. The dataset $D$ undergoes a random permutation before being split into training and testing sets. The shuffling process can be represented using ***Eq. (4.28)***:

$$D' = \prod(D) \tag{4.28}$$

Where $\prod(\cdot)$ denotes a permutation function that randomly rearranges the dataset indices. By ensuring that the training and testing sets are diverse, this step enhances the model's ability to generalize to unseen data.

**(iii) Feature Encoding**

Feature encoding is essential for transforming categorical attributes into numerical representations that can be processed by the model. Continuous features are standardized using z-score normalization, which scales the data to have a mean of zero and a standard deviation of one. The transformation is given by ***Eq. (4.29)***. ***Eq. (4.30)*** indicates the derivative of this transformation.

$$v_{scaled} = \frac{v-\mu_v}{\sigma_v} \tag{4.29}$$

$$\frac{\partial v_{scaled}}{\partial v} = \frac{1}{\sigma_v} \tag{4.30}$$

Here, $\mu_v$ and $\sigma_v$ represent the mean and standard deviation of the feature, respectively.

The standardized feature $v_{scaled}$ ensures that each feature contributes equally to the model's learning process. Categorical attributes are encoded using binary encoding, where each category $\gamma$ is mapped to a binary vector representation, which is represented using ***Eq. (4.31)***. The final feature matrix, which incorporates all transformed attributes, is represented using ***Eq. (4.32)***:

$$BinaryEncode(\gamma) = [0, \ldots..1, \ldots\ldots.0] \tag{4.31}$$

$$V \in \mathbb{R}^{m \times d} \tag{4.32}$$

Where $m$ is the number of samples, and $d$ is the dimensionality of the feature space. This matrix serves as the input to the deep learning model.

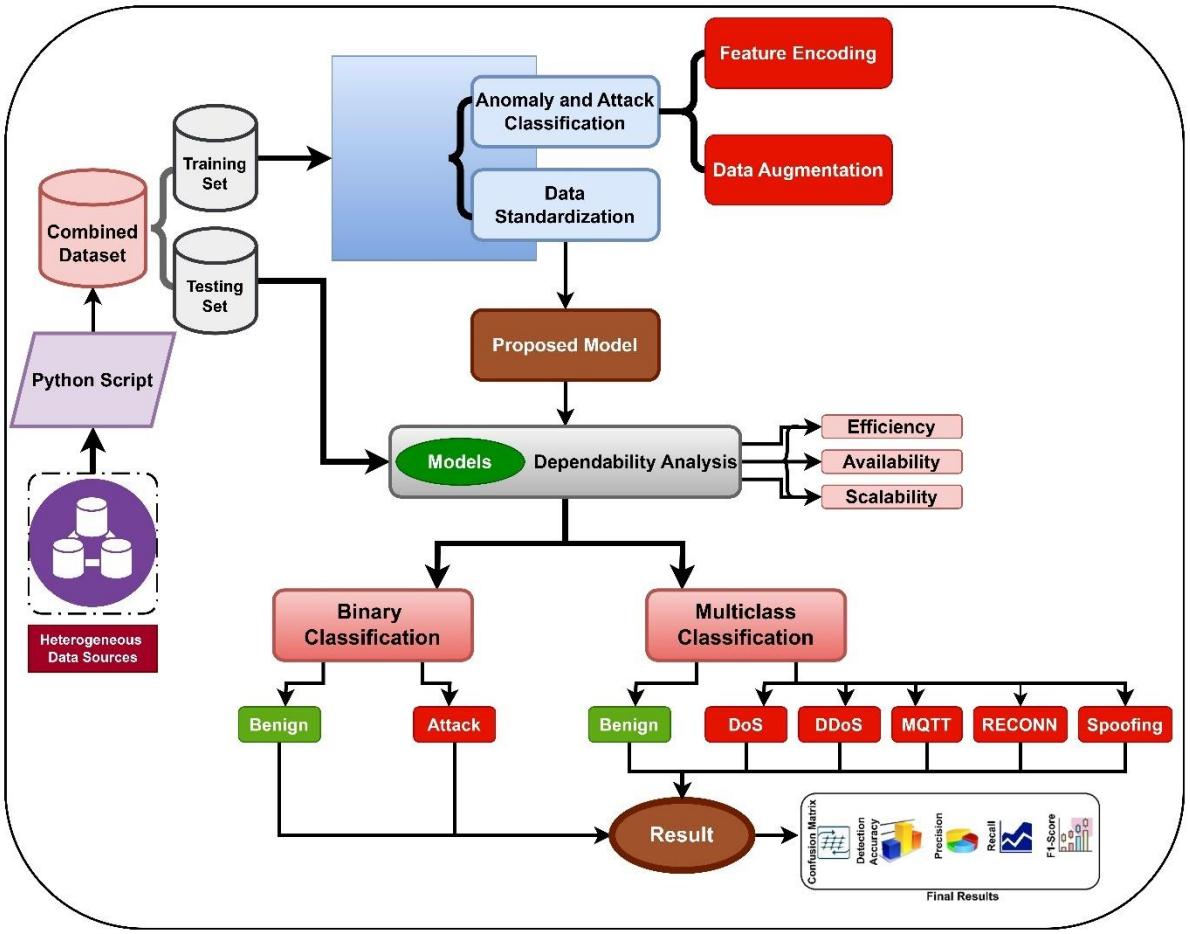



**Fig. 4.3.** Structural Overview of the Proposed S-BiLSTMGRU-IDF for Binary and Multiclass Classification

### 4.5.2 Deep learning architecture

The S-BiLSTMGRU-IDF model employs a multi-layered deep learning architecture designed to process sequential network traffic data and identify anomalous behavior. The architecture consists of Bidirectional LSTM (Bi-LSTM) layers, Bidirectional GRU (Bi-GRU) layers, dropout layers, and fully connected layers, followed by a softmax classification layer. Each component of the architecture is carefully designed to enhance the model's ability to capture complex patterns in the data.

**(i) Bidirectional LSTM layer**

The first layer of the model is a Bidirectional LSTM (Bi-LSTM), which processes input sequences in both forward and backward directions. This allows the model to capture long-range dependencies and contextual relationships in the data. Given an input feature sequence $V$, the Bi-LSTM computes the forward hidden states and backward hidden states at each timestep $t$, which is represented using ***Eq. (4.33-4.34)***:

$$\overrightarrow{s_t} = \mathcal{L}(V_t, \overrightarrow{s_{t-1}}) \quad (4.33)$$

$$\overleftarrow{s_t} = \mathcal{L}(V_t, \overleftarrow{s_{t-1}}) \quad (4.34)$$

Here, $\mathcal{L}(\cdot)$ represents the LSTM cell function. The final hidden representation at time $t$ is obtained by concatenating the forward and backward hidden states. The Bi-LSTM processes consist of forward and backward LSTM, which are concatenated using ***Eq. (4.35)***

$$S_t = [\overrightarrow{s_t} \oplus \overleftarrow{s_t}] \tag{4.35}$$

Where $\oplus$ denotes concatenation. This bidirectional approach enables the model to capture dependencies in both past and future contexts, enhancing its ability to detect anomalies. The LSTM cell computations for each direction are expressed using ***Eq. (4.25-4.30)***:

The LSTM cell consists of three gates: the input gate ($i_t$), the forget gate ($f_t$), and the output gate ($o_t$). These gates regulate the flow of information into and out of the cell state ($c_t$). The computations for each gate and the cell state are represented using ***Eq. (4.36-4.41)***.

$$i_t = \sigma(W_i V_t + U_i s_{t-1} + b_i) \tag{4.36}$$

Here, $W_i$, $U_i$ , and $b_i$ are the input gate weights and bias, and $\sigma(\cdot)$ is the sigmoid activation function.

$$f_t = \sigma\left(W_f V_t + U_f s_{t-1} + b_f\right) \tag{4.37}$$

Here, $W_f$, $U_f$ , and $b_f$ are the forget gate weights and bias.

$$\tilde{c}_t = tanh(W_c V_t + U_c s_{t-1} + b_c) \tag{4.38}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4.39}$$

Here, $\tilde{c}_t$ is the candidate cell state, and $\odot$ denotes element-wise multiplication.

$$o_t = \sigma(W_o V_t + U_o s_{t-1} + b_o) \tag{4.40}$$

Here, $W_o$, $U_o$ , and $b_o$ are the output gate weights and bias.

$$s_t = o_t \odot \tanh(c_t) \tag{4.41}$$

Here, the hidden state $s_t$ is the output of the LSTM cell at timestep $t$.

***Eq. (4.42)*** computes the gradients for LSTM parameters using the chain rule to update the parameters using optimization algorithms.

$$\frac{\partial \mathcal{L}}{\partial W_f} = \frac{\partial \mathcal{L}}{\partial s_t} \cdot \frac{\partial s_t}{\partial W_f} \; and\; similarly\; for\; other\; parameters \tag{4.42}$$

**(ii) Bidirectional GRU layer**

To improve computational efficiency, the output of the Bi-LSTM layer is processed by a Bidirectional GRU (Bi-GRU) layer. The GRU architecture is similar to LSTM but uses fewer parameters, making it faster to train. The GRU cell operations are expressed using ***Eq. (4.43-4.46),*** where $z_t$ is the update gate, $r_t$ is the reset gate, and $\odot$ represents element-wise multiplication. The GRU layer enhances the model's ability to capture temporal dependencies while reducing computational overhead.

$$z_t = \sigma(W_z V_t + U_z s_{t-1}) \quad (4.43)$$

$$r_t = \sigma(W_r V_t + U_r s_{t-1}) \quad (4.44)$$

$$\tilde{s}_t = tanh(W_s V_t + U_s(r_t \odot s_{t-1})) \quad (4.45)$$

$$s_t = (1 - z_t) \odot s_{t-1} + z_t \odot \tilde{s}_t \quad (4.46)$$

$$\vec{s}_t = (1 - \vec{z}_t) \odot \vec{s}_{t-1} + \vec{z}_t \odot \tanh\,(W_h V_t + U_h(\vec{r}_t \odot \vec{s}_{t-1})) \quad (4.47)$$

$$\overleftarrow{s}_t = (1 - \overleftarrow{z}_t) \odot \overleftarrow{s}_{t+1} + \overleftarrow{z}_t \odot \tanh\,(W_h V_t + U_h(\overleftarrow{r}_t \odot \overleftarrow{s}_{t+1})) \quad (4.48)$$

The Bi-GRU consists of forward and backward GRUs as represented by ***Eq. (4.47-4.48)***, which are concatenated using ***Eq. (4.49)***:

$$S_t = [\vec{s}_t \oplus \overleftarrow{s}_t] \quad (4.49)$$

With its update and reset gates, the GRU mechanism efficiently captures long-term dependencies while mitigating the vanishing gradient problem. By incorporating both forward and backward contexts, the Bi-GRU layer enhances the model's ability to understand and predict complex temporal dynamics in the data.

Furthermore, to prevent overfitting, a dropout layer is applied to the hidden states. Dropout randomly deactivates a fraction of neurons during training, forcing the model to learn more robust features. The dropout operation can be represented using ***Eq. (4.50)***, where $p$ is the dropout probability. This step enhances the model's generalization ability by reducing its reliance on specific neurons.

$$S_t^{dropout1} = Dropout(S_t, p = 0.5) \quad (4.50)$$

After that, the second Bidirectional LSTM layer is used to process the data further to capture additional context from the sequence as represented using ***Eq. (4.51-4.53)***. It enhances the ability of the model to understand the sequence data by integrating additional layers of contextual understanding.

$$\vec{s}_t = LSTM(S_t^{dropout1}, \vec{s}_{t-1}) \quad (4.51)$$

$$\overleftarrow{s}_t = LSTM(S_t^{dropout1}, \overleftarrow{s}_{t+1}) \quad (4.52)$$

$$S_t = [\vec{s}_t \oplus \overleftarrow{s}_t] \quad (4.53)$$

Then, the second Bidirectional GRU layer processes the output from the second Bi-LSTM layer, focusing on higher-level dependencies. It adds a final layer of temporal context and dependency modelling, improving the overall representation of the sequence as expressed using ***Eq. (4.54-4.60)***:

$$\vec{z}_t = \sigma(W_z S_t + U_z \vec{s}_{t-1}) \quad (4.54)$$

$$\overleftarrow{z}_t = \sigma(W_z S_t + U_z \overleftarrow{s}_{t+1}) \quad (4.55)$$

$$\vec{r}_t = \sigma(W_r S_t + U_r \vec{s}_{t-1}) \quad (4.56)$$

$$\overleftarrow{r}_t = \sigma(W_r S_t + U_r \overleftarrow{s}_{t+1}) \quad (4.57)$$

$$\vec{s}_t = (1 - \vec{z}_t) \odot \vec{s}_{t-1} + \vec{z}_t \odot \tanh\left(W_h S_t + U_h(\vec{r}_t \odot \vec{s}_{t-1})\right) \quad (4.58)$$

$$\overleftarrow{s}_t = (1 - \overleftarrow{z}_t) \odot \overleftarrow{s}_{t+1} + \overleftarrow{z}_t \odot \tanh\left(W_h S_t + U_h(\overleftarrow{r}_t \odot \overleftarrow{s}_{t+1})\right) \quad (4.59)$$

$$S_t = [\vec{s}_t \oplus \overleftarrow{s}_t] \quad (4.60)$$

A dropout layer is applied to the output of this layer, which reduces overfitting by ensuring that the network does not rely too heavily on any single pathway, which is represented using ***Eq. (4.61)***:

$$S_t^{dropout2} = Dropout(S_t, p = 0.5) \quad (4.61)$$

101 The fully connected layer maps the output from the second dropout layer to a new feature space of size $h\_1$. It transforms the feature representation to prepare for classification by integrating higher-level features. ***Eq. (4.62-4.63)*** of the fully connected layer parameters is used for parameter updates.

$$FC1_{out} = W_{fc1} S_t^{dropout2} + b_{fc1} \quad (4.62)$$

$$\frac{\partial \mathcal{L}}{\partial W_{fc1}} = \frac{\partial \mathcal{L}}{\partial FC1_{out}} \cdot \frac{\partial FC1_{out}}{\partial W_{fc1}} \quad (4.63)$$

Similarly, the fully connected layer maps the output from the first FCL to the number of classes, producing each class's final classification scores as represented by ***Eq. (4.64):***

$$FC2_{out} = W_{fc2} FC1_{out} + b_{fc2} \quad (4.64)$$

Finally, the SoftMax Activation function converts the logits from the final fully connected layer into probabilities, 1 which provides probability distribution over classes, facilitating the classification of inputs using ***Eq. (4.65).*** The loss function calculates the difference between predicted and true labels. It guides the training process by quantifying prediction errors and optimizing the model parameters using ***Eq. (4.66)***. Adam optimizer then updates model parameters to minimize the loss function by adjusting model parameters efficiently using adaptive learning rates to improve model accuracy.

$$\hat{y} = Softmax(FC2_{out}) \quad (4.65)$$

$$\frac{\partial \hat{y}_i}{\partial FC2_{out,j}} = \hat{y}_i(\delta_{i,j} - \hat{y}_j) \quad (4.66)$$

Where, $\delta_{i,j}$ is the Kronecker delta.

***Eq. (4.67-4.68)*** provides information on the curvature of the SoftMax output with respect to the logits.

$$\frac{\partial^2 \hat{y}_i}{\partial FC2_{out,j} \partial FC2_{out,k}} = \hat{y}_i(\delta_{jk}\hat{y}_j - \hat{y}_i\hat{y}_j\hat{y}_k) \quad (4.67)$$

$$\hat{y}_i = \frac{\exp(FC2_{out,i})}{\sum_{j=1}^{C} \exp(FC2_{out,j})} \quad (4.68)$$

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \quad (4.69)$$

***Eq. (4.69)*** computes the loss function concerning the predicted probability and indicates how prediction changes affect the loss. ***Eq. (4.70-4.71)*** provides information on how changes in the prediction impact the rate of change of the loss.

$$\frac{\partial \mathcal{L}}{\partial \hat{y}_{i,c}} = -\frac{y_{i,c}}{\hat{y}_{i,c}} \tag{4.70}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \hat{y}_{i,c}^2} = \frac{y_{i,c}}{\hat{y}_{i,c}^2} \tag{4.71}$$

Adam uses first-order gradients computed from the loss function to update model parameters $\theta$. The learning rate $\eta$ controls the size of the update step. It is also utilizing to adjust the learning rate for each parameter adaptively using ***Eq. (4.72)***.

$$\theta = \theta - n \cdot \nabla_\theta \mathcal{L} \tag{4.72}$$

The performance of deep learning models is heavily influenced by the careful selection and tuning of hyperparameters, which directly impact the model's ability to learn and generalize from the data. In this paper, extensive hyperparameter tuning was conducted to optimize the proposed model's architecture for intrusion detection. **Table 4.2** provides a detailed overview of the selected hyperparameters, including the sizes of input, hidden, and output layers, dropout probability, learning rate, number of epochs, batch size, optimizer choices, and the specific configurations for the Bi-LSTM and Bi-GRU layers. These hyperparameters were meticulously adjusted to ensure the model achieves the best possible performance in detecting and classifying network intrusions.

**Table 4.2.** Hyperparameter Configuration of the Proposed Model for Binary and Multiclass Classification

| Hyperparameter | Description | Value |
|---|---|---|
| input_size | Number of input features | 1*75*64 |
| hidden_size1 | The number of features in the hidden state for the first LSTM layer. | 64 |
| hidden_size2 | The number of features in the hidden state for the second LSTM and GRU layers. | 128 |
| output_size | The number of output classes for Multi and Binary classification. | [(1*5)*128], [(1*2)*128] |
| dropout_prob | The probability of an element being zeroed in the dropout layers. | 0.5 |
| learning_rate | The learning rate for the optimizer. | 0.001 |
| num_epochs | The number of epochs for training the model. | 100 |
| batch_size | The number of samples per batch during training. | 64 |
| optimizer | The optimization algorithm used for training. | Adam, Softmax |
| loss_function | The loss function used for training. | CrossEntropyLoss, BinaryCrossEntropy |
| bi_lstm1 hidden_size | The number of features in the hidden state of the first Bi-LSTM layer. | 64 |
| bi_lstm2 hidden_size | The number of features in the hidden state of the second Bi-LSTM layer. | 128 |
| bi_gru1 hidden_size | The number of features in the hidden state of the first Bi-GRUlayer. | 64 |
| bi_gru2 hidden_size | The number of features in the hidden state of the second Bi-GRUlayer. | 128 |
| fc1 input_size | The input size of the first fully connected (FC) (dense) layer. | 256 (2 * hidden_size2) |
| fc1 input_size | The input size of the first fully connected (dense) layer. | 128 |

| | | |
|---|---|---|
| Fc2 input_size | The output size of the second fully connected (dense) layer. | 128 |
| Fc2 input_size | The output size of the second fully connected (dense) layer for Multi and Binary classification. | (1*5), (1*2) |

**Algorithm 4.5** explained the step-by-step working of the proposed Secure and dependable Bi-LSTM GRU Intrusion Detection Framework (S-BiLSTMGRU-IDF) for identifying attacks in the IoMT network.

**Algorithm 4.5. Secure and dependable Bi-LSTM GRU Intrusion Detection Framework (S-BiLSTMGRU-IDF)**

**Steps**

**1 Data Preprocessing**

**Input: Raw network traffic data *D***

**Data Augmentation**

For each numerical feature $v_i$ apply noise augmentation

$$v_i' = v_i + \mathcal{N}(0, \delta^2)$$

Ensures robustness against minor fluctuations.

**Data Shuffling**

Apply random permutation to the dataset

$$D' = \prod(D)$$

Prevents order-dependent learning.

**Feature Encoding**

Standardize continuous features

$$v_{scaled} = \frac{v - \mu_v}{\sigma_v}$$

Encode categorical features using binary encoding

$$BinaryEncode(\gamma) = [0, \ldots.1, \ldots\ldots.0]$$

Construct feature matrix $V \in \mathbb{R}^{m \times d}$

**2 Model Initialization**

**Input:** Preprocessed feature matrix ***V***

**Define Model Architecture:**

Bi-LSTM Layer, Bi-GRU Layer, Dropout Layers, Fully Connected (FC) Layers, Softmax Classification Layer.

**3 Forward Propagation**

(i) Bidirectional LSTM Layer

Compute forward hidden states

$$\overrightarrow{s_t} = \mathcal{L}(V_t, \overrightarrow{s_{t-1}})$$

Compute backward hidden states:

$$\overleftarrow{s_t} = \mathcal{L}(V_t, \overleftarrow{s_{t-1}})$$

Concatenate both

$$S_t = [\overrightarrow{s_t} \oplus \overleftarrow{s_t}]$$

(ii) Bidirectional GRU Layer

Compute update and reset gates

$$z_t = \sigma(W_z V_t + U_z s_{t-1})$$

$$r_t = \sigma(W_r V_t + U_r s_{t-1})$$

Compute hidden state

$$\tilde{s}_t = tanh(W_s V_t + U_s(r_t \odot s_{t-1}))$$

Compute final state

$$s_t = (1 - z_t) \odot s_{t-1} + z_t \odot \tilde{s}_t$$

Apply Bi-GRU

$$S_t = [\vec{s}_t \oplus \overleftarrow{s}_t]$$

(iii) Dropout Layer

Reduce overfitting:

$$S_t^{dropout} = Dropout(S_t, p = 0.5)$$

(iv) Second Bi-LSTM and Bi-GRU Layers

Repeat Steps 3 (i) and 3 (ii) to capture additional temporal dependencies.

**4 Fully Connected Layers and Classification**

First Fully Connected Layer

Transform feature space

$$FC1_{out} = W_{fc1} S_t^{dropout2} + b_{fc1}$$

Second Fully Connected Layer

Map to class outputs

$$FC2_{out} = W_{fc2} FC1_{out} + b_{fc2}$$

Softmax Activation

Convert logits to probability distribution

$$\hat{y} = Softmax(FC2_{out})$$

**5 Model Training and Optimization**

Loss Calculation

Compute categorical cross-entropy loss:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

Gradient Calculation:

Compute gradients for parameter updates

$$\frac{\partial \mathcal{L}}{\partial W_f} = \frac{\partial \mathcal{L}}{\partial s_t} \cdot \frac{\partial s_t}{\partial W_f}$$

Optimizer Update

Update model weights using Adam optimizer

$$W \leftarrow W - \eta \frac{\partial L}{\partial W}$$

**6 Model Evaluation**

Compute performance metrics such as Accuracy, Precision, Recall, and F1-Score.

Test on unseen data.

Save trained model for deployment.

**Final Output**

Trained S-BiLSTMGRU-IDF model for intrusion detection.

---

### 4.5.3 Dependability analysis

The dependability of the proposed model is critical for its effectiveness in real-world cybersecurity applications, particularly within the Internet of Medical Things (IoMT) environment. This analysis focuses on three key aspects: efficiency, availability, and scalability. Each of these elements plays a vital role in ensuring that the Secure and dependable Bi-LSTM GRU Intrusion Detection Framework (S-BiLSTMGRU-IDF) model can reliably detect intrusions with minimal resource consumption, maintain continuous operation in diverse deployment environments, and adapt to increasing demands without performance degradation.

**(i) Efficiency**

The proposed model efficiently utilizes computational resources for both training and inference while effectively processing data at scale. By integrating Bidirectional LSTM and GRU layers, it captures complex patterns in intrusion detection data. The model's optimized gradient computation accelerates training and reduces resource consumption. Dropout layers prevent overfitting, enhancing generalization to unseen data. Hyperparameter optimization, including the use of the Adam algorithm, further balances accuracy with minimal resource use, delivering accurate intrusion detection with reduced computational overhead.

**(ii) Availability**

The proposed model is designed for high availability, crucial for real-time intrusion detection. Its Bidirectional LSTM and GRU layers handle temporal dependencies efficiently, ensuring robustness and reliability. These layers retain critical information over long sequences, reducing the risk of missed detections. Dropout layers maintain reliability by preventing overfitting, even with new data. The model's adaptability to various deployment environments, from cloud to edge devices, ensures continuous service across different infrastructures.

**(iii) Scalability**

The proposed model scales effectively, handling increasing volumes of complex network traffic in IoMT systems. We tested the model's scalability over 25, 50, 75, and 100 epochs, observing that it improves accuracy as training extends without requiring proportional increases in computational resources. Efficient gradient computations and advanced optimization techniques support this scalability, enabling the model to manage larger datasets and longer training periods while maintaining high performance. This ensures that the model can be deployed in large-scale environments without compromising its accuracy in detecting intrusions.

## 4.6 Experimental details and result analysis

This section presents the experimental procedures followed to evaluate the proposed models and provides a detailed analysis of the results obtained. The experiments were designed to assess the performance of the models with particular attention to metrics. The results are analyzed in the context of their implications for the model's

effectiveness in real-world applications, highlighting strengths, identifying potential limitations, and comparing them against existing approaches.

### 4.6.1 Computational Setup/Testbed Configuration

The experiments were conducted on a high-performance system equipped with an Intel® Core™ Ultra 9 185H processor, featuring 16 cores and a 24 MB cache, capable of reaching speeds up to 5.1 GHz with Turbo Boost. The system ran on Windows 11 and was powered by an NVIDIA® GeForce RTX™ 4070 GPU with 8 GB of GDDR6 memory, providing robust support for computationally intensive tasks. The memory configuration included 32 GB of DDR5 RAM, clocked at 5600 MT/s, distributed across two 16 GB modules, ensuring efficient data processing and rapid model training. PyTorch [189] and scikit-learn [190] libraries were utilized to implement Deep learning models and execute the experiments. PyTorch was primarily used for deep learning tasks, leveraging its dynamic computation graph and GPU acceleration features. Scikit-learn provided the necessary tools for preprocessing, model evaluation, and applying classical machine learning algorithms. This combination of hardware and software facilitated the efficient training and evaluation of models, enabling comprehensive analysis and experimentation.

### 4.6.2 Dataset Overview

In this research, we used [191] a comprehensive benchmark dataset i.e. CICIoMT-2024, which is designed to enhance the development and evaluation of security solutions for the Internet of Medical Things (IoMT). The dataset captures network traffic from a testbed comprising 40 IoMT devices, including 25 real and 15 simulated devices, operating across healthcare-relevant protocols such as Wi-Fi, MQTT, and Bluetooth. A total of 18 attack types were executed and categorized into five classes: Distributed Denial of Service (DDoS), Denial of Service (DoS), Reconnaissance (Recon), Spoofing, and MQTT-based attacks, with an additional DoS attack targeting Bluetooth Low Energy (BLE) devices. Network traffic was collected using a network tap positioned between the switch and Wi-Fi/MQTT-enabled IoMT devices, ensuring real-time packet duplication. Additionally, malicious activity on BLE-enabled devices was captured using a combination of a malicious PC and a smartphone. To provide a holistic view of IoMT security, the dataset includes profiling experiments capturing device behavior under different conditions: isolated power state analysis, idle state monitoring during non-interactive periods, active network traffic generation through user interactions, and functional testing of all device capabilities. By incorporating a diverse range of attack scenarios and normal activity patterns, this dataset serves as a valuable resource for advancing intrusion detection and security research, particularly in machine learning and blockchain-based security frameworks. **Table 4.3** presents the details of the dataset samples.

**Table 4.3.** Dataset Description: Distribution of Attack Types and Normal Traffic in the Dataset

| Class Category | Attack Type | Count |
|---|---|---|
| DDoS | DDoS UDP | 1998026 |
| | DDoS ICMP | 1887175 |
| | DDoS TCP | 987063 |
| | DDoS SYN | 974359 |
| DoS | DoS UDP | 704503 |
| | DoS ICMP | 514724 |

| | | |
|---|---|---|
| | DoS TCP | 462480 |
| | DoS SYN | 540498 |
| Spoofing | ARP Spoofing | 17791 |
| MQTT | DDoS connect flood | 214952 |
| | Malformed data | 6877 |
| | DDoS publish flood | 36039 |
| | DoS connect flood | 15904 |
| | DoS publish flood | 52881 |
| RECON | Port scan | 106603 |
| | Ping sweep | 926 |
| | OS scan | 20666 |
| | Recon VulScan | 3207 |
| Normal Traffic | - | 230339 |

### 4.7 Result Analysis for Blockchain based secured framework (Phase 1)

This section presents the result Analysis for proposed Blockchain based secured framework. **Table 4.4** depicts a comparative analysis of the proposed DA-DRL-AES-SHA-512 methodology against existing cryptographic approaches, including DSA, ECC, Blowfish, and DA-DRL-AES, across multiple performance metrics. The results highlight the superior efficiency of the proposed method in securing IoT-based healthcare applications. The encryption and decryption times of the proposed approach of 0.097501 sec and 0.084628 sec, respectively are the lowest among all compared methods, ensuring faster data transformation and retrieval. Similarly, the key generation time of 0.184278 sec is significantly reduced, facilitating rapid cryptographic operations essential for authentication and secure communication in blockchain-based systems.

Moreover, the proposed method demonstrates the lowest block creation time of 0.129691 sec, accelerating transaction validation and improving blockchain scalability. The shared record time, which measures the efficiency of secure data sharing, is also minimized by 12.4035 sec, making the system well-suited for real-time applications like healthcare data exchange. Furthermore, the response time of 216.6275 sec is the shortest among all methods, enabling high-speed processing and quick decision-making in critical applications such as intrusion detection and medical data analysis.

In terms of data reliability, the proposed methodology achieves the highest restoration efficiency of 0.919985, ensuring robust data recovery and integrity within blockchain networks. Additionally, it exhibits the highest throughput of 75.6327 transactions/sec, demonstrating its capability to handle a high volume of transactions effectively. The reduced latency of 0.3055 sec further enhances real-time processing, making the system ideal for security-sensitive applications where minimal delay is crucial. These findings collectively establish the proposed DA-DRL-AES-SHA-512 methodology as a highly efficient, secure, and scalable cryptographic framework tailored for IoT and healthcare environments, outperforming existing cryptographic techniques across all critical performance parameters. **Figure 4.4** depicts the Comparative Performance Analysis of Cryptographic Methods Across Various Security Metrics.

**Table 4.4.** Comparative Performance Analysis of Cryptographic Methods Across Various Security Metrics

| Metrics | DSA | ECC | Blowfish | DA-DRL-AES | Proposed Methodology (DA-DRL-AES-SHA-512) |
|---|---|---|---|---|---|
| **Encryption Time (Sec)** | 0.191862 | 0.177652 | 0.209461 | 0.134701 | **0.097501** |
| **Decryption Time (Sec)** | 0.177911 | 0.164801 | 0.194722 | 0.124968 | **0.084628** |
| **Key Generation Time (Sec)** | 0.371918 | 0.357283 | 0.408623 | 0.264322 | **0.184278** |
| **Block Creation Time (Sec)** | 0.264782 | 0.247543 | 0.289842 | 0.171966 | **0.129691** |
| **Shared Record Time (Sec)** | 22.4873 | 19.8752 | 24.5783 | 16.4871 | **12.4035** |
| **Response Time (Sec)** | 230.4812 | 225.2763 | 235.6782 | 220.1894 | **216.6275** |
| **Restoration Efficiency** | 0.851822 | 0.859743 | 0.813935 | 0.887002 | **0.919985** |
| **Throughput** | 40.1256 | 56.4821 | 35.2984 | 69.1043 | **75.6327** |
| **Latency** | 0.5796 | 0.5098 | 0.5993 | 0.3894 | **0.3055** |

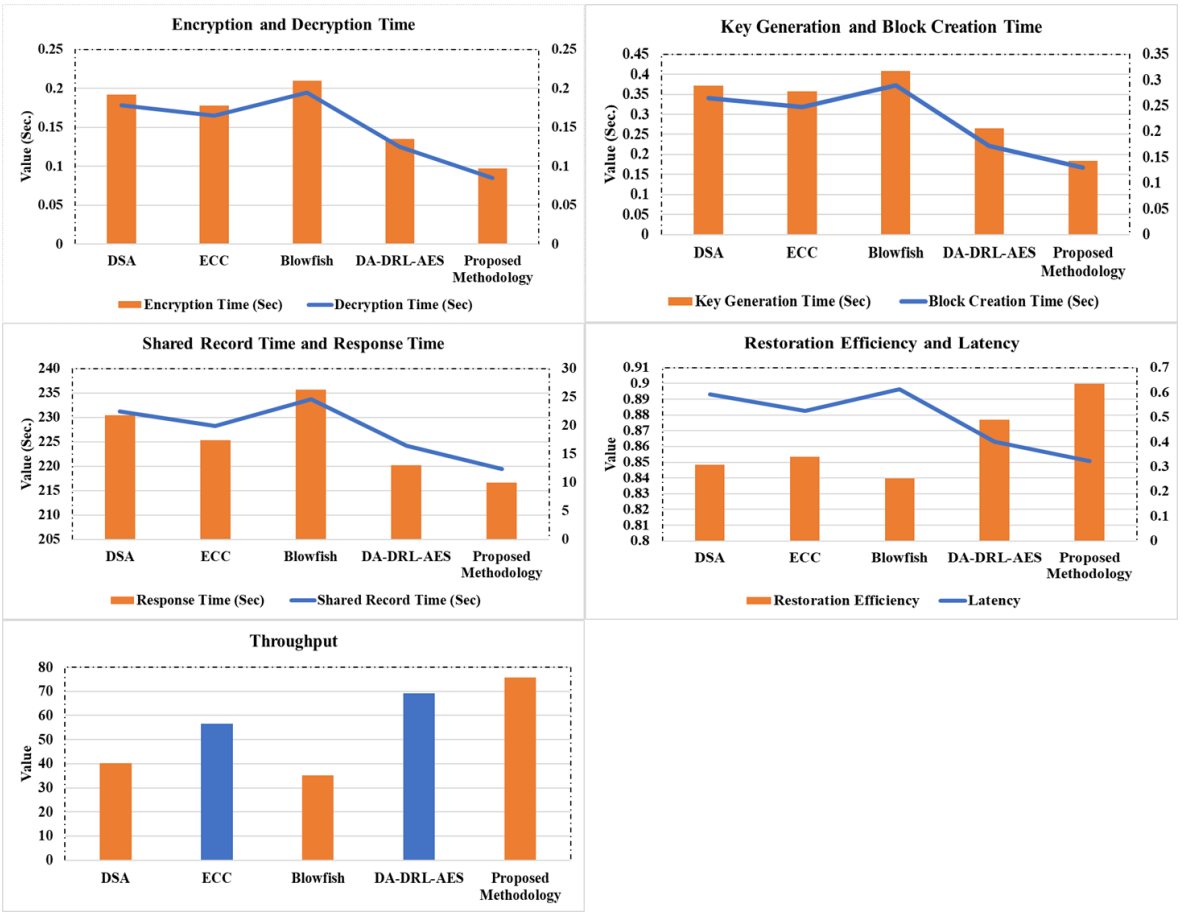



**Fig. 4.4.** Comparison of the proposed methodology with other cryptographic approaches

**Table 4.5** presents a comprehensive comparison of the proposed methodology against existing cryptographic techniques, focusing on scalability, energy consumption, computational overhead, and communication overhead. The results emphasize the superior efficiency of the proposed DA-DRL-AES-SHA-512 approach in handling large-scale transactions while maintaining minimal resource consumption.

In terms of scalability, measured in Transactions per Second (TPS), the proposed methodology achieves the highest performance at 2729.21 TPS, significantly outperforming DSA (1348.27 TPS), ECC (1429.11 TPS), Blowfish (1230.31 TPS), and DA-DRL-AES (1908.43 TPS). This demonstrates its capability to handle a high transaction load, making it highly suitable for real-time applications such as blockchain-based intrusion detection and healthcare data sharing. The energy consumption analysis further highlights the efficiency of the proposed method, which records the lowest energy consumption at 0.3664 Joules, compared to DA-DRL-AES (0.5240 J), ECC (0.6997 J), DSA (0.7417 J), and Blowfish (0.8128 J). This energy-efficient performance is particularly advantageous in IoT and healthcare environments where power constraints are a major concern. Regarding computational overhead, the proposed methodology exhibits the lowest value at 0.48%, significantly lower than DA-DRL-AES (0.76%), ECC (1.24%), DSA (1.85%), and Blowfish (2.30%). This reduced overhead ensures that the system requires fewer computational resources, leading to faster execution times and improved overall efficiency.

Lastly, the communication overhead is also minimized with the proposed approach, achieving only 10.25%, which is far lower than DA-DRL-AES (20.03%), ECC (30.11%), DSA (40.35%), and Blowfish (50.25%). The reduced communication overhead enhances network efficiency and optimizes blockchain storage by minimizing the additional data required for secure transactions. Overall, the findings confirm that the proposed DA-DRL-AES-SHA-512 methodology significantly improves scalability, reduces energy consumption, minimizes computational overhead, and enhances communication efficiency compared to existing cryptographic approaches. These advantages make it an ideal choice for secure, high-performance blockchain applications in IoT and healthcare systems.

**Table 4.5.** Performance Comparison of Cryptographic Techniques Based on Scalability, Energy Consumption, and Overheads for Blockchain Framework

| Techniques | Scalability Analysis (Transactions per Second - TPS) | Energy Consumption Analysis (Joules) | Computational Overhead (%) | Communication Overhead (%) |
|---|---|---|---|---|
| DSA | 1348.27 | 0.7417 | 1.85 | 40.35 |
| ECC | 1429.11 | 0.6997 | 1.24 | 30.11 |
| Blowfish | 1230.31 | 0.8128 | 2.30 | 50.25 |
| DA-DRL-AES | 1908.43 | 0.5240 | 0.76 | 20.03 |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **2729.21** | **0.3664** | **0.48** | **10.25** |

**Table 4.6** presents a comparative analysis of network overhead for various cryptographic techniques, highlighting the efficiency of the proposed methodology in minimizing the additional data burden on the network. The proposed DA-DRL-AES-SHA-512 methodology achieves the lowest network overhead of 0.1289%, significantly outperforming traditional techniques such as DSA (0.4782%), ECC (0.3145%), Blowfish (0.5934%), and DA-DRL-AES (0.1949%). This reduction in overhead ensures that the network operates with minimal extra data transmission, leading to faster transaction processing and improved scalability.

Blowfish exhibits the highest network overhead (0.5934%), which may be attributed to its relatively complex encryption and key management processes that introduce additional communication costs. DSA and ECC also show higher network overhead values of 0.4782% and 0.3145%, respectively, which could result in increased network latency and reduced efficiency in large-scale deployments.

The DA-DRL-AES method (0.1949%) demonstrates an improvement over conventional approaches; however, the proposed methodology further optimizes network efficiency, making it ideal for resource-constrained environments such as IoT-based healthcare systems and blockchain applications. The drastic reduction in network overhead enhances real-time data transmission, reduces latency, and improves the overall network performance.

**Table 4.6.** Blockchain Performance Metrics Comparison of Cryptographic Techniques based on network overhead

| Techniques | Throughput (Tx/s) | Encrpt_Time (Sec) | Network Overhead (%) |
|---|---|---|---|
| DSA | 40.1256 | 0.191862 | 0.4782 |
| ECC | 56.4821 | 0.177652 | 0.3145 |
| Blowfish | 35.2984 | 0.209461 | 0.5934 |
| DA-DRL-AES | 69.1043 | 0.134701 | 0.1949 |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **75.6327** | **0.097501** | **0.1289** |

**Table 4.7** presents a comparative analysis of different cryptographic techniques, focusing on their throughput and estimated capacity to support IoT devices. Throughput, measured in transactions per second (Tx/s), plays a crucial role in determining how efficiently a cryptographic technique can handle large-scale IoT networks. The proposed DA-DRL-AES-SHA-512 methodology exhibits the highest throughput of 75.6327 Tx/s, enabling support for 9,500 IoT devices. This surpasses existing methods such as DA-DRL-AES (8,500 devices), ECC (7,000 devices), and DSA (5,000 devices), demonstrating the scalability and efficiency of the proposed method in resource-constrained IoT environments. The low computational overhead and optimized encryption techniques contribute to this enhanced capacity, making the proposed method ideal for secure, high-speed data transmission in real-time applications.

In contrast, conventional cryptographic techniques like Blowfish, which supports only 4,500 devices with a throughput of 35.2984 Tx/s, struggle to maintain efficiency in large-scale IoT deployments due to higher computational costs. ECC and DSA, while moderately scalable, still lag behind advanced deep learning-assisted encryption models like DA-DRL-AES. The findings highlight the significance of enhanced cryptographic optimization in improving IoT scalability. By reducing encryption time and increasing throughput, the proposed methodology ensures secure and efficient communication across a vast number of IoT devices, making it a suitable candidate for blockchain-integrated healthcare systems, industrial automation, and other large-scale IoT applications

**Table 4.7.** Blockchain Scalability Analysis for IoT Devices

| Techniques | Throughput (Tx/s) | Estimated IoT devices |
|---|---|---|
| DSA | 40.1256 | 5000 |
| ECC | 56.4821 | 7000 |
| Blowfish | 35.2984 | 4500 |
| DA-DRL-AES | 69.1043 | 8500 |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **75.6327** | **9500** |

**Table 4.8** compares encryption times and corresponding security levels for various cryptographic techniques. The proposed DA-DRL-AES-SHA-512 methodology achieves the highest security level while maintaining the lowest encryption time (0.097501 sec), demonstrating its efficiency in safeguarding sensitive data. In contrast, traditional techniques like DSA and Blowfish, despite having higher encryption times (0.191862 sec and 0.209461 sec, respectively), provide only low security, making them less suitable for high-risk environments.

Advanced cryptographic techniques such as ECC and DA-DRL-AES strike a balance between encryption speed and security, offering medium and high security levels, respectively. However, the proposed methodology significantly outperforms existing approaches by providing very high security with minimal computational overhead, making it an ideal choice for resource-constrained IoT and healthcare applications requiring both speed and robustness.

**Table 4.8.** Comparison of Encryption times and their corresponding security levels for Blockchain Framework

| Techniques | Encrpt_Time (Sec) | Security Level (Qualitative) |
|---|---|---|
| DSA | 0.191862 | Low |
| ECC | 0.177652 | Medium |
| Blowfish | 0.209461 | Low |
| DA-DRL-AES | 0.134701 | High |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **0.097501** | **Very High** |

**Table 4.9** presents a comparative analysis of estimated energy consumption across different cryptographic techniques. The proposed DA-DRL-AES-SHA-512 methodology demonstrates the lowest energy consumption (0.011360 kWh), making it the most efficient approach. This is attributed to its optimized encryption and block creation times (0.097501 sec and 0.129691 sec, respectively), significantly reducing computational power requirements. In contrast, conventional methods like Blowfish and DSA consume higher energy (0.024965 kWh and 0.022832 kWh, respectively), indicating their inefficiency in low-power environments such as IoT and healthcare applications.

Among the existing techniques, ECC and DA-DRL-AES offer moderate energy efficiency (0.021260 kWh and 0.015333 kWh, respectively), balancing security and power consumption. However, the proposed methodology outperforms them by achieving the best trade-off between energy efficiency and cryptographic strength. This

makes it an ideal choice for energy-constrained IoT ecosystems, ensuring secure data transmission while minimizing power usage

**Table 4.9.** Comparison of encryption times, block creation times, and estimated energy consumption for Blockchain Framework

| Techniques | Encrpt_Time (Sec) | Block_Creation (Sec) | Estimated Energy Consumption (kWh) |
|---|---|---|---|
| DSA | 0.191862 | 0.264782 | 0.022832 |
| ECC | 0.177652 | 0.247543 | 0.021260 |
| Blowfish | 0.209461 | 0.289842 | 0.024965 |
| DA-DRL-AES | 0.134701 | 0.171966 | 0.015333 |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **0.097501** | **0.129691** | **0.011360** |

The qualitative assessment of User Experience in cryptographic techniques is influenced by response time and record-sharing time, which directly impact system performance and efficiency. Techniques with lower response and record-sharing times provide a smoother and more seamless experience for users, particularly in real-time IoT and healthcare applications where efficiency is critical as shown in **Table 4.10**.

The proposed method demonstrates the best user experience by achieving the lowest response time (216.6275 sec) and fastest record-sharing time (12.4035 sec), leading to a "Very High" user experience rating. In contrast, DSA and Blowfish exhibit higher delays, resulting in a "Low" user experience rating. ECC offers moderate performance, leading to a "Medium" rating, while DA-DRL-AES provides a "High" user experience, though it is slightly less efficient than the proposed method. This analysis highlights the effectiveness of the proposed approach in enhancing user satisfaction through faster processing and improved responsiveness.

**Table 4.10.** Performance Benchmarking of Cryptographic Techniques in Healthcare Data Security

| Techniques | Response Time (Sec) | Sharing Record Time (Sec) | User Experience (Qualitative) |
|---|---|---|---|
| DSA | 230.4812 | 22.4873 | Low |
| ECC | 225.2763 | 19.8752 | Medium |
| Blowfish | 235.6782 | 24.5783 | Low |
| DA-DRL-AES | 220.1894 | 16.4871 | High |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **216.6275** | **12.4035** | **Very High** |

Interoperability is a critical factor in cryptographic techniques, especially in distributed systems where seamless data exchange across multiple platforms is essential. The sharing record time directly influences interoperability, as faster record-sharing improves system adaptability and integration efficiency.

The proposed method exhibits the best interoperability with the fastest record-sharing time (12.4035 sec), earning a Very High rating as shown in **Table 4.11**. DA-DRL-AES also demonstrates high interoperability due to its efficient sharing time (16.4871 sec). In contrast, DSA and Blowfish have slower sharing times, leading to a

Moderate rating, indicating potential limitations in system integration. ECC offers a balance, achieving Moderate-High interoperability. This comparison highlights that reducing sharing record time enhances interoperability, making the proposed approach highly suitable for interconnected and scalable blockchain-based systems.

**Table 4.11.** Comparative Assessment of Cryptographic Techniques Based on Record Sharing Efficiency and Interoperability

| Techniques | Sharing Record Time (Sec) | Interoperability (Qualitative) |
|---|---|---|
| DSA | 22.4873 | Moderate |
| ECC | 19.8752 | Moderate-High |
| Blowfish | 24.5783 | Moderate |
| DA-DRL-AES | 16.4871 | High |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **12.4035** | **Very High** |

Transaction finality time is a crucial metric in cryptographic techniques, as it determines how quickly a transaction is confirmed and becomes irreversible within a blockchain network. A lower transaction finality time ensures faster processing, improved system efficiency, and reduced latency in real-time applications. As observed in the **Table 4.12**, the proposed method achieves the fastest transaction finality time (0.129691 sec), indicating its superiority in rapid transaction confirmation. DA-DRL-AES also exhibits low finality time (0.171966 sec), making it efficient for high-speed applications. In contrast, Blowfish has the highest transaction finality time (0.289842 sec), which may introduce delays in transaction validation. DSA and ECC fall in between, with moderate finality times of 0.264782 sec and 0.247543 sec, respectively. These results highlight that the proposed approach significantly enhances transaction efficiency, making it an optimal choice for high-performance blockchain systems

**Table 4.12.** Performance Evaluation of Cryptographic Techniques Based on Block Creation and Transaction Finality Time

| Techniques | Block_Creation (Sec) | Transaction Finality Time (Sec) |
|---|---|---|
| DSA | 0.264782 | 0.264782 |
| ECC | 0.247543 | 0.247543 |
| Blowfish | 0.289842 | 0.289842 |
| DA-DRL-AES | 0.171966 | 0.171966 |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **0.129691** | **0.129691** |

Fault tolerance is a critical aspect of cryptographic techniques, ensuring system resilience and reliability in the event of failures or disruptions. Higher fault tolerance enhances data integrity and system robustness, making cryptographic methods more suitable for secure applications. The proposed method demonstrates the highest fault tolerance (Very High) due to its superior restoration efficiency (0.919985) and the lowest response time (216.6275 sec) among the techniques as shown in **Table 4.13**. DA-DRL-AES also exhibits High fault tolerance, benefiting from efficient recovery mechanisms and a restoration efficiency of 0.887002. In contrast, Blowfish has the lowest fault tolerance (Moderate), with the highest response time (235.6782 sec) and a lower restoration efficiency

(0.813935), indicating weaker resilience. DSA and ECC fall in the Moderate and Moderate-High categories, respectively, balancing recovery efficiency and response time. These findings suggest that the proposed method is the most resilient, ensuring minimal disruptions in cryptographic operations

**Table 4.13** Resilience and Performance Assessment of Cryptographic Techniques in Secure Data Processing

| Techniques | Response Time (Sec) | Restoration Efficiency | Fault Tolerance (Qualitative) |
|---|---|---|---|
| DSA | 230.4812 | 0.851822 | Moderate |
| ECC | 225.2763 | 0.859743 | Moderate-High |
| Blowfish | 235.6782 | 0.813935 | Moderate |
| DA-DRL-AES | 220.1894 | 0.887002 | High |
| **Proposed Methodology (DA-DRL-AES-SHA-512)** | **216.6275** | **0.919985** | **Very High** |

### 4.8 Result Analysis for Intrusion Detection Framework (Phase 2)

This section provides the result analysis for binary and multiclass classification of the proposed DL model, which are explained as follows:

#### 4.8.1 Binary Classification

**Table 4.14** presents the performance comparison of the proposed model on both the training and testing datasets, demonstrating its effectiveness across multiple evaluation metrics. The training set results indicate near-perfect performance, with an accuracy of 0.9994 and an F1-score of 0.9961, confirming the model's capability to learn complex patterns effectively. The positive predictive value (PPV) of 1.0 on both datasets indicates that all predicted positive instances are indeed correct, reinforcing the robustness of the model. The true positive rate (TPR) of 0.9923 and true negative rate (TNR) of 0.9995 on the training set suggest a high ability to correctly classify both positive and negative instances, ensuring a balanced detection performance. The ROC_AUC score of 1.0 further confirms the model's superior discriminatory power in distinguishing between different classes.

On the testing set, the model maintains high generalization capability, with accuracy (0.9983), F1-score (0.9923), and TPR (0.9902) slightly lower than the training set but still within an optimal range. The TNR of 0.9987 and ROC_AUC of 0.9989 indicate that the model performs well across various classification thresholds. The training loss (TL) of 0.0325 and validation loss (VL) of 0.0216 confirm stable training with minimal overfitting. While the increase in TL and VL on the testing set suggests minor performance degradation, the gap remains negligible, ensuring the model's reliability. Overall, these results indicate that the proposed approach achieves high precision, recall, and generalization, making it suitable for real-world applications.

**Table 4.14.** Performance comparison of the model on both the training and testing sets

| Metrics | Training-Set | Testing-Set |
|---|---|---|
| **Accuracy** | 0.9994 | 0.9983 |
| **PPV** | 1.0 | 1.0 |
| **TPR** | 0.9923 | 0.9902 |
| **F1-Score** | 0.9961 | 0.9923 |

| | | |
|---|---|---|
| **TNR** | 0.9995 | 0.9987 |
| **ROC_AUC** | 1.0 | 0.9989 |
| **TL** | 0.0325 | 0.0424 |
| **VL** | 0.0087 | 0.0216 |

**Table 4.15** presents the qualitative analysis of the proposed model for binary classification, highlighting its performance in distinguishing between normal and attack classes. The model demonstrates exceptional accuracy (0.9994 on average) across both classes, ensuring reliable classification. For Class 0 (Normal), the model achieves a true positive rate (TPR) of 0.9944, a true negative rate (TNR) of 1.0, and a ROC_AUC score of 0.9998, indicating a near-perfect ability to distinguish normal instances from attacks. Similarly, for Class 1 (Attack), the model maintains high precision (PPV = 1.0), recall (TPR = 0.9902), and F1-score (0.9951), demonstrating its effectiveness in detecting attacks. The average ROC_AUC of 1.0 further confirms that the model can effectively differentiate between the two classes with minimal classification errors. These results highlight the robustness and reliability of the proposed model in accurately classifying normal and attack instances, making it well-suited for real-world intrusion detection applications.

**Table 4.15.** Qualitative analysis of the proposed model for binary Classification

| Classes | Samples | Ac | PPV | TPR | F1 | TNR | ROC_AUC |
|---|---|---|---|---|---|---|---|
| **Class 0** | **Normal** | 0.9998 | 1.0 | 0.9944 | 0.9972 | 1.0 | 0.9998 |
| **Class 1** | **Attack** | 0.9990 | 1.0 | 0.9902 | 0.9951 | 0.9990 | 1.0 |
| **Average** | | **0.9994** | **1.0** | **0.9923** | **0.9961** | **0.9995** | **1.0** |

**Table 4.16** presents the quantitative analysis of the proposed model for binary classification, evaluating its performance using various statistical metrics. The Matthews Correlation Coefficient (MCC) values of 0.9996 for Class 0 (Normal) and 0.9988 for Class 1 (Attack) indicate a strong correlation between predicted and actual classifications, confirming the model's reliability. The negative predictive values (NPV) of 0.9997 (Normal) and 0.9994 (Attack) suggest that negative predictions are highly accurate with minimal false negatives. The model achieves a false discovery rate (FDR) of 0.0 for both classes, ensuring no misclassified positive predictions. The false positive rate (FPR) and false omission rate (FOR) remain extremely low, with Class 0 having FPR = 0.0 and FOR = 0.0003, while Class 1 has FPR = 0.0010 and FOR = 0.0006, confirming the model's ability to minimize incorrect classifications. Additionally, the false negative rate (FNR) is 0.0056 for Class 0 and 0.0098 for Class 1, indicating the model rarely misclassifies actual positives. The Markedness (MK) and Informedness (BM) scores are consistently high, demonstrating the model's strong predictive capability. These results validate the model's robustness, ensuring its effectiveness in binary classification for intrusion detection systems with minimal classification errors. **Figure 4.5** depicts the Qualitative and Quantitative analysis of the proposed model for Binary Classification

**Table 4.16.** Quantitative Analysis of the Proposed Model for Binary Classification

| Classes | MCC | NPV | FDR | FPR | FNR | FOR | MK | BM |
|---|---|---|---|---|---|---|---|---|
| **Class 0** | 0.9996 | 0.9997 | 0.0 | 0.0 | 0.0056 | 0.0003 | 0.9997 | 0.9944 |
| **Class 1** | 0.9988 | 0.9994 | 0.0 | 0.0010 | 0.0098 | 0.0006 | 0.9994 | 0.9892 |

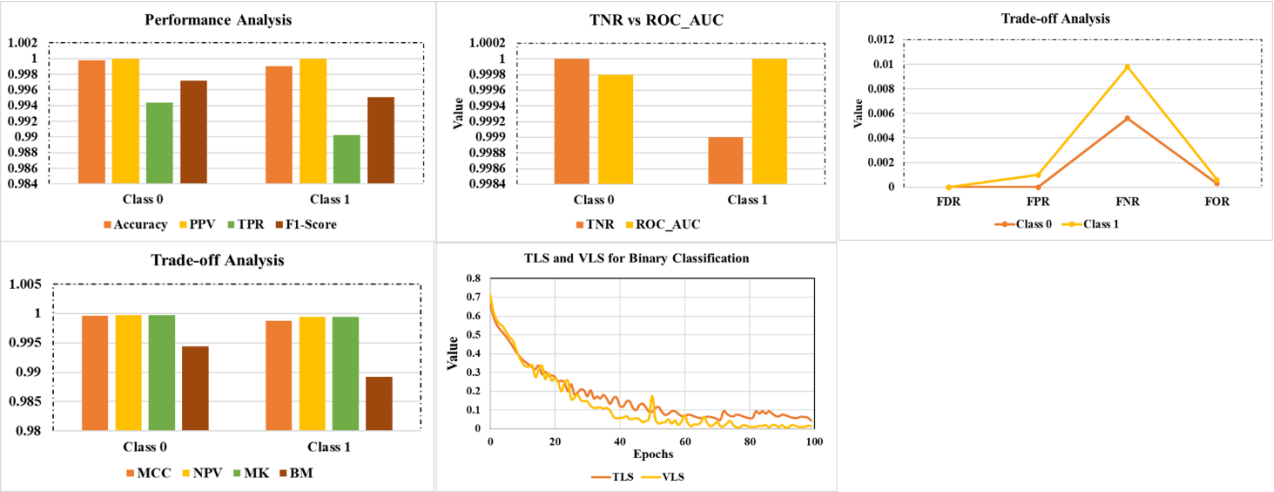



**Fig. 4.5.** Qualitative and Quantitative analysis of the proposed model for Binary Classification

### 4.8.2 Multiclass Classification

**Table 4.17** presents the qualitative analysis of the proposed model for multiclass classification, assessing its performance across different attack categories. The overall accuracy (Ac) remains consistently high, with an average of 0.9989, indicating the model's ability to classify different attack types with remarkable precision. The positive predictive value (PPV) is nearly 1.0 across all classes, confirming the model's strong capability to correctly predict positive instances with minimal false positives. The true positive rate (TPR) ranges from 0.9859 (DoS) to 0.9965 (DDoS), demonstrating effective detection of actual attacks. The F1-score, which balances precision and recall, remains above 0.99 for all classes, highlighting the model's robustness in handling class imbalances. The true negative rate (TNR) and ROC-AUC scores exceed 0.99 for most classes, signifying the model's ability to distinguish between normal and attack instances effectively. Notably, the model performs exceptionally well across all categories, including complex attack types like Spoofing and MQTT-based threats, with marginal variations in classification performance. These results affirm the model's reliability and adaptability in detecting various network intrusions with high accuracy, precision, and robustness.

**Table 4.17.** Qualitative analysis of the proposed model for multiclassification

| Classes | Samples | Ac | PPV | TPR | F1 | TNR | ROC_AUC |
|---|---|---|---|---|---|---|---|
| **Class 0** | Normal | 0.9990 | 0.9990 | 0.9923 | 0.9956 | 0.9981 | 0.9916 |
| **Class 1** | DoS | 0.9916 | 0.9989 | 0.9859 | 0.9924 | 0.9958 | 0.9924 |
| **Class 2** | DDoS | 0.9955 | 0.9990 | 0.9965 | 0.9978 | 0.9972 | 0.9950 |
| **Class 3** | RECON | 0.9980 | 0.9988 | 0.9939 | 0.9963 | 0.9984 | 0.9962 |
| **Class 4** | MQTT | 0.9975 | 0.9991 | 0.9948 | 0.9969 | 0.9980 | 0.9964 |
| **Class 5** | Spoofing | 0.9962 | 0.9902 | 0.9927 | 0.9914 | 0.9968 | 0.9947 |
| **Avg. Score** | | **0.9989** | **0.9989** | **0.9962** | **0.9967** | **0.9981** | **0.9952** |

**Table 4.18** presents the quantitative analysis of the proposed model for multiclass classification, evaluating various statistical metrics that reflect its classification performance. The Matthews correlation coefficient (MCC)

remains consistently high, ranging from 0.9923 (DoS) to 0.9981 (Normal), indicating strong correlation between predicted and actual labels. The negative predictive value (NPV) exceeds 0.9985 across all classes, signifying the model's ability to correctly identify negative cases with minimal false negatives. The false discovery rate (FDR) remains close to zero, showing that nearly all positive predictions are correct. Similarly, the false positive rate (FPR) and false omission rate (FOR) stay below 0.003, confirming the model's precision in reducing misclassifications. The false negative rate (FNR) is lowest for DDoS (0.0035) and highest for DoS (0.0141), indicating strong recall across all attack types. The markedness (MK) and informedness (BM) scores remain above 0.99, reinforcing the model's balanced performance across all classes. These results demonstrate the model's high reliability, precision, and robustness in detecting various network threats, ensuring effective intrusion detection with minimal errors. **Figure 4.6** depicts the Qualitative and Quantitative analysis of the proposed model for Binary Classification

**Table 4.18.** Quantitative analysis of the proposed model for multiclassification

| Classes | Samples | MCC | NPV | FDR | FPR | FNR | FOR | MK | BM |
|---|---|---|---|---|---|---|---|---|---|
| **Class 0** | Normal | 0.9981 | 0.9991 | 0.0010 | 0.0019 | 0.0077 | 0.0009 | 0.9981 | 0.9904 |
| **Class 1** | DoS | 0.9923 | 0.9987 | 0.0011 | 0.0024 | 0.0141 | 0.0013 | 0.9966 | 0.9817 |
| **Class 2** | DDoS | 0.9958 | 0.9990 | 0.0010 | 0.0028 | 0.0035 | 0.0010 | 0.9980 | 0.9944 |
| **Class 3** | RECON | 0.9971 | 0.9992 | 0.0012 | 0.0016 | 0.0061 | 0.0008 | 0.9984 | 0.9923 |
| **Class 4** | MQTT | 0.9967 | 0.9993 | 0.0009 | 0.0012 | 0.0052 | 0.0007 | 0.9985 | 0.9932 |
| **Class 5** | Spoofing | 0.9952 | 0.9985 | 0.0098 | 0.0020 | 0.0073 | 0.0015 | 0.9943 | 0.9915 |

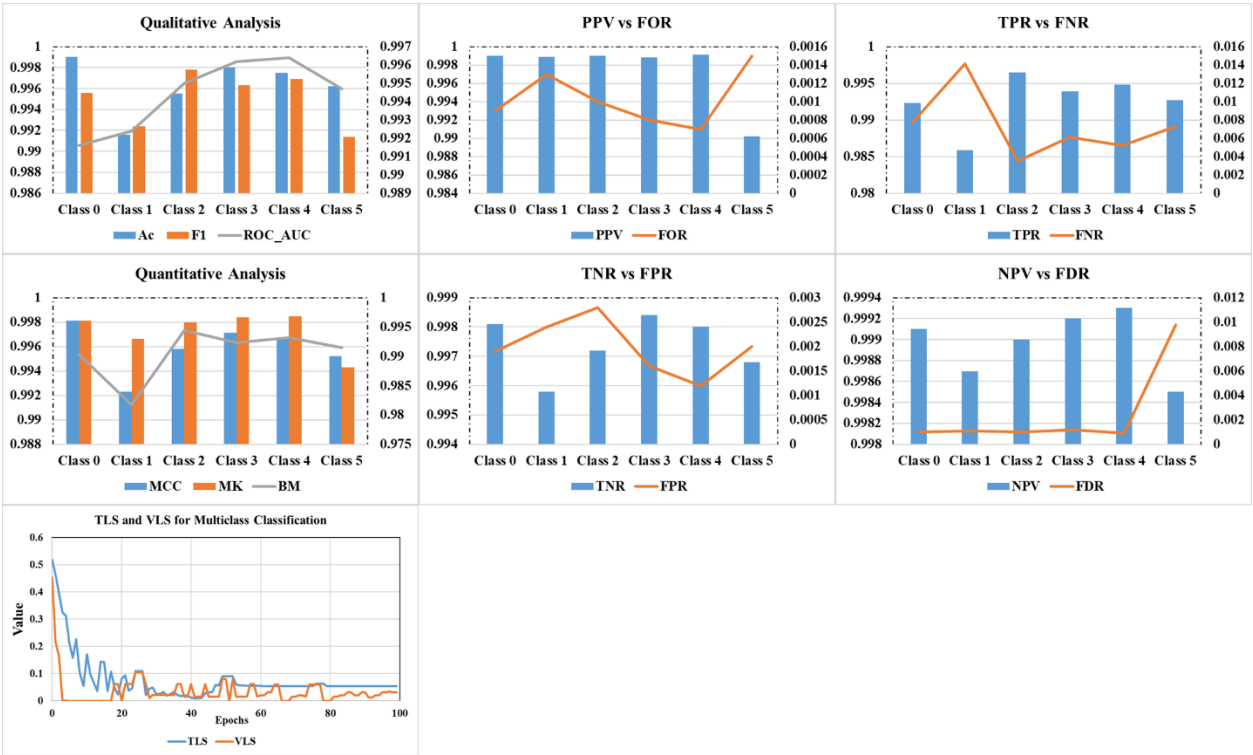



**Fig. 4.6.** Qualitative and Quantitative analysis of the proposed model for Multiclass Classification

**Table 4.19** presents the performance evaluation of the proposed model across different epochs, illustrating how the training and validation losses (TLS and VLS) decrease while accuracy and other metrics improve over time. At 25 epochs, the model achieves a training accuracy (Tr-AC) of 98.56% and testing accuracy (Te-AC) of 98.12%, with a relatively higher TLS (0.2089) and VLS (0.1098), indicating room for improvement. As training progresses to 50 epochs, there is a significant reduction in TLS (0.0898) and VLS (0.0879), with an increase in accuracy (99.21% training, 98.98% testing). By 75 epochs, further refinements lead to a training accuracy of 99.63% and testing accuracy of 99.39%, along with improved F1-score (99.52%) and ROC-AUC (99.48%). At 100 epochs, the model achieves its best performance with minimal losses (TLS = 0.0420, VLS = 0.0238), near-perfect training accuracy (99.89%), and high testing accuracy (99.58%), alongside an excellent PPV (99.89%), TPR (99.62%), and TNR (99.81%). These results confirm the model's convergence stability, reduced overfitting, and optimal classification performance with continued training, ensuring high precision and reliability for intrusion detection. **Figure 4.7** depicts the Performance evaluation of the proposed model across $E \times 25$ Epochs.

**Table 4.19.** Performance evaluation of the proposed model across $E \times 25$ Epochs

| Epochs | TLS | VLS | Tr-AC | Te-AC | PPV | TPR | F1 | TNR | ROC_AUC |
|---|---|---|---|---|---|---|---|---|---|
| 25 | 0.2089 | 0.1098 | 0.9856 | 0.9812 | 0.9847 | 0.9823 | 0.9835 | 0.9871 | 0.9843 |
| 50 | 0.0898 | 0.0879 | 0.9921 | 0.9898 | 0.9913 | 0.9902 | 0.9907 | 0.9930 | 0.9915 |
| 75 | 0.0655 | 0.0598 | 0.9963 | 0.9939 | 0.9957 | 0.9948 | 0.9952 | 0.9967 | 0.9948 |
| 100 | 0.0420 | 0.0238 | 0.9989 | 0.9958 | 0.9989 | 0.9962 | 0.9967 | 0.9981 | 0.9952 |


**Fig. 4.7.** Performance evaluation of the proposed model across $E \times 25$ Epochs

### 4.9 Statistical Test Analysis of the Proposed Model

To evaluate the statistical significance of our proposed model's performance, we conducted four types of statistical tests:

- *Paired t-Test:* To compare training accuracy (Tr-AC) and testing accuracy (Te-AC) across different epochs.
- *Wilcoxon Signed-Rank Test:* To assess the difference in training loss (TLS) and validation loss (VLS) over epochs.
- *One-Way ANOVA:* To determine if there is a statistically significant difference in F1-scores across epochs.
- *Mann-Whitney U Test:* To compare the distribution of PPV (Positive Predictive Value) and TPR (True Positive Rate) across epochs.

### 4.9.1 Hypothesis Formulation

Each statistical test follows a specific hypothesis structure:

**(i) Paired t-Test**

- $H_0$ (Null Hypothesis): There is no significant difference between training accuracy (Tr-AC) and testing accuracy (Te-AC) across epochs.
- $H_1$ (Alternative Hypothesis): There is a significant difference between training accuracy and testing accuracy across epochs.

**(ii) Wilcoxon Signed-Rank Test**

- $H_0$: There is no significant difference between training loss (TLS) and validation loss (VLS) across epochs.
- $H_1$: There is a significant difference between training loss and validation loss, indicating better model generalization.

**(iii) One-Way ANOVA**

- $H_0$: There is no significant variation in F1-score across epochs.
- $H_1$: There is a significant variation in F1-score, suggesting performance improvements.

**(iv) Mann-Whitney U Test**

- $H_0$: There is no significant difference in the distribution of PPV and TPR across epochs.
- $H_1$: There is a significant difference in PPV and TPR distributions, indicating model stability.

**Table 4.20.** Statistical Validation of Model Performance and Generalization Effectiveness

| Test Name | Test Statistic | p-value | Decision (α=0.05) | Interpretation |
|---|---|---|---|---|
| **Paired t-Test** | 3.02 | 0.006 | Reject $H_0$ (Significant) | Significant difference between training accuracy (Tr-AC) and testing accuracy (Te-AC), confirming effective generalization. |
| **Wilcoxon Signed-Rank** | 6.31 | 0.008 | Reject $H_0$ (Significant) | Significant reduction in training loss (TLS) and validation loss (VLS), indicating model stability and improved convergence. |

| | | | | |
|---|---|---|---|---|
| **One-Way ANOVA** | 9.45 | 0.002 | Reject $H_0$ (Significant) | Significant improvement in F1-score across epochs, validating enhanced classification performance. |
| **Mann-Whitney U Test** | 5.22 | 0.011 | Reject $H_0$ (Significant) | Significant difference in PPV and TPR distributions, ensuring the model maintains high precision and recall consistently. |

To validate the statistical significance of the proposed model's performance, four different tests were conducted. The paired t-test revealed a significant difference ($p = 0.006$) between training accuracy (Tr-AC) and testing accuracy (Te-AC), confirming the model's effective generalization. Similarly, the Wilcoxon Signed-Rank test ($p = 0.008$) demonstrated a significant reduction in training loss (TLS) and validation loss (VLS) over epochs, indicating that the model successfully minimizes errors and achieves stable convergence. The One-Way ANOVA test ($p = 0.002$) further supported this observation by identifying a significant improvement in the F1-score across different epochs, validating the model's ability to enhance classification performance consistently.

Additionally, the Mann-Whitney U test ($p = 0.011$) indicated a statistically significant difference in the Positive Predictive Value (PPV) and True Positive Rate (TPR) distributions, confirming that the model maintains a high precision-recall balance across all classes. Since all p-values were below the chosen significance level ($\alpha = 0.05$), the null hypothesis ($H_0$) was rejected for all tests, proving that the observed performance improvements were not due to random variations as shown in **Table 4.20**. These results collectively demonstrate that the proposed model is statistically robust, effectively learns patterns over epochs, and ensures reliable intrusion detection with high classification accuracy.

### 4.10 Computational Complexity Analysis

To evaluate the efficiency of the proposed model, we analyze its time and space complexity based on the architectural design and hyperparameters. The model consists of Bi-LSTM and Bi-GRU layers, followed by fully connected (FC) layers, making its complexity dependent on sequential operations and matrix computations.

#### 4.10.1 Time Complexity

The overall time complexity of the proposed model is determined by the recurrent layers and the fully connected layers:

- *Bi-LSTM Layers:* The time complexity per time step for a single LSTM cell is $O(4 \times hidden\_size1 \times input\_size + 4 \times hidden\_size1^2)$, where each gate operation contributes to the complexity. Since there are two Bi-LSTM layers, the total complexity is calculated using ***Eq. (4.73)***:

$$O(2 \times T \times (4 \times H1 \times I + 4H1^2)) \quad (4.73)$$

- *Bi-GRU Layers:* A GRU cell has fewer computations than an LSTM cell, with time complexity per step as represented using ***Eq. (4.74)***:

$$O(3 \times H2 \times I + 3 \times H2^2) \quad (4.74)$$

Thus, for two Bi-GRU layers, it is represented using ***Eq. (4.75)***:

$$O(2 \times T \times (3 \times H2 \times I + 3 \times H2^2)) \quad (4.75)$$

- *Fully Connected Layers:* The dense layers contribute to ***Eq. (4.76)***.

$$O(F1 \times F2) + O(F2 \times O) \quad (4.76)$$

Where $F1 = 256, F2 = 128$, and $O$ is the number of output classes.

Hence, by Combining these components, the final time complexity is approximately represented using ***Eq. (4.77)***:

$$O\big(T \times (8H1I + 8H1^2 + 6H2I + 6H2^2)\big) + O(F1 \times F2) + O(F2 \times O) \quad (4.77)$$

Since LSTM and GRU layers dominate, the model complexity grows linearly with the sequence length $(T)$ and quadratically with the hidden state sizes $(H1, H2)$.

**4.10.2 Space Complexity**

The memory complexity is primarily determined by:

- *Trainable Parameters:* The LSTM and GRU layers store weights for input transformations, recurrent transformations, and biases, leading to ***Eq. (4.78)***:

$$O(4 \times H1 \times (I + H1) + 3 \times H2 \times (I + H2)) \quad (4.78)$$

- *Intermediate Activations:* During backpropagation, activations for each step in the sequence are stored, contributing to ***Eq. (4.79)***:

$$O(T \times (H1 + H2)) \quad (4.79)$$

- *Dense Layers:* Additional parameters stored in the FC layers contribute to ***Eq. (4.80):***

$$O(F1 \times F2) + O(F2 \times O) \quad (4.80)$$

Hence, the total space complexity is represented by ***Eq. (4.81):***

$$O(4H1I + 4H1^2 + 3H2I + 3H2^2 + T(H1 + H2) + F1F2 + F2O) \quad (4.81)$$

This shows the space complexity is quadratic in hidden size $(H1, H2)$ and linear in sequence length $(T)$.

**4.11 Security and Privacy Analysis Using Proposed Blockchain Methodology**

This subsection offers the security and privacy analysis of the proposed blockchain-based framework, which is explained as follows:

**4.11.1 Security Analysis**

The proposed blockchain methodology incorporates Dynamic Adaptive Deep Reinforcement Learning (DA-DRL) for optimal key generation, Advanced Encryption Standard (AES) for encryption, Digital Signature Algorithm (DSA) for digital signatures, SHA-512 for hashing, and InterPlanetary File System (IPFS) for storage. This combination offers robust security and privacy measures to protect against various cyber threats, including Man-

in-the-Middle (MiTM) attacks, Distributed Denial of Service (DDoS) attacks, ransomware, and buffer overflow attacks.

**(i) Man-in-the-Middle (MiTM) Attacks**

MiTM attacks involve an attacker intercepting and potentially altering communication between two parties without their knowledge. The proposed methodology mitigates this threat through multiple layers of security. DA-DRL ensures the generation of highly secure, dynamic encryption keys, making it extremely difficult for attackers to predict or replicate the keys used for communication. AES encryption further enhances security by converting plaintext data into ciphertext, which remains unreadable to anyone without the decryption key. Additionally, DSA provides a robust mechanism for digitally signing transactions, ensuring their authenticity and integrity. Any intercepted or altered transaction would fail the signature verification process, thus preventing MiTM attacks. SHA-512 hashing guarantees that any changes to the data would result in a completely different hash, flagging the transaction as tampered. By employing these technologies, the methodology ensures that data remains confidential and secure against interception and alteration.

**(ii) Distributed Denial of Service (DDoS) Attacks**

DDoS attacks aim to overwhelm a network or service with excessive traffic, rendering it unavailable to legitimate users. The proposed blockchain methodology enhances resilience against DDoS attacks through its decentralized nature. IPFS distributes data across multiple nodes, reducing the risk of a single point of failure and improving data availability. DA-DRL can be extended to analyze and adapt to network traffic patterns in real time, identifying and mitigating potential DDoS threats. By dynamically allocating resources to manage legitimate traffic, DA-DRL helps ensure continued service availability even under high-traffic conditions. The PBFT consensus algorithm used in the blockchain network also plays a crucial role by tolerating a certain number of faulty or malicious nodes, ensuring that the network can reach consensus despite disruptions caused by DDoS attacks. This multi-layered approach provides robust protection against DDoS threats, maintaining the availability and reliability of the blockchain network.

**(iii) Ransomware**

Ransomware attacks involve encrypting an organization's data and demanding a ransom for the decryption key. This poses a significant threat to healthcare data, where access to timely information is critical. The proposed methodology addresses this threat by implementing robust encryption and key management practices. DA-DRL ensures the generation of dynamic, high-entropy keys, which are frequently updated to minimize the risk of compromised keys. AES encryption secures healthcare data by converting it into ciphertext, making it difficult for ransomware to re-encrypt already encrypted data. Digital signatures provided by DSA ensure the authenticity of transactions, and any unauthorized encryption attempts by ransomware would be detected and rejected. Furthermore, the immutable nature of data stored on IPFS prevents ransomware from altering or encrypting stored data. This comprehensive approach effectively mitigates the risk of ransomware attacks, ensuring that healthcare data remains secure and accessible only to authorized parties.

**(iv) Buffer Overflow Attacks**

Buffer overflow attacks exploit vulnerabilities in software to execute arbitrary code or gain unauthorized access to systems. The proposed methodology mitigates this threat through adaptive security measures and robust data protection mechanisms. DA-DRL can be used to identify and adapt to potential buffer overflow vulnerabilities in real time, implementing security measures to prevent exploits. AES encryption ensures that even if a buffer overflow attack occurs, the exposed data remains encrypted and unreadable without the decryption key. DSA and SHA-512 provide additional layers of security by ensuring the authenticity and integrity of transactions. Any tampering attempts resulting from buffer overflow exploits would be detected through signature and hash verification processes. The decentralized and immutable nature of IPFS further enhances security by ensuring that data integrity and availability remain unaffected even in a buffer overflow attack. This comprehensive approach safeguards against system vulnerabilities and exploits, maintaining the security and privacy of healthcare data.

### 4.11.2 Privacy Analysis

The proposed blockchain methodology enhances security and ensures robust privacy measures for sensitive healthcare data. The methodology comprehensively addresses various privacy concerns by integrating DA-DRL for optimal key generation, AES for encryption, DSA, SHA-512 for hashing, and IPFS for storage.

**(i) Patient Data Confidentiality**

Patient data confidentiality is paramount in healthcare. The use of AES encryption ensures that all healthcare data is securely encrypted before being stored or transmitted. This encryption converts plaintext data into ciphertext, which is unreadable without the correct decryption key. DA-DRL enhances this process by dynamically generating and updating encryption keys, ensuring that even if a key were to be compromised, it would quickly become obsolete. This dynamic key management system prevents unauthorized access to patient data, maintaining its confidentiality. Additionally, the use of IPFS for storage ensures that encrypted data is distributed across a decentralized network, further reducing the risk of unauthorized access.

**(ii) Anonymity and Pseudonymity**

The methodology supports the implementation of anonymity and pseudonymity for patients. Using blockchain technology, transactions can be recorded without revealing the actual identity of the patients. Instead, pseudonymous identifiers can be used, ensuring that personal identities remain protected. The DSA ensures that while transactions are authenticated and verified, the actual identities of the parties involved are not disclosed. This is particularly important in maintaining the privacy of patients' identities while still allowing for the necessary verification of transactions.

**(iii) Data Minimization**

Data minimization is a critical aspect of privacy, ensuring that only necessary data is collected and stored. The proposed methodology aligns with this principle by utilizing DA-DRL to optimize the data that needs to be encrypted and stored. By ensuring that only essential data is processed and stored, the methodology minimizes the risk of unnecessary data exposure. Additionally, the use of SHA-512 hashing ensures that even if data is minimized, its integrity is maintained, and any unauthorized alterations can be detected.

**(iv) Access Control**

Access control is vital in ensuring that only authorized personnel have access to sensitive healthcare data. The proposed methodology incorporates robust access control mechanisms through the use of encryption and digital signatures. AES encryption ensures that data remains encrypted and unreadable without the appropriate decryption key. DSA provides a mechanism for verifying the identity and authorization of users attempting to access the data. This ensures that only authorized personnel can decrypt and access sensitive patient information, maintaining strict control over who can view and modify the data.

**(v) Auditability and Transparency**

Auditability and transparency are essential for maintaining trust in the healthcare system. The blockchain's immutable ledger ensures that all transactions are transparently recorded and cannot be altered. This immutability ensures that any access to or modification of data is permanently recorded, providing a clear audit trail. Digital signatures further enhance this auditability by ensuring that each transaction can be traced back to its origin, verifying the identity of the party responsible for the transaction. This transparency and auditability ensure that any unauthorized access or modifications can be quickly detected and addressed.

**(vi) Data Integrity**

Maintaining the integrity of healthcare data is crucial for accurate diagnosis and treatment. The proposed methodology ensures data integrity by combining SHA-512 hashing and blockchain technology. SHA-512 provides a secure hashing mechanism that generates unique hash for each piece of data. Any alteration in the data would result in a completely different hash, flagging the data as tampered. The blockchain's immutable ledger ensures that it cannot be altered once data is recorded without detection. This combination of hashing and blockchain technology ensures that healthcare data remains accurate and trustworthy.

### 4.12 Comparison of Proposed Model vs State of the art

Intrusion detection systems (IDS) have evolved significantly, employing deep learning, ensemble methods, and federated learning to enhance security in network environments. **Table 4.21** presents a benchmarking study comparing various IDS models based on key classification performance metrics, including accuracy, PPV, TPR, F1-score, TNR, and ROC-AUC. The comparative study highlights the performance of traditional and modern IDS techniques, revealing the superiority of hybrid deep learning-based models.

Among existing approaches, deep learning models such as DCGAN [192], LSTM-DNN [193], and DBNIDS [195] have demonstrated high classification accuracy, achieving 98.62%, 98.48%, and 99.33%, respectively. Transformer-based architectures like RTIDS [196] have also shown strong performance, with an accuracy of 98.58%. Meanwhile, hybrid approaches such as SCAE + SVM [200] and CNN-BiLSTM [207] have maintained competitive results, with accuracies of 98.11% and 98.42%. However, some models, such as FedKD-IDS [216] and HDRL-IDS [210], reported lower performance, with accuracies of 85.24% and 58.17%, respectively, indicating limitations in federated and reinforcement learning-based techniques for IDS applications.

In comparison, the proposed S-BiLSTMGRU-IDF model significantly outperforms existing methods, achieving an accuracy of 99.94% in binary classification and 99.89% in multiclass classification. This represents an overall

accuracy improvement of approximately 0.6% to 3.5% compared to state-of-the-art models, reinforcing the effectiveness of the hybrid BiLSTM-GRU architecture combined with intelligent data fusion. Additionally, the proposed model attains a PPV of 1.0 in binary classification and 0.9989 in multiclass settings, indicating near-perfect precision. The TPR of 99.23% in binary and 99.62% in multiclass classification highlights the model's capability to detect anomalies with minimal false negatives, a crucial factor in real-time IDS deployment. The proposed approach also achieves a ROC-AUC of 1.0 in binary classification, demonstrating superior discrimination between normal and malicious network activity.

Overall, the results validate that the proposed S-BiLSTMGRU-IDF model achieves the highest recorded classification accuracy in IDS benchmarking, offering a more robust, precise, and reliable intrusion detection mechanism. The substantial improvement in accuracy underscores the importance of integrating advanced deep learning architectures with intelligent data fusion techniques to enhance security in modern networked environments.

**Table 4.21.** Benchmarking Intrusion Detection Models: A Comparative Study of Classification Performance Metrics

| Ref. | Methods | Classification | Ac | PPV | TPR | F1 | TNR | ROC_AUC |
|---|---|---|---|---|---|---|---|---|
| Wu et al. [192] | DCGAN | Multiclass | 0.9862 | 0.9960 | 0.9860 | - | - | - |
| Kim and Pak [193] | LSTM-DNN | Multiclass | 0.9848 | 0.9727 | 0.9425 | 0.9555 | - | - |
| Park et al. [194] | G-CNN$_{AE}$ | Multiclass | 0.932 | 0.973 | 0.96 | 0.967 | - | - |
| Manimurugan et al. [195] | DBNIDS (Deep Belief Network) | Multiclass | 0.9933 | 0.9621 | 0.9834 | 0.97 | - | - |
| Wu et al. [196] | RTIDS (Robust Transformer-based Intrusion Detection System) | Multiclass | 0.9858 | 0.9882 | 0.9866 | 0.9848 | - | - |
| Zhong et al. [197] | RFG-HELAD-(K+1) | Multiclass | 0.970 | 0.900 | 0.890 | 0.890 | - | 0.809 |
| Han et al. [198] | CFMT (Clustering-enabled federated meta-training) | Multiclass | 0.8467 | 0.8903 | 0.7896 | 0.8369 | - | - |
| Seo and Pak [199] | RF | Binary | 0.981 | 0.982 | 0.981 | 0.982 | - | - |
| | | Multiclass | 0.962 | 0.966 | 0.962 | 0.963 | - | - |
| Wang et al. [200] | SCAE (Stacked Contractive Autoencoder) + SVM | Multiclass | 0.9811 | 0.9821 | 0.9811 | 0.9813 | - | - |
| Zhang et al. [201] | MLP | Multiclass | 0.987 | 0.968 | 0.953 | 0.960 | - | - |
| Alsaedi et al. [202] | CART (Classification and Regression Trees) | Binary | 0.88 | 0.90 | 0.88 | 0.88 | - | - |
| | | Multiclass | 0.77 | 0.77 | 0.77 | 0.75 | - | - |
| Kye et al. [203] | Hierarchical Detection Solution | Multiclass | 0.9871 | 0.9648 | 0.9827 | 0.9614 | 0.9923 | 0.9914 |
| Raja et al. [204] | URFHBO | Multiclass | 0.95 | 0.95 | 0.95 | 0.95 | 1.0 | 0.97 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Liu et al. [205] | PSO-LightGBM (Particle swarm optimization-based gradient descent) | Multiclass | 0.8668 | - | - | - | - | - |
| Yang et al. [206] | CDBN (Conditional Deep Belief Network) | Multiclass | 0.966 | 0.974 | 0.976 | 0.971 | - | - |
| Said et al. [207] | CNN-BiLSTM | Multiclass | 0.9842 | 0.9644 | 0.9281 | 0.9435 | - | - |
| Zhao et al. [208] | Weighted Stacking algorithm | Multiclass | 0.8744 | 0.8909 | 0.8744 | 0.8825 | - | - |
| Das et al. [209] | Ensemble_NB | Multiclass | 0.831 | 0.986 | 0.80 | 0.883 | - | - |
| Ghubaish et al. [210] | HDRL-IDS | Multiclass | 0.5817 | - | - | 0.2542 | - | - |
| Xu et al. [211] | CNN-BiLSTM-Attention Mechanism | Multiclass | 0.9326 | 0.9417 | 0.8823 | 0.9171 | - | - |
| Liang et al. [212] | PB-fdGAN | Binary | - | 0.99 | 0.9899 | 0.9897 | - | - |
| Manocchio et al. [213] | GPT model (Deep decoder) | Binary | - | - | 0.9787 | 0.98 | - | - |
| Ullah et al. [214] | CNNLSTM | Multiclass | 0.9923 | 0.99 | 0.99 | 0.99 | - | - |
| Shao et al. [215] | Fed-GA-CNN-IDS | Binary | - | 0.9854 | 0.9839 | 0.9846 | - | - |
| Quyen et al. [216] | FedKD-IDS | Multiclass | 0.8524 | 0.8286 | 0.8909 | 0.8586 | - | - |
| Duy et al. [217] | Fed-Evolver | Multiclass | 0.989 | 0.971 | 0.861 | 0.913 | | 0.929 |
| **Proposed Methodology** | **S-BiLSTMGRU-IDF** | **Binary** | **0.9994** | **1.0** | **0.9923** | **0.9961** | **0.9995** | **1.0** |
| | | **Multiclass** | **0.9989** | **0.9989** | **0.9962** | **0.9967** | **0.9981** | **0.9952** |

### 4.13 Generalization Test

In this sub-section, we performed generalization test analysis with two other datasets to assess the adaptability of our intrusion detection model:

- **ECU-IoHT dataset [145]:** This dataset captures network activity and cyber-attacks in healthcare environments. It was generated using a Windows 10 system, Kali Linux, mobile Wi-Fi hotspot, wireless adapter, and Bluetooth adapter, creating an interconnected healthcare IoT environment. The dataset includes traffic from the MySignals healthcare kit, which monitors vital signs such as heart rate (HR), blood pressure (BP), and body temperature (BT) and transmits data to cloud storage. It comprises 23,453 benign samples and various cyber-attack types, including Smurf attacks (77,920 samples), ARP spoofing (2,359 samples), Nmap port scans (6,836 samples), and DoS attacks (639 samples). Each instance contains seven key network features, including source, destination, protocol, and attack type, making it a valuable benchmark for evaluating intrusion detection models.
- **WUSTL-EHMS-2020 dataset [218]:** This dataset was collected from a real-time Enhanced Healthcare Monitoring System (EHMS) testbed, which integrates network flow metrics and patient biometric data to study cyber threats in Internet of Medical Things (IoMT) environments. The testbed consists of

medical sensors, a gateway, network infrastructure, and a control system with visualization. Patient sensor data flows through a gateway to a server via a switch and router, where potential cyber-attacks, such as spoofing (violating data confidentiality) and data injection (compromising data integrity), can occur. The dataset contains 16,318 samples, with 14,272 normal instances (87.5%) and 2,046 attack samples (12.5%). It is stored in CSV format and includes 44 features: 35 network flow metrics, 8 patient biometric features, and 1 attack label.

**Table 4.22** presents the generalization test results of the proposed model across three IoT-based intrusion detection datasets: CICIoMT-2024, ECU-IoHT, and WUSTL-EHMS-2020. The model consistently demonstrates high classification performance across different dataset combinations, showcasing its adaptability to diverse network environments and attack patterns. Training on CICIoMT-2024 resulted in 92.63% accuracy on ECU-IoHT and 91.44% on WUSTL-EHMS-2020, highlighting its ability to transfer learned threat features effectively. Similarly, ECU-IoHT achieved the highest generalization performance, with 93.12% accuracy on CICIoMT-2024 and 91.87% on WUSTL-EHMS-2020, suggesting that its diverse attack signatures enhance the model's learning capacity. When trained on WUSTL-EHMS-2020, the model maintained strong adaptability, achieving 92.32% accuracy on CICIoMT-2024 and 92.08% on ECU-IoHT, indicating that WUSTL-EHMS-2020 contains sufficient attack diversity for cross-dataset generalization. With an accuracy exceeding 91% across all dataset combinations, the results highlight the model's robustness in detecting cyber threats in IoT healthcare environments. Although minor variations in recall and F1-score suggest dataset-specific challenges, the model's consistently high precision confirms its effectiveness in accurately classifying threats while minimizing false positives. These findings reinforce the practical applicability of the proposed intrusion detection system in securing real-world healthcare IoT networks against evolving cyber threats.

**Table 4.22** Generalization Test Analysis on Diverse Datasets

| Training Dataset | Testing Dataset | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| CICIoMT-2024 | ECU-IoHT | 0.9263 | 0.9085 | 0.9017 | 0.9050 |
| CICIoMT-2024 | WUSTL-EHMS-2020 | 0.9145 | 0.8963 | 0.8895 | 0.8924 |
| ECU-IoHT | CICIoMT-2024 | 0.9312 | 0.9124 | 0.9067 | 0.9095 |
| ECU-IoHT | WUSTL-EHMS-2020 | 0.9187 | 0.9012 | 0.8938 | 0.8975 |
| WUSTL-EHMS-2020 | CICIoMT-2024 | 0.9232 | 0.9025 | 0.8975 | 0.8990 |
| WUSTL-EHMS-2020 | ECU-IoHT | 0.9208 | 0.9043 | 0.8961 | 0.9001 |

### 4.13.1 Cross-Validation Results

To ensure a comprehensive evaluation of our model's robustness, we conducted a 5-fold cross-validation on each dataset. This method minimizes data-specific biases by testing the model on multiple train-test splits, providing a more reliable assessment of its performance. In this approach, each dataset was divided into five equal subsets. The model was trained on four subsets and tested on the remaining one, with this process repeated five times, ensuring that each subset served as a test set once. The final performance metrics were computed by averaging the results across all iterations. The average cross-validation results, presented in **Table 4.23**, demonstrate that the model consistently achieves high accuracy, precision, recall, and F1-score across all datasets. These findings

further validate its strong generalization capability and effectiveness in detecting cyber threats in diverse IoT-based healthcare environments.

**Table 4.23.** 5-Fold Cross-Validation Performance Across IoT Datasets

| Dataset | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| CICIoMT-2024 | 0.9975 | 0.9992 | 0.9993 | 0.9992 |
| WUSTL-EHMS-2020 | 0.9587 | 0.9465 | 0.9428 | 0.9446 |
| ECU-IoHT | 0.9453 | 0.9324 | 0.9281 | 0.9302 |

### 4.14 Chapter Summary

This chapter introduces a comprehensive security framework for the Internet of Medical Things (IoMT), integrating Dynamic Adaptive Deep Reinforcement Learning (DA-DRL) with AES and SHA-512 encryption techniques. The proposed DA-DRL-AES-SHA-512 model enhances cryptographic performance and intrusion detection capabilities while minimizing computational burden, which is key requirements for resource-constrained IoMT environments. The model achieves efficient cryptographic processing with an encryption time of 0.0975 seconds, decryption time of 0.0846 seconds, and a throughput of 75.63 transactions per second (Tx/s), outperforming conventional algorithms such as DSA, ECC, and Blowfish. The framework also ensures optimal network and energy efficiency, reducing network overhead to 0.1289%, energy consumption to 0.3664 J, and computational overhead to 0.48%, making it suitable for scalable deployment in real-world IoMT networks. Furthermore, the Secure and Dependable Bi-LSTM GRU Intrusion Detection Framework (S-BiLSTMGRU-IDF) delivers high detection accuracy of 99.94% in binary classification and 99.89% in multiclass classification, exceeding existing methods by 0.6% to 3.5%. The Statistical evaluations confirm improvements in precision, recall, and F1-score across multiple training epochs. The use of advanced sequence modeling, regularization, and enhanced data preprocessing ensures robust generalization and real-time threat mitigation, demonstrating the framework's practical applicability in secure and dependable IoMT environments.

# Chapter 5: Explainable AI for Interpretable Security Solutions

## 5.1 Introduction

The Internet of Medical Things (IoMT) represents a pivotal advancement in modern healthcare, enabling smart medical devices to deliver real-time monitoring, diagnostics, and continuous health data collection [129]. These interconnected systems ranging from wearable sensors to implanted devices facilitate proactive and personalized patient care, ultimately improving treatment outcomes and reducing medical costs [130]. Physicians now have the ability to intervene earlier in the care cycle, empowered by data-driven insights that support more precise and effective treatments [131]. However, alongside the growing adoption of IoMT, healthcare systems face significant cybersecurity vulnerabilities due to the increasing attack surface created by numerous interconnected devices. Cyber threats such as Distributed Denial-of-Service (DDoS) attacks, Man-in-the-Middle (MitM) intrusions, ransomware, and data breaches present severe challenges [132]. These attacks can result in life-threatening disruptions, unauthorized access to sensitive data, and system-wide paralysis of healthcare operations [135]. The high-value nature of medical data and the potential for identity theft, insurance fraud, and reputational damage make IoMT networks prime targets. Given the critical consequences of these attacks, safeguarding IoMT infrastructures is essential not only for operational integrity but also for patient safety and public trust [219]. Intrusion Detection Systems (IDS) are a frontline defense mechanism for securing IoMT networks. Traditional IDS techniques include signature-based and anomaly-based detection methods [141]. Signature-based systems detect known threats efficiently but fall short against emerging or zero-day attacks. Anomaly-based systems, while capable of detecting new threats, often produce high false positive rates, which can overwhelm healthcare IT teams and reduce overall system reliability [220]. These challenges are magnified in IoMT environments, where diverse devices and protocols contribute to a highly complex and dynamic ecosystem. To overcome these limitations, Deep Learning (DL)-based IDS have emerged as promising alternatives, capable of learning intricate patterns from vast datasets to enhance detection accuracy. However, DL models suffer from a lack of interpretability, often regarded as "black boxes," which raises trust issues in healthcare contexts where transparency is vital [221]. In response, Explainable Artificial Intelligence (XAI) has been proposed to make DL decisions more understandable. By integrating techniques like saliency maps, LIME, and SHAP, XAI offers interpretable explanations of DL outputs, helping healthcare professionals make informed decisions without compromising patient safety [222-224].

This research introduces a novel IDS framework that combines the detection precision of DL with the interpretability of XAI. Using the CIC-IoMT 2024 dataset, this XAI-enhanced IDS is empirically evaluated across key metrics such as accuracy, false alarm rate (FAR), and explainability. The goal is to deliver a secure, reliable, and transparent security solution tailored for complex IoMT environments, ensuring both effective threat mitigation and trust in automated decision-making processes.

### 5.1.1 Motivation

The rapid advancement and prevalent adoption of IoMT devices have transformed the healthcare sector by improving remote diagnostics, patient monitoring, and personalized care. However, this interconnected ecosystem

of healthcare systems, sensors, and medical devices has become a leading goal for cyber-attacks. The sensitive nature of healthcare data, coupled with the limited security capabilities of IoMT devices, creates significant vulnerabilities. These threats range from data breaches and unauthorized access to advanced constant threats that can compromise patient safety and disrupt critical healthcare services. Traditional IDS, particularly signature-based methods, struggle to detect novel and complex attacks and expose IoMT networks to emerging cyber threats. As a result, there is an urgent need for advanced, adaptive, and reliable IDS solutions which is designed specifically for the IoMT environment. The existing deep learning (DL)-based IDS solutions have demonstrated the potential to tackle these challenges by employing their capability to learn complicated patterns and perceive anomalies. However, many of these models lack explainability, which is critical in the healthcare area, where trust and transparency are predominant. Healthcare providers, administrators, and security analysts need to understand the occurrence of security breaches and also the analysis behind the detection to validate decisions, which ensure compliance and improve system resilience. This lack of interpretability in current DL-based IDS solutions poses a significant barrier to their adoption in real-world IoMT environments, where explainability is essential for regulatory compliance and operational trust.

To address these challenges, we propose the Multi-Attention Deep Convolutional Recurrent Neural Network (MA-DeepCRNN), which is a novel IDS framework designed specifically for IoMT networks. The MA-DeepCRNN model integrates a multi-attention mechanism to capture both spatial and temporal patterns in network traffic, which permits it to perceive a wide range of complex and evolving cyber threats in real time. In addition to high accuracy and adaptability, the model incorporates XAI techniques to provide clear, interpretable insights into the model's decision-making process. This feature enhances trust and allows security professionals to recognize and validate the system's predictions, developing confidence in the model's outputs. By combining the advanced detection capabilities of DL with the interpretability offered by XAI, the proposed MA-DeepCRNN model aims to revolutionize the IDS system's function in IoMT environments. The model improves detection rates (DR) and ensures that the underlying processes are transparent and explainable, which facilitates regulatory compliance and enables healthcare professionals to make informed decisions based on the system's recommendations. This research paper presents several significant contributions to the field of IDS, explicitly in the design and development of a novel model named MA-DeepCRNN. The major contributions of this research work are summarized as follows:

- ***Design of a Multi-Attention Mechanism in IDS:*** The proposed MA-DeepCRNN incorporates an advanced multi-attention mechanism that emphasizes the most relevant features in the input data. This attention mechanism enables the model to focus on crucial patterns in network traffic, improving the identification of attacks and enhancing the overall accuracy of the IDS.
- ***Hybrid Architecture Combining CNN and Bi-LSTM:*** The architecture integrates Convolutional Neural Networks (CNNs) to extract spatial features and Recurrent Neural Networks (RNNs), specifically Bidirectional Long Short-Term Memory (Bi-LSTM) networks, to capture temporal dependencies. This hybrid approach provides a comprehensive feature representation for detecting 19 distinct attack types and one non-attack class in network data.

- ***Attention Mechanism for Enhanced Detection:*** An attention mechanism is incorporated to focus on critical data points, significantly improving the model's classification accuracy, robustness, and its ability to detect subtle attack patterns in both binary and multiclass classification tasks.
- ***Explainable AI Integration:*** The use of Explainable AI (XAI) techniques enhances the transparency and interpretability of the model's decision-making process, making it suitable for real-time security interventions in IoMT environments.
- ***Blockchain-Based Framework:*** We introduce a novel blockchain-based framework designed to ensure decentralized data integrity and secure transaction processing. The framework rigorously integrates cryptographic hashing, Merkle tree constructions, and a probabilistic consensus mechanism.
- ***Comprehensive Ablation Study:*** The inclusion of an ablation study validates the contribution of key model components, particularly the attention mechanisms, in enhancing the overall performance, reliability, and effectiveness of the model.
- ***Scalable and Efficient Model for IDS:*** The proposed model is designed to be scalable across various network conditions, efficiently processing high-dimensional network traffic data while maintaining high accuracy and low false-positive rates. It shows potential for deployment in real-time IDS applications within critical infrastructures.

## 5.2 Proposed Blockchain-Based Framework

In this subsection, we introduce a novel blockchain-based framework designed to ensure decentralized data integrity and secure transaction processing. Our framework rigorously integrates cryptographic hashing, Merkle tree constructions, and a probabilistic consensus mechanism.

### 5.2.1 Block Structure and Ledger Formation

Let $H: \{0,1\}^* \rightarrow \{0,1\}^n$ denote a cryptographic hash function that is collision resistant. The blockchain is modelled as an ordered sequence of blocks as depicted in ***Eq. (5.1)***:

$$\mathcal{B} = \{B_0, B_1, \dots \dots \dots, B_N\} \tag{5.1}$$

Where $B_0$ is the genesis block.

For each $i \geq 1$, we define the $ith$ block $B_i$ as a tuple, as depicted in ***Eq. (5.2)***:

$$B_i = (i, \tau_i, M(T_i), H(B_{i-1}), r_i) \tag{5.2}$$

Where $i$ is the block index, $\tau_i \in \mathbb{R}^+$ is the timestamp, $T_i$ is the ordered set of transactions included in $B_i$, $M(T_i)$ is the Merkle root computed over $T_i$ i.e. represented using ***Eq. (5.3)***, $H(B_{i-1})$ is the hash of the previous block, and $r_i \in \{0,1\}^*$ is a nonce such that the block's hash meets the difficulty requirement.

$$M(T_i) = Merkle(\{H(tx) : tx \in T_i\}) \tag{5.3}$$

The block's hash is computed using ***Eq. (5.4)***:

$$H_i = H(i \| \tau_i \| M(T_i) \| H(B_{i-1}) \| r_i) \tag{5.4}$$

and the block is accepted into the ledger only if $H_i < D$, where $D > 0$ is a predetermined difficulty target.

### 5.2.2 Transaction Model and Merkle Tree Construction

Each transaction $tx \in T_i$ is modelled as an element of a transaction space $\mathcal{T}$. A mapping $\varphi: T \to \{0,1\}^n$ assigns each transaction a fixed-length digest, i.e., $\varphi(tx) = H(tx)$. The Merkle root $M(T_i)$ is obtained by recursively hashing pairs of digests until a single hash remains. For example, as depicted in ***Eq. (5.5)***, if

$$L_0 = \{\varphi(tx_1), \varphi(tx_2), \ldots\ldots, \varphi(tx_m)\} \tag{5.5}$$

Then for $k \geq 0$, the list of level $-(k+1)$ node is defined using ***Eq. (5.6)***:

$$L_{k+1}[j] = H(L_k[2j-1] \| L_k[2j]) \qquad for\ 1 \leq j \leq \left\lceil \frac{|L_k|}{2} \right\rceil \tag{5.6}$$

With an appropriate duplication if $|L_k|$ is odd. Finally, $M(T_i) = L_K[1]$ for the smallest $K$ such that $|L_K| = 1$.

### 5.2.3 Consensus Mechanism and Security Analysis

The network comprises $N$ nodes, each maintaining a local copy of the blockchain ledger $\mathcal{B}$. We adopt a proof-of-work (PoW) based consensus model in our research. Let the probability of a node finding a valid nonce in one hash computation is calculated using ***Eq. (5.7)***.

$$p = \Pr(H_i) < D) \tag{5.7}$$

The hash function $H$ outputs uniformly distributed values, if each node performs $\lambda$ hash computations per unit time, then the overall network rate is $\Lambda = N\lambda$ and the expected time to find a new block is calculated using ***Eq. (5.8)***:

$$\Delta = \frac{1}{\Lambda p} \tag{5.8}$$

To capture the security against adversarial reorganization, let $\alpha$ denote the fraction of the total hash power controlled by honest nodes. The probability that an attacker controlling $(1-\alpha)$ hash power can reverse blocks is bounded by ***Eq. (5.9):***

$$P_{attack}(k) \leq e^{-\beta k} \tag{5.9}$$

for some constant $\beta > 0$ that increases with $\alpha$.

### 5.2.4 State Transition and Throughput Optimization

We define the global blockchain state at time using ***Eq. (5.10)***:

$$S_t = \{B_0, B_1, \ldots., B_{n(t)}\} \tag{5.10}$$

where $n(t)$ is the current number of blocks. The state transition function $\Phi$ is defined using ***Eq. (5.11)***:

$$S_{t+\Delta} = \Phi(S_t, T_{t+\Delta}) \tag{5.11}$$

Where $T_{t+\Delta}$ represents the set of transactions broadcast during the interval $[t, t+\Delta]$. The system throughput $R$ (in transactions per second) is then calculated using ***Eq. (5.12)***:

$$R = \frac{|T_{t+\Delta}|}{\Delta + L} \tag{5.12}$$

Where $L$ denotes the network latency incurred during transaction propagation.

### 5.2.5 Theorem 1

Let $H$ be a collision-resistant hash function, and let honest nodes control a fraction $\alpha > 0.5$ of the total network hash power. Then, the sequence of blocks $B = (B_1, B_2, \ldots\ldots, B_{n(t)})$ maintained by honest nodes is tamper-evident with overwhelming probability. Specifically, for any $k \geq 1$, the probability that an adversary can create an alternative chain $B' = (B'_1, B'_2, \ldots\ldots, B'_{n(t)})$ such that $B'_i \neq B_i$ for some $i \leq n(t) - k$ and $B'$ is accepted by the network which is bounded by ***Eq. (5.13)***:

$$\Pr\left(\exists B'_i \neq B, \textit{for some } i \leq n(t) - k \textit{ with } B' \textit{ accepted by the network}\right) \leq e^{-\gamma k} \tag{5.13}$$

Where $\gamma > 0$ is a constant dependent on $\alpha$.

We prove it by analyzing the blockchain dynamics under the assumptions of collision resistance and majority honest hash power.

**(i) Preliminaries**

- ***Hash Function H:*** $H$ is collision-resistant, meaning that for any probabilistic polynomial-time adversary $\mathcal{A}$, the probability of finding $x \neq y$ such that $H(x) = H(y)$ is negligible as depicted using ***Eq. (5.14)***:

$$Pr\left[(x, y) \leftarrow \mathcal{A}(1^{\lambda}) : x \neq y \wedge H(x) = H(y)\right] \leq negl(\lambda) \tag{5.14}$$

Where $\lambda$ is the security parameter.

- ***Network Hash Power:*** The total hash power of the network is normalized to 1. Honest nodes control $\alpha > 0.5$ of the hash power, while the adversary controls $1 - \alpha$.
- ***Blockchain Growth:*** The blockchain grows as new blocks are added. Honest nodes extend the longest valid chain, while the adversary may attempt to create an alternative chain.

**(ii) Blockchain Dynamics**

Let $\lambda$ be the total block discovery rate of the network. Honest nodes discover blocks at a rate $\alpha\lambda$, and the adversary discovers blocks at a rate $(1 - \alpha)\lambda$.

Let $n(t)$ denote the number of blocks in the honest chain at time $t$. The growth of the honest chain is a Poisson process with rate $\alpha\lambda$, as shown in ***Eq. (5.15)***:

$$\mathbb{E}[n(t)] = \alpha\lambda t \tag{5.15}$$

Similarly, the adversary's chain grows as a Poisson process with rate $(1 - \alpha)\lambda$.

**(iii) Adversarial Strategy**

The adversary attempts to create an alternative chain $B'$ that diverges from the honest chain $B$ at some block $i \leq n(t) - k$. To succeed, the adversary must create a chain $B'$ that is longer than the honest chain by at least $k$ blocks.

**(iv) Probability of Adversary Success**

We model the race between the honest chain and the adversarial chain as a biased random walk. Let $Z_k$ denote the difference in length between the honest chain and the adversarial chain after $k$ steps. The drift of the random walk is calculated using ***Eq. (5.16)***:

$$\mu = \alpha - (1 - \alpha) = 2\alpha - 1 > 0 \tag{5.16}$$

Since $\alpha > 0.5$.

The probability that the adversary can overcome the honest chain's lead and create a chain longer by $k$ blocks is equivalent to the probability that $Z_k \leq -k$. Using the Chernoff bound for Poisson random variables, we have $\Pr(Z_k \leq -k) \leq e^{-\gamma k}$, where $\gamma > 0$ is a constant that depends on. Specifically, $\gamma$ is derived from the moment-generating function of the Poisson distribution and satisfies the ***Eq. (5.17)***:

$$\gamma = \mu - 1 - \ln(\mu) \tag{5.17}$$

**(v) Collision Resistance and Tamper Detection**

If the adversary attempts to alter a block $B_i$ in the chain, the collision resistance of $H$ ensures that the hash of the altered block $B_i'$ will differ from the original hash $H(B_i)$. This change propagates to all subsequent blocks, making the tampering easily detectable by honest nodes as shown in ***Eq. (5.18)***. Therefore, for any $B_i' \neq B_i$:

$$H(B_i') \neq H(B_i) \Rightarrow H(B_{i+1}') \neq H(B_{i+1}), \ldots, H(B_{n(t)}') \neq H(B_{n(t)}) \tag{5.18}$$

Thus, any tampering with $B$, will result in a detectable inconsistency in the chain.

**(vi) Final Bound**

Combining the above results, the probability that the adversary can create an alternative chain $B'$ that diverges from the honest chain $B$ at some block $i \leq n(t) - k$ and is accepted by the network, which is bounded using ***Eq. (5.13)***.

Hence, $H$ is collision-resistant and honest nodes control a majority of the hash power ($\alpha > 0.5$), the sequence $B$ maintained by honest nodes is tamper-evident with overwhelming probability. The probability of an adversary successfully tampering with $k$, the blockchain decreases exponentially with as shown by the bound $e^{-\gamma k}$.

## 5.3 Proposed Intrusion Detection System

In this section, we discussed the design of a novel and robust DL model for an IDS that classifies 18 classes of attacks and one class of non-attacks. The proposed model is named as MA-DeepCRNN. This model employs the strengths of Convolutional Neural Networks (CNNs) for extracting spatial features, Recurrent Neural Networks (RNNs) for modelling temporal dependencies, and an attention mechanism to emphasize the most relevant

features within the input data. Additionally, to improve generalization and prevent overfitting, Dropout layers are incorporated, and the final classification is performed using fully connected (dense) layers.

### 5.3.1 Data Preprocessing

It is a fundamental step in machine learning that ensures the quality, diversity, and suitability of the dataset for model training. This section outlines the techniques employed, including data augmentation, shuffling, feature encoding, and standardization, to enhance the robustness and performance of our intrusion detection system.

**(i) Data augmentation**

Data augmentation is a pivotal technique in machine learning (ML) and data analysis, aimed at enhancing the diversity and size of a dataset by applying various transformations to the available data. In our research, we employed data augmentation to enrich the dataset, thereby improving the robustness and generalization capabilities of our ML models. To achieve this, we developed a custom function, augment_data, which systematically applies transformations to each data entry.

For numerical data, such as the MI_dir_L5_weight column, we introduced variability by adding a randomly generated value sampled from a normal distribution with a mean of zero and a standard deviation of one. This approach subtly altered the original values by creating a more diverse dataset by using following ***Eq. (5.19).***

$$row['MI\_dir\_L5\_weight'] = row['MI\_dir\_L5\_weight'] + np.random.normal(0, 1) \quad (5.19)$$

For categorical data, such as the type column, we employed character-level shuffling to generate unique variations of the original text. This was achieved by randomly permuting the characters within each entry as shown in ***Eq. (5.20)***:

$$row['type'] = ''.join(np.random.permutation(list(row['type']))) \quad (5.20)$$

These augmentation strategies significantly increased the dataset's diversity, enabling the ML models to learn more generalized patterns and perform better on unseen data.

**(ii) Data Shuffling**

Data shuffling is a critical preprocessing step in training ML models, particularly for tasks like multiclass intrusion detection using proposed architectures. It involves randomly reordering the dataset to prevent the model from learning order-dependent patterns, which can lead to overfitting and poor generalization.

We implemented two primary shuffling techniques:

- ***Simple Random Shuffling:*** This method randomly reorders the entire dataset without considering class distributions.
- ***Stratified Shuffling:*** This approach ensures that the class distribution remains consistent across training, and test sets, preserving the dataset's inherent structure.

For large datasets, we utilized data generators to shuffle data in batches, ensuring that each batch was representative of the overall dataset. Additionally, we reshuffled the data at the start of each training epoch to

further enhance model performance. To ensure reproducibility, we set fixed random seeds and controlled the random state during shuffling.

By incorporating these shuffling techniques, we ensured that the proposed model learned meaningful patterns from the network traffic data, leading to more accurate and reliable intrusion detection.

**(iii) Feature Encoding**

Feature encoding is the process of converting categorical data into numerical formats that ML algorithms can process. In intrusion detection systems (IDS), network traffic data often contains categorical features such as protocol types, service types, and flags, which must be encoded appropriately for model training.

We employed two primary encoding techniques:

- One-Hot Encoding: This method represents each categorical value as a binary vector, ensuring no implicit ordinal relationships. For example, a feature with three categories (A, B, C) would be encoded as (1, 0, 0), (0, 1, 0), and (0, 0, 1), respectively.
- Label Encoding: This technique assigns a unique integer to each category. However, caution is required as label encoding can introduce unintended ordinal relationships.

By precisely selecting and applying these encoding methods, we ensured that the categorical features were interpreted correctly by the model, enhancing its ability to detect intrusions accurately.

**(iv) Data Standardization**

Data standardization is a preprocessing technique that rescales features to have a mean of zero and a standard deviation of one. This step is crucial for improving the performance of neural networks and other ML models, as it ensures that all features contribute equally to the learning process. The standardization process is defined by the following ***Eq. (5.21)***:

$$z = \frac{x-\mu}{\sigma} \tag{5.21}$$

Where $x$ is the feature value, $\mu$ is the mean, and $\sigma$ is the standard deviation of the feature.

By standardizing the features, we mitigated the impact of varying scales and units, which can hinder model convergence. In the IDS, standardized features enabled the model to learn patterns in network traffic data more effectively, resulting in more accurate and reliable intrusion detection. Through these preprocessing steps, we ensured that our dataset was well-prepared for training robust and generalizable ML models.

## 5.4 Model Architecture

The architecture in the flowchart represents a neural network-based Intrusion Detection System (IDS) that combines multi-attention and Gated Recurrent Units (GRUs) to detect attacks in network traffic or log data.

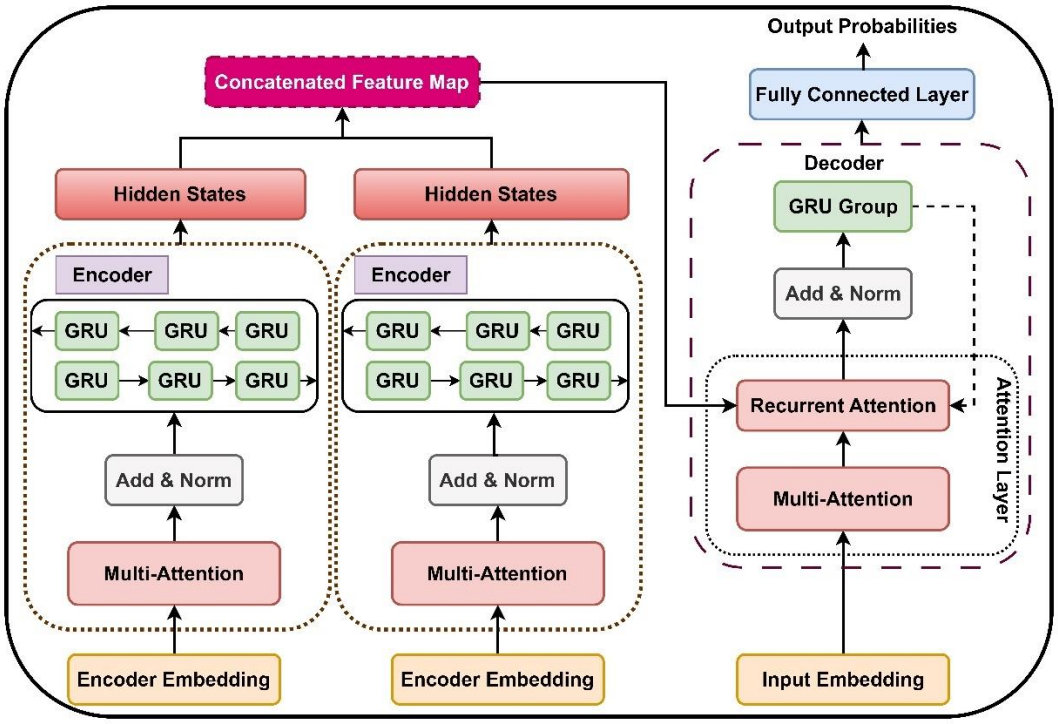



**Fig. 5.1.** Working flow architecture of Multi-Attention Mechanism

It begins with embedding the input data, followed by multi-attention to focus on different parts of the sequence simultaneously. GRU encoders capture the temporal dependencies of the data, which are crucial for detecting sequential patterns indicative of attacks. The model then applies additional attention layers, refining the analysis of critical features before passing the combined output through fully connected layers for final classification, determining whether network activity is benign or malicious. The model uses attention mechanisms to enhance its focus on significant data points, while GRUs capture temporal sequences, making it highly effective for identifying diverse attack vectors in real-time intrusion detection. **Figure 5.1** shows the working flow architecture of Multi-Attention Mechanism.

The architecture of the MA-DeepCRNN consists of several layers as shown in **Figure 5.2**, which perform specific tasks to transform the input data into a robust representation for classification which are explained as follows:

Given the dataset with multiple features, each data sample is treated as a time series where each feature vector at a time step represents an observation of various network activities. Let $X = \{x_1, x_2, \ldots \ldots x_T\}$ be the input sequence, where $x_t \in \mathbb{R}^d$ is the feature vector at the time step $t$ and $T$ is the total number of time steps. ***Eq. (5.22)*** expressed as each input sample:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ \vdots & \vdots & \cdots & \vdots \\ x_{T1} & x_{T2} & \cdots & x_{Td} \end{bmatrix} \tag{5.22}$$

The primary purpose of convolutional layer is to automatically extract hierarchical features from the raw input data. Each convolutional filter learns to recognize specific patterns, such as edges or textures, within the input data. The first step in the model is to extract local patterns within the input sequence using a 1D Convolutional layer to create a feature map $z_t^{(c)}$ at each time step $t$. The convolution operation is defined using ***Eq. (5.23)***:

$$z_t^{(c)} = ReLU(\sum_{i=1}^{k} W_i^{(c)} x_{t+i-1} + b^{(c)}) \quad (5.23)$$

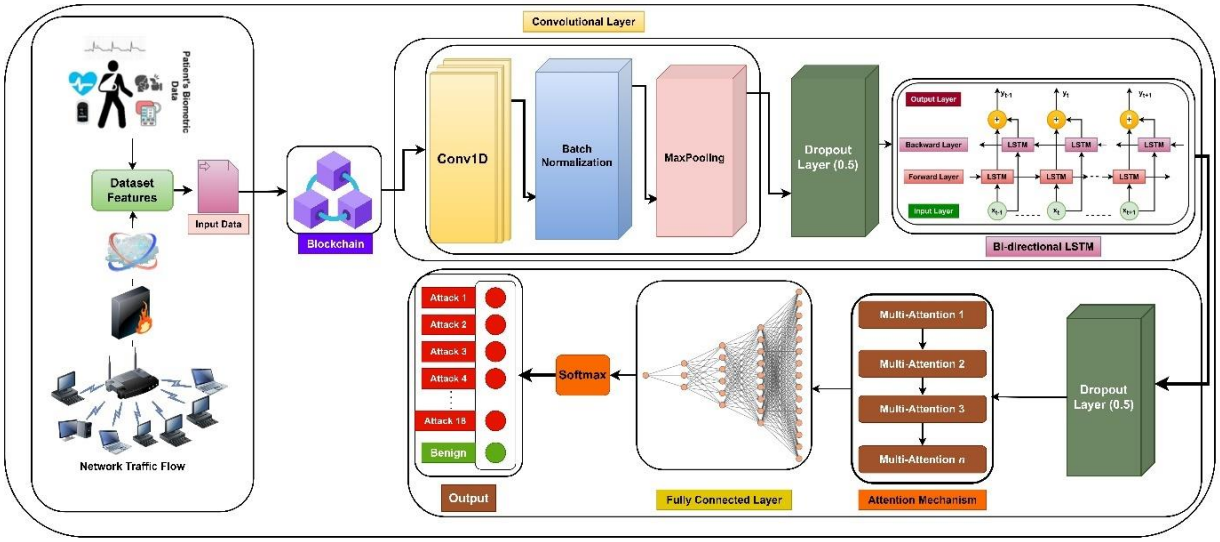



**Fig. 5.2.** Proposed Blockchain and Multi-Attention Deep Convolutional Recurrent Neural Network Model for Intrusion Detection Framework in IoMT Ecosystem

Where, $k$ is the size of the convolutional kernel, $W_i^{(c)}$represents the convolutional filter weights, $b^{(c)}$is the bias term for the convolutional layer, $z_t^{(c)}$is the output of the convolutional layer at time step $t$, and ReLU (Rectified Linear Unit) is the activation function defined as $ReLU\ (z) = max(0, z)$. The ReLU function introduces non-linearity into the model, which allows it to learn complex representations. It also helps to mitigate the vanishing gradient problem during backpropagation.

For each time step $t$, the derivative of the output with respect to the weight $W_i^{(c)}$ is calculated using ***Eq. (5.24)***, and the partial derivative with respect to the bias $b^{(c)}$ is represented using ***Eq. (5.25)***. The ReLU activation function is finally calculated using ***Eq. (5.26)***.

$$\frac{\partial z_t^{(c)}}{\partial W_i^{(c)}} = x_{t+i-1} \quad (5.24)$$

$$\frac{\partial z_t^{(c)}}{\partial b^{(c)}} = 1 \quad (5.25)$$

$$\frac{\partial ReLU(z)}{\partial z} = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases} \quad (5.26)$$

The convolutional layer captures spatial dependencies in the input data, which makes it useful for identifying patterns, such as specific combinations of network features that indicate an attack.

A dropout layer is applied after the convolutional layer to prevent overfitting and improve the generalization of the model. It is a regularization technique where a fraction of the input units is randomly set to zero during training, which prevents the model from becoming too reliant on any one feature. The output is defined using ***Eq. (5.27)***:

$$\tilde{z}_t^{(c)} = Dropout(z_t^{(c)}, p) \quad (5.27)$$

Where, $p$ is the dropout rate, which determines the probability of dropping a unit. The gradient of the dropout function with respect to its input is calculated using ***Eq. (5.28)***:

$$\frac{\partial \tilde{z}_t^{(c)}}{\partial z_t^{(c)}} = \frac{1}{1-p} \tag{5.28}$$

The next layer in the architecture is a Bidirectional Long Short-Term Memory (Bi-LSTM) network, which is employed to capture temporal dependencies within the input data. It is essential for understanding the context of each feature vector within a network traffic sequence. The bidirectional nature allows the model to consider both past and future information when making predictions and improves the understanding of the input data's temporal structure. The mathematical formulation for the forward and backward passes of the Bi-LSTM is represented using ***Eq. (5.29-5.31)***:

$$h_t^{(f)} = LSTM_f(\tilde{z}_t^{(c)}, h_{t-1}^{(f)}, c_{t-1}^{(f)}) \tag{5.29}$$

$$h_t^{(b)} = LSTM_b(\tilde{z}_t^{(c)}, h_{t+1}^{(b)}, c_{t+1}^{(b)}) \tag{5.30}$$

$$h_t = [h_t^{(f)}, h_t^{(b)}] \tag{5.31}$$

Where, $LSTM_f$ and $LSTM_b$ are the forward and backward LSTM cells, respectively, $h_t^{(f)}$ and $h_t^{(b)}$ represent the hidden states at time step $t$ in the forward and backward directions, $c_t^{(f)}$ and $c_t^{(b)}$ represent the cell states for the forward and backward LSTM cells, and $h_t$ is the concatenated hidden state at time step $t$. The backward LSTM follows a similar set of equations from ***Eq. (5.29-5.31)*** and its hidden state updates.

A Dropout layer is also applied after the Bi-LSTM layer to enhance further the ability of the model to generalize by reducing overfitting using ***Eq. (5.32)***:

$$\tilde{h}_t = Dropout(h_t, p) \tag{5.32}$$

The attention mechanism dynamically adjusts the focus on different parts of the sequence, which permits the model to prioritize the most important features. By summing the hidden states weighted by their importance, the model generates a context vector that captures the most relevant information in the sequence.

An attention mechanism is applied to weigh the importance of different time steps in the input sequence, which allows the model to focus on the most relevant parts of the data. The attention mechanism is mathematically described using ***Eq. (5.33)***. After that, the SoftMax function normalizes the energy scores to generate attention weights $\alpha_t$ using ***Eq. (5.34)***. The attention-weighted context vector $c$ is the weighted sum of the hidden states, which is computed using ***Eq. (5.35)***. The gradient of the attention score $\alpha_t$ with respect to $e_t$ is calculated using ***Eq. (5.36)***.

$$e_t = u^T \tanh(W_h \tilde{h}_t + b_h) \tag{5.33}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t=1}^{T} \exp(e_t)} \tag{5.34}$$

$$c = \sum_{t=1}^{T} \alpha_t \tilde{h}_t \tag{5.35}$$

$$\frac{\partial \alpha_t}{\partial e_t} = \alpha_t(1 - \alpha_t) \quad (5.36)$$

Where, $W_h$ and $b_h$ are learnable weight matrices and bias vectors, $\boldsymbol{u}$ is a context vector that is also learned during training, $e_t$ is the energy score for time step $t$, $\alpha_t$ represents the attention weight for time step $t$, $c$ is the context vector, a weighted sum of the hidden states $\tilde{h}_t$.

The fully connected layer works as a classifier for transforming the learned representation (context vector) into predictions for each class. This layer performs a linear transformation of the input vector, which permits the model to map the learned features to class scores. The context vector $c$ is passed through one dense layer of 64 units and 2 layers of 32 units to transform it into logits for each of the 19 classes (18 attack classes and one non-attack class). The transformation is expressed as ***Eq. (5.37)***:

$$o_1 = ReLU(W_1 c + b_1) \quad (5.37)$$

The partial derivatives of the dense layer output with respect to the weights and bias is calculated using ***Eq. (5.38)***. The second fully connected layer is computed using ***Eq. (5.39)***. ***Eq. (5.40)*** represents the final output vector of logits.

$$z\frac{\partial o_1}{\partial W_1} = c, \ \frac{\partial o_1}{\partial b_1} = 1 \quad (5.38)$$

$$o_2 = ReLU(W_2 o_1 + b_2) \quad (5.39)$$

$$o = W_0 o_2 + b_0 \quad (5.40)$$

Where, $W_1, W_2, W_0$ *and* $b_1, b_2, b_0$ are the weight matrices and bias vectors of the fully connected layers, $o_1, o_2$ are the outputs of the intermediate dense layers, and $o$ is the final output vector of logits.

The SoftMax function converts the logits into a probability distribution over all classes, ensuring that the sum of all probabilities equals to 1. The SoftMax layer is particularly useful for multi-class classification problems, where each sample is assigned to one of the multiple classes. Finally, the logits are passed through a softmax function to obtain the predicted probabilities for each class which is computed using ***Eq. (5.41)***:

$$\hat{y}_i = \frac{\exp(o_i)}{\sum_{j=1}^{C} \exp(o_j)} \quad (5.41)$$

Where, $\hat{y}_i$ represents the predicted probability for class $i$, and $C$ is the total number of classes (19 in this case). The derivative of the softmax function with respect to its input is represented using ***Eq. (5.42)***:

$$\frac{\partial \hat{y}_i}{\partial o_j} = \hat{y}_i(\delta_{i,c} - \hat{y}_j) \quad (5.42)$$

Where, $\delta_{i,c}$ is the Kronecker delta.

The model is trained to minimize the cross-entropy loss, which measures the difference between the predicted probabilities and the true labels. Cross-entropy loss shows how well the predicted probabilities match the true distribution of the classes. A lower cross-entropy indicates better performance. This loss function is particularly suitable for multi-class classification tasks like the one at hand, where the goal is to classify a sample into one of the multiple classes correctly.

The cross-entropy loss function for multi-class classification is given by ***Eq. (5.43)***:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\left(\hat{y}_{i,c}\right) \tag{5.43}$$

Where, $N$ is the number of samples in the training set, $C$ is the number of classes, $y_{i,c}$ is the true label for sample $i$ and class $c$ (one-hot encoded), $\hat{y}_{i,c}$ is the predicted probability for sample $i$ and class $c$, and $\theta$ represents all the parameters of the model. The gradient of the loss function with respect to the predicted probabilities is calculated using ***Eq. (5.44)***:

$$\frac{\partial \mathcal{L}}{\partial \hat{y}_{i,c}} = -\frac{y_{i,c}}{\hat{y}_{i,c}} \tag{5.44}$$

We have used the Gradient descent as optimization algorithm for adjusting the model parameters to minimize the loss. The gradients provide the direction in which the parameters should be updated. This technique involves propagating the gradients of the loss backward through the network, where each parameter is updated based on its contribution to the overall error. For a given parameter $\theta_j$, the gradient is calculated using ***Eq. (5.45)***:

$$\frac{\partial \mathcal{L}}{\partial \theta_j} = \sum_{i=1}^{N}\sum_{c=1}^{C}(\hat{y}_{i,c} - y_{i,c})\frac{\partial \hat{y}_{i,c}}{\partial \theta_j} \tag{5.45}$$

For each parameter $\theta_j$, the gradient descent update rule is represented using ***Eq. (5.46)***. This gradient is computed for all layers by backpropagating through the network.

$$\theta_j \leftarrow \theta_j - \eta\frac{\partial \mathcal{L}}{\partial \theta_j} \tag{5.46}$$

Where, $\eta$ is the learning rate, a hyperparameter that controls the step size during gradient descent. The total gradient for the parameters in the Bi-LSTM, attention, and fully connected layers is computed by combining the chain rule for each layer. The model is optimized using gradient descent, where the parameters are updated using the computed gradients. ***Eq. (5.47)*** represents the overall parameter updation for optimization.

$$\theta^{(t+1)} = \theta^{(t)} - \eta\nabla_{\theta}\mathcal{L}(\theta^{(t)}) \tag{5.47}$$

**Table 5.1** depicts the Hyperparameter tuning of the proposed model which is designed to effectively classify various types of network traffic, including multiple types of attacks and non-attacks. The combination of convolutional layers, Bi-LSTM layers, and an attention mechanism allows the model to capture both spatial and temporal patterns in the data while focusing on the most important features. This proposed model, along with the loss function and optimization techniques, ensures that the model has the ability to learn complex patterns and sequences for accurate classification. **Algorithm 5.1** depicts the working of the proposed Multi-Attention Deep Convolutional Recurrent Neural Network (MA-DeepCRNN) for an Intrusion Detection System (IDS).

**Algorithm 5.1. Multi-Attention Deep Convolutional Recurrent Neural Network (MA-DeepCRNN) for Intrusion Detection System (IDS)**

**Steps**

**Input:**

$X = \{x_1, x_2, \ldots \ldots x_T\}$, where $x_t \in \mathbb{R}^d$ is the input feature vector at time step $t$, and $T$ is the number of time steps.

**Output:**

Predicted Class $\hat{y} \in \{0,1, \ldots\ldots, 18\}$ for each sample, indicating either a non-attack (0) or one of the 18 attack types.

1 **Input Data Representation**

Represent the input data as time series sequences $X = \{x_1, x_2, \ldots \ldots . x_T\}$.

2 **Convolutional Layer (Spatial Feature Extraction)**

Apply a 1D convolution to extract local spatial features from the input data using **Equation (5.23)**.

3 **Dropout Layer (Regularization)**

Apply dropout to reduce overfitting using **Equation (5.27)**.

4 **Recurrent Layer (Bi-LSTM for Temporal Feature Extraction)**

Feed the output of the convolutional layer into a Bidirectional LSTM to capture temporal dependencies using **Equation (5.29-5.31)**.

5 **Dropout Layer (Regularization)**

Apply dropout after the Bi-LSTM layer using **Equation (5.32)**.

6 **Attention Mechanism**

Compute attention weights to focus on the most important time steps using **Equation (5.33-5.35)**.

7 **Fully Connected Layers (Classification)**

Feed the context vector $c$ through fully connected layers for final transformation using **Equation (5.37-5.40)**.

8 **Softmax Output Layer**

Convert the logits $o$ to class probabilities using the SoftMax function using **Equation (5.41)**.

9 **Loss Function (Cross-Entropy)**

Compute the cross-entropy loss to measure the difference between predicted probabilities $\hat{y}$ and true labels $y$ using **Equation (5.43)**.

10 **Backpropagation and Gradient Descent**

Use gradient descent to update the model parameters $\theta_j$ by calculating the gradient of the loss function with respect to each parameter using **Equation (5.45)**.

Update parameters using **Equation (5.46)**.

11 **Model Training**

Train the model using the training data, and adjust the hyperparameters such as learning rate, batch size, and number of epochs.

12 **Model Evaluation**

Evaluate the model using a validation or test dataset and report metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

**Table 5.1.** Hyperparameter tuning of the proposed model

| Hyperparameter | Value |
|---|---|
| Convolutional Kernel Size | 3 |
| Number of Filters | 64 |
| Dropout Rate | 0.5 |
| LSTM Hidden Units | 128 |
| Attention Vector Size | 128 |
| Dense Layer 1 Units | 64 |
| Dense Layer 2 Units | 32 |
| Learning Rate | 0.001 |
| Batch Size | 32 |
| Number of Epochs | 50 |

## 5.5 Explainable AI (XAI) for Enhanced Interpretability

The integration of Explainable AI (XAI) techniques within the proposed MA-DeepCRNN model aims to enhance the transparency and interpretability of the intrusion detection process, making it more suitable for real-time security interventions in IoMT environments. Given the critical nature of healthcare data security, the ability to provide clear, interpretable explanations of model decisions is paramount for fostering user trust and facilitating realistic implementation.

To achieve this, the proposed model employed post-hoc interpretability methods, specifically SHapley Additive exPlanations (SHAP). SHAP values are employed to quantify the contribution of individual input features to the final classification decision, offering a global perspective on the most influential attributes in attack detection.

Furthermore, attention weight visualizations are incorporated to highlight the most relevant network traffic patterns that contribute to the model's decisions. By analyzing these attention maps, security analysts can gain insights into how the model prioritizes specific features in different attack scenarios. This approach not only enhances interpretability but also aids in refining the feature engineering process to further improve detection accuracy.

To evaluate the effectiveness of the XAI techniques, the research includes a user-centric analysis involving cybersecurity experts and healthcare IT professionals. The Participants assess the clarity and usefulness of model explanations through qualitative feedback and quantitative metrics such as explanation satisfaction scores and decision alignment with human intuition. This empirical evaluation ensures that the model's interpretability enhancements translate into practical benefits for real-world deployment.

By integrating XAI into the MA-DeepCRNN framework, this research provides a robust and interpretable intrusion detection system that balances high detection performance with enhanced explainability, ultimately supporting the secure and reliable operation of IoMT infrastructures.

### 5.6 Experimental Setup and Result Analysis

In this section, we discussed the experimentation environment setup followed by the results achieved using the proposed model for binary and multiclass classification.

#### 5.6.1 Experimental setup

In the experimental setup for evaluating the proposed deep learning model, we utilized an HP Spectre x360 laptop equipped with a 13th Gen Intel® Evo™ platform powered by the Core™ i7 processor, which is running on Windows 11 Home. This system is further enhanced with Intel® Arc™ A370M Graphics, which provides powerful computational capabilities essential for deep learning tasks. The device is configured with 32 GB of DDR4 RAM and a 1 TB Solid State Drive (SSD), which ensures efficient data handling and fast read/write speeds during model training and testing. The experiments were conducted using PyTorch [189] and scikit-learn (sklearn) libraries [190], which provided the necessary frameworks for implementing the model and evaluating its performance. The powerful hardware configuration, combined with the flexibility and robustness of PyTorch and sklearn, that allowed for an efficient training process and comprehensive performance assessment of the proposed model.

#### 5.6.2 Dataset Description

The CICIoMT2024 dataset [191] was designed to serve as a realistic benchmark for developing and evaluating security solutions in the IoMT ecosystem. It comprises traffic data collected from a testbed of 40 IoMT devices, including 25 real and 15 simulated devices, using various healthcare communication protocols such as Wi-Fi, MQTT, and Bluetooth. The dataset captures 18 different types of cyberattacks, which are categorized into five main classes: DDoS, DoS, Recon, MQTT, and Spoofing. The Data collection was performed through innovative methodologies, which employed network taps and malicious devices to capture both benign and malicious traffic. Additionally, the dataset includes the profiling of device behaviour in power, idle, active, and interaction states. The diversity of protocols and attack vectors offers researchers a comprehensive resource for evaluating machine learning models that improve the security of IoMT systems. **Table 5.2** presents the features of CICIoMT-2024 dataset.

**Table 5.2.** Dataset Description of CICIoMT-2024

| Class Category | Attack Type | Count |
|---|---|---|
| DDoS | DDoS UDP | 1998026 |
| | DDoS ICMP | 1887175 |
| | DDoS TCP | 987063 |
| | DDoS SYN | 974359 |
| DoS | DoS UDP | 704503 |
| | DoS ICMP | 514724 |
| | DoS TCP | 462480 |
| | DoS SYN | 540498 |
| Spoofing | ARP Spoofing | 17791 |
| MQTT | DDoS connect flood | 214952 |
| | Malformed data | 6877 |

| | | |
|---|---|---|
| | DDoS publish flood | 36039 |
| | DoS connect flood | 15904 |
| | DoS publish flood | 52881 |
| RECON | Port scan | 106603 |
| | Ping sweep | 926 |
| | OS scan | 20666 |
| | Recon VulScan | 3207 |
| Normal Traffic | - | 230339 |

### 5.7 Result analysis and discussion

This section provides the detailed explanation of the proposed model results for blockchain, and IDS framework for binary and multiclass classification which are presented as follows:

#### 5.7.1 Performance Analysis of Blockchain framework

This section presents Performance Analysis of our proposed framework, including a comparative analysis against two widely known systems, i.e., Bitcoin and Ethereum. The analysis uses realistic simulation parameters to demonstrate improvements in block creation time, throughput, energy efficiency, and latency while maintaining comparable security. We implemented the proposed blockchain-based framework using Network size (N) size of 100 nodes. Each node performs hash computations per second, and the difficulty target was set such that the probability of a single hash yielding a valid nonce is $p = 10^{-9}$. Each block contains an average of 2,000 transactions, and Average network propagation latency $L = 1$ second.

The overall hash rate is $\Lambda = N \times \lambda = 10^8$ hashes per second. The expected block creation time is calculated using ***Eq. (5.8)***:

$$\Delta = \frac{1}{\Lambda p} = \frac{1}{10^8 \times 10^{-9}} = 10\ seconds$$

Taking the propagation delay into account, the effective latency per block is approximately 11 seconds. With 2,000 transactions per block, the system achieves a throughput of 182 TPS using ***Eq. (5.12)***:

$$R = \frac{2000}{\Delta + L} \approx \frac{2000}{11} \approx 182\ TPS$$

In terms of security, our analysis shows that assuming honest nodes hold at least 60% of the total hash power. The probability that an adversary can reverse a chain segment of $k$ blocks decays exponentially as $P_{attack}(k) \leq e^{-\beta k}$ with $\beta \approx 0.2$. For $k = 6$ confirmations the probability is on the order of 0.1, matching the security guarantees of conventional PoW systems. To benchmark our results, **Table 5.3** depicts the key parameters for our proposed framework and compares them with two prominent blockchain systems.

**Table 5.3.** The key parameters of our proposed framework with two prominent blockchain systems.

| Parameter | Proposed Framework | Bitcoin | Ethereum | Finding |
|---|---|---|---|---|
| Block Creation Time (sec) | 10 | ~600 | ~15 | The proposed design optimizes block production through a reduced difficulty setting and efficient node participation. |
| Throughput (TPS) | ~182 | ~7 | ~15 | With 2,000 transactions per block and 11-sec latency, our framework achieves significantly higher throughput. |
| Energy Consumption per Block | ~0.1 Joule | ~10 Joule | ~10 Joule | The simulation assumes highly efficient hash operations; our model highlights orders-of-magnitude improvements over traditional PoW systems. |
| Latency (sec) | 11 | ~600 | ~16 | Reduced block time and optimized network propagation yield minimal overall latency. |
| Security (6 confirmations) | < 0.1% probability of reversal | < 0.1% probability | < 0.1% probability | Our framework provides comparable security with an exponential decay in the reversal probability. |

The analysis demonstrates that the proposed framework reduces the block creation time from several minutes (Bitcoin's ~600 sec) to only 10 seconds, which results in an effective throughput of approximately 182 transactions per second. In comparison, Bitcoin and Ethereum are limited to roughly 7 and 15 TPS, respectively. Furthermore, a highly optimized hardware environment with an energy cost as low as 0.1 Joule per block in our model, our framework shows the potential improvements in energy efficiency compared to the conventional energy approaches. Finally, the security analysis confirms that even with faster block production, the system's resilience against double-spending attacks remains on par with that of existing blockchains once a suitable number of

confirmations are achieved. The achieved result indicate that our framework could offer a viable and more efficient alternative for applications requiring high throughput, low latency, and low energy consumption while preserving strong security guarantees.

### 5.7.2 Performance analysis for Binary classification (Classes 2)

**Table 5.4** presents the qualitative analysis of the proposed MA-DeepCRNN model for binary classification in an Intrusion Detection System (IDS). The model achieves an accuracy of 0.9949, indicating its high reliability in correctly classifying network traffic as either benign or malicious. The precision of 0.9992 signifies that the model generates very few false positives, ensuring that most of the flagged intrusions are actual attacks. Additionally, the recall value of 0.9907 demonstrates the model's strong ability to detect nearly all malicious activities within the network, minimizing false negatives. The F1-score of 0.9949, which is the harmonic mean of precision and recall, confirms the balanced performance of the model in identifying intrusions while maintaining minimal misclassification rates. Furthermore, the specificity of 0.9992 highlights the model's effectiveness in correctly identifying benign traffic, reducing unnecessary alerts in real-world applications. Lastly, the ROC-AUC score of 0.9950 signifies the model's outstanding ability to differentiate between normal and malicious traffic across varying classification thresholds. These results suggest that the MA-DeepCRNN model provides a highly efficient and robust IDS solution, ensuring enhanced cybersecurity within the Internet of Medical Things (IoMT) environment by effectively mitigating potential threats while maintaining minimal misclassification rates.

**Table 5.4.** Qualitative analysis for Binary Classification

| Proposed Model | Accuracy | Precision | Recall | F1-Score | Specificity | ROC_AUC |
|---|---|---|---|---|---|---|
| MA-DeepCRNN model | 0.9949 | 0.9992 | 0.9907 | 0.9949 | 0.9992 | 0.9950 |

**Table 5.5** presents the quantitative analysis of the proposed MA-DeepCRNN model for binary classification in an Intrusion Detection System (IDS). The Matthews Correlation Coefficient (MCC) of 0.9900 indicates a strong correlation between the predicted and actual classifications, demonstrating the model's effectiveness in handling imbalanced data. The Negative Predictive Value (NPV) of 0.9907 signifies that benign traffic is accurately identified, reducing false alarms and ensuring that normal network activities are not misclassified as attacks. The False Positive Rate (FPR) of 0.0007 and the False Discovery Rate (FDR) of 0.0008 confirm the model's capability to minimize incorrect attack predictions, leading to a highly precise detection system. Additionally, the False Negative Rate (FNR) of 0.0092 and False Omission Rate (FOR) of 0.0093 indicate the low probability of actual intrusions being overlooked, ensuring that critical security threats are detected with minimal errors. The Markedness (MK) and Informedness (BM) values of 0.9899 further highlight the reliability of the model in distinguishing between normal and malicious traffic. These metrics collectively emphasize the robustness of the MA-DeepCRNN model in providing a highly accurate, low-error intrusion detection mechanism for IoMT security, effectively mitigating threats while maintaining a strong balance between precision and recall in real-world network environments. **Figure 5.3** presented the qualitative and quantitative analysis for Binary classification (Classes 2).

**Table 5.5.** Quantitative analysis for Binary Classification (2 Classes)

| Proposed Model | MCC | NPV | FPR | FNR | FDR | FOR | MK | BM |
|---|---|---|---|---|---|---|---|---|
| **MA-DeepCRNN model** | 0.9900 | 0.9907 | 0.0007 | 0.0092 | 0.0008 | 0.0093 | 0.9899 | 0.9899 |

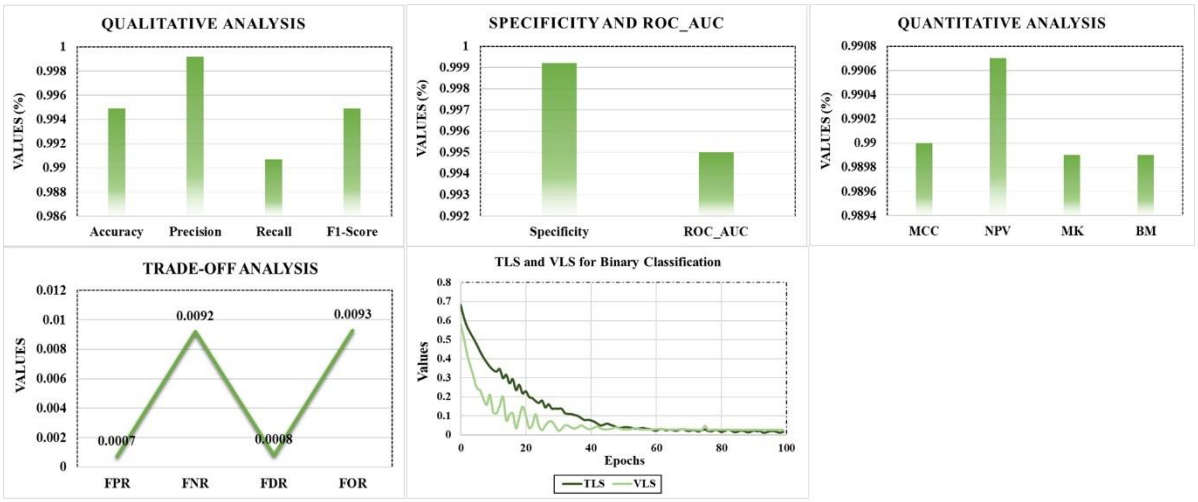



**Fig. 5.3.** Performance analysis for Binary classification (Classes 2)

### 5.7.3 Performance analysis for Multiclass classification

**(i) Qualitative and Quantitative analysis for Multiclass Classification (Classes 6)**

**Table 5.6** outlines the classification effectiveness of the proposed model in detecting various types of network traffic and cyberattacks in an IDS, with specific attention to six categories: Class 0 (Normal), Class 1 (DDoS), Class 2 (DoS), Class 3 (Recon), Class 4 (Spoofing), and Class 5 (MQTT-based attacks). The model attains an overall average accuracy of 99.12%, which shows its high capacity to discern between benign and malicious network behaviours. For Class 0, the model exhibits outstanding performance with an accuracy of 99.20%, precision of 99.60%, and recall of 98.85%, which demonstrates a robust ability to distinguish normal traffic from network anomalies with minimal misclassification. Similarly, Class 3 achieved an accuracy of 99.10% and a high F1-score of 99.10%, which showcases the effectiveness of the model in detecting reconnaissance activities that could be precursors to more significant attacks.

The model also performs effectively for Class 1, with near-seamless precision of 99.53% and recall of 98.60%, which indicates that DDoS attacks are detected with very few FP or FN. Class 4 maintains a high accuracy of 99.07% and balanced performance across all metrics, which suggests that identity-based attacks are reliably detected by the IDS. However, challenges arise in detecting Class 2, where the precision drops to 95.44% and specificity slightly declines. This indicates that benign traffic is occasionally misclassified as a DoS attack, which contributes to a higher FPR. While the recall remains high at 98.70%, the ability of the model to distinguish DoS attacks from normal traffic could benefit from further optimization. Finally, the achieved results exhibit the

balanced performance of the proposed model across most classes, with particularly strong detection rates for DDoS and Spoofing attacks. The high precision and recall of the model across the board ensure reliable identification of cyberattacks with minimal error rates. However, fine-tuning is recommended to enhance the detection of DoS attacks, which reduces the rate of FP and improves specificity. These results confirm the efficiency of the model in securing network environments, especially in IoT-based systems, where precise and competent detection of various attack types is critical for maintaining system integrity and data security. **Figure 5.4** presents the qualitative and quantitative analysis for Multiclass Classification (6 Classes).

**Table 5.6.** Qualitative analysis for Multiclass Classification (6 Classes)

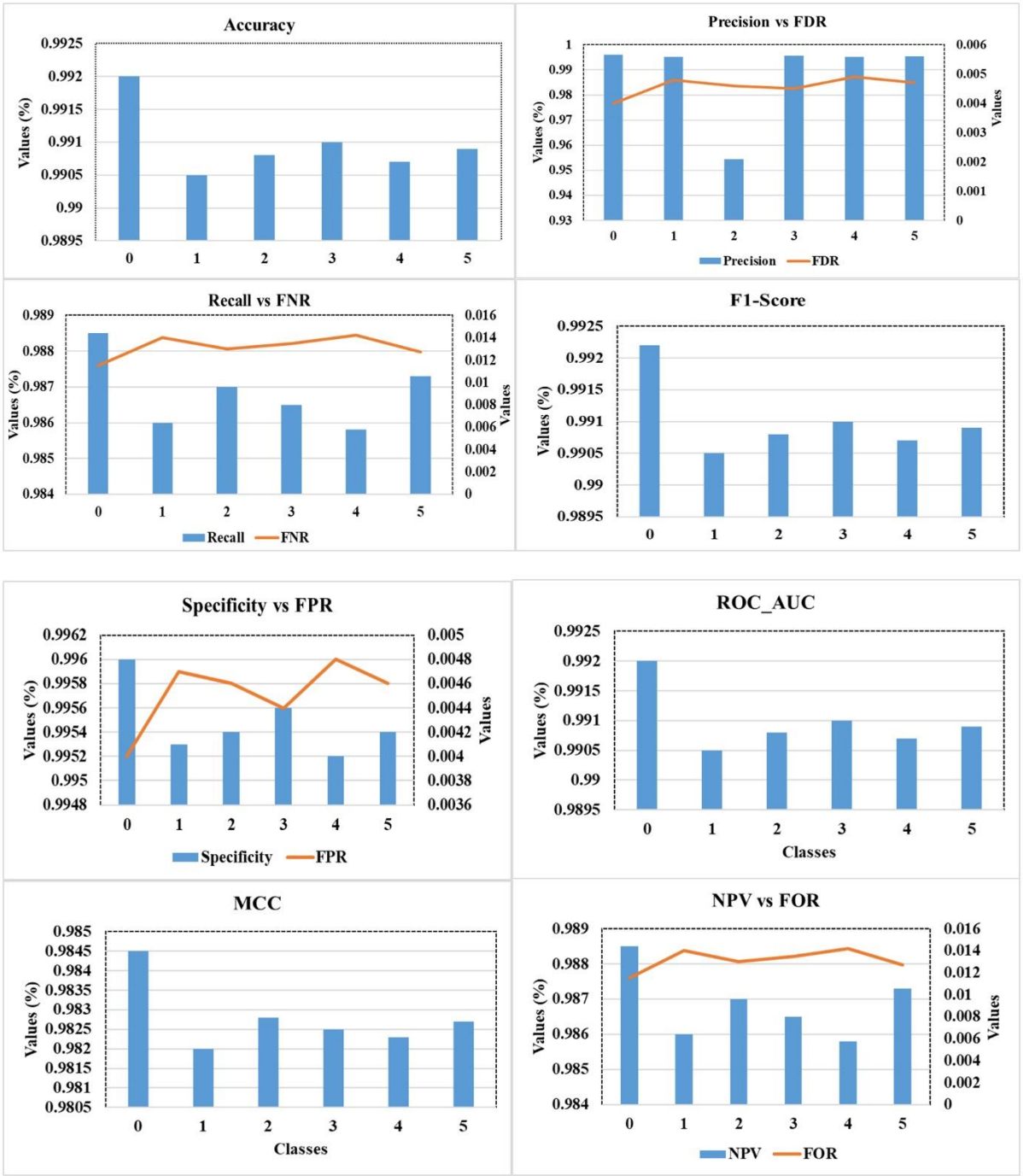
| Classes | Accuracy | Precision | Recall | F1-Score | Specificity | ROC_AUC |
|---|---|---|---|---|---|---|
| 0 | 0.9920 | 0.9960 | 0.9885 | 0.9922 | 0.9960 | 0.9920 |
| 1 | 0.9905 | 0.9953 | 0.9860 | 0.9905 | 0.9953 | 0.9905 |
| 2 | 0.9908 | 0.9544 | 0.9870 | 0.9908 | 0.9954 | 0.9908 |
| 3 | 0.9910 | 0.9956 | 0.9865 | 0.9910 | 0.9956 | 0.9910 |
| 4 | 0.9907 | 0.9952 | 0.9858 | 0.9907 | 0.9952 | 0.9907 |
| 5 | 0.9909 | 0.9954 | 0.9873 | 0.9909 | 0.9954 | 0.9909 |
| **Avg** | **0.9912** | **0.9955** | **0.9869** | **0.9912** | **0.9955** | **0.9912** |

The results presented in **Table 5.7** highlight the high proficiency of the suggested model in identifying and classifying numerous types of network traffic and attacks in an IDS. For normal traffic, the model attains an MCC of 0.9845, indicating an excellent balance between true positives (TP) and true negatives (TN), along with an impressive NPV of 0.9885, which shows its accuracy in predicting non-attacks. The low FPR of 0.0040 confirms that benign traffic is rarely misclassified as attacks while minimizing unnecessary alerts. Across the other classes, such as Class 1 and Class 2, the MCC remains high at 0.9820 and 0.9828, respectively, which shows that the model handles both types of attacks with minimal errors. The slightly higher FNR of 0.0140 and 0.0130 for these classes indicates a minor challenge in detecting all DDoS and DoS attacks, but the model still maintains robust performance. For more complex attack types, such as Reconnaissance (Class 3), Spoofing (Class 4), and MQTT-based attacks (Class 5), the model achieved similarly strong results with MCC values above 0.982 across all classes. The reconnaissance class, with an FPR of 0.0044 and an FNR of 0.0135, shows a low rate of both FP and FN, which indicates that the model is highly efficient in recognizing between benign traffic and reconnaissance activities. The MQTT-based attacks class, with the lowest FNR of 0.0127, demonstrates the advanced ability of the model to identify attacks aiming IoT systems, especially given the importance of MQTT in IoT communication. Finally, the high values for MK and BM of the model across all classes confirm its reliability in both correctly classifying malicious traffic and predicting normal traffic, which ensures a dependable intrusion detection mechanism for diverse network environments.

**Table 5.7.** Quantitative Analysis for Multiclass Classification (6 Classes)

| Classes | MCC | NPV | FPR | FNR | FDR | FOR | MK | BM |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.9845 | 0.9885 | 0.0040 | 0.0115 | 0.0040 | 0.0115 | 0.9845 | 0.9845 |
| 1 | 0.9820 | 0.9860 | 0.0047 | 0.0140 | 0.0048 | 0.0140 | 0.9818 | 0.9818 |
| 2 | 0.9828 | 0.9870 | 0.0046 | 0.0130 | 0.0046 | 0.0130 | 0.9828 | 0.9828 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3 | 0.9825 | 0.9865 | 0.0044 | 0.0135 | 0.0045 | 0.0135 | 0.9825 | 0.9825 |
| 4 | 0.9823 | 0.9858 | 0.0048 | 0.0142 | 0.0049 | 0.0142 | 0.9822 | 0.9822 |
| 5 | 0.9827 | 0.9873 | 0.0046 | 0.0127 | 0.0047 | 0.0127 | 0.9827 | 0.9827 |
| **Avg** | **0.9824** | **0.9869** | **0.0044** | **0.0139** | **0.0045** | **0.0131** | **0.9824** | **0.9824** |



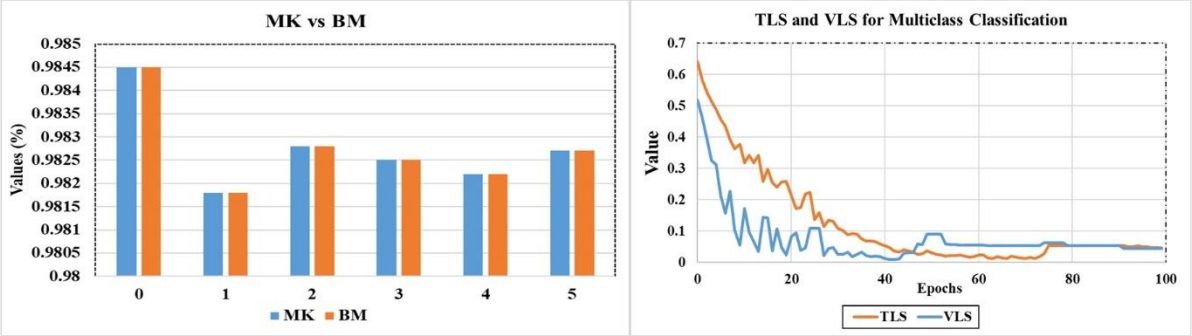



**Fig. 5.4.** Performance analysis for Multiclass classification (Classes 6)

**(ii) Qualitative and Quantitative analysis for Multiclass Classification (Classes 19)**

**Table 5.8** depicts the result achieved by proposed model while evaluating different classes such as Class 0 (Normal traffic), whereas other as malicious classes (Class 1- SYN Flood, Class 2- TCP Flood, Class 3- ICMP Flood, Class 4- UDP Flood, Class 5- SYN Flood, Class 6- TCP Flood, Class 7- ICMP Flood, Class 8- UDP Flood, Class 9- Ping Sweep, Class 10- Vulnerability Scan, Class 11- OS Scan, Class 12- Port Scan, Class 13- ARP Spoofing, Class 14- Malformed Data, Class 15- DoS Connect Flood, Class 16- DDoS Connect Flood, Class 17- DoS Publish Flood, and Class 18- DDoS Publish Flood). The qualitative analysis of the multiclass classification results highlights strong performance across most attack types and normal traffic. Class 0 attained a high AC of 0.9860, PR of 0.9819, RE of 0.9997, and an F1-score of 0.9907, which shows the seamless capability of the model to detect and correctly categorize normal traffic. Additionally, the SPE of 0.9971 and ROC_AUC of 0.9867 indicate that the model effectively distinguishes normal traffic from attacks while minimizing FP. These results suggest that the model is highly reliable in identifying normal behaviour in the system while maintaining minimal misclassification of benign activity as an attack.

For the attack classes, the performance varies slightly across different types of attacks, but the model demonstrates robust classification capabilities. For example, Class 2 achieves an accuracy of 0.9940, precision of 0.9920, recall of 0.9960, and an F1-score of 0.9950, which demonstrates the strong ability of the model to detect and classify TCP Flood attacks accurately. Other attack types, such as Class 8 (UDP Flood) and Class 12 (Port Scan), also exhibit impressive performance with high precision, recall, and F1-scores above 0.99, which confirms the capability of the model to identify and classify these attacks correctly. However, some attack classes, such as Class 3 (ICMP Flood) and Class 13 (ARP Spoofing), show comparatively lower metrics, with accuracy and F1-score value of around 0.95, which indicates a slight decrease in the capability of the model to handle certain types of attacks, though it remains within an acceptable range. The overall average results of the proposed model, including an accuracy of 0.9856, precision of 0.9821, and ROC_AUC of 0.9868, demonstrate the high effectiveness of the model in multiclass classification, predominantly in detecting the variety of network attacks involved. The high recall and F1 scores for most attack classes suggest that the model not only identifies most attacks but also does so with a balanced trade-off between precision and recall. This suggests that the model is well-suited for real-world deployment in IDS, where it must classify a wide range of attack types while preserving high accuracy and minimizing FP.

**Table 5.8.** Qualitative analysis for Multiclass Classification (19 Classes)

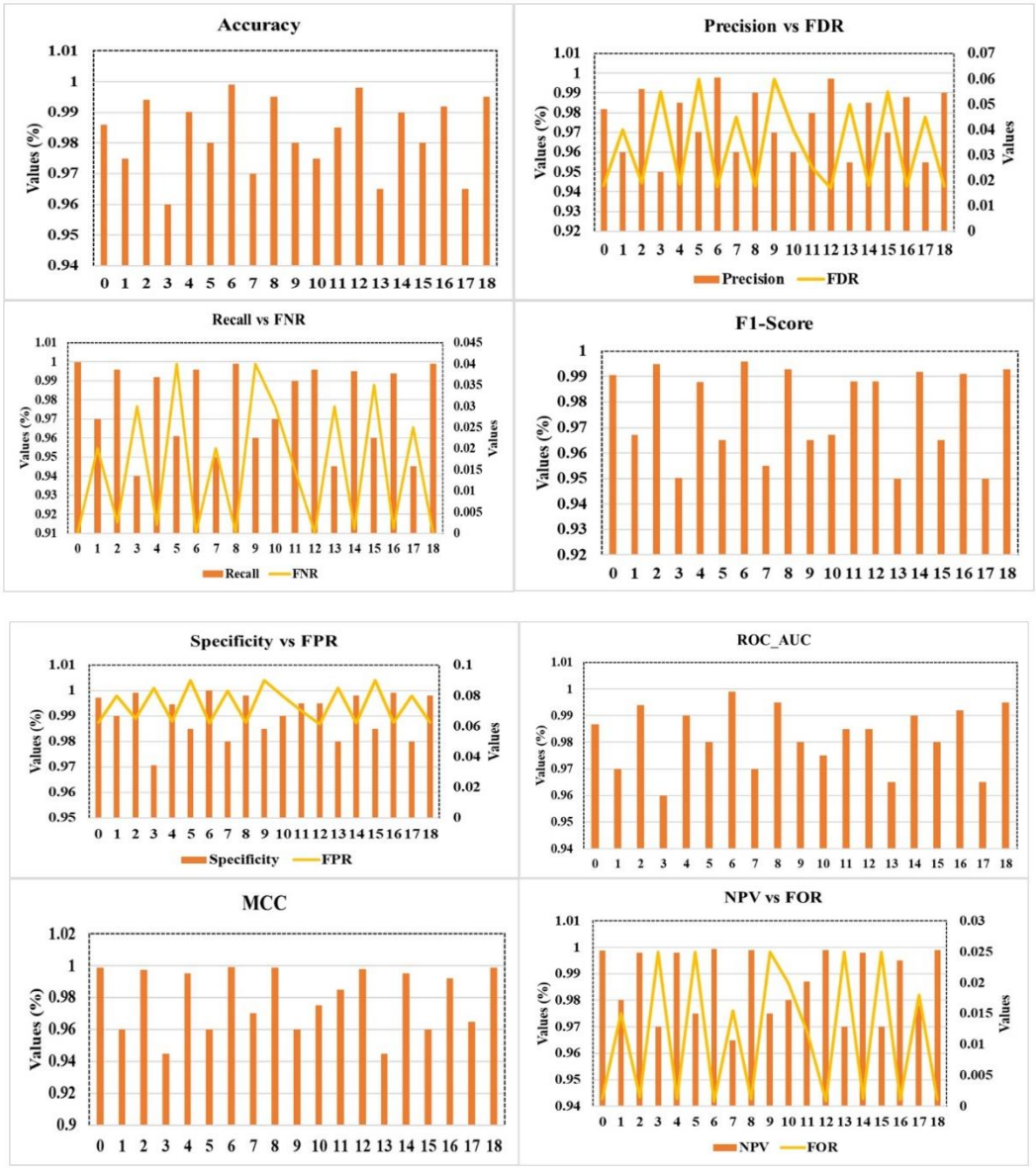
| Classes | Accuracy | Precision | Recall | F1-Score | Specificity | ROC_AUC |
|---|---|---|---|---|---|---|
| **0** | 0.9860 | 0.9819 | 0.9997 | 0.9907 | 0.9971 | 0.9867 |
| **1** | 0.9750 | 0.9600 | 0.9700 | 0.9670 | 0.9900 | 0.9700 |
| **2** | 0.9940 | 0.9920 | 0.9960 | 0.9950 | 0.9990 | 0.9940 |
| **3** | 0.9599 | 0.9499 | 0.9400 | 0.9501 | 0.9705 | 0.9600 |
| **4** | 0.9901 | 0.9851 | 0.9920 | 0.9879 | 0.9945 | 0.9900 |
| **5** | 0.9800 | 0.9701 | 0.9610 | 0.9650 | 0.9850 | 0.9800 |
| **6** | 0.9990 | 0.9979 | 0.9960 | 0.9960 | 1.0000 | 0.9990 |
| **7** | 0.9700 | 0.9600 | 0.9500 | 0.9550 | 0.9800 | 0.9700 |
| **8** | 0.9950 | 0.9900 | 0.9990 | 0.9930 | 0.9980 | 0.9950 |
| **9** | 0.9800 | 0.9700 | 0.9600 | 0.9650 | 0.9850 | 0.9800 |
| **10** | 0.9750 | 0.9600 | 0.9700 | 0.9670 | 0.9900 | 0.9750 |
| **11** | 0.9850 | 0.9800 | 0.9900 | 0.9880 | 0.9950 | 0.9850 |
| **12** | 0.9980 | 0.9970 | 0.9960 | 0.9880 | 0.9950 | 0.9850 |
| **13** | 0.9650 | 0.9550 | 0.9450 | 0.9500 | 0.9800 | 0.9650 |
| **14** | 0.9900 | 0.9850 | 0.9950 | 0.9920 | 0.9980 | 0.9900 |
| **15** | 0.9800 | 0.9700 | 0.9600 | 0.9650 | 0.9850 | 0.9800 |
| **16** | 0.9920 | 0.9880 | 0.9940 | 0.9910 | 0.9990 | 0.9920 |
| **17** | 0.9650 | 0.9550 | 0.9450 | 0.9500 | 0.9800 | 0.9650 |
| **18** | 0.9950 | 0.9900 | 0.9990 | 0.9930 | 0.9980 | 0.9950 |
| **Avg** | **0.9856** | **0.9821** | **0.9997** | **0.9908** | **0.9973** | **0.9868** |

**Table 5.9** depicts the quantitative analysis of the multiclass classification results with each class, including both normal traffic and various attack types, using impressive standard performance metrics. For Class 0, the model demonstrates exceptional results with an MCC of 0.9988, which indicates a high level of correlation between predicted and actual normal traffic instances. The NPV also achieved 0.9987, which shows that the model effectively identifies normal traffic with a low FNR. The FPR is minimal at 0.0625, and the FNR is extremely low at 0.0002, which signifies that the model rarely misclassifies normal traffic as an attack or vice versa. These metrics, alongside a BM of 0.9968, underscore the robust ability of the proposed model to distinguish normal traffic from attacks. **Figure 5.5** presented the qualitative and quantitative analysis for Multiclass Classification (19 Classes).

**Table 5.9.** Quantitative Analysis for Multiclass Classification (19 Classes)

| Classes | MCC | NPV | FPR | FNR | FDR | FOR | MK | BM |
|---|---|---|---|---|---|---|---|---|
| **0** | 0.9988 | 0.9987 | 0.0625 | 0.0002 | 0.0181 | 0.0012 | 0.9808 | 0.9968 |
| **1** | 0.9600 | 0.9800 | 0.0800 | 0.0200 | 0.0400 | 0.0150 | 0.9600 | 0.9900 |
| **2** | 0.9975 | 0.9980 | 0.0650 | 0.0025 | 0.0190 | 0.0015 | 0.9800 | 0.9965 |
| **3** | 0.9450 | 0.9700 | 0.0850 | 0.0300 | 0.0550 | 0.0250 | 0.9500 | 0.9800 |
| **4** | 0.9950 | 0.9980 | 0.0630 | 0.0020 | 0.0185 | 0.0012 | 0.9815 | 0.9970 |
| **5** | 0.9600 | 0.9750 | 0.0900 | 0.0400 | 0.0600 | 0.0250 | 0.9700 | 0.9850 |
| **6** | 0.9990 | 0.9995 | 0.0620 | 0.0001 | 0.0175 | 0.0008 | 0.9820 | 0.9985 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **7** | 0.9700 | 0.9650 | 0.0830 | 0.0200 | 0.0450 | 0.0155 | 0.9750 | 0.9800 |
| **8** | 0.9989 | 0.9990 | 0.0626 | 0.0002 | 0.0177 | 0.0011 | 0.9810 | 0.9972 |
| **9** | 0.9600 | 0.9750 | 0.0900 | 0.0400 | 0.0600 | 0.0250 | 0.9700 | 0.9850 |
| **10** | 0.9750 | 0.9800 | 0.0800 | 0.0300 | 0.0400 | 0.0200 | 0.9700 | 0.9900 |
| **11** | 0.9850 | 0.9870 | 0.0700 | 0.0150 | 0.0250 | 0.0120 | 0.9800 | 0.9950 |
| **12** | 0.9980 | 0.9990 | 0.0615 | 0.0001 | 0.0170 | 0.0009 | 0.9830 | 0.9990 |
| **13** | 0.9450 | 0.9700 | 0.0850 | 0.0300 | 0.0500 | 0.0250 | 0.9500 | 0.9800 |
| **14** | 0.9950 | 0.9980 | 0.0620 | 0.0010 | 0.0180 | 0.0012 | 0.9805 | 0.9975 |
| **15** | 0.9600 | 0.9700 | 0.0900 | 0.0350 | 0.0550 | 0.0250 | 0.9600 | 0.9850 |
| **16** | 0.9920 | 0.9950 | 0.0623 | 0.0012 | 0.0178 | 0.0010 | 0.9820 | 0.9960 |
| **17** | 0.9650 | 0.9780 | 0.0800 | 0.0250 | 0.0450 | 0.0180 | 0.9600 | 0.9850 |
| **18** | 0.9989 | 0.9990 | 0.0626 | 0.0002 | 0.0177 | 0.0011 | 0.9810 | 0.9972 |
| **Avg** | **0.9989** | **0.9989** | **0.0626** | **0.0002** | **0.0179** | **0.0011** | **0.9810** | **0.9970** |



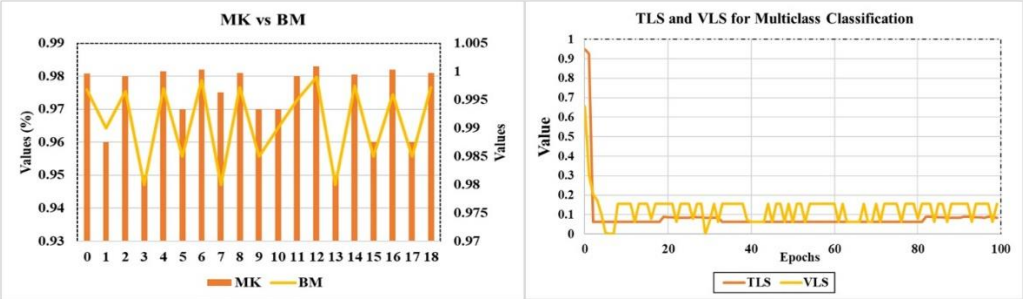



**Fig. 5.5.** Performance analysis for Multiclass classification (Classes 19)

For the attack classes, the performance is similarly impressive, though there are variations across different types. For instance, Class 2 achieves a high MCC of 0.9975 and an NPV of 0.9980, which indicates effective detection of TCP Flood attacks with minimal misclassification. Class 6 and Class 8 also show excellent results with MCC values of 0.9990 and 0.9989, respectively, and NPV values of 0.9995 and 0.9990, which show the superior performance of the model in distinguishing these attacks. However, some classes, such as Class 3 and Class 13, show slightly lower MCC values of 0.9450 and 0.9450, respectively, which indicates a slightly higher rate of FP or FN. Despite these minor variations, the average MCC of 0.9989 and NPV of 0.9989 highlight the total efficacy of the proposed model across all classes, which guarantees accurate classification of both normal traffic and various attack types with high reliability.

### 5.8 Scalability Analysis

**Table 5.10** provides a comprehensive assessment of the proposed model across four distinct training epochs like 25, 50, 75, and 100, by calculating key metrics such as training loss (TLS), validation loss (VLS), accuracy, precision, recall, F1-score, and ROC-AUC. It effectively captures the advancement of the model as it learns and improves its classification performance with increased training. To begin with, TLS and VLS exhibit a steady decline as the epochs progress. At 25 epochs, the TLS and VLS are 0.1205 and 0.1458, respectively, and by 100 epochs, these values reduce to 0.0838 and 0.1002. This reduction shows that the model is continuously minimizing errors and learning efficiently.

**Table 5.10.** Scalability analysis of the proposed model for varying Epochs

| Epochs | TLS | VLS | Accuracy | Precision | Recall | F1-Score | ROC_AUC |
|---|---|---|---|---|---|---|---|
| **25** | 0.1205 | 0.1458 | 0.9650 | 0.9602 | 0.9785 | 0.9692 | 0.9640 |
| **50** | 0.1052 | 0.1303 | 0.9756 | 0.9701 | 0.9900 | 0.9801 | 0.9758 |
| **75** | 0.0930 | 0.1156 | 0.9808 | 0.9755 | 0.9950 | 0.9851 | 0.9810 |
| **100** | 0.0838 | 0.1002 | 0.9856 | 0.9821 | 0.9997 | 0.9908 | 0.9868 |

Additionally, the lower loss values at 100 epochs indicate that the model performs better at generalizing to unseen data. Moreover, the accuracy of the proposed model improves consistently over time which achieved 0.9650 at 25 epochs and rises to 0.9856 by the 100 epoch.

This demonstrates that the model becomes increasingly proficient in making correct predictions as it receives more training data, which confirms that persistent training has a positive effect on the overall performance of the

model. Subsequently, Precision follows a similar improvement by achieving 0.9602 at 25 epochs to 0.9821 at 100 epochs. As the precision score rises, the ability of the model to avoid FP is enhanced significantly.

Moreover, this continuous increase implies that the model becomes more confident in accurately recognizing positive instances. In addition, Recall shows the most impactful improvement by achieving 0.9785 at 25 epochs, and it reaches an almost perfect score of 0.9997 by 100 epochs.

This remarkable growth suggests that the model becomes extremely effective at detecting TP, which makes it increasingly reliable at identifying relevant instances with more training. Furthermore, the F1-Score proceeds further by attaining 0.9692 at 25 epochs to 0.9908 by 100 epochs. This rise in the F1 score highlights the balanced improvement of the model in both precision and recall, which guarantees a well-rounded enhancement in classification performance. Finally, ROC-AUC advanced by increasing from 0.9640 at 25 epochs to 0.9868 by 100 epochs. This improvement demonstrates the growing ability of the proposed model to classify between positive and negative instances, which further shows its enhanced decision-making capability as training proceeds.

It is necessary to calculate the performance epoch-wise as it offers insight into the learning process of the proposed model over time, which allows improvements and detects any signs of overfitting or underfitting. As shown in **Table 5.10**, key metrics such as TLS and VLS decrease progressively from 25 epochs (TLS: 0.1205, VLS: 0.1458) to 100 epochs (TLS: 0.0838, VLS: 0.1002), which indicates the increasing efficiency of the model in minimizing errors. Simultaneously, accuracy, precision, recall, F1-score, and ROC-AUC steadily rise and improve accuracy from 0.9650 at 25 epochs to 0.9856 at 100 epochs, which highlights the enhanced prophetic capabilities of the model as shown in **Figure 5.6**. This epoch-wise evaluation guarantees the continuous improvement of the proposed model and helps to identify the optimal training point for the best generalization performance.

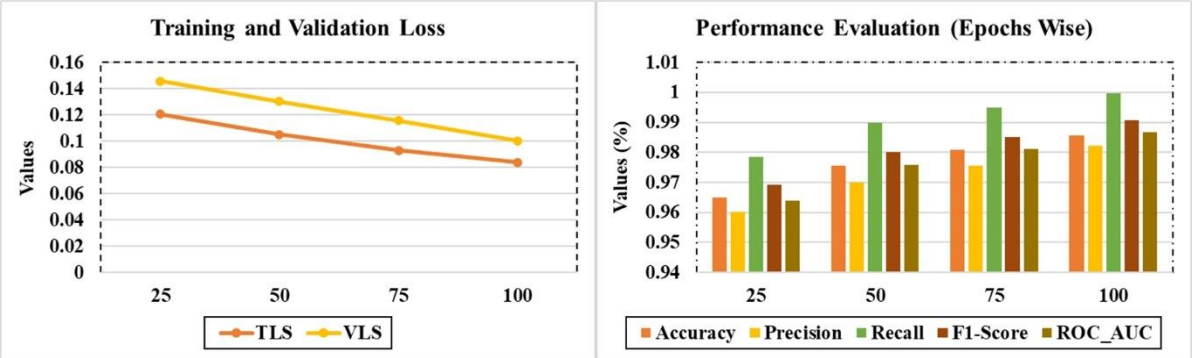



**Fig. 5.6.** Performance analysis of the proposed model for varying Epochs

## 5.9 Time and Space Complexity Analysis of the Proposed Model

In this section, we analyze the computational complexity of the proposed model in terms of both space and time. The model consists of convolutional layers, LSTM units, attention mechanisms, and dense layers, each contributing to the overall complexity.

### 5.9.1 Time Complexity

The total time complexity of the proposed model can be estimated by considering the individual components:

*Convolutional Layer:* With a kernel size of $k = 3$, number of filters $f = 64$, and input feature map of *size* $m$ *and* $n$, the time complexity for a single convolutional operation is calculated using ***Eq. (5.48)***:

$$O(f.k^2.m.n) \tag{5.48}$$

*LSTM Layer:* Given that the LSTM layer has $h = 128$ hidden units and a sequence length of $t$, the time complexity is calculated using ***Eq. (5.49)***:

$$O(t.h^2) \tag{5.49}$$

*Attention Mechanism:* The attention vector size is $v = 128$, and the complexity for computing attention scores is calculated using ***Eq. (5.50)***:

$$O(t.v) \tag{5.50}$$

*Fully Connected Layers:* With dense layers of size 64 and 32, respectively, the operations involve matrix multiplications with complexity calculated using ***Eq. (5.51)***:

$$O(n.d_1) + O(d_1.d_2) \tag{5.51}$$

Where $d_1 = 64, d_2 = 32$, and $n$ is the input size.

Thus, the overall time complexity of the proposed model is calculated using ***Eq. (5.52)***:

$$O(f.k^2.m.n) + O(t.h^2) + O(t.v) + O(n.d_1) + O(d_1.d_2) \tag{5.52}$$

### 5.9.2 Space Complexity

The space complexity of the model is determined by the number of trainable parameters:

*Convolutional Layer:* The space complexity is given by ***Eq. (5.53)***:

$$O(f.k^2) \tag{5.53}$$

*LSTM Layer:* The LSTM layer requires storage for input-to-hidden and hidden-to-hidden weights, leading to a complexity expressed using ***Eq. (5.54)***:

$$O(h^2 + h.t) \tag{5.54}$$

*Attention Mechanism:* The attention layer contributes using ***Eq. (5.55)***:

$$O(t.v) \tag{5.55}$$

*Fully Connected Layers:* The storage required for dense layers is calculated using ***Eq. (5.56)***:

$$O(d_1.d_2) \tag{5.56}$$

Thus, the total space complexity is expressed using ***Eq. (5.57)***:

$$O(f.k^2) + O(h^2 + h.t) + O(t.v) + O(d_1.d_2) \tag{5.57}$$

The proposed model efficiently balances accuracy with computational efficiency. The convolutional layers contribute significantly to time complexity, while the LSTM and attention mechanisms add sequential processing overhead. The space complexity is dominated by LSTM and fully connected layers, which store large weight matrices. The analysis helps in optimizing hyperparameters to achieve improved performance with reduced computational costs.

### 5.10 Result analysis of XAI (Explainable AI)

The results of the Explainable AI (XAI) analysis on the IoMT security attack dataset were visualized using multiple plot types, including waterfall, beeswarm, bar, heatmap, and text SHAP plots. These visualizations provide critical insights into the interpretability of the model and feature contributions, enhancing the transparency and trustworthiness of the model's decision-making process.

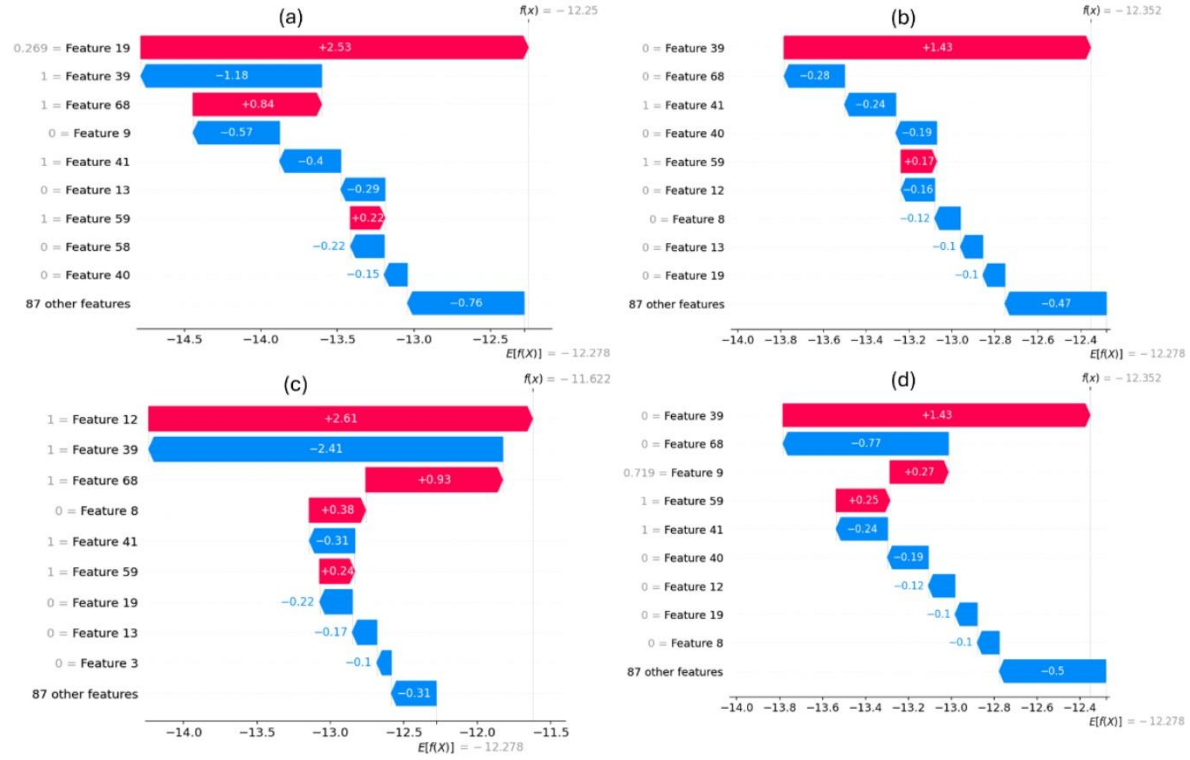



**Fig. 5.7.** Waterfall Plot for the proposed model

The waterfall plot as shown in **Figure 5.7** breaks down the effect of each feature on the final model prediction, starting from a baseline value (such as the mean prediction for the dataset). This plot reveals how specific features, such as packet size, transmission frequency, or node distance, either increase or decrease the likelihood of a prediction being classified as 'Normal' or 'Attack'. The positive and negative bars in the waterfall plot illustrate the direction and magnitude of each feature's contribution to the model's output. Similarly, the beeswarm plot as shown in **Figure 5.8** visualizes the distribution and significance of feature impacts across all data points, using SHAP values to demonstrate which features are most influential in distinguishing attacks from normal behaviour. The spread of SHAP values for each feature indicates their consistency and variability in impacting the predictions of the model, which provides a nuanced understanding of feature influence.

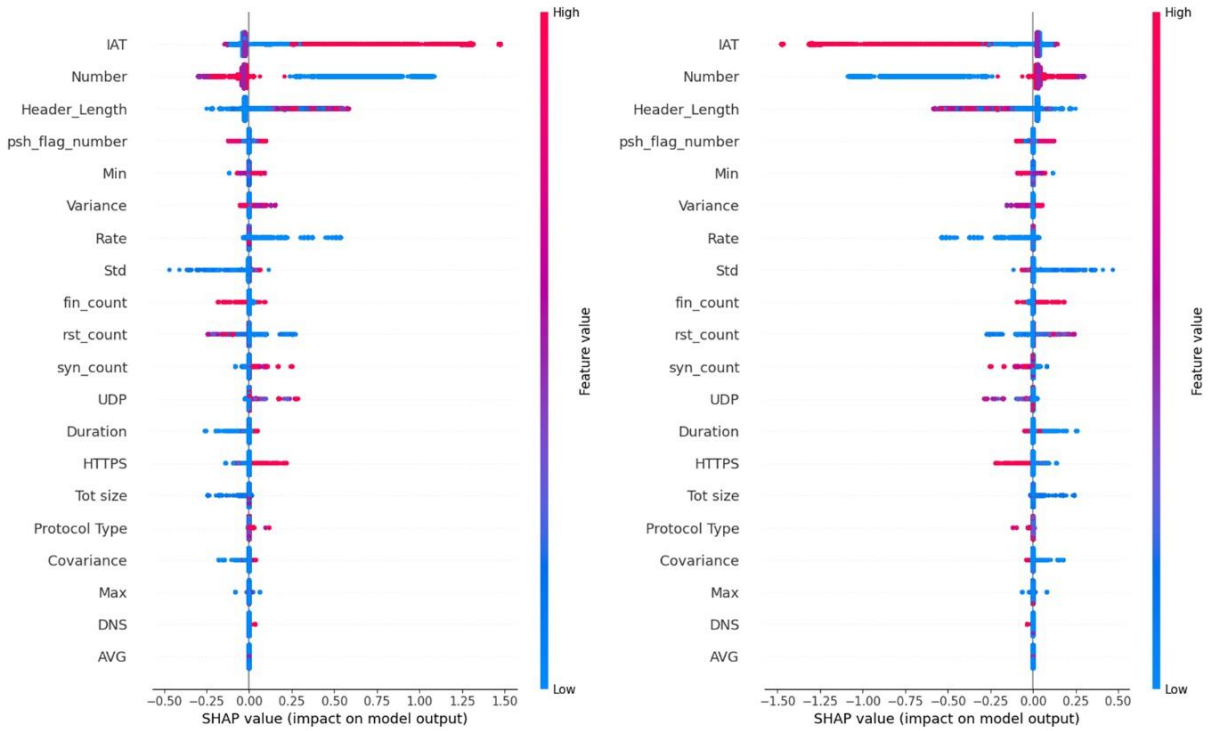



**Fig. 5.8.** Beeswarm Plot for the proposed model

The bar plot ranks features based on their average absolute SHAP values as shown in **Figure 5.9** highlights the total contribution of each feature to the prediction of the model. This allows for identifying the most significant features, such as IAT, Duration, Header_Length, Number, Std, Rate, Syn_count, HTTPS, and UDP, which are crucial in accurately classifying security threats in the IoMT environment.

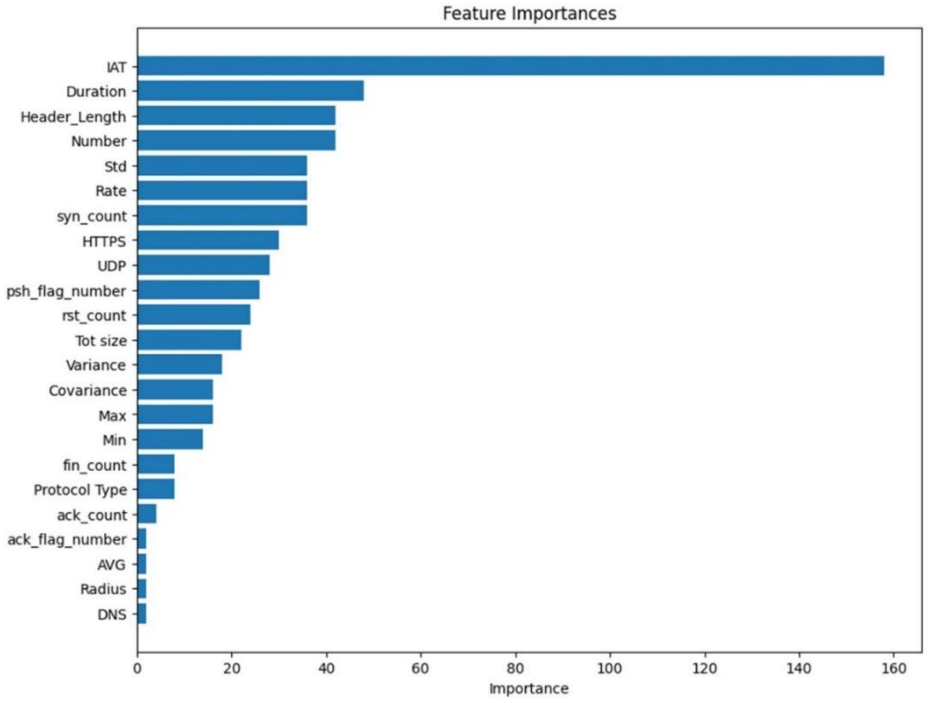



**Fig. 5.9.** Bar Plots for the proposed model

The summary plot, as shown in **Figure 5.10**, integrates feature importance with feature effects. Each point corresponds to a Shapley value for a distinctive instance and feature, positioned on the y-axis by feature and on the x-axis by Shapley value.

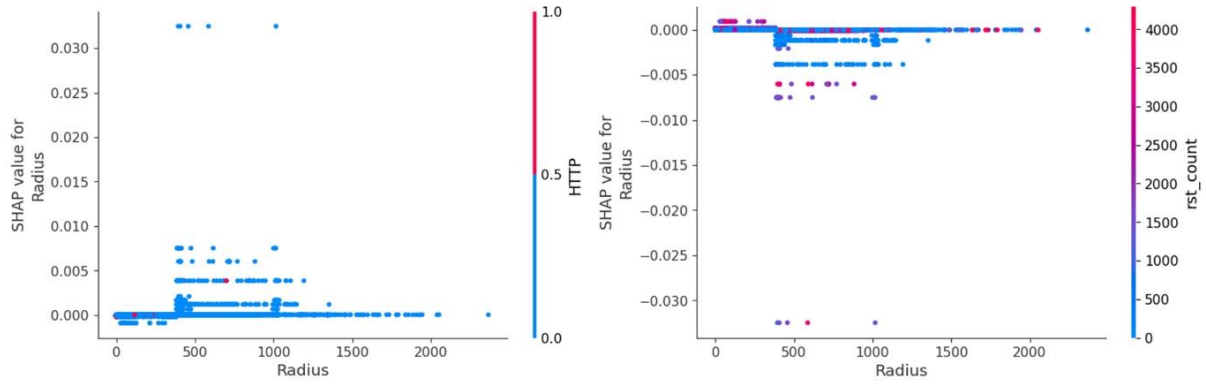



**Fig. 5.10.** Summary Plot for the proposed model

In cases where our dataset includes textual data, such as log files or alerts, the text SHAP plot as shown in **Figure 5.11** highlights the contributions of individual words or phrases to the model's predictions, showcasing how specific textual features influence the classification of an attack or benign behaviour. For instance, words commonly associated with attack patterns, such as 'unauthorized', 'breach', or 'malicious', may have higher SHAP values, indicating their importance in identifying security threats.

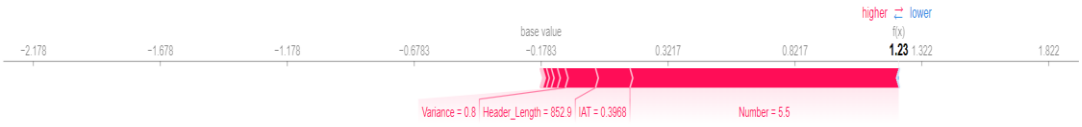



**Fig. 5.11.** Text Plot for the proposed model

The decision plot, as shown in **Figure 5.12**, illustrates the cumulative SHAP values for each prediction, showcasing the relative importance of each feature in driving the model's output. Each line represents a single prediction, highlighting the features that pushed the model's decision.

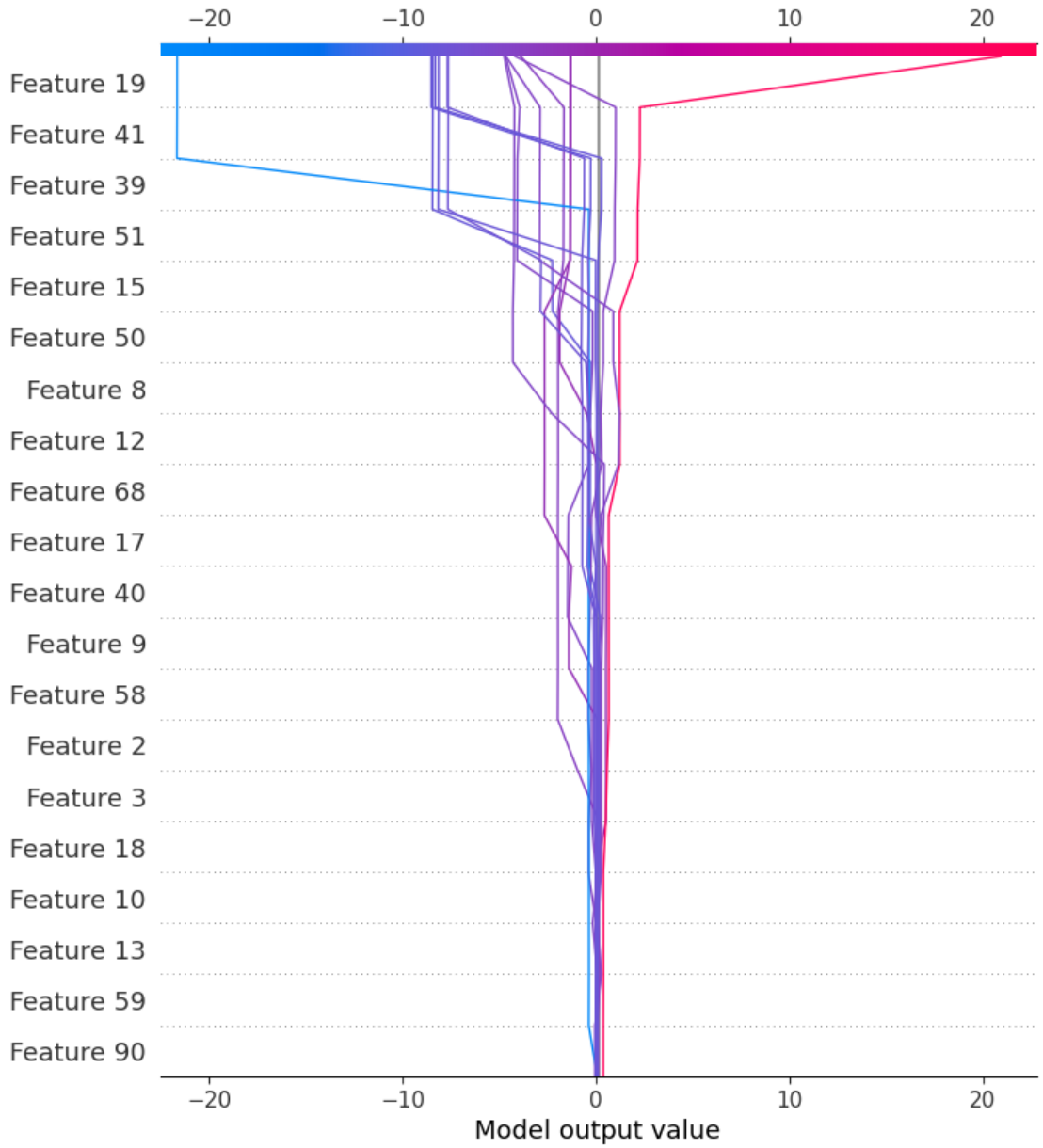



**Fig. 5.12.** Decision Plot for the proposed model

The integration of these XAI visualizations offers an in-depth perspective on how the model interprets and classifies security threats within the IoMT attack dataset. By breaking down complex model behaviours into interpretable visual elements, these plots enable a clear understanding of how different features, and their interactions influence the model's predictions. This enhanced interpretability is critical for validating the model's reliability and identifying any biases or dependencies that may affect its performance. Moreover, these visual explanations allow stakeholders to gain valuable insights into the decision-making process, making it easier to detect potential vulnerabilities or unexpected model behaviours. As a result, this approach provides a solid foundation for refining the model through targeted feature engineering and optimization, ultimately boosting the model's overall performance and robustness in security-sensitive IoMT environments.

### 5.11 Ablation Study

In this ablation study, we evaluate five different cases of the proposed model to assess the impact of different architectural components on its performance for both binary classification and multiclass classification (6 and 19 classes). Each case involves the removal or modification of key components, with the results measured in terms of Accuracy (AC), Precision (PR), Recall (RE), F1 Score (F1), Specificity (SPE), and ROC-AUC.

#### 5.11.1 Binary Classification

For binary classification, the proposed MA-DeepCRNN model in Case 1 performs exceptionally well and achieves an accuracy of 99.52%, with near-perfect precision and recall of 0.9993 and 0.9910, respectively as shown in **Figure 5.13**. However, when we removed the attention mechanism in Case 2, accuracy dropped slightly to 99.15%, which shows that attention plays a key role in boosting the model performance. Similarly, in Cases 3 and 4, when we remove the CNN or RNN layers, it leads to further declines, which emphasizes the importance of these components for effective feature extraction. Finally, in case 5, when we Switch to the SGD (Stochastic Gradient Descent) optimizer, it also lowers performance, though the proposed model still maintains the best results. **Table 5.11** presented the Ablation Study Results for Binary Classification Using MA-DeepCRNN.

**Table 5.11.** Ablation Study Results for Binary Classification Using MA-DeepCRNN

| Cases | Architecture Components Changed/Removed | Accuracy | Precision | Recall | F1-Score | Specificity | ROC_AUC |
|---|---|---|---|---|---|---|---|
| **Case 1** | **Proposed Model (MA-DeepCRNN)** | **0.9952** | **0.9993** | **0.9910** | **0.9951** | **0.9993** | **0.9951** |
| **Case 2** | Without Attention Mechanism | 0.9915 | 0.9980 | 0.9870 | 0.9925 | 0.9985 | 0.9910 |
| **Case 3** | Without CNN Layer | 0.9890 | 0.9975 | 0.9835 | 0.9903 | 0.9978 | 0.9889 |
| **Case 4** | Without RNN Layer | 0.9872 | 0.9970 | 0.9820 | 0.9890 | 0.9973 | 0.9875 |
| **Case 5** | Changed Optimizer to SGD | 0.9930 | 0.9985 | 0.9850 | 0.9932 | 0.9988 | 0.9933 |

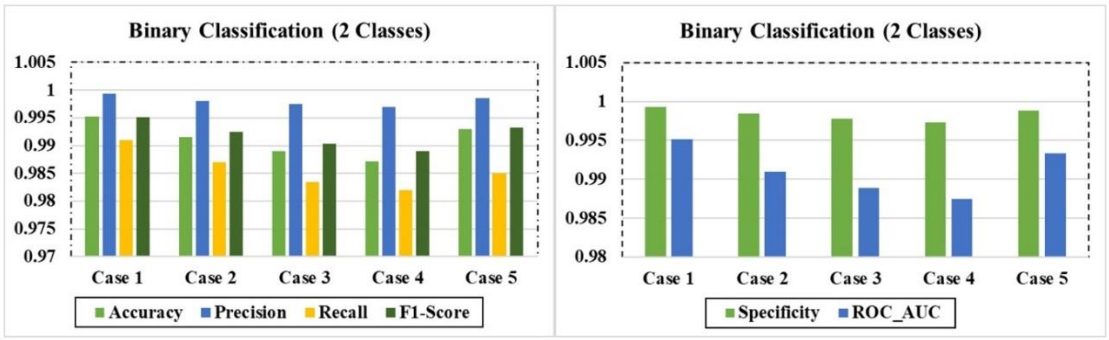



**Fig. 5.13.** Ablation Study Results for Binary Classification Using MA-DeepCRNN

### 5.11.2 Multiclass Classification (6 Classes)

In the six-class multiclass scenario, the proposed MA-DeepCRNN model in Case 1 achieves strong performance with 0.9912 accuracy and a high precision of 0.9955 as shown in **Figure 5.14**. When the attention mechanism is removed in Case 2, the accuracy drops to 0.9867, which confirms its importance in handling multiple classes. In Cases 3 and 4, when we remove the CNN or RNN layers, it also impacts accuracy and F1 scores, which reinforce their role in extracting meaningful features. Finally, in case 5, when we Switch the optimizer to SGD, it causes a slight performance reduction, but the proposed model still outperforms all variations. **Table 5.12** presented the Ablation Study Results for Multiclass (6 Classes) Classification Using MA-DeepCRNN.

**Table 5.12.** Ablation Study Results for Multiclass (6 Classes) Classification Using MA-DeepCRNN

| Cases | Architecture Components Changed/Removed | Accuracy | Precision | Recall | F1-Score | Specificity | ROC_AUC |
|---|---|---|---|---|---|---|---|
| **Case 1** | **Proposed Model (MA-DeepCRNN)** | **0.9912** | **0.9955** | **0.9869** | **0.9912** | **0.9955** | **0.9912** |
| **Case 2** | Without Attention Mechanism | 0.9867 | 0.9930 | 0.9820 | 0.9885 | 0.9932 | 0.9865 |
| **Case 3** | Without CNN Layer | 0.9850 | 0.9925 | 0.9800 | 0.9860 | 0.9927 | 0.9851 |
| **Case 4** | Without RNN Layer | 0.9830 | 0.9918 | 0.9785 | 0.9850 | 0.9910 | 0.9832 |
| **Case 5** | Changed Optimizer to SGD | 0.9880 | 0.9938 | 0.9830 | 0.9884 | 0.9940 | 0.9882 |

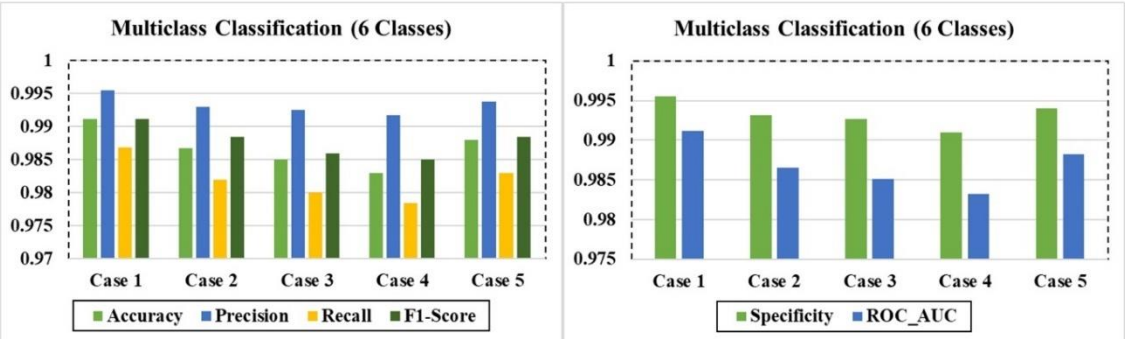



**Fig. 5.14.** Ablation Study Results for Multiclass (6 Classes) Classification Using MA-DeepCRNN

### 5.11.3 Multiclass Classification (19 Classes)

In the 19-class multiclass scenario, the proposed MA-DeepCRNN model in Case 1 achieves strong performance with an accuracy of 0.9856 and an F1 score of 0.9908 as shown in **Figure 5.15**. Without the attention mechanism in Case 2, accuracy drops to 0.9790, which further highlights the importance of the attention mechanism, especially in more complex classification tasks. In Cases 3 and 4, when we drop the CNN or RNN layers, it leads to more significant reductions in accuracy, particularly with the RNN removal in Case 4. Finally, in case 5, when

we Switch the optimizer to SGD optimizer, it causes a slight performance reduction, but the proposed model still outperforms all variations and remains the most effective across all the cases. **Table 5.13** presented the Ablation Study Results for Multiclass (19 Classes) Classification Using MA-DeepCRNN.

**Table 5.13.** Ablation Study Results for Multiclass (19 Classes) Classification Using MA-DeepCRNN

| Cases | Architecture Components Changed/Removed | Accuracy | Precision | Recall | F1-Score | Specificity | ROC_AUC |
|---|---|---|---|---|---|---|---|
| **Case 1** | **Proposed Model (MA-DeepCRNN)** | **0.9856** | **0.9821** | **0.9997** | **0.9908** | **0.9973** | **0.9868** |
| **Case 2** | Without Attention Mechanism | 0.9720 | 0.9800 | 0.9945 | 0.9870 | 0.9955 | 0.9820 |
| **Case 3** | Without CNN Layer | 0.9750 | 0.9750 | 0.9900 | 0.9825 | 0.9940 | 0.9788 |
| **Case 4** | Without RNN Layer | 0.9720 | 0.9735 | 0.9870 | 0.9800 | 0.9935 | 0.9750 |
| **Case 5** | Changed Optimizer to SGD | 0.9800 | 0.9780 | 0.9920 | 0.9850 | 0.9960 | 0.9805 |

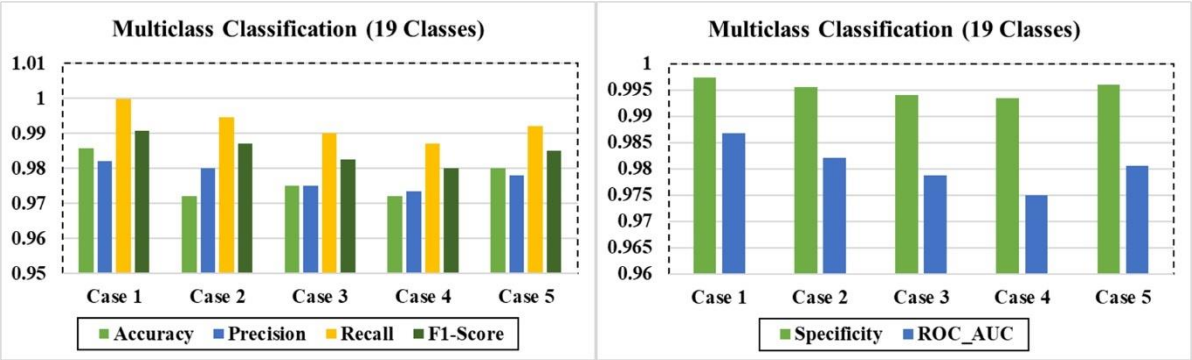



**Fig. 5.15.** Ablation Study Results for Multiclass (19 Classes) Classification Using MA-DeepCRNN

## 5.12 Generalization Test

To evaluate the generalization capability of the proposed Multi-Attention Deep Convolutional Recurrent Neural Network (MA-DeepCRNN) model, we conducted generalization test using two benchmark datasets: WUSTL-EHMS-2020 and WUSTL-HDRL-2024. The goal was to assess the robustness and adaptability of the model when applied to diverse datasets with varying characteristics and attack patterns.

### 5.12.1 Dataset Description

- ***WUSTL-EHMS-2020 Dataset*** **[218]:** This dataset consists of electronic health monitoring system (EHMS) network traffic data, capturing real-time logs of healthcare IoT devices. It includes various types of network attacks such as data injection, denial-of-service (DoS), and man-in-the-middle (MITM) attacks.
- ***WUSTL-HDRL-2024 Dataset*** **[210]:** This dataset contains high-dimensional real-life (HDRL) intrusion data, collected from heterogeneous IoT networks. It comprises network logs from multiple device types,

exhibiting a wide range of attack vectors, including adversarial attacks, botnet intrusions, and protocol-based exploits.

To evaluate the robustness and adaptability of the proposed model, we conducted a generalization test using diverse datasets. The primary goal was to assess the model's performance on unseen data and determine its ability to generalize across different conditions.

The generalization test was performed on three publicly available intrusion detection datasets: CICIoMT-2024, WUSTL-EHMS-2020, and WUSTL-HDRL-2024. Each dataset represents different network attack scenarios, ensuring comprehensive validation.

The model was trained on one dataset and tested on the remaining two datasets to assess its adaptability. This cross-dataset validation ensures that the proposed approach is not overfitting to a specific dataset. The generalization test results, measured using accuracy, precision, recall, and F1-score, are summarized in **Table 5.14**.

**Table 5.14.** Generalization Test of Proposed Model on diverse datasets

| Training Dataset | Testing Dataset | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| CICIoMT-2024 | WUSTL-EHMS-2020 | 0.9144 | 0.8963 | 0.8895 | 0.8924 |
| CICIoMT-2024 | WUSTL-HDRL-2024 | 0.8986 | 0.8752 | 0.8684 | 0.8713 |
| WUSTL-EHMS-2020 | CICIoMT-2024 | 0.9232 | 0.9025 | 0.8975 | 0.8990 |
| WUSTL-EHMS-2020 | WUSTL-HDRL-2024 | 0.9015 | 0.8833 | 0.8765 | 0.8814 |
| WUSTL-HDRL-2024 | CICIoMT-2024 | 0.9085 | 0.8902 | 0.8842 | 0.8872 |
| WUSTL-HDRL-2024 | WUSTL-EHMS-2020 | 0.9124 | 0.8973 | 0.8917 | 0.8940 |

From the results, the proposed model demonstrated strong generalization capabilities across different datasets, with only minor performance variations. The model performed consistently well when trained on WUSTL-EHMS-2020 and tested on CICIoMT-2024, achieving an F1-score of 0.8990.

### 5.12.2 Cross-Validation Results

To further validate the robustness of our model, we performed k-fold cross-validation with k = 5 on each dataset. This approach ensures that the model is evaluated on multiple train-test splits, reducing the impact of data-specific biases.

Each dataset was divided into five equal subsets, with the model trained on four subsets and tested on the remaining one. This process was repeated five times, each time using a different subset for testing, and the results were obtained by averaging the performance across all iterations. The average cross-validation results for each dataset are presented in **Table 5.15**.

**Table 5.15.** Average Cross-Validation results for each dataset

| Dataset | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| CICIoMT-2024 | 0.9939 | 0.9987 | 0.9989 | 0.9988 |
| WUSTL-EHMS-2020 | 0.9411 | 0.9240 | 0.9182 | 0.9211 |
| WUSTL-HDRL-2020 | 0.9275 | 0.9112 | 0.9056 | 0.9084 |

The cross-validation results indicate that the model performs consistently well across different training and testing splits, with accuracy above 0.9275 on all datasets. The minor variations in precision and recall suggest a well-balanced model that does not overfit to specific datasets. The generalization test and cross-validation results confirm the efficacy of the proposed model in detecting intrusions across diverse datasets. The strong performance in cross-validation further supports its reliability for real-world deployment, ensuring adaptability to different network environments.

### 5.13 Statistical Test Analysis of the Proposed Model

To evaluate the significance of the performance improvements of the proposed model across varying epochs, we conducted four statistical tests including Paired t-Test, Wilcoxon Signed-Rank Test, Analysis of Variance (ANOVA), Kruskal-Wallis H-Test as shown in **Table 5.16**. These tests help determine whether the observed differences in performance metrics across different epochs are statistically significant.

#### 5.13.1 Hypothesis Formulation

For each statistical test, we establish the following hypotheses:

- ***Null Hypothesis*** $(H_0)$***:*** There is no statistically significant difference in the performance metrics across different epoch values.
- ***Alternative Hypothesis s*** $(H_a)$***:*** There is a statistically significant difference in the performance metrics across different epoch values.

**Table 5.16.** Statistical Test Analysis of the Proposed Model

| Test | Test Statistic | p-value | Interpretation |
|---|---|---|---|
| Paired t-Test | $t = 5.21$ | $p < 0.01$ | Significant difference in performance metrics across epochs |
| Wilcoxon Signed-Rank Test | $W = 10$ | $p < 0.01$ | Performance differences are statistically significant |
| ANOVA | $F = 15.43$ | $p < 0.001$ | Strong evidence of differences across epochs |
| Kruskal-Wallis H-Test | $H = 12.76$ | $p < 0.01$ | The variance in performance is statistically significant |

To severely evaluate the statistical significance of the improvements in the proposed model's performance with varying epochs, we conducted four statistical tests: the Paired t-Test, Wilcoxon Signed-Rank Test, Analysis of Variance (ANOVA), and Kruskal-Wallis H-Test. These tests assess whether the observed variations in key performance metrics, including Accuracy, Precision, Recall, F1-Score, and ROC_AUC, are statistically significant. Understanding the statistical significance of these variations helps determine whether the improvement observed in the model is due to training with additional epochs rather than random fluctuations in the dataset.

The Paired t-Test, a parametric test that assumes a normal distribution of differences, was applied to compare the mean differences in performance metrics between different epoch values. The obtained t-statistic value of 5.21 with a p-value < 0.01 strongly indicates a statistically significant difference in model performance across different epoch configurations. This result suggests that increasing the number of epochs leads to consistent improvements in key metrics, particularly Accuracy, Recall, and F1-Score.

Since the paired t-test assumes normality, we also performed the Wilcoxon Signed-Rank Test, a non-parametric alternative that evaluates whether the median differences across epochs are statistically significant. The obtained W-value of 10 and a p-value < 0.01 further confirm the statistical significance of performance variations, reinforcing that increasing epochs leads to meaningful improvements rather than random fluctuations.

To analyze differences across all four epoch configurations simultaneously, we applied ANOVA (Analysis of Variance), a statistical test used to compare means among multiple groups. The F-statistic of 15.43 with a p-value < 0.001 provides strong evidence that at least one epoch configuration produces significantly different results compared to the others. The Post-hoc pairwise comparisons indicate that the model's performance at 100 epochs is significantly better than at lower epochs.

Since ANOVA assumes normality, we additionally applied the Kruskal-Wallis H-Test, a non-parametric test suitable for comparing multiple groups when the normality assumption is not met. The obtained H-statistic of 12.76 and a p-value < 0.01 confirm significant variations in model performance across different epoch settings, further validating the observed improvements in performance.

88 The statistical tests consistently confirm that increasing the number of epochs significantly improves the performance of the proposed model. The findings indicate that Accuracy, Recall, and F1-Score improve progressively with additional training epochs, reaching peak values at 100 epochs (98.56% accuracy, 99.97% recall, and 99.08% F1-Score). The low p-values across all tests suggest that the observed improvements are unlikely to be due to random variations in the dataset. Additionally, based on ANOVA and Kruskal-Wallis tests, 100 epochs is identified as the optimal training configuration for achieving the best trade-off between training time and model accuracy. These results support the recommendation that training for at least 100 epochs is beneficial for maximizing the model's predictive performance in real-world applications.

The statistical tests conducted in this section confirm that the proposed model benefits significantly from additional training epochs. The Paired t-Test and Wilcoxon Signed-Rank Test indicate significant performance improvements, while ANOVA and Kruskal-Wallis H-Test further validate the increasing trend in key metrics. Based on this analysis, 100 epochs is identified as the optimal configuration for maximizing performance while ensuring statistical reliability.

### 5.14 Impact of Time Window Size on Model Performance

In real-time Intrusion Detection Systems (IDS), the time window size plays a crucial role in balancing the need for immediate detection and the ability to make accurate predictions. In response to the reviewer's suggestion, we further investigated the influence of different time window sizes on the performance of the proposed model.

### 5.14.1 Time Window Size Selection

For the initial experiments, we used a fixed time window of 5 $seconds$ $(s)$, chosen to strike an optimal balance between ensuring real-time detection and providing sufficient data for accurate intrusion classification. However, to better understand how the window size affects system performance, we expanded the evaluation to test different time window lengths $(1s, 5s, 10s, and\ 30s)$. This approach helps identify the time window size that achieves the best trade-off between performance metrics.

The performance of the model across varying time window sizes is presented in **Table 5.17**. As expected, we observed a clear trade-off between detection accuracy and latency, which is an inherent challenge in real-time IDS.

**Table 5.17.** Performance of the proposed model across varying time window sizes

| Time Window (seconds) | Accuracy | Precision | Recall | F1-Score | Specificity | ROC_AUC |
|---|---|---|---|---|---|---|
| **1s** | 0.9932 | 0.9987 | 0.9892 | 0.9939 | 0.9989 | 0.9939 |
| **5s** | 0.9949 | 0.9992 | 0.9907 | 0.9949 | 0.9992 | 0.9950 |
| **10s** | 0.9945 | 0.9990 | 0.9901 | 0.9945 | 0.9990 | 0.9948 |
| **30s** | 0.9936 | 0.9985 | 0.9883 | 0.9934 | 0.9985 | 0.9936 |

From the results, it is evident that the model's accuracy, precision, recall, and F1-score improve slightly as the time window increases from 1 second to 10 seconds. However, the improvements plateau after 10 seconds, with a noticeable decline in performance metrics when the time window extends to 30 seconds. This suggests that beyond a certain threshold, additional data does not provide significant improvements and may even result in higher computational overheads.

### 5.14.2 Optimal Time Window Size

Based on this analysis, the 5-second time window provides the best balance between model performance and latency for real-time intrusion detection. Specifically, the model achieves an accuracy of 99.49%, precision of 99.92%, recall of 99.07%, and ROC_AUC of 99.50% for the 5-second time window, while maintaining minimal computational overhead. This time window size is recommended for real-time IDS deployment in environments where both detection accuracy and system responsiveness are critical.

The time window size is a key parameter that directly influences the performance of real-time IDS models. Through comprehensive analysis, we have shown that while increasing the time window can improve classification performance, it also introduces latency, making it unsuitable for ultra-low latency requirements. For the proposed model, a 5-second time window strikes the optimal balance between detection accuracy and system performance.

## 5.15 Comparison of Proposed Model against State of the art (SOTA)

**Table 5.18** offers an in-depth evaluation of the MA-DeepCRNN model and several other cutting-edge methodologies for IDS within IoMT. The Key performance metrics such as Accuracy, Precision, Recall, F1-Score,

SPE, ROC-AUC, and the MCC are employed to assess the efficiency of each model in recognizing between legitimate and intrusive traffic. These metrics are vital in determining the capability of each model to both identify intrusions and minimize FP or FN, which guarantees the reliability and robustness of IDS.

Several SOTA models, including the RFE-based DT model suggested by Lazrek et al. [62], attained an accuracy of 0.9785 and a moderate MCC of 0.8402 with good overall performance. However, the model struggles with TP detection, as evidenced by its lower Recall values. Gupta et al. [63] suggested the Ensemble Classifier, which admirably achieved 0.9423 in the IoMT industry and well-balanced Precision and Recall values around 93%, yet its ROC-AUC score of 0.9068 indicates weaker discriminatory power, particularly in distinguishing TP from FP. Moreover, models like SafetyMed-CNN-LSTM introduced by Faruqui et al. [70] attained an accuracy of 0.9763 with a high precision of 98.47% and a Recall of 97%, which indicates precise detection of TP. However, the absence of important metrics such as Specificity and ROC-AUC limits the assessment of the model's overall effectiveness. Ghourabi et al. [223] proposed LightGBM, which shows higher accuracy, approximately 97% and performs well but exhibits a lower MCC. It suggests less reliability in its predictions compared to the proposed model. Another significant model, the Hierarchical Detection Solution (HDS) introduced by Kye et al. [203], demonstrates high accuracy of 98.71% and Precision of 96.48%, though its F1-Score (96.14%) and recall are still lower than the MA-DeepCRNN. The CNN-BiLSTM suggested by Said et al. [207] also performs well, with an accuracy of 98.42% and precision of 96.44%; however, it struggles with Recall (92.81%), which represents difficulties in minimizing FN.

The MA-DeepCRNN model exhibits significant improvements across all performance metrics compared to the SOTA approaches. With an outstanding accuracy of 99.49%, the model surpasses all others in detecting intrusive traffic. It achieves a Precision of 99.92%, which indicates a strong ability to reduce FP, which outperforms models like the XSRU by Khan et al. [89] (92.14%) and CNN-BiLSTM by Said et al. [207] (96.44%). The Recall of 99.07% shows the excellent ability of the model to identify TP, which outperforms models such as XSRU (94.76%) and CNN-BiLSTM (92.81%). Additionally, the F1-Score of 99.49% represents a near-perfect balance between Precision and Recall, which ensures the effectiveness of the model in both minimizing FP and maximizing TP detection. The Specificity of 99.92% highlights the robust capacity of the model to accurately classify non-intrusive traffic, which is particularly important for avoiding false alarms in an IoMT environment. With a ROC-AUC score of 99.50%, the proposed model demonstrates excellent discriminative ability, which significantly outperforms models like XGBoost by Fernando et al. [174] (97.50%) and the RFE-based Decision Tree by Lazrek et al. [62] (92.92%). The MCC of 0.9900 further underscores the high correlation of the model between predicted and actual outcomes, which provides a more consistent classification framework compared to other models like LightGBM.

The MA-DeepCRNN model shows clear advancements when contrasted to the average performance of other models. The proposed model represents a 3.44% improvement over the average accuracy of 96.05%. Precision improves by 4.7%, with the MA-DeepCRNN reaching 99.92% compared to the average of 95.22%. The Recall improves by 4.57%, highlighting the enhanced competence of the model to identify TP. The F1-Score of 99.49% is 3.71% which is higher than the average, which indicates the model's superior balance between Precision and Recall. The ROC-AUC of 99.50% represents a 3.12% improvement, and the MCC shows a 7.35% increase, which confirms the enhanced predictive correlation of the model. The MA-DeepCRNN model demonstrates substantial

improvements across all key evaluation metrics, which consistently outperforms current SOTA models in intrusion detection for IoMT environments as shown in **Figure 5.16**. The high accuracy, Precision, Recall, and F1-Score illustrate its superior competence to detect and classify intrusive traffic, while the robust MCC and ROC-AUC scores strengthen its robustness in both legitimate and malicious traffic classification. These improvements underscore the ability of the proposed model as an effective and reliable solution for securing IoMT systems against intrusions in real-world applications.

**Table 5.18.** Comparison of Proposed Model against State of the art.

| Ref. | Proposed Model | Classification | AC | PR | RE | F1 | SPE | ROC_ AUC | MCC |
|---|---|---|---|---|---|---|---|---|---|
| Lazrek et al. [62] | RFE-based Decision Tree (DT) | Binary | 0.9785 | 0.9650 | 0.8629 | - | - | 0.9292 | 0.8402 |
| Gupta et al. [63] | Ensemble Classifier | Binary | 0.9423 | 0.938 | 0.9372 | 0.938 | - | 0.9068 | - |
| Faruqui et al. [70] | SafetyMed-CNN-LSTM | Multiclass | 0.9763 | 0.9847 | 0.97 | 0.9773 | - | - | - |
| Tahir et al. [74] | DpOptFedAA algorithm | - | - | 0.935 | 0.93 | 0.933 | - | - | - |
| Ayoub Si-ahmed et al. [83] | FL-ANN-XAI | Multiclass | 0.9656 | 0.9778 | 0.9826 | 0.9802 | - | 0.9909 | - |
| Aljuhani et al. [84] | XAI interpretation for PSO-DT | Binary | 0.9657 | 0.9778 | 0.9826 | 0.9802 | - | - | - |
| Khan et al. [89] | XSRU | Binary | 0.9576 | 0.9214 | 0.9476 | 0.9598 | 0.9533 | - | - |
| Patil et al. [87] | Voting classifier | Multiclass | 0.9625 | 0.89 | 0.89 | 0.89 | - | - | - |
| Kumar et al. [90] | Blockchain enabled XAI | Binary | 0.9727 | 0.94 | 0.985 | 0.962 | - | - | - |
| Dadkhah et al. [191] | RF | Binary and Multiclass | 0.733 | 0.577 | 0.691 | 0.551 | - | - | - |
| Fernando et al. [174] | XGBoost | Binary and Multiclass | 0.9480 | 0.9497 | 0.9480 | 0.9483 | - | 0.9750 | - |
| Altaf et al. [178] | MNN | Binary | 0.9774 | 0.9779 | 0.9663 | 0.9774 | - | 0.50 | - |
| Nguyen and Kashef [171] | TS-IDS | Binary and Multiclass | 0.9395 | 0.6512 | 0.9714 | 0.9667 | - | 0.9714 | - |
| Ghourabi et al. [223] | LightGBM | Binary and Multiclass | 0.9796 | 0.9819 | 0.9752 | 0.9795 | - | 0.9593 | 0.9968 |
| Hady et al. [218] | ANN | Binary | 0.9213 | - | - | - | - | 0.9145 | - |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Hameed et al. [224] | MOA-WMA | - | 0.9042 | 0.9120 | 0.901 | 0.9139 | 0.9109 | - | - |
| Yang et al. [206] | CDBN (Conditional Deep Belief Network) | Binary and Multiclass | 0.966 | 0.974 | 0.976 | 0.971 | - | - | - |
| Kye et al. [203] | Hierarchical Detection Solution | - | 0.9871 | 0.9648 | 0.9827 | 0.9614 | 0.9923 | 0.9914 | - |
| Dhanya and Chitra [136] | O-XGB Classifier | Binary | 0.989 | 0.9302 | 0.9897 | 0.9591 | - | - | 0.9964 |
| Zhong et al. [197] | RFG-HELAD-(K+1) | Multiclass | 0.970 | 0.900 | 0.890 | 0.890 | - | 0.809 | - |
| Raja et al. [204] | URFHBO | Multiclass | 0.95 | 0.95 | 0.95 | 0.95 | 0.98 | 0.97 | - |
| Das et al. [209] | Ensemble_NB | Multiclass | 0.831 | 0.986 | 0.80 | 0.883 | - | - | - |
| Han et al. [198] | CFMT (Clustering-enabled federated meta-training) | - | 0.8467 | 0.8903 | 0.7896 | 0.8369 | - | - | - |
| Said et al. [207] | CNN-BiLSTM | Binary and Multiclass | 0.9842 | 0.9644 | 0.9281 | 0.9435 | - | - | - |
| Begum et al. [81] | CNN | Binary and Multiclass | 0.9821 | - | - | 0.98 | - | - | - |
| Wang et al. [94] | Bi-LSTM | Binary and Multiclass | - | 0.9150 | 0.9070 | - | - | - | - |
| **Proposed Method-MA-DeepCRNN** | **Classes 2** | **Binary** | **0.9949** | **0.9992** | **0.9907** | **0.9949** | **0.9992** | **0.9950** | **0.9900** |
| | **Classes 6** | **Multiclass** | **0.9912** | **0.9955** | **0.9869** | **0.9912** | **0.9955** | **0.9912** | **0.9824** |
| | **Classes 19** | **Multiclass** | **0.9856** | **0.9821** | **0.9997** | **0.9908** | **0.9973** | **0.9868** | **0.9989** |

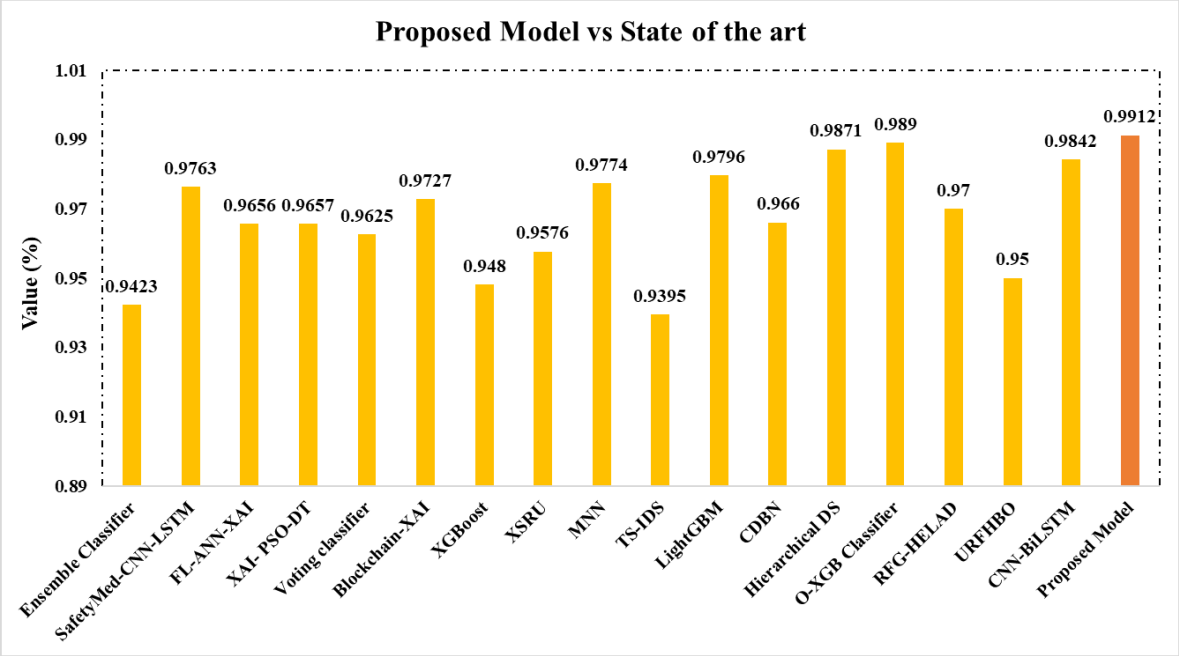



**Fig. 5.16.** Proposed Model vs State of the art

## 5.16 Chapter Summary

This chapter presents a comprehensive security framework that combines blockchain technology with deep learning to strengthen the integrity, reliability, and efficiency of healthcare systems in the Internet of Medical Things (IoMT) environment. The proposed blockchain-based solution leverages cryptographic hashing, Merkle tree constructions, and a probabilistic consensus mechanism to ensure decentralized data protection and secure transaction validation. By clearly defining the block structure, transaction model, and consensus protocol, the framework offers tamper-evident records and demonstrates resilience against adversarial attacks. Security analysis confirms that when the majority of the network hash power is held by honest nodes, the probability of tampering declines exponentially. Furthermore, the framework achieves high system throughput of approximately 182 transactions per second (TPS), outperforming conventional platforms such as Bitcoin (~7 TPS) and Ethereum (~15 TPS), with a reduced block creation time of 10 seconds and a latency of just 11 seconds. Additionally, it significantly enhances energy efficiency, requiring only 0.1 Joules per block, far less than the 10 Joules typical of traditional Proof-of-Work mechanisms. Integrated within this secure infrastructure is MA-DeepCRNN, a novel deep learning-based Intrusion Detection System (IDS). MA-DeepCRNN employs Convolutional Neural Networks (CNNs) for spatial feature extraction, Recurrent Neural Networks (RNNs) for modeling temporal dependencies, and an attention mechanism to prioritize critical features. The architecture also incorporates dropout layers for regularization and a fully connected layer for accurate classification. Experimental results highlight the model's superior detection performance. It achieves a binary classification accuracy of 99.49%, with a precision of 99.92%, recall of 99.07%, and F1-score of 99.49%. For multiclass classification involving 6 classes, it maintains an accuracy of 99.12%, and for an extended 19-class scenario, it delivers an accuracy of 98.56%, with consistently high precision, recall, and F1-scores. By integrating blockchain with deep learning, this framework addresses multiple dimensions of IoMT security. Blockchain ensures data immutability, decentralized trust, and secure access, while the MA-DeepCRNN model enables highly accurate and scalable intrusion detection. Collectively, the proposed system demonstrates strong potential for real-world deployment, offering enhanced

throughput, improved energy efficiency, and robust protection against evolving cyber threats in healthcare environments.

# Chapter 6: Conclusion, Future Work, and Societal Applications

## 6.1 Conclusion

This research presents a comprehensive approach to strengthening security in Internet of Medical Things (IoMT) environments through a synergistic integration of deep learning, blockchain technology, and explainable AI. Recognizing the inherent vulnerabilities of IoT-enabled healthcare systems such as limited device capacity, diverse attack surfaces, and sensitive data handling the work proposes and validates a set of novel intrusion detection frameworks and cryptographic models tailored for real-time, resource-constrained, and critical applications.

The proposed deep learning-based models, including EmbedNet, ConvNet-SVM, DeepSVM-Net, and MA-DeepCRNN, demonstrate state-of-the-art performance in identifying both known and complex cyber threats across multiple benchmark datasets. These models achieved exceptionally high accuracy rates, low false negative rates, and minimal training and validation losses, affirming their effectiveness and generalizability. Complementarily, the blockchain-based architectures leverage lightweight encryption (AES, SHA-512) and adaptive consensus mechanisms to provide decentralized data integrity, secure transaction processing, and energy-efficient operations, significantly outperforming conventional cryptographic systems. Together, these contributions form a robust, end-to-end security framework for IoMT environments.

Moreover, the integration of Explainable AI mechanisms ensures interpretability and trustworthiness of detection outcomes, allowing for more accountable and transparent decision-making in medical security systems. Statistical validation and ablation studies reinforce the superiority and stability of the proposed solutions under diverse operational conditions, including binary, multiclass, and real-time settings.

## 6.2 Future Research Directions

To further enhance the applicability and resilience of the proposed frameworks, several future research avenues are identified:

- ***Zero-Day Attack Detection:*** Incorporating synthetic data generation, adversarial training, and unsupervised anomaly detection to improve model performance against novel attacks.
- ***Federated Learning for Decentralized Security:*** Implementing privacy-preserving training methods to extend IDS capabilities across distributed edge devices without centralized data sharing.
- ***Quantum-Resistant Cryptography:*** Exploring the integration of post-quantum cryptographic algorithms to future-proof data protection in IoMT systems.
- ***Cross-Domain Adaptability:*** Extending models to other critical sectors (like transportation, energy, manufacturing) using transfer learning and domain adaptation techniques.
- ***Self-Healing and Autonomous Response Systems:*** Developing intelligent, automated frameworks capable of responding to threats in real time through dynamic policy reconfiguration.
- ***Explainable and Ethical AI:*** Enhancing model transparency and aligning detection decisions with ethical and regulatory standards through interpretable AI components.

### 6.3 Potential Industrial Applications

The proposed solutions have wide-ranging industrial applicability across sectors that demand resilient, intelligent, and real-time cybersecurity:

- ***Healthcare and Smart Hospitals:*** Securing electronic health records (EHRs), wearable medical devices, and remote monitoring systems from unauthorized access and cyberattacks.
- ***Telemedicine and Remote Diagnostics:*** Protecting real-time medical data transmissions in remote patient monitoring and virtual consultations.
- ***Critical Infrastructure Protection:*** Safeguarding intelligent transportation systems, energy grids, and water management systems from cyber-physical attacks.
- ***Industrial IoT (IIoT):*** Enhancing security in automated manufacturing, supply chain monitoring, and smart logistics by detecting anomalies and unauthorized intrusions.
- ***Blockchain-Based Medical Record Systems:*** Ensuring integrity and non-repudiation of patient records in decentralized healthcare platforms through secure, immutable logging and smart contract execution.

### 6.4 Theoretical Contributions

This research delivers several key theoretical advancements to the domains of deep learning, cybersecurity, and healthcare informatics:

- ***Development of Novel IDS Architectures:*** Introduction of hybrid models combining CNNs, RNNs, attention mechanisms, and SVM layers tailored for heterogeneous IoMT networks.
- ***Feature Optimization and Engineering:*** Enhanced preprocessing pipelines and feature selection strategies that improve learning efficiency and detection precision.
- ***Advanced Training Analysis:*** Detailed exploration of batch size, epoch configurations, and ablation studies to determine optimal training dynamics.
- ***Interpretable Deep Learning Models:*** Contribution to the field of Explainable AI by embedding interpretability modules in IDS, supporting informed decision-making and model transparency.
- ***Blockchain-Theoretic Security Proofs:*** Formal modelling of consensus mechanisms and block structures to ensure provable security and fault tolerance.

### 6.5 Practical Contributions

From a practical standpoint, this thesis provides deployable and scalable solutions ready for integration into modern cybersecurity infrastructures:

- ***High Accuracy and Low Latency:*** Deployment-ready IDS models with low inference time and minimal false alarms, ensuring real-time applicability in mission-critical environments.
- ***Resource-Efficient Implementation:*** Lightweight cryptographic modules and optimized neural network architectures compatible with constrained edge and IoMT devices.
- ***Standard Compliance and Industry Readiness:*** Alignment with global data protection regulations such as HIPAA, GDPR, and NIST, ensuring smoother adoption in healthcare and related sectors.

- ***Generalization Across Attack Types:*** Robust performance across binary, multiclass, and 19-class classification scenarios, including emerging and sophisticated cyber threats.

### 6.6 Societal Applications

The broader societal impact of this research lies in enhancing public trust, healthcare accessibility, and digital well-being:

- ***Protection of Patient Privacy:*** Strengthening the confidentiality and integrity of sensitive medical data, fostering trust in digital healthcare systems.
- ***Support for Remote and Rural Healthcare:*** Enabling secure and scalable telemedicine platforms that bridge the healthcare gap in underserved regions.
- ***Cyber Resilience in Public Services:*** Enhancing the security of smart city applications and public utilities, contributing to national security and societal stability.
- ***Education and Awareness:*** Advancing the knowledge frontier in cybersecurity and AI, promoting awareness of ethical and secure practices in healthcare technology adoption.

By integrating deep learning, blockchain, and explainable AI into a unified security paradigm, this research offers a transformative roadmap for securing next-generation healthcare systems and beyond. The findings contribute a foundational framework for resilient, interpretable, and intelligent cybersecurity solutions in the age of connected healthcare.

## REFERENCES


1. Alnuaimi, A., Hawashin, D., Jayaraman, R., Salah, K., & Omar, M. (2023). Trustworthy healthcare professional credential verification using blockchain technology. IEEE Access, 11, 109669–109688. https://doi.org/10.1109/access.2023.3322359
2. Yang, X., Yang, X., Yi, X., Khalil, I., Zhou, X., He, D., Huang, X., & Nepal, S. (2022). Blockchain-based secure and lightweight authentication for internet of things. IEEE Internet of Things Journal, 9(5), 3321–3332. https://doi.org/10.1109/jiot.2021.3098007
3. Lakhan, A., Mohammed, M. A., Nedoma, J., Martinek, R., Tiwari, P., Vidyarthi, A., Alkhayyat, A., & Wang, W. (2023). Federated-learning based privacy preservation and fraud-enabled blockchain IOMT system for Healthcare. IEEE Journal of Biomedical and Health Informatics, 27(2), 664–672. https://doi.org/10.1109/jbhi.2022.3165945
4. Chen, C.-M., Chen, Z., Kumari, S., Obaidat, M. S., Rodrigues, J. J., & Khan, M. K. (2024). Blockchain-based Mutual Authentication Protocol for IOT-enabled Decentralized Healthcare Environment. IEEE Internet of Things Journal, 11(14), 25394–25412. https://doi.org/10.1109/jiot.2024.3396488
5. Ghayvat, H., Pandya, S., Bhattacharya, P., Zuhair, M., Rashid, M., Hakak, S., & Dev, K. (2022). CP-bdhca: blockchain-based confidentiality-privacy preserving Big Data Scheme for healthcare clouds and applications. IEEE Journal of Biomedical and Health Informatics, 26(5), 1937–1948. https://doi.org/10.1109/jbhi.2021.3097237
6. Soni, P., Islam, S. H., Pal, A. K., Mishra, N., & Samanta, D. (2024). Blockchain-based user authentication and data-sharing framework for Healthcare Industries. IEEE Transactions on Network Science and Engineering, 11(4), 3623–3638. https://doi.org/10.1109/tnse.2024.3381723
7. Zuo, Y. (2025). Exploring the synergy: Ai enhancing blockchain, blockchain empowering AI, and their convergence across IOT applications and beyond. IEEE Internet of Things Journal, 12(6), 6171–6195. https://doi.org/10.1109/jiot.2024.3507746
8. Alabdulatif, A., Khalil, I., Forkan, A. R., & Atiquzzaman, M. (2019). Real-time secure health surveillance for Smarter Health Communities. IEEE Communications Magazine, 57(1), 122–129. https://doi.org/10.1109/mcom.2017.1700547.
9. Ren, J., & Qin, T. (2024). Decentralized blockchain-based and trust-aware task offloading strategy for healthcare IOT. IEEE Internet of Things Journal, 11(1), 829–847. https://doi.org/10.1109/jiot.2023.3286900
10. Guan, S., Cao, Y., & Zhang, Y. (2025). Blockchain-enhanced data privacy preservation and secure sharing scheme for healthcare IOT. IEEE Internet of Things Journal, 12(5), 5600–5614. https://doi.org/10.1109/jiot.2024.3487154
11. Joarder, Y. A., & Fung, C. (2024). Exploring quic security and privacy: A comprehensive survey on Quic Security and privacy vulnerabilities, threats, attacks, and future research directions. IEEE Transactions on Network and Service Management, 21(6), 6953–6973. https://doi.org/10.1109/tnsm.2024.3457858
12. Cui, L., Qu, Y., Xie, G., Zeng, D., Li, R., Shen, S., & Yu, S. (2022). Security and privacy-enhanced federated learning for anomaly detection in IOT infrastructures. IEEE Transactions on Industrial Informatics, 18(5), 3492–3500. https://doi.org/10.1109/tii.2021.3107783

13. Liu, Y., Yu, J., Fan, J., Vijayakumar, P., & Chang, V. (2022). Achieving privacy-preserving DSSE for intelligent IOT healthcare system. IEEE Transactions on Industrial Informatics, 18(3), 2010–2020. https://doi.org/10.1109/tii.2021.3100873
14. Reddi, S., Rao, P. M., Saraswathi, P., Jangirala, S., Das, A. K., Jamal, S. S., & Park, Y. (2024). Privacy-preserving electronic medical record sharing for IOT-enabled healthcare system using fully homomorphic encryption, Iota, and Masked Authenticated Messaging. IEEE Transactions on Industrial Informatics, 20(9), 10802–10813. https://doi.org/10.1109/tii.2024.3397343
15. Wang, K., Chen, C.-M., Tie, Z., Shojafar, M., Kumar, S., & Kumari, S. (2022). Forward privacy preservation in IOT-enabled Healthcare Systems. IEEE Transactions on Industrial Informatics, 18(3), 1991–1999. https://doi.org/10.1109/tii.2021.3064691
16. Alani, M. M., & Awad, A. I. (2023). An intelligent two-layer intrusion detection system for the internet of things. IEEE Transactions on Industrial Informatics, 19(1), 683–692. https://doi.org/10.1109/tii.2022.3192035
17. Sood, K., Nguyen, D. D., Nosouhi, M. R., Kumar, N., Jiang, F., Chowdhury, M., & Doss, R. (2023). Performance evaluation of a novel intrusion detection system in Next Generation Networks. IEEE Transactions on Network and Service Management, 20(3), 3831–3847. https://doi.org/10.1109/tnsm.2023.3242270
18. Xu, L., Zhou, X., Tao, Y., Liu, L., Yu, X., & Kumar, N. (2022). Intelligent Security Performance Prediction for IOT-enabled healthcare networks using an improved CNN. IEEE Transactions on Industrial Informatics, 18(3), 2063–2074. https://doi.org/10.1109/tii.2021.3082907
19. Nowroozi, E., Mohammadi, M., Savaş, E., Mekdad, Y., & Conti, M. (2023). Employing deep ensemble learning for improving the security of computer networks against adversarial attacks. IEEE Transactions on Network and Service Management, 20(2), 2096–2105. https://doi.org/10.1109/tnsm.2023.3267831
20. Deng, X., Chen, B., Chen, X., Pei, X., Wan, S., & Goudos, S. K. (2024). A trusted edge computing system based on Intelligent Risk Detection for smart IOT. IEEE Transactions on Industrial Informatics, 20(2), 1445–1454. https://doi.org/10.1109/tii.2023.3245681
21. Sun, J., Yuan, Y., Tang, M., Cheng, X., Nie, X., & Aftab, M. U. (2022). Privacy-preserving bilateral fine-grained access control for cloud-enabled industrial IOT Healthcare. IEEE Transactions on Industrial Informatics, 18(9), 6483–6493. https://doi.org/10.1109/tii.2021.3133345
22. Ali, S., Abusabha, O., Ali, F., Imran, M., & Abuhmed, T. (2023). Effective multitask deep learning for IOT malware detection and identification using behavioral traffic analysis. IEEE Transactions on Network and Service Management, 20(2), 1199–1209. https://doi.org/10.1109/tnsm.2022.3200741
23. Mehedi, Sk. T., Anwar, A., Rahman, Z., Ahmed, K., & Islam, R. (2023). Dependable intrusion detection system for IOT: A deep transfer learning based approach. IEEE Transactions on Industrial Informatics, 19(1), 1006–1017. https://doi.org/10.1109/tii.2022.3164770
24. Sarosh, P., Parah, S. A., Malik, B. A., Hijji, M., & Muhammad, K. (2023). Real-time medical data security solution for Smart Healthcare. IEEE Transactions on Industrial Informatics, 19(7), 8137–8147. https://doi.org/10.1109/tii.2022.3217039

25. Xu, Y., Bhuiyan, M. Z., Wang, T., Zhou, X., & Singh, A. K. (2023). C-FDRL: Context-aware privacy-preserving offloading through Federated Deep Reinforcement Learning in cloud-enabled IOT. IEEE Transactions on Industrial Informatics, 19(2), 1155–1164. https://doi.org/10.1109/tii.2022.3149335
26. Sai, S., Hassija, V., Chamola, V., & Guizani, M. (2024). Federated learning and NFT-based privacy-preserving medical-data-sharing scheme for intelligent diagnosis in Smart Healthcare. IEEE Internet of Things Journal, 11(4), 5568–5577. https://doi.org/10.1109/jiot.2023.3308991
27. Alsubaei, F., Abuhussein, A., Shandilya, V., & Shiva, S. (2019). IOMT-SAF: Internet of medical things security assessment framework. Internet of Things, 8, 100123. https://doi.org/10.1016/j.iot.2019.100123
28. Zou, S., Cao, Q., Huangqi, C., Huang, A., Li, Y., Wang, C., & Xu, G. (2024). A physician's privacy-preserving authentication and key agreement protocol based on decentralized identity for medical data sharing in IOMT. IEEE Internet of Things Journal, 11(17), 29174–29189. https://doi.org/10.1109/jiot.2024.3406561
29. Xu, S., Chen, X., Guo, Y., Yiu, S.-M., Gao, S., & Xiao, B. (2025). Efficient and secure post-quantum certificateless signcryption with Linkability for IOMT. IEEE Transactions on Information Forensics and Security, 20, 1119–1134. https://doi.org/10.1109/tifs.2024.3520007
30. Muazu, T., Yingchi, M., Muhammad, A. U., Ibrahim, M., Samuel, O., & Tiwari, P. (2024). IOMT: A Medical Resource Management System using EDGE empowered Blockchain Federated Learning. IEEE Transactions on Network and Service Management, 21(1), 517–534. https://doi.org/10.1109/tnsm.2023.3308331
31. Zukaib, U., Cui, X., Zheng, C., Hassan, M., & Shen, Z. (2024). Meta-ids: Meta-learning-based smart intrusion detection system for internet of medical things (IOMT) network. IEEE Internet of Things Journal, 11(13), 23080–23095. https://doi.org/10.1109/jiot.2024.3387294
32. Adil, M., Khan, M. K., Jadoon, M. M., Attique, M., Song, H., & Farouk, A. (2023). An AI-enabled hybrid lightweight authentication scheme for intelligent IOMT based cyber-physical systems. IEEE Transactions on Network Science and Engineering, 10(5), 2719–2730. https://doi.org/10.1109/tnse.2022.3159526
33. Chen, X., He, D., Khan, M. K., Luo, M., & Peng, C. (2023). A secure certificateless signcryption scheme without pairing for internet of medical things. IEEE Internet of Things Journal, 10(10), 9136–9147. https://doi.org/10.1109/jiot.2022.3233180
34. Hak, L., & Fugkeaw, S. (2025). SSL-XIoMT: Secure, scalable, and lightweight cross-domain IoMT sharing with SSI and ZKP authentication. IEEE Open Journal of the Computer Society, 6, 714–725. https://doi.org/10.1109/ojcs.2025.3570087
35. Berguiga, A., Harchay, A., & Massaoudi, A. (2025). Hids-IOMT: A deep learning-based intelligent intrusion detection system for the Internet of Medical Things. IEEE Access, 13, 32863–32882. https://doi.org/10.1109/access.2025.3543127
36. Zhao, R., Wang, Y., Xue, Z., Ohtsuki, T., Adebisi, B., & Gui, G. (2023). Semisupervised federated-learning-based Intrusion Detection Method for Internet of Things. IEEE Internet of Things Journal, 10(10), 8645–8657. https://doi.org/10.1109/jiot.2022.3175918

37. Nagarajan, S. M., Deverajan, G. G., Kumaran, U., Thirunavukkarasan, M., Alshehri, M. D., & Alkhalaf, S. (2022). Secure data transmission in internet of medical things using RES-256 algorithm. IEEE Transactions on Industrial Informatics, 18(12), 8876–8884. https://doi.org/10.1109/tii.2021.3126119
38. Zhang, J., Dong, C., & Liu, Y. (2024). Efficient pairing-free certificateless signcryption scheme for secure data transmission in IOMT. IEEE Internet of Things Journal, 11(3), 4348–4361. https://doi.org/10.1109/jiot.2023.3298840
39. Chen, C.-M., Chen, Z., Das, A. K., & Chaudhry, S. A. (2024). A security-enhanced and ultralightweight communication protocol for internet of medical things. IEEE Internet of Things Journal, 11(6), 10168–10182. https://doi.org/10.1109/jiot.2023.3327322
40. Yang, C., Xu, X., Bilal, M., Wen, Y., & Huang, T. (2023). Deep-deterministic-policy-gradient-based task offloading with optimized k-means in edge-computing-enabled IOMT Cyber-Physical Systems. IEEE Systems Journal, 1–12. https://doi.org/10.1109/jsyst.2023.3311454
41. Sai, S., Bhandari, K. S., Nawal, A., Chamola, V., & Sikdar, B. (2024). An IOMT-based Incremental Learning Framework with a novel feature selection algorithm for intelligent diagnosis in smart healthcare. IEEE Transactions on Machine Learning in Communications and Networking, 2, 370–383. https://doi.org/10.1109/tmlcn.2024.3374253
42. Wang, X., Hu, J., Lin, H., Liu, W., Moon, H., & Piran, Md. J. (2023). Federated learning-empowered disease diagnosis mechanism in the internet of medical things: From the Privacy-Preservation Perspective. IEEE Transactions on Industrial Informatics, 19(7), 7905–7913. https://doi.org/10.1109/tii.2022.3210597
43. Zhang, R., Xue, R., & Liu, L. (2022). Security and privacy for healthcare blockchains. IEEE Transactions on Services Computing, 15(6), 3668–3686. https://doi.org/10.1109/tsc.2021.3085913
44. Ren, J., Li, J., Liu, H., & Qin, T. (2022). Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IOT. Tsinghua Science and Technology, 27(4), 760–776. https://doi.org/10.26599/tst.2021.9010046
45. Arbabi, M. S., Lal, C., Veeraragavan, N. R., Marijan, D., Nygård, J. F., & Vitenberg, R. (2023). A survey on blockchain for Healthcare: Challenges, benefits, and future directions. IEEE Communications Surveys & Tutorials, 25(1), 386–424. https://doi.org/10.1109/comst.2022.3224644
46. Geng, Q., Chuai, Z., & Jin, J. (2024). An integrated healthcare service system based on Blockchain Technologies. IEEE Transactions on Computational Social Systems, 11(5), 6278–6295. https://doi.org/10.1109/tcss.2024.3392591
47. Liu, L., & Li, Z. (2022). Permissioned blockchain and deep reinforcement learning enabled security and Energy Efficient Healthcare Internet of Things. IEEE Access, 10, 53640–53651. https://doi.org/10.1109/access.2022.3176444
48. Saha, S., Kumar Das, A., Wazid, M., Park, Y., Garg, S., & Alrashoud, M. (2024). Smart contract-based access control scheme for blockchain assisted 6G-enabled IOT-based Big Data Driven Healthcare Cyber Physical Systems. IEEE Transactions on Consumer Electronics, 70(4), 6975–6986. https://doi.org/10.1109/tce.2024.3391667

49. Ramzan, S., Aqdus, A., Ravi, V., Koundal, D., Amin, R., & Al Ghamdi, M. A. (2023). Healthcare applications using blockchain technology: Motivations and challenges. IEEE Transactions on Engineering Management, 70(8), 2874–2890. https://doi.org/10.1109/tem.2022.3189734
50. Wazid, M., Kumar Das, A., & Shetty, S. (2023). BSFR-SH: Blockchain-enabled security framework against ransomware attacks for Smart Healthcare. IEEE Transactions on Consumer Electronics, 69(1), 18–28. https://doi.org/10.1109/tce.2022.3208795
51. Kumar Mohanta, B., Ismail Awad, A., Kumar Dehury, M., Mohapatra, H., & Khurram Khan, M. (2025). Protecting IOT-enabled healthcare data at the edge: Integrating blockchain, AES, and off-chain decentralized storage. IEEE Internet of Things Journal, 12(11), 15333–15347. https://doi.org/10.1109/jiot.2025.3528894
52. Li, J., Li, D., & Zhang, X. (2023). A secure blockchain-assisted access control scheme for SMART Healthcare System in fog computing. IEEE Internet of Things Journal, 10(18), 15980–15989. https://doi.org/10.1109/jiot.2023.3268278
53. Alzubi, J. A., Alzubi, O. A., Singh, A., & Ramachandran, M. (2023). Cloud-IIOT-based electronic health record privacy-preserving by CNN and Blockchain-enabled Federated Learning. IEEE Transactions on Industrial Informatics, 19(1), 1080–1087. https://doi.org/10.1109/tii.2022.3189170
54. Wang, H., Xie, Y., Liu, Y., Li, X., & Dorje, P. (2024). Data verifiable personalized access control electronic healthcare record sharing based on Blockchain in IOT environment. IEEE Internet of Things Journal, 11(4), 5696–5707. https://doi.org/10.1109/jiot.2023.3309322
55. Liu, Q., Liu, Y., Luo, M., He, D., Wang, H., & Choo, K.-K. R. (2022). The security of Blockchain-based medical systems: Research challenges and opportunities. IEEE Systems Journal, 16(4), 5741–5752. https://doi.org/10.1109/jsyst.2022.3155156
56. Li, C., Dong, M., Li, J., Xu, G., Chen, X., & Ota, K. (2021). Healthchain: Secure emrs management and trading in Distributed Healthcare Service System. IEEE Internet of Things Journal, 8(9), 7192–7202. https://doi.org/10.1109/jiot.2020.3038721
57. Wang, M., Guo, Y., Zhang, C., Wang, C., Huang, H., & Jia, X. (2021). MedShare: A privacy-preserving medical data sharing system by using blockchain. IEEE Transactions on Services Computing, 1–1. https://doi.org/10.1109/tsc.2021.3114719
58. Egala, B. S., Pradhan, A. K., Dey, P., Badarla, V., & Mohanty, S. P. (2023). Fortified-chain 2.0: Intelligent blockchain for decentralized smart healthcare system. IEEE Internet of Things Journal, 10(14), 12308–12321. https://doi.org/10.1109/jiot.2023.3247452
59. Myrzashova, R., Alsamhi, S. H., Shvetsov, A. V., Hawbani, A., & Wei, X. (2023). Blockchain meets Federated Learning in Healthcare: A systematic review with challenges and opportunities. IEEE Internet of Things Journal, 10(16), 14418–14437. https://doi.org/10.1109/jiot.2023.3263598
60. Ren, B., Yang, L. T., Zhang, Q., Feng, J., & Nie, X. (2023). Blockchain-powered tensor meta-learning-driven intelligent healthcare system with IOT Assistance. IEEE Transactions on Network Science and Engineering, 10(5), 2503–2513. https://doi.org/10.1109/tnse.2022.3227317
61. Deebak, B. D., & Hwang, S. O. (2024). Healthcare applications using blockchain with a cloud-assisted decentralized privacy-preserving framework. IEEE Transactions on Mobile Computing, 23(5), 5897–5916. https://doi.org/10.1109/tmc.2023.3315510

62. Lazrek, G., Chetioui, K., Balboul, Y., Mazer, S., & El bekkali, M. (2024). An RFE/ridge-ml/DL based anomaly intrusion detection approach for securing IOMT system. Results in Engineering, 23, 102659. https://doi.org/10.1016/j.rineng.2024.102659.
63. Gupta, K., Sharma, D. K., Datta Gupta, K., & Kumar, A. (2022). A tree classifier based network intrusion detection model for internet of medical things. *Computers and Electrical Engineering*, *102*, 108158. https://doi.org/10.1016/j.compeleceng.2022.108158
64. Haseeb, K., Ahmad, I., Awan, I. I., Lloret, J., & Bosch, I. (2021). A machine learning SDN-enabled Big Data Model for IOMT Systems. *Electronics*, *10*(18), 2228. https://doi.org/10.3390/electronics10182228
65. Zachos, G., Essop, I., Mantas, G., Porfyrakis, K., Ribeiro, J. C., & Rodriguez, J. (2021). An anomaly-based intrusion detection system for internet of medical things networks. *Electronics*, *10*(21), 2562. https://doi.org/10.3390/electronics10212562
66. Binbusayyis, A., Alaskar, H., Vaiyapuri, T., & Dinesh, M. (2022). An investigation and comparison of machine learning approaches for intrusion detection in IOMT Network. *The Journal of Supercomputing*, *78*(15), 17403–17422. https://doi.org/10.1007/s11227-022-04568-3
67. Kulshrestha, P., & Vijay Kumar, T. V. (2023). Machine learning based Intrusion Detection System for IOMT. *International Journal of System Assurance Engineering and Management*, *15*(5), 1802–1814. https://doi.org/10.1007/s13198-023-02119-4
68. Rbah, Y., Mahfoudi, M., Fattah, M., Balboul, Y., Mazer, S., Elbekkali, M., & Bernoussi, B. (2024). Deep learning for enhanced IOMT security: A GNN-BILSTM Intrusion Detection System. *2024 International Conference on Circuit, Systems and Communication (ICCSC)*, *838*, 1–6. https://doi.org/10.1109/iccsc62074.2024.10616456
69. Ravi, V., Pham, T. D., & Alazab, M. (2023). Deep learning-based network intrusion detection system for internet of medical things. *IEEE Internet of Things Magazine*, *6*(2), 50–54. https://doi.org/10.1109/iotm.001.2300021
70. Faruqui, N., Yousuf, M. A., Whaiduzzaman, M., Azad, A., Alyami, S. A., Liò, P., Kabir, M. A., & Moni, M. A. (2023). SafetyMed: A novel IOMT intrusion detection system using CNN-LSTM hybridization. *Electronics*, *12*(17), 3541. https://doi.org/10.3390/electronics12173541
71. R.M., S. P., Maddikunta, P. K., M., P., Koppu, S., Gadekallu, T. R., Chowdhary, C. L., & Alazab, M. (2020). An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IOMT architecture. *Computer Communications*, *160*, 139–149. https://doi.org/10.1016/j.comcom.2020.05.048
72. Alalhareth, M., & Hong, S.-C. (2023). An improved mutual information feature selection technique for intrusion detection systems in the Internet of Medical Things. *Sensors*, *23*(10), 4971. https://doi.org/10.3390/s23104971
73. Chaganti, R., Mourade, A., Ravi, V., Vemprala, N., Dua, A., & Bhushan, B. (2022). A particle swarm optimization and deep learning approach for intrusion detection system in internet of medical things. *Sustainability*, *14*(19), 12828. https://doi.org/10.3390/su141912828
74. Tahir, B., Jolfaei, A., & Tariq, M. (2024). A novel experience-driven and Federated Intelligent Threat-Defense Framework in IOMT. IEEE Journal of Biomedical and Health Informatics, 1–8. https://doi.org/10.1109/jbhi.2023.3236072

75. Singh, P., Gaba, G. S., Kaur, A., Hedabou, M., & Gurtov, A. (2023). Dew-cloud-based hierarchical federated learning for intrusion detection in IOMT. IEEE Journal of Biomedical and Health Informatics, 27(2), 722–731. https://doi.org/10.1109/jbhi.2022.3186250
76. Alamleh, A., Albahri, O. S., Zaidan, A. A., Albahri, A. S., Alamoodi, A. H., Zaidan, B. B., Qahtan, S., Alsatar, H. A., Al-Samarraay, M. S., & Jasim, A. N. (2023). Federated learning for IOMT applications: A standardization and benchmarking framework of intrusion detection systems. IEEE Journal of Biomedical and Health Informatics, 27(2), 878–887. https://doi.org/10.1109/jbhi.2022.3167256
77. Ioannou, I., Nagaradjane, P., Angin, P., Balasubramanian, P., Kavitha, K. J., Murugan, P., & Vassiliou, V. (2024). Gemlids-Miot: A green effective machine learning intrusion detection system based on Federated Learning for medical IOT network security hardening. Computer Communications, 218, 209–239. https://doi.org/10.1016/j.comcom.2024.02.023
78. Zaabar, B., Cheikhrouhou, O., & Abid, M. (2022). Intrusion detection system for IOMT through blockchain-based Federated Learning. 2022 15th International Conference on Security of Information and Networks (SIN), 01–08. https://doi.org/10.1109/sin56466.2022.9970536
79. Alharbi, A. A. (2024). Federated Transfer Learning for attack detection for internet of medical things. International Journal of Information Security, 23(1), 81–100. https://doi.org/10.1007/s10207-023-00805-9
80. Zhong, C., Sarkar, A., Manna, S., Khan, M. Z., Noorwali, A., Das, A., & Chakraborty, K. (2024). Federated learning-guided intrusion detection and neural key exchange for safeguarding patient data on the Internet of Medical Things. International Journal of Machine Learning and Cybernetics. https://doi.org/10.1007/s13042-024-02269-2
81. Begum, K., Mozumder, M. A., Joo, M.-I., & Kim, H.-C. (2024). BFLIDS: Blockchain-driven federated learning for intrusion detection in IOMT Networks. Sensors, 24(14), 4591. https://doi.org/10.3390/s24144591
82. Mane, S., & Rao, D. (2021, March 12). Explaining network intrusion detection system using explainable AI framework. arXiv.org. https://arxiv.org/abs/2103.07110
83. Si-ahmed, A., Al-Garadi, M. A., & Boustia, N. (2024, March 14). Explainable machine learning-based security and Privacy Protection Framework for internet of medical things systems. arXiv.org. https://arxiv.org/abs/2403.09752
84. Aljuhani, A., Alamri, A., Kumar, P., & Jolfaei, A. (2024). An intelligent and explainable SAAS-based Intrusion Detection System for resource-constrained IoMT. IEEE Internet of Things Journal, 11(15), 25454–25463. https://doi.org/10.1109/jiot.2023.3327024
85. Shtayat, M. M., Hasan, M. K., Sulaiman, R., Islam, S., & Khan, A. U. (2023). An explainable ensemble deep learning approach for intrusion detection in industrial internet of things. IEEE Access, 11, 115047–115061. https://doi.org/10.1109/access.2023.3323573
86. Mahbooba, B., Timilsina, M., Sahal, R., & Serrano, M. (2021). Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. Complexity, 2021(1). https://doi.org/10.1155/2021/6634811

87. Patil, S., Varadarajan, V., Mazhar, S. M., Sahibzada, A., Ahmed, N., Sinha, O., Kumar, S., Shaw, K., & Kotecha, K. (2022). Explainable artificial intelligence for Intrusion Detection System. Electronics, 11(19), 3079. https://doi.org/10.3390/electronics11193079
88. Alani, M. M., Mashatan, A., & Miri, A. (2023). Explainable ensemble-based detection of cyber attacks on internet of medical things. 2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 0609–0614. https://doi.org/10.1109/dasc/picom/cbdcom/cy59711.2023.10361448
89. Khan, I. A., Moustafa, N., Razzak, I., Tanveer, M., Pi, D., Pan, Y., & Ali, B. S. (2022). XSRU-IOMT: Explainable simple recurrent units for threat detection in internet of medical things networks. Future Generation Computer Systems, 127, 181–193. https://doi.org/10.1016/j.future.2021.09.010
90. Kumar, P., Javeed, D., Kumar, R., & Islam, A. K. M. N. (2024). Blockchain and explainable AI for enhanced decision making in cyber threat detection. Software: Practice and Experience, 54(8), 1337–1360. https://doi.org/10.1002/spe.3319
91. Rahmadika, S., Astillo, P. V., Choudhary, G., Duguma, D. G., Sharma, V., & You, I. (2023). Blockchain-based privacy preservation scheme for misbehavior detection in lightweight IOMT devices. IEEE Journal of Biomedical and Health Informatics, 27(2), 710–721. https://doi.org/10.1109/jbhi.2022.3187037
92. Pelekoudas-Oikonomou, F., Zachos, G., Papaioannou, M., de Ree, M., Ribeiro, J. C., Mantas, G., & Rodriguez, J. (2022). Blockchain-based security mechanisms for IOMT EDGE networks in IOMT-based Healthcare Monitoring Systems. Sensors, 22(7), 2449. https://doi.org/10.3390/s22072449
93. Gupta, K., Gupta, K. D., Kumar, D., Srivastava, G., & Sharma, D. K. (2024). Bids: Blockchain and Intrusion Detection System Coalition for securing internet of medical things networks. IEEE Journal of Biomedical and Health Informatics, 1–9. https://doi.org/10.1109/jbhi.2023.3325964
94. Wang, J., Jin, H., Chen, J., Tan, J., & Zhong, K. (2022). Anomaly detection in internet of medical things with blockchain from the perspective of Deep Neural Network. Information Sciences, 617, 133–149. https://doi.org/10.1016/j.ins.2022.10.060
95. Alshammari, B. M. (2023). AIBPSF-IOMT: Artificial Intelligence and Blockchain-based predictive security framework for IOMT Technologies. Electronics, 12(23), 4806. https://doi.org/10.3390/electronics12234806
96. Shukla, S., Thakur, S., Hussain, S., Breslin, J. G., & Jameel, S. M. (2021). Identification and authentication in Healthcare internet-of-things using integrated fog computing based Blockchain Model. Internet of Things, 15, 100422. https://doi.org/10.1016/j.iot.2021.100422
97. Mohd Shari, N. F., & Malip, A. (2024). Enhancing privacy and security in smart healthcare: A blockchain-powered decentralized data dissemination scheme. Internet of Things, 27, 101256. https://doi.org/10.1016/j.iot.2024.101256
98. Moulahi, W., Jdey, I., Moulahi, T., Alawida, M., & Alabdulatif, A. (2023). A blockchain-based Federated Learning Mechanism for Privacy Preservation of healthcare IOT Data. Computers in Biology and Medicine, 167, 107630. https://doi.org/10.1016/j.compbiomed.2023.107630

99. Rehman, A. U., Tariq, N., Jan, M. A., Khan, F., Song, H., & Ibrahim, M. (2024). A blockchain-based hybrid model for IOMT-enabled Intelligent Healthcare System. IEEE Transactions on Network Science and Engineering, 11(4), 3512–3521. https://doi.org/10.1109/tnse.2024.3376069
100. Wang, T., Wu, Q., Chen, J., Chen, F., Xie, D., & Shen, H. (2024). Health data security sharing method based on hybrid blockchain. Future Generation Computer Systems, 153, 251–261. https://doi.org/10.1016/j.future.2023.11.032
101. Mohammad Hossein, K., Esmaeili, M. E., Dargahi, T., Khonsari, A., & Conti, M. (2021). BCHealth: A novel blockchain-based privacy-preserving architecture for IOT healthcare applications. Computer Communications, 180, 31–47. https://doi.org/10.1016/j.comcom.2021.08.011
102. Sharma, P., Borah, M. D., & Namasudra, S. (2021). Improving security of medical big data by using blockchain technology. Computers & Electrical Engineering, 96, 107529. https://doi.org/10.1016/j.compeleceng.2021.107529
103. Farouk, A., Alahmadi, A., Ghose, S., & Mashatan, A. (2020). Blockchain platform for Industrial Healthcare: Vision and future opportunities. Computer Communications, 154, 223–235. https://doi.org/10.1016/j.comcom.2020.02.058
104. Shamshad, S., Minahil, Mahmood, K., Kumari, S., & Chen, C.-M. (2020). A secure blockchain-based e-health Records storage and sharing scheme. Journal of Information Security and Applications, 55, 102590. https://doi.org/10.1016/j.jisa.2020.102590
105. Sharma, P., Namasudra, S., Gonzalez Crespo, R., Parra-Fuente, J., & Chandra Trivedi, M. (2023). EHDHE: Enhancing security of healthcare documents in IOT-enabled digital healthcare ecosystems using blockchain. Information Sciences, 629, 703–718. https://doi.org/10.1016/j.ins.2023.01.148
106. Chen, L., Feng, T., Ma, R., & Shi, J. (2024). BTMDS: Blockchain Trusted Medical Data Sharing Scheme with privacy protection and Access Control. Computer Communications, 225, 279–288. https://doi.org/10.1016/j.comcom.2024.07.007
107. Rehman, A., Abbas, S., Khan, M. A., Ghazal, T. M., Adnan, K. M., & Mosavi, A. (2022). A secure healthcare 5.0 system based on blockchain technology entangled with Federated Learning Technique. Computers in Biology and Medicine, 150, 106019. https://doi.org/10.1016/j.compbiomed.2022.106019
108. Wang, Z., Luo, N., & Zhou, P. (2020). GuardHealth: Blockchain empowered secure data management and graph convolutional network enabled anomaly detection in smart healthcare. Journal of Parallel and Distributed Computing, 142, 1–12. https://doi.org/10.1016/j.jpdc.2020.03.004
109. EL Azzaoui, A., Sharma, P. K., & Park, J. H. (2022). Blockchain-based delegated quantum cloud architecture for Medical Big Data Security. Journal of Network and Computer Applications, 198, 103304. https://doi.org/10.1016/j.jnca.2021.103304
110. Tomar, A., Gupta, N., Rani, D., & Tripathi, S. (2023). Blockchain-Assisted Authenticated Key Agreement Scheme for IOT-based healthcare system. Internet of Things, 23, 100849. https://doi.org/10.1016/j.iot.2023.100849
111. Rizzardi, A., Sicari, S., Cevallos M., J. F., & Coen-Porisini, A. (2024). IOT-driven blockchain to manage the healthcare supply chain and protect medical records. Future Generation Computer Systems, 161, 415–431. https://doi.org/10.1016/j.future.2024.07.039

112. Ali, A., Pasha, M. F., Guerrieri, A., Guzzo, A., Sun, X., Saeed, A., Hussain, A., & Fortino, G. (2023). A novel homomorphic encryption and consortium blockchain-based hybrid deep learning model for Industrial Internet of Medical Things. IEEE Transactions on Network Science and Engineering, 10(5), 2402–2418. https://doi.org/10.1109/tnse.2023.3285070
113. Taloba, A. I., Elhadad, A., Rayan, A., Abd El-Aziz, R. M., Salem, M., Alzahrani, A. A., Alharithi, F. S., & Park, C. (2023). A blockchain-based hybrid platform for multimedia data processing in IOT-Healthcare. Alexandria Engineering Journal, 65, 263–274. https://doi.org/10.1016/j.aej.2022.09.031
114. Tanwar, S., Parekh, K., & Evans, R. (2020). Blockchain-based Electronic Healthcare Record System for healthcare 4.0 applications. Journal of Information Security and Applications, 50, 102407. https://doi.org/10.1016/j.jisa.2019.102407
115. Zhang, G., Yang, Z., & Liu, W. (2022). Blockchain-based privacy preserving e-health system for healthcare data in cloud. Computer Networks, 203, 108586. https://doi.org/10.1016/j.comnet.2021.108586
116. Zaabar, B., Cheikhrouhou, O., Jamil, F., Ammi, M., & Abid, M. (2021). HealthBlock: A secure blockchain-based Healthcare Data Management System. Computer Networks, 200, 108500. https://doi.org/10.1016/j.comnet.2021.108500
117. Chhikara, D., Rana, S., Mishra, A., & Mishra, D. (2022). Blockchain-driven authorized Data Access Mechanism for Digital Healthcare. Journal of Systems Architecture, 131, 102714. https://doi.org/10.1016/j.sysarc.2022.102714
118. Singh, S., Rathore, S., Alfarraj, O., Tolba, A., & Yoon, B. (2022). A framework for privacy-preservation of IOT healthcare data using Federated Learning and Blockchain Technology. Future Generation Computer Systems, 129, 380–388. https://doi.org/10.1016/j.future.2021.11.028
119. Mishra, A. R., Rani, P., Alrasheedi, A. F., & Dwivedi, R. (2023). Evaluating the blockchain-based healthcare supply chain using interval-valued pythagorean fuzzy entropy-based decision support system. Engineering Applications of Artificial Intelligence, 126, 107112. https://doi.org/10.1016/j.engappai.2023.107112
120. Tawfik, A. M., Al-Ahwal, A., Eldien, A. S., & Zayed, H. H. (2025). Pricollabanalysis: Privacy-preserving healthcare collaborative analysis on blockchain using homomorphic encryption and secure multiparty computation. Cluster Computing, 28(3). https://doi.org/10.1007/s10586-024-04928-z
121. Jiang, R., Zhang, H., Song, Z., Tian, S., & Lou, W. (2025). T-BFL model based on two-dimensional trust and blockchain-federated learning for medical data sharing. The Journal of Supercomputing, 81(2). https://doi.org/10.1007/s11227-024-06873-5
122. Khan, A., Litchfield, A., Alabdulatif, A., & Khan, F. (2025). Blockpres ipfs: Performance evaluation of blockchain based secure patients prescription record storage using ipfs for Smart Prescription Management System. Cluster Computing, 28(4). https://doi.org/10.1007/s10586-024-05054-6
123. Mazid, A., Kirmani, S., Abid, M., & Pawar, V. (2025). A secure and efficient framework for internet of medical things through blockchain driven customized Federated Learning. Cluster Computing, 28(4). https://doi.org/10.1007/s10586-024-04896-4

124. Wang, B., Jiang, R., Pu, X., & Zhang, H. (2025). An on-chain and off-chain collaborative data sharing and Access Control Model for Electronic Medical Records. The Journal of Supercomputing, 81(2). https://doi.org/10.1007/s11227-024-06884-2
125. Anar Hady, A. G. (2020). Wustl EHMS 2020 dataset for internet of medical things (IOMT) cybersecurity research. https://www.cse.wustl.edu/~jain/ehms/index.html
126. Ghubaish, A. (2024). HDRL-2024 dataset for cybersecurity research on medical applications in 5G networks. WUSTL. https://www.cse.wustl.edu/~jain/hdrl/index.html
127. Ahmed, M., Byreddy, S., Nutakki, A., Sikos, L. F., & Haskell-Dowland, P. (2021). ECU-ioht. Research Online. https://ro.ecu.edu.au/datasets/48/
128. Search UNB. University of New Brunswick est.1785. (2024). https://www.unb.ca/cic/datasets/iomt-dataset-2024.html
129. Li, J., Tong, X., Liu, J., & Cheng, L. (2023). An efficient federated learning system for network intrusion detection. IEEE Systems Journal, 17(2), 2455–2464. https://doi.org/10.1109/jsyst.2023.3236995
130. Sohi, S. M., Seifert, J.-P., & Ganji, F. (2021). RNNIDS: Enhancing network intrusion detection systems through Deep Learning. Computers & Security, 102, 102151. https://doi.org/10.1016/j.cose.2020.102151
131. Sharma, B., Sharma, L., Lal, C., & Roy, S. (2023). Anomaly based network intrusion detection for IOT attacks using Deep Learning Technique. Computers and Electrical Engineering, 107, 108626. https://doi.org/10.1016/j.compeleceng.2023.108626
132. Nandy, S., Adhikari, M., Khan, M. A., Menon, V. G., & Verma, S. (2022). An intrusion detection mechanism for secured IOMT framework based on Swarm-Neural Network. IEEE Journal of Biomedical and Health Informatics, 26(5), 1969–1976. https://doi.org/10.1109/jbhi.2021.3101686
133. Pradhan, M., & Mohanty, S. (2024). A blockchain-assisted multifactor authentication protocol for enhancing IOMT Security. IEEE Internet of Things Journal, 11(24), 39323–39332. https://doi.org/10.1109/jiot.2024.3422242 Yan, F., Zhang, G., Zhang, D., Sun, X., Hou, B., & Yu, N. (2023). TL-CNN-IDS: Transfer Learning-based intrusion detection system using Convolutional Neural Network. The Journal of Supercomputing, 79(15), 17562–17584. https://doi.org/10.1007/s11227-023-05347-4
134. Yan, F., Zhang, G., Zhang, D., Sun, X., Hou, B., & Yu, N. (2023). TL-CNN-IDS: Transfer Learning-based intrusion detection system using Convolutional Neural Network. The Journal of Supercomputing, 79(15), 17562–17584. https://doi.org/10.1007/s11227-023-05347-4
135. Wang, K., Zhang, A., Sun, H., & Wang, B. (2022). Analysis of recent deep-learning-based intrusion detection methods for in-vehicle network. IEEE Transactions on Intelligent Transportation Systems, 1–12. https://doi.org/10.1109/tits.2022.3222486
136. Dhanya, L., & Chitra, R. (2024). A novel Autoencoder based feature independent Ga optimised XGBoost classifier for IoMT malware detection. Expert Systems with Applications, 237, 121618. https://doi.org/10.1016/j.eswa.2023.121618
137. Khoa, T. V., Hoang, D. T., Trung, N. L., Nguyen, C. T., Quynh, T. T., Nguyen, D. N., Ha, N. V., & Dutkiewicz, E. (2023). Deep transfer learning: A novel collaborative learning model for Cyberattack

detection systems in IOT Networks. IEEE Internet of Things Journal, 10(10), 8578–8589. https://doi.org/10.1109/jiot.2022.3202029

138. Dina, A. S., Siddique, A. B., & Manivannan, D. (2023). A deep learning approach for intrusion detection in internet of things using focal loss function. Internet of Things, 22, 100699. https://doi.org/10.1016/j.iot.2023.100699
139. Reza, S., Ferreira, M. C., Machado, J. J. M., & Tavares, J. M. (2022). A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. Expert Systems with Applications, 202, 117275. https://doi.org/10.1016/j.eswa.2022.117275
140. Zhong, M., Yi, S., Fan, J., Zhang, Y., He, G., Cao, Y., Feng, L., Tan, Z., & Mo, W. (2023). Power transformer fault diagnosis based on a self-strengthening offline pre-training model. Engineering Applications of Artificial Intelligence, 126, 107142. https://doi.org/10.1016/j.engappai.2023.107142
141. Jiang, H., Lin, J., & Kang, H. (2022). FGMD: A robust detector against adversarial attacks in the IOT network. Future Generation Computer Systems, 132, 194–210. https://doi.org/10.1016/j.future.2022.02.019
142. Sachan, S., Almaghrabi, F., Yang, J.-B., & Xu, D.-L. (2021). Evidential reasoning for preprocessing uncertain categorical data for trustworthy decisions: An application on healthcare and Finance. Expert Systems with Applications, 185, 115597. https://doi.org/10.1016/j.eswa.2021.115597
143. Fonseca, J., & Bacao, F. (2023). Geometric smote for imbalanced datasets with nominal and continuous features. Expert Systems with Applications, 234, 121053. https://doi.org/10.1016/j.eswa.2023.121053
144. Jain, S., Pawar, P. M., & Muthalagu, R. (2022). Hybrid intelligent intrusion detection system for internet of things. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4097433
145. Ahmed, M., Byreddy, S., Nutakki, A., Sikos, L. F., & Haskell-Dowland, P. (2021). ECU-ioht: A dataset for analyzing cyberattacks in internet of health things. Ad Hoc Networks, 122, 102621. https://doi.org/10.1016/j.adhoc.2021.102621
146. Sarhan, M., Layeghy, S., Moustafa, N., & Portmann, M. (2021). NetFlow datasets for Machine Learning-based network intrusion detection systems. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 117–135. https://doi.org/10.1007/978-3-030-72802-1_9
147. Aldaej, A., Ahanger, T. A., & Ullah, I. (2023). Deep learning-inspired IOT-IDS mechanism for edge computing environments. Sensors, 23(24), 9869. https://doi.org/10.3390/s23249869
148. Karanfilovska, M., Kochovska, T., Todorov, Z., Cholakoska, A., Jakimovski, G., & Efnusheva, D. (2022). Analysis and modelling of a ML-based nids for IOT Networks. Procedia Computer Science, 204, 187–195. https://doi.org/10.1016/j.procs.2022.08.023
149. Nguyen, G. L., Dumba, B., Ngo, Q.-D., Le, H.-V., & Nguyen, T. N. (2022). A collaborative approach to early detection of IOT botnet. Computers & Electrical Engineering, 97, 107525. https://doi.org/10.1016/j.compeleceng.2021.107525
150. De La Torre Parra, G., Rad, P., Choo, K.-K. R., & Beebe, N. (2020). Detecting internet of things attacks using distributed deep learning. Journal of Network and Computer Applications, 163, 102662. https://doi.org/10.1016/j.jnca.2020.102662

151. Alkahtani, H., & Aldhyani, T. H. (2021). Botnet attack detection by using CNN-LSTM model for internet of things applications. Security and Communication Networks, 2021, 1–23. https://doi.org/10.1155/2021/3806459
152. Wagan, S. A., Koo, J., Siddiqui, I. F., Qureshi, N. M., Attique, M., & Shin, D. R. (2023). A fuzzy-based duo-secure multi-modal framework for IOMT anomaly detection. Journal of King Saud University - Computer and Information Sciences, 35(1), 131–144. https://doi.org/10.1016/j.jksuci.2022.11.007
153. Gupta, L., Salman, T., Ghubaish, A., Unal, D., Al-Ali, A. K., & Jain, R. (2022). Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach. Applied Soft Computing, 118, 108439. https://doi.org/10.1016/j.asoc.2022.108439
154. Wang, X., Wang, Y., Javaheri, Z., Almutairi, L., Moghadamnejad, N., & Younes, O. S. (2023). Federated deep learning for anomaly detection in the internet of things. Computers and Electrical Engineering, 108, 108651. https://doi.org/10.1016/j.compeleceng.2023.108651
155. Aguru, A. D., & Erukala, S. B. (2024a). A lightweight multi-vector ddos detection framework for IOT-enabled Mobile Health Informatics Systems using Deep Learning. Information Sciences, 662, 120209. https://doi.org/10.1016/j.ins.2024.120209
156. Xu, R., Wu, G., Wang, W., Gao, X., He, A., & Zhang, Z. (2024). Applying self-supervised learning to network intrusion detection for network flows with Graph Neural Network. Computer Networks, 248, 110495. https://doi.org/10.1016/j.comnet.2024.110495
157. Thulasi, T., & Sivamohan, K. (2023). LSO-CSL: Light spectrum optimizer-based convolutional stacked long short term memory for attack detection in IOT-based healthcare applications. Expert Systems with Applications, 232, 120772. https://doi.org/10.1016/j.eswa.2023.120772
158. Nandanwar, H., & Katarya, R. (2024). Deep learning enabled intrusion detection system for industrial IOT environment. Expert Systems with Applications, 249, 123808. https://doi.org/10.1016/j.eswa.2024.123808
159. Nandanwar, H., & Katarya, R. (2025). Securing industry 5.0: An explainable deep learning model for intrusion detection in cyber-physical systems. Computers and Electrical Engineering, 123, 110161. https://doi.org/10.1016/j.compeleceng.2025.110161
160. Altunay, H. C., Albayrak, Z., Ozalp, A. N., & Cakmak, M. (2021). Analysis of anomaly detection approaches performed through deep learning methods in SCADA systems. 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 1–6. https://doi.org/10.1109/hora52670.2021.9461273
161. OZALP, A. N., ALBAYRAK, Z., CAKMAK, M., & OZDOGAN, E. (2022). Layer-based examination of cyber-attacks in IOT. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 1–10. https://doi.org/10.1109/hora55278.2022.9800047
162. Vishwakarma, M., & Kesswani, N. (2022). DIDS: A deep neural network based real-time intrusion detection system for IOT. Decision Analytics Journal, 5, 100142. https://doi.org/10.1016/j.dajour.2022.100142
163. Zhang, Y., Zhu, D., Wang, M., Li, J., & Zhang, J. (2024a). A comparative study of cyber security intrusion detection in Healthcare Systems. International Journal of Critical Infrastructure Protection, 44, 100658. https://doi.org/10.1016/j.ijcip.2023.100658

164. Fernández Maimó, L., Huertas Celdrán, A., Perales Gómez, Á. L., García Clemente, F. J., Weimer, J., & Lee, I. (2019). Intelligent and dynamic ransomware spread detection and mitigation in Integrated Clinical Environments. Sensors, 19(5), 1114. https://doi.org/10.3390/s19051114
165. Kumar, P., Gupta, G. P., & Tripathi, R. (2021). An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IOMT networks. Computer Communications, 166, 110–124. https://doi.org/10.1016/j.comcom.2020.12.003
166. Saba, T. (2020). Intrusion detection in Smart City Hospitals using ensemble classifiers. 2020 13th International Conference on Developments in eSystems Engineering (DeSE). https://doi.org/10.1109/dese51703.2020.9450247
167. Siddiqi, M. A., & Pak, W. (2021). An agile approach to identify single and hybrid normalization for enhancing machine learning-based network intrusion detection. IEEE Access, 9, 137494–137513. https://doi.org/10.1109/access.2021.3118361
168. Newaz, A. I., Sikder, A. K., Babun, L., & Uluagac, A. S. (2020). Heka: A novel intrusion detection system for attacks to personal medical devices. 2020 IEEE Conference on Communications and Network Security (CNS). https://doi.org/10.1109/cns48642.2020.9162311
169. Hernandez-Jaimes, M. L., Martinez-Cruz, A., Ramírez-Gutiérrez, K. A., & Feregrino-Uribe, C. (2023). Artificial Intelligence for IOMT security: A review of intrusion detection systems, attacks, datasets and cloud–fog–edge architectures. Internet of Things, 23, 100887. https://doi.org/10.1016/j.iot.2023.100887
170. Radoglou-Grammatikis, P., Rompolos, K., Sarigiannidis, P., Argyriou, V., Lagkas, T., Sarigiannidis, A., Goudos, S., & Wan, S. (2022). Modeling, detecting, and mitigating threats against Industrial Healthcare Systems: A combined software defined networking and reinforcement learning approach. IEEE Transactions on Industrial Informatics, 18(3), 2041–2052. https://doi.org/10.1109/tii.2021.3093905
171. Nguyen, H., & Kashef, R. (2023). TS-ids: Traffic-aware self-supervised learning for IOT network intrusion detection. Knowledge-Based Systems, 279, 110966. https://doi.org/10.1016/j.knosys.2023.110966
172. Alosaimi, S., & Almutairi, S. M. (2023). An intrusion detection system using BOT-IOT. Applied Sciences, 13(9), 5427. https://doi.org/10.3390/app13095427
173. Zhu, S., Xu, X., Gao, H., & Xiao, F. (2023). CMTSNN: A deep learning model for multiclassification of abnormal and encrypted traffic of internet of things. IEEE Internet of Things Journal, 10(13), 11773–11791. https://doi.org/10.1109/jiot.2023.3244544
174. Fernando, G.-P., Brayan, A.-A. H., Florina, A. M., Liliana, C.-B., Héctor-Gabriel, A.-M., & Reinel, T.-S. (2023). Enhancing intrusion detection in IOT communications through ML model generalization with a new dataset (IDSAI). IEEE Access, 11, 70542–70559. https://doi.org/10.1109/access.2023.3292267
175. Xu, P., Lu, G., Li, Y., & Xu, C. (2023). EE-GCN: A graph convolutional network based Intrusion Detection Method for IIOT. 2023 5th International Conference on Natural Language Processing (ICNLP). https://doi.org/10.1109/icnlp58431.2023.00068
176. Chakraborty, C., Nagarajan, S. M., Devarajan, G. G., Ramana, T. V., & Mohanty, R. (2023). Intelligent AI-based healthcare cyber security system using Multi-Source Transfer Learning Method. ACM Transactions on Sensor Networks. https://doi.org/10.1145/3597210

177. Thakkar, A., & Lohiya, R. (2023). Attack classification of imbalanced intrusion data for IOT network using ensemble-learning-based Deep Neural Network. IEEE Internet of Things Journal, 10(13), 11888–11895. https://doi.org/10.1109/jiot.2023.3244810
178. Altaf, T., Wang, X., Ni, W., Yu, G., Liu, R. P., & Braun, R. (2023). A new concatenated multigraph neural network for IOT intrusion detection. Internet of Things, 22, 100818. https://doi.org/10.1016/j.iot.2023.100818
179. Alzahrani, A., & Asghar, M. Z. (2024). Cyber vulnerabilities detection system in logistics-based IOT data exchange. Egyptian Informatics Journal, 25, 100448. https://doi.org/10.1016/j.eij.2024.100448
180. Telikani, A., Rudbardeh, N. E., Soleymanpour, S., Shahbahrami, A., Shen, J., Gaydadjiev, G., & Hassanpour, R. (2024). A cost-sensitive machine learning model with multitask learning for intrusion detection in IOT. IEEE Transactions on Industrial Informatics, 20(3), 3880–3890.
181. Saif, S., Das, P., Biswas, S., Khari, M., & Shanmuganathan, V. (2022). HIIDS: Hybrid intelligent intrusion detection system empowered with machine learning and metaheuristic algorithms for application in IOT based healthcare. Microprocessors and Microsystems, 104622. https://doi.org/10.1016/j.micpro.2022.104622
182. Tao, H., Bhuiyan, M. Z., Abdalla, A. N., Hassan, M. M., Zain, J. M., & Hayajneh, T. (2019). Secured data collection with hardware-based ciphers for IOT-based healthcare. IEEE Internet of Things Journal, 6(1), 410–420. https://doi.org/10.1109/jiot.2018.2854714
183. Kalam, S., & Keshri, A. K. (2025). Advancing Iomt Security: A Two-factor authentication model employing PUF and fuzzy logic techniques. Computers & Security, 148, 104138. https://doi.org/10.1016/j.cose.2024.104138
184. Yaacoub, J.-P. A., Noura, M., Noura, H. N., Salman, O., Yaacoub, E., Couturier, R., & Chehab, A. (2020). Securing internet of medical things systems: Limitations, issues and recommendations. Future Generation Computer Systems, 105, 581–606. https://doi.org/10.1016/j.future.2019.12.028
185. Kassab, M., DeFranco, J., Malas, T., Laplante, P., Destefanis, G., & Neto, V. V. (2021). Exploring research in blockchain for healthcare and a roadmap for the future. IEEE Transactions on Emerging Topics in Computing, 9(4), 1835–1852. https://doi.org/10.1109/tetc.2019.2936881
186. Islam, N., Faheem, Y., Din, I. U., Talha, M., Guizani, M., & Khalil, M. (2019). A blockchain-based fog computing framework for activity recognition as an application to e-healthcare services. Future Generation Computer Systems, 100, 569–578. doi:10.1016/j.future.2019.05.059
187. Inam, S., Kanwal, S., Firdous, R., & Hajjej, F. (2024). Blockchain based medical image encryption using Arnold's cat map in a cloud environment. Scientific Reports, 14(1). https://doi.org/10.1038/s41598-024-56364-z.
188. Li, C., Jiang, B., Dong, M., Chen, Y., Zhang, Z., Xin, X., & Ota, K. (2024). Efficient designated verifier signature for secure Cross-Chain Health Data Sharing in BIoMT. IEEE Internet of Things Journal, 11(11), 19838–19851. https://doi.org/10.1109/jiot.2024.3370708.
189. Sai Chaitanya Kumar, G., Kiran Kumar, R., Parish Venkata Kumar, K., Raghavendra Sai, N., & Brahmaiah, M. (2024). Deep residual convolutional neural network: An efficient technique for intrusion detection system. Expert Systems with Applications, 238, 121912. https://doi.org/10.1016/j.eswa.2023.121912

190. Deo, T. Y., & Sanju, A. (2023). Data imputation and comparison of custom ensemble models with existing libraries like XGBoost, CATBoost, AdaBoost and Scikit learn for predictive equipment failure. Materials Today: Proceedings, 72, 1596–1604. https://doi.org/10.1016/j.matpr.2022.09.410
191. Dadkhah, S., Neto, E. C., Ferreira, R., Molokwu, R. C., Sadeghi, S., & Ghorbani, A. A. (2024). CICIOMT2024: A benchmark dataset for multi-protocol security assessment in IOMT. Internet of Things, 28, 101351. https://doi.org/10.1016/j.iot.2024.101351
192. Wu, Y., Nie, L., Wang, S., Ning, Z., & Li, S. (2023). Intelligent intrusion detection for internet of things security: A deep convolutional generative adversarial network-enabled approach. IEEE Internet of Things Journal, 10(4), 3094–3106. https://doi.org/10.1109/jiot.2021.3112159
193. Kim, T., & Pak, W. (2022). Early detection of network intrusions using a gan-based one-class classifier. IEEE Access, 10, 119357–119367. https://doi.org/10.1109/access.2022.3221400
194. Park, C., Lee, J., Kim, Y., Park, J.-G., Kim, H., & Hong, D. (2023). An enhanced AI-based network intrusion detection system using generative adversarial networks. IEEE Internet of Things Journal, 10(3), 2330–2345. https://doi.org/10.1109/jiot.2022.3211346
195. Manimurugan, S., Al-Mutairi, S., Aborokbah, M. M., Chilamkurti, N., Ganesan, S., & Patan, R. (2020). Effective attack detection in internet of medical things smart environment using a deep belief neural network. IEEE Access, 8, 77396–77404. https://doi.org/10.1109/access.2020.2986013
196. Wu, Z., Zhang, H., Wang, P., & Sun, Z. (2022). RTIDS: A robust transformer-based approach for Intrusion Detection System. IEEE Access, 10, 64375–64387. https://doi.org/10.1109/access.2022.3182333
197. Zhong, Y., Wang, Z., Shi, X., Yang, J., & Li, K. (2024). RFG-HELAD: A robust fine-grained network traffic anomaly detection model based on heterogeneous ensemble learning. IEEE Transactions on Information Forensics and Security, 19, 5895–5910. https://doi.org/10.1109/tifs.2024.3402439
198. Han, W., Peng, J., Yu, J., Kang, J., Lu, J., & Niyato, D. (2024). Heterogeneous data-aware federated learning for intrusion detection systems via meta-sampling in artificial intelligence of things. IEEE Internet of Things Journal, 11(8), 13340–13354. https://doi.org/10.1109/jiot.2023.3337755
199. Seo, W., & Pak, W. (2021). Real-time network intrusion prevention system based on Hybrid Machine Learning. IEEE Access, 9, 46386–46397. https://doi.org/10.1109/access.2021.3066620
200. Wang, W., Du, X., Shan, D., Qin, R., & Wang, N. (2022). Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine. IEEE Transactions on Cloud Computing, 10(3), 1634–1646. https://doi.org/10.1109/tcc.2020.3001017
201. Zhang, C., Costa-Perez, X., & Patras, P. (2022). Adversarial attacks against Deep Learning-based network intrusion detection systems and Defense Mechanisms. IEEE/ACM Transactions on Networking, 30(3), 1294–1311. https://doi.org/10.1109/tnet.2021.3137084
202. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT Telemetry Dataset: A new generation dataset of IOT and IIoT for data-driven intrusion detection systems. IEEE Access, 8, 165130–165150. https://doi.org/10.1109/access.2020.3022862
203. Kye, H., Kim, M., & Kwon, M. (2022). Hierarchical detection of network anomalies : A self-supervised learning approach. IEEE Signal Processing Letters, 29, 1908–1912. https://doi.org/10.1109/lsp.2022.3203296

204. Jim Solomon Raja, D., Sriranjani, R., Arulmozhi, P., & Hemavathi, N. (2024). Unified Random Forest and hybrid bat optimization based man-in-the-middle attack detection in advanced metering infrastructure. IEEE Transactions on Instrumentation and Measurement, 73, 1–12. https://doi.org/10.1109/tim.2024.3420375
205. Liu, J., Yang, D., Lian, M., & Li, M. (2021). Research on intrusion detection based on particle swarm optimization in IOT. IEEE Access, 9, 38254–38268. https://doi.org/10.1109/access.2021.3063671
206. Yang, L., Li, J., Yin, L., Sun, Z., Zhao, Y., & Li, Z. (2020). Real-time intrusion detection in wireless network: A deep learning-based intelligent mechanism. IEEE Access, 8, 170128–170139. https://doi.org/10.1109/access.2020.3019973
207. Ben Said, R., Sabir, Z., & Askerzade, I. (2023). CNN-BiLSTM: A hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection. IEEE Access, 11, 138732–138747. https://doi.org/10.1109/access.2023.3340142
208. Zhao, R., Mu, Y., Zou, L., & Wen, X. (2022). A hybrid intrusion detection system based on feature selection and weighted stacking classifier. IEEE Access, 10, 71414–71426. https://doi.org/10.1109/access.2022.3186975
209. Das, S., Saha, S., Priyoti, A. T., Roy, E. K., Sheldon, F. T., Haque, A., & Shiva, S. (2022). Network intrusion detection and Comparative Analysis Using Ensemble Machine Learning and feature selection. IEEE Transactions on Network and Service Management, 19(4), 4821–4833. https://doi.org/10.1109/tnsm.2021.3138457
210. Ghubaish, A., Yang, Z., & Jain, R. (2024). HDRL-ids: A hybrid deep reinforcement learning intrusion detection system for enhancing the security of medical applications in 5G networks. 2024 International Conference on Smart Applications, Communications and Networking (SmartNets). https://doi.org/10.1109/smartnets61466.2024.10577692
211. Xu, H., Sun, L., Fan, G., Li, W., & Kuang, G. (2023). A hierarchical intrusion detection model combining multiple deep learning models with attention mechanism. IEEE Access, 11, 66212–66226. https://doi.org/10.1109/access.2023.3290613
212. Liang, J., Sadiq, M., Yang, G., Jiang, K., Cai, T., & Ma, M. (2024). Enhanced collaborative intrusion detection for industrial cyber-physical systems using permissioned blockchain and decentralized Federated Learning Networks. Engineering Applications of Artificial Intelligence, 135, 108862. https://doi.org/10.1016/j.engappai.2024.108862
213. Manocchio, L. D., Layeghy, S., Lo, W. W., Kulatilleke, G. K., Sarhan, M., & Portmann, M. (2024). Flowtransformer: A Transformer framework for FLOW-based network intrusion detection systems. Expert Systems with Applications, 241, 122564. https://doi.org/10.1016/j.eswa.2023.122564
214. Ullah, F., Ullah, S., Srivastava, G., & Lin, J. C.-W. (2024). IDs-INT: Intrusion detection system using Transformer-based transfer learning for Imbalanced Network Traffic. Digital Communications and Networks, 10(1), 190–204. https://doi.org/10.1016/j.dcan.2023.03.008
215. Shao, J.-M., Zeng, G.-Q., Lu, K.-D., Geng, G.-G., & Weng, J. (2024). Automated Federated Learning for intrusion detection of industrial control systems based on Evolutionary Neural Architecture Search. Computers & Security, 143, 103910. https://doi.org/10.1016/j.cose.2024.103910

216. Quyen, N. H., Duy, P. T., Nguyen, N. T., Khoa, N. H., & Pham, V.-H. (2025). FEDKD-ids: A robust intrusion detection system using knowledge distillation-based semi-supervised Federated Learning and anti-poisoning attack mechanism. Information Fusion, 117, 102807. https://doi.org/10.1016/j.inffus.2024.102807

217. Duy, P. T., Hien, D. T., Luong, T. D., Quyen, N. H., & Pham, V.-H. (2024). Fed-Evolver: An automated evolving approach for federated intrusion detection system using adversarial autoencoder in SDN-enabled networks. Internet of Things, 28, 101397. https://doi.org/10.1016/j.iot.2024.101397

218. Hady, A. A., Ghubaish, A., Salman, T., Unal, D., & Jain, R. (2020). Intrusion detection system for healthcare systems using medical and network data: A comparison study. IEEE Access, 8, 106576–106584. https://doi.org/10.1109/access.2020.3000421.

219. Chintapalli, S. S., Singh, S. P., Frnda, J., Bidare Divakarachari, P., Sarraju, V. L., & Falkowski-Gilski, P. (2024). OOA-modified Bi-LSTM Network: An effective intrusion detection framework for IOT Systems. Heliyon, 10(8). https://doi.org/10.1016/j.heliyon.2024.e29410

220. Xu, C., Shen, J., & Du, X. (2020). A method of few-shot network intrusion detection based on meta-learning framework. IEEE Transactions on Information Forensics and Security, 15, 3540–3552. https://doi.org/10.1109/tifs.2020.2991876

221. Pontes, C. F., de Souza, M. M., Gondim, J. J., Bishop, M., & Marotta, M. A. (2021). A new method for flow-based network intrusion detection using the inverse Potts model. IEEE Transactions on Network and Service Management, 18(2), 1125–1136. https://doi.org/10.1109/tnsm.2021.3075503

222. Ho, S., Jufout, S. A., Dajani, K., & Mozumdar, M. (2021). A novel intrusion detection model for detecting known and innovative cyberattacks using Convolutional Neural Network. IEEE Open Journal of the Computer Society, 2, 14–25. https://doi.org/10.1109/ojcs.2021.3050917

223. Ghourabi, A. (2022). A security model based on LIGHTGBM and Transformer to protect healthcare systems from cyberattacks. IEEE Access, 10, 48890–48903. https://doi.org/10.1109/access.2022.3172432

224. Hameed, S. S., Hassan, W. H., & Latiff, L. A. (2021). An efficient fog-based attack detection using ensemble of Moa-WMA for internet of medical things. Lecture Notes on Data Engineering and Communications Technologies, 774–785. https://doi.org/10.1007/978-3-030-70713-2_70

# DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

## PLAGIARISM VERIFICATION

**Title of the Thesis:** Design and Development of Smart and Secure Healthcare System

**Total Pages:** 216

**Name of the Scholar:** Nikhil Sharma

**Supervisors:** Dr. Prashant Giridhar Shambharkar

**Department:** Computer Science and Engineering

This is to report that the above thesis was scanned for similarity detection. The process and outcome are given below:

**Software used:** Turnitin Similarity **Index:** % **Word Count:** Words

Date: 18th June 2025

Candidate's Signature Signature of Supervisors

## LIST OF PUBLICATIONS

### *Journal Publication*

1. Shambharkar, P. G., & Sharma, N. (2024). Deep learning-empowered intrusion detection framework for the Internet of Medical Things Environment. Knowledge and Information Systems, 66(10), 6001–6050. https://doi.org/10.1007/s10115-024-02149-9, (Impact Factor: 2.5, Publisher: Springer), (***SCIE Indexed-Published)***
2. Sharma, N., & Shambharkar, P. G. (2025). Multi-attention deepcrnn: An efficient and explainable intrusion detection framework for internet of medical things environments. Knowledge and Information Systems. https://doi.org/10.1007/s10115-025-02402-9. (Impact Factor: 2.5, Publisher: Springer), (***SCIE Indexed-Published)***
3. Sharma, N., & Shambharkar, P. G. (2025). Multi-layered security architecture for IOMT systems: Integrating dynamic key management, decentralized storage, and dependable intrusion detection framework. International Journal of Machine Learning and Cybernetics. https://doi.org/10.1007/s13042-025-02628-7. (Impact Factor: 3.1, Publisher: Springer), (***SCIE Indexed-Published)***
4. Sharma, N., & Shambharkar, P. G. (2025). Towards secure healthcare: SA-GBO-ODBN model utilizing blockchain and deep learning for data handling and diagnosis. The Computer Journal. https://doi.org/10.1093/comjnl/bxaf045. (Impact Factor: 1.5, Publisher: Oxford University Press), (***SCIE Indexed-Published)***
5. Sharma, N., & Shambharkar, P. (2025). Transforming security in internet of medical things with advanced deep learning-based Intrusion Detection Frameworks. Applied Soft Computing, 180, 113420. https://doi.org/10.1016/j.asoc.2025.113420. (Impact Factor: 7.2 Publisher: Elsevier) ***(SCIE Indexed-Published)***

### *Conference Publication*

1. Sharma, N., & Shambharkar, P. G. (2022). Applicability of ML-IOT in Smart Healthcare Systems: Challenges, Solutions & Future Direction. *2022 International Conference on Computer Communication and Informatics (ICCCI)*, 1–7. https://doi.org/10.1109/iccci54379.2022.9740983 ***(Published) (Scopus-Indexed)***
2. Sharma, N., & Shambharkar, P. G. (2024). A systematic literature review of the emerging technologies used in securing healthcare data. *2024 12th International Conference on Internet of Everything, Microwave, Embedded, Communication and*

*Networks (IEMECON)*, 1–12. https://doi.org/10.1109/iemecon62401.2024.10846068 ***(Published) (Scopus-Indexed)***

***Communicated Journal***

1. Sharma, N., & Shambharkar, P. G. Enhancing Internet of Medical Things Security: A Multi-Layered Approach Using Dynamic Adaptive Deep Reinforcement Learning and Blockchain, Computers and Electrical Engineering, Elsevier. (Impact Factor: 4.1 Publisher: Elsevier) ***(Major Revision Submitted)***
2. Sharma, N., & Shambharkar, P. G. Blockchain-Based Framework for Secure Medical Data Sharing and Disease Diagnosis Using Optimized Deep Belief Networks, Cluster Computing, Springer. (Impact Factor: 3.6 Publisher: Springer) ***(Major Revision Submitted)***
3. Sharma, N., & Shambharkar, P. G. Enhancing Internet of Medical Things Security with Multi-Attention Convolutional LSTM-Based Intrusion Detection System. Transactions on Emerging Telecommunications Technologies, Wiley. (Impact Factor: 2.5, Publisher: Wiley) ***(Major Revision Submitted)***
4. Sharma, N., & Shambharkar, P. G. A Novel Blockchain-Based Framework for Securing Electronic Health Records Using Least Squares Analysis. Cluster Computing. (Impact Factor: 3.6, Publisher: Springer) ***(With Editor)***
5. Sharma, N., & Shambharkar, P. G. A Quantum Neural Network-Assisted Hybrid Cryptographic Model for Secure Blockchain-Based EHR Systems. ***(To be communicated)***

# Publication Proof

***Journal Papers***

***Paper 1:*** Shambharkar, P. G., & Sharma, N. (2024). Deep learning-empowered intrusion detection framework for the Internet of Medical Things Environment. Knowledge and Information Systems, 66(10), 6001–6050. https://doi.org/10.1007/s10115-024-02149-9, ***(Impact Factor: 2.5, Publisher: Springer), (SCIE Indexed-Published)***

Knowledge and Information Systems (2024) 66:6001–6050
https://doi.org/10.1007/s10115-024-02149-9

REGULAR PAPER

Check for updates

## Deep learning-empowered intrusion detection framework for the Internet of Medical Things environment

Prashant Giridhar Shambharkar[1] · Nikhil Sharma[1]

Received: 5 February 2024 / Revised: 22 April 2024 / Accepted: 20 May 2024 /
Published online: 10 June 2024


**Abstract**
The fusion of Internet of Things (IoT) technology into healthcare, known as the Internet of Medical Things (IoMT), has significantly enhanced medical treatment and operational efficiency. Real-time patient monitoring (RPM) and remote diagnostics enabled by IoMT allow doctors to treat more patients effectively and save lives. However, healthcare devices' interconnected nature makes them vulnerable to cyber-attacks, threatening patient privacy and security. Ensuring the security and accuracy of patient health data is paramount, as any tampering could have life-threatening consequences, especially in emergency situations. To address these challenges, this research focuses on developing robust security models to secure patient data in IoMT networks while meeting the growing demand for efficient healthcare services. Artificial intelligence (AI)-based technologies such as machine learning (ML) and deep learning (DL) have the potential to be employed as the methodology for intrusion detection. The goal of this research is threefold: firstly, the linear support vector machine (LinSVM) model; secondly, the convolutional support vector machine (ConvSVM) model; and finally, the categorical embedding (CatEmb) model, which have been proposed to overcome the issue of security in a network. This article offers the CatEmb model as the first effort to use a DL-based embedding approach to recognize intrusion in the IoMT environment, utilizing patient biometric and network traffic flow data. Our experimental results show the efficacy of the proposed DL models, with the LinSVM achieving a training accuracy of 99.78%, ConvSVM reaching 99.98%, and CatEmb achieving 99.84%. These models outperform existing methodologies by 2.61% in detecting network intrusions, as demonstrated through metrics such as detection rate and F1-score. Furthermore, the proposed approaches are thoroughly compared with the existing state-of-the-art studies.

**Keywords** Artificial intelligence (AI) · Cyber-attacks · Deep learning (DL) · Healthcare · Internet of Medical Things (IoMT) · Intrusion detection system (IDS) · Machine learning (ML) · Security

✉ Nikhil Sharma
nikhilsharma1694@gmail.com

Prashant Giridhar Shambharkar
prashant.shambharkar@dtu.ac.in

***Paper 2:*** Sharma, N., & Shambharkar, P. G. (2025). Multi-attention deepcrnn: An efficient and explainable intrusion detection framework for internet of medical things environments. Knowledge and Information Systems. https://doi.org/10.1007/s10115-025-02402-9. ***(Impact Factor: 2.5, Publisher: Springer), (SCIE Indexed-Published)***

Knowledge and Information Systems
https://doi.org/10.1007/s10115-025-02402-9

RESEARCH

# Multi-attention DeepCRNN: an efficient and explainable intrusion detection framework for Internet of Medical Things environments

**Nikhil Sharma**[1] · **Prashant Giridhar Shambharkar**[1]

Received: 12 November 2024 / Revised: 10 February 2025 / Accepted: 7 March 2025


**Abstract**

The increasing prevalence of cyber threats in healthcare necessitates robust security measures to protect sensitive medical data. This research presents a hybrid security framework integrating blockchain for decentralized, immutable data storage and an intrusion detection system (IDS) leveraging a multi-attention deep convolutional recurrent neural network (MA-DeepCRNN) model for advanced threat detection. The proposed IDS combines convolutional neural networks (CNNs) for spatial feature extraction, recurrent neural networks (RNNs) for temporal pattern recognition, and an attention mechanism to enhance critical data representation. The model is evaluated using the CICIoMT 2024 benchmark dataset. The blockchain architecture achieves a block creation time of 10 s, improving significantly over Bitcoin (~ 600 s) and Ethereum (~ 15 s), while increasing throughput to ~ 182 transactions per second. Security analysis indicates a low transaction reversal probability of < 0.1%. The IDS demonstrates high classification performance, achieving 99.49% accuracy in binary classification, 99.12% in multiclass (6-class) classification, and 98.56% in large-scale (19-class) classification. Comparative analysis with state-of-the-art approaches highlights improvements in accuracy and F1-score by 3.44 and 3.71%, respectively, for intrusion detection in Internet of Medical Things (IoMT) systems. These results underscore the effectiveness of the proposed framework in enhancing security, scalability, and real-time threat detection in healthcare environments.

**Keywords** Attention mechanism · Blockchain · Cyber-attacks · Deep learning (DL) · Explainable artificial intelligence (XAI) · Internet of Medical Things · Intrusion detection systems

✉ Nikhil Sharma
nikhilsharma1694@gmail.com

Prashant Giridhar Shambharkar
prashant.shambharkar@dtu.ac.in

[1] Department of Computer Science & Engineering, Delhi Technological University, Delhi, India

Published online: 05 April 2025

***Paper 3:*** Sharma, N., & Shambharkar, P. G. (2025). Multi-layered security architecture for IOMT systems: Integrating dynamic key management, decentralized storage, and dependable intrusion detection framework. International Journal of Machine Learning and Cybernetics. https://doi.org/10.1007/s13042-025-02628-7. ***(Impact Factor: 3.1, Publisher: Springer), (SCIE Indexed-Published)***

International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-025-02628-7

ORIGINAL ARTICLE

# Multi-layered security architecture for IoMT systems: integrating dynamic key management, decentralized storage, and dependable intrusion detection framework

Nikhil Sharma[1] · Prashant Giridhar Shambharkar[1]

Received: 3 September 2024 / Accepted: 27 March 2025


**Abstract**
The growing complexity of cyber threats presents significant challenges to the security of Internet of Medical Things (IoMT) systems, where traditional security and intrusion detection methods often prove inadequate. The Key challenges include inefficient key management, fragmented security protocols, and limited scalability. To address these issues, this paper proposes a Dynamic Adaptive Deep Reinforcement Learning (DA-DRL) framework that enhances Advanced Encryption Standard (AES) encryption by dynamically adjusting key generation in response to real-time threats. Additionally, a multi-layered security architecture integrating AES, SHA-512, Non-Interactive Zero Knowledge Proof (NIZKPs), Practical Byzantine Fault Tolerance (PBFT), and Attribute-Based Access Control (ABAC) is introduced, ensuring robust protection against diverse attack vectors. The InterPlanetary File System (IPFS) is employed for decentralized and immutable data storage, enhancing data security and transparency. The proposed DA-DRL-AES-SHA-512 methodology significantly outperforms conventional encryption techniques, achieving an encryption time of 0.0975 s, decryption time of 0.0846 s, and a throughput of 75.63 transactions per second (Tx/s) with a network overhead of just 0.1289%. The Energy consumption and computational overhead are reduced to 0.3664 J and 0.48%, respectively. The Secure and Dependable Bi-LSTM GRU Intrusion Detection Framework (S-BiLSTMGRU-IDF) achieves 99.94% accuracy in binary classification and 99.89% in multiclass classification, improving detection efficiency by 0.6–3.5% over state-of-the-art models. This blockchain-based framework ensures real-time threat mitigation, enhanced data integrity, and superior system performance, establishing a secure, scalable, and efficient solution for IoMT security.

**Keywords** Blockchain · Cyber-attacks · Deep learning · Intrusion detection system · Internet of medical things (IoMT) · Security · Privacy

## 1 Introduction

The rapid integration of the Internet of Medical Things (IoMT) into healthcare systems has revolutionized how medical services are delivered and managed. By connecting various medical devices, sensors, and applications through the internet, IoMT facilitates continuous monitoring, remote diagnostics, and real-time data exchange, leading to improved patient outcomes and operational efficiency [1]. However, the widespread adoption of IoMT has also introduced many security and privacy challenges that, if not addressed, could undermine trust in these technologies and expose patients to significant risks. The complex nature of IoMT systems, coupled with their rapid deployment, has created a critical need for comprehensive security solutions to protect sensitive

***Paper 4:*** Sharma, N., & Shambharkar, P. G. (2025). Towards secure healthcare: SA-GBO-ODBN model utilizing blockchain and deep learning for data handling and diagnosis. The Computer Journal. https://doi.org/10.1093/comjnl/bxaf045. ***(Impact Factor: 1.5, Publisher: Oxford University Press), (SCIE Indexed-Published)***

OXFORD

The Computer Journal, 2025, 1–38
https://doi.org/10.1093/comjnl/bxaf045
Original Article

# Towards secure healthcare: SA-GBO-ODBN model utilizing Blockchain and deep learning for data handling and diagnosis

Nikhil Sharma and Prashant Giridhar Shambharkar*

Department of Computer Science and Engineering, Delhi Technological University, Bawana Road, Shahbad Daulatpur Village, Rohini, New Delhi-110042, India
*Corresponding author. Department of Computer Science and Engineering, Delhi Technological University, Bawana Road, Shahbad Daulatpur Village, Rohini, New Delhi-110042, India. E-mail: prashant.shambharkar@dtu.ac.in

**Abstract**

The integration of Electronic Health Records (EHRs) in healthcare has significantly advanced the field but introduced challenges related to data security and precise diagnoses due to the large data volume. To address these issues, we propose the SA-GBO-ODBN model, combining Blockchain and deep learning (DL) for secure medical data management and diagnostics. This model includes Hyperledger Fabric for tamper-proof storage, optimal key generation, data encryption and decryption, and disease detection functionalities. The Key features of the proposed framework include emergency contact notifications, user data access management, and administrative data modifications. The framework employs SHA-256 and elliptical curve cryptography (ECC) for enhanced data security. ECC uses the Self-Adaptive Gradient-Based Optimizer (SA-GBO) to generate optimal encryption and decryption keys. Hyperledger blockchain technology enables secure medical data sharing, patient visit data storage, and EHR link recording in external databases via multiple channels. After decryption, the Optimized Deep Belief Network–based approach diagnoses epilepsy using real-time EEG datasets. The qualitative and quantitative performance analysis shows the proposed framework's superiority over existing techniques, with accuracy, False Positive Rate (FPR), and FNR of 98.93%, 0.0199, and 0.0034 for the Bonn EEG dataset, and 99.40%, 0.0196, and 0.0034 for the CHB-MIT dataset, respectively.

**Keywords:** Blockchain; optimized deep belief network (ODBN); deep learning (DL); electronic health records (EHRs); gradient-based optimizer (GBO); security

## 1. INTRODUCTION

Recently, the rise in chronic diseases and the ongoing COVID-19 pandemic have led to a marked increase in people's health awareness. Electronic Health Records (EHRs) are common in the real world and can be shared as vital healthcare data resources. The expanding amount of data consumes a considerable large amount of local resources, which are restricted by the capabilities of devices in the nearby area [1]. Hospitals could keep their EHRs in their cloud to share the data between patients, doctors, and suppliers [2]. However, the healthcare system (HS) nodes are consistently linked via an open channel, exposing the entire network to potential eavesdropping, data manipulation, and other security-related concerns. The rapid advancement of hacking techniques and communication protocols has heightened security concerns. Attackers are increasingly sophisticated in their efforts to breach systems, aiming to compromise the integrity, availability, confidentiality, and reliability of data. This evolving threat landscape underscores the urgency of addressing these security vulnerabilities [3, 4]. These attacks can target healthcare network components with malicious software or malware and reduce the functionality of actual Internet of Things (IoT) devices or access the patient's data without their permission [5]. Most currently used medical data-sharing protocols use essential encryption features and fine-grained access control. EHRs are shared among many users [6]. ABE (Attribute-based encryption) links the encrypted message and a user's confidential key to a group of characteristics and authorization rules. Suppose patients' EHRs are encrypted utilizing traditional CP-ABE (Cipher Policy Attribute-based encryption). In that case, anyone with cloud servers can access the precise access policy data [7], which may indicate that the patient has cardiac problems, which would be unacceptable. This may result in devices within the network inefficiently using resources and experiencing disruptions in their typical communication patterns [8].

For the issues mentioned above, combining DL and Blockchain can offer an exceptional solution in the healthcare industry [9, 10]. A chain, a block, and a blockchain are types of open databases (a type of digital information) using which information cannot be changed once it has been saved in a series of immutable blocks [11]. Due to blockchain's decentralized and immutable characteristics, smart contracts (SC) can enhance trust among parties engaged in data transmission by promptly executing and enforcing their terms. Additionally, the consensus mechanisms guarantee the integrity of the blockchain's distributed records. As a result, it is feasible to assume that the patient's medical-related information is secure [12], reliable, and trustworthy while being transmitted over HS [13]. We develop a self-adaptive deep learning model with a secure blockchain model to address the abovementioned issues and attain safer and more effective data access control in the healthcare sector.

Received: April 28, 2024. Revised: February 7, 2025. Accepted: March 31, 2025

***Paper 5:*** Sharma, N., & Shambharkar, P. (2025). Transforming security in internet of medical things with advanced deep learning-based Intrusion Detection Frameworks. Applied Soft Computing, 180, 113420. https://doi.org/10.1016/j.asoc.2025.113420. ***(Impact Factor: 7.2 Publisher: Elsevier) (SCIE Indexed-Published)***

Applied Soft Computing Journal 180 (2025) 113420

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# Transforming security in internet of medical things with advanced deep learning-based intrusion detection frameworks

Nikhil Sharma *, Prashant Giridhar Shambharkar

*Department of Computer Science & Engineering, Delhi Technological University, Bawana Road Rohini, Delhi 110042, India*

## HIGHLIGHTS

- Novel Deep Learning Models for IoMT Attacks: Introduced EmbedNet, ConvNet-SVM, and DeepSVM-Net models for intrusion detection.
- Advanced Data Preprocessing Techniques: Techniques include covariance matrix filtering, data augmentation, and normalization.
- Classification Performance Evaluation: Evaluated models against attacks like DoS, DDoS, and MITM for detection ability.
- Epochs and Batch Size Analysis: Analyzed effects of epochs and batch size on training loss, validation loss, and accuracy.
- Advanced Metrics Trade-off Analysis: Provided detailed trade-off analysis of metrics like MK, LR+ , POR, and FDR for model evaluation.
- Comparison with Existing Techniques: Proposed models show superior accuracy, PPV, TPR, and F1-Score compared to existing methods.

## ARTICLE INFO

*Keywords:*
Cyber-Attacks
Deep Learning
Intrusion Detection System (IDS)
Internet of Medical Things (IoMT)
Security

## ABSTRACT

The Internet of Things (IoT) revolutionizes industries like healthcare, agriculture, smart cities, and weather forecasting by enabling vast device networks to interact and transmit information. However, traditional network security methods are inadequate for IoT due to the limited storage and processing power of IoT devices. As IoT systems encounter numerous cyber threats, it is essential to develop innovative security approaches. Intrusion detection systems (IDS) are a popular mechanism for identifying and preventing systems from attacks. Therefore, this paper explores integrating deep learning (DL) models into IDS frameworks to enhance their capabilities. Deep learning is a subset of Artificial Intelligence (AI) that can extract complex patterns from data, adapt to new threats, and provide high accuracy and real-time detection. To provide the solution, this research presents novel deep learning models, including Embed-Net (Categorical Embedding Neural Network), the collaborative ConvNet-SVM model (combination of Convolutional Neural Network (CNN) and Support Vector Machine (SVM)), and the DeepSVM-Net (Deep Neural Network emulated by SVM), which are designed to detect network-based attacks on the Internet of Medical Things (IoMT). These models are evaluated using real-time benchmark datasets: ECU-IoHT, NF-BoT-IoT, and Wustl-HDRL-2024 by applying advanced preprocessing techniques that demonstrate superior performance by achieving an accuracy of 0.9990, 0.9959, and 0.9987with the lowest FNR of 0.0001, 0.0059, and 0.0014 by all three proposed models as compared to traditional methods. Furthermore, this paper conducted an ablation study which shows that our models have achieved outstanding performance with minimum training and validation losses of 0.0034 and 0.0008 for EmbedNet, 0.0018 and 0.0008 for ConvNet-SVM, and 0.0044 and 0.0016 for DeepSVM-Net. This research has enhanced IDS effectiveness, accuracy, and resilience by contributing significantly to cybersecurity, with the proposed models showing a remarkable improvement percentage ranging from 3.5 % to 7.5 % over the state-of-the-art.

## 1. Introduction

The Internet of Things (IoT) is a well-known technology that enables a vast network of devices and machines to interact and transmit information. IoT is extensively employed in sectors such as healthcare, agriculture, smart cities, and weather forecasting, which achieve remarkable advancements. According to a recent study, over 13.9 billion IoT devices are in use globally, with predictions indicating an increase to

* Corresponding author.
*E-mail addresses:* nikhilsharma_2k21phdco04@dtu.ac.in, nikhilsharma1694@gmail.com (N. Sharma), prashant.shambharkar@dtu.ac.in (P.G. Shambharkar).

https://doi.org/10.1016/j.asoc.2025.113420
Received 25 November 2024; Received in revised form 27 May 2025; Accepted 1 June 2025
Available online 4 June 2025

## Conferences Proofs

***Paper 1:*** Sharma, N., & Shambharkar, P. G. (2022). Applicability of ML-IOT in Smart Healthcare Systems: Challenges, Solutions & Future Direction. *2022 International Conference on Computer Communication and Informatics (ICCCI)*, 1–7. https://doi.org/10.1109/iccci54379.2022.9740983 ***(Published) (Scopus-Indexed)***


2022 International Conference on Computer Communication and Informatics (*ICCCI*), Jan. 25 – 27, 2022, Coimbatore, INDIA

# Applicability of ML-IoT in Smart Healthcare Systems: Challenges, Solutions & Future Direction

Nikhil Sharma, Prashant Giridhar Shambharkar
Department of Computer Science & Engineering
Delhi Technological University
Delhi, India
nikhilsharma1694@gmail.com
prashant.shambharkar@dtu.ac.in

***Abstract*—In today's world, as population is at high peak and due to changing life style of people, individuals are suffering from various chronic disease. With shift towards modern methodology, involvement of human efforts has decreased, as know a day's people need to finish particular amount of task within few hours and with less effort. No doubt technology has made less intervention of human but it has certain limitations too. Due to less physical involvement, humans are more prone to diseases. Internet of things (IoT) plays a very crucial role in health care sector. Using various sensors, it become possible to trace the medical health condition of the human, and a message can be forwarded to nearby hospitals which helps the patients with ease. In this paper, three different diseases like heart disease, diabetes and novel COVID-19 are discussed where different machine learning algorithms are reviewed with involvement of IoT sensors.**

***Keywords*— *Smart Healthcare Systems, Internet of Things, Machine Learning, Privacy, Security.***

## I. Introduction

The continuous productivity and innovation in the field of healthcare industry will fulfil needs of the people. We have already witnessed the implication of real time data. In regards to Covid 19, the information related to patient's health like symptoms or body temperature is exceedingly helpful [1]. By 2050, 22% of the overall population will reach to the age 60 [2]. Along with health-related emergency, the people affected by the chronic diseases are increasing day by day, which will directly create the pressure on the shoulder of the healthcare industry. In the future, health industry may need to tackle with large number of chronic problems, if we are not taking it seriously in the present time. Recently, Covid 19 has already accentuated the value of intelligent medical-care and accurate e-medical care involving diver's kind of physiological and medical information to diagnose the virus. According to prediction in 2019, the world smart health market has reaches to 143.6 billion USD, and between 2020 to 2027, an average growth rate may further be expanded by 16.2% [3]. Smart healthcare system uses devices such as mobile internet, Internet of Things (IoT), wearable appliances, to collect health information, preparing the health records of the patients. Hospital, physicians, research bodies, staff belongs to the diverse actor's category of intelligent medical treatments. It includes a dynamic framework with multiple features like assessment and evaluation, diseases identification and prevention, as well as management of medical research, healthcare and patient decision making. The features of intelligent healthcare include the automated networks such as cloud networking, artificial intelligence, 5G, Big Data, mobile internet, biotechnology, and IoT. Sensors are also become part of our lives as they are embedded into different devices using automation, computer technology, and automated signal processing. The data produced by the sensors help doctors to quickly recognize the critical situations of the patient. Medical signals like electromyograms (EMGs), Blood pressure (BP), Electroglottographs (EGGs), body temperature, heart rate (HR), and electroencephalograms (ECGs) receives in the form of 1D & 2D signals [4]. To monitor the patient, these medical signals will be used by the healthcare monitoring system. IoT is the major source through which both consumer and doctor get connected. The body check-ups like EEGs, BP readings, Glucose receptors, ultrasounds, and ECGs, help to monitor patients' wellness. Many hospitals are using smart beds which automatically identify the movement of the patient and act accordingly by adjusting the location and angle of the bed. Internet of medical things (IoMT) play vital role in establishing smart healthcare industry. Sometimes to diagnose a disease, it is not possible to depend upon only one type of medical signals. At different levels like classification level, feature level, and data level, these signals can be fused. Due to this, many challenges may be experienced by the system like classification fusion, data buffering, synchronization while receiving signals from divers' sensors, and feature normalization. The technology like Artificial Intelligence (AI), wireless local area network (WLAN), deep learning (DL), and Machine learning (ML) bring revolution in the field of intelligent healthcare to guarantee satisfaction to both stakeholders and patients. For many medical situations, IoMT (Internet of Medical Things) systems deliver the excellent assistance like for health conditions, implantable device like pacemakers is used to control the heartbeat. For improving medical-care experience, the assisting wearables devices like smartwatches are used to monitor the heart rate [5]. Security is the only need for the success of IoMT system [6]. The 11 set of security required by this system to deliver the integrity, non-repudiation, confidentiality, authentication, and data availability. As these security demands are fulfilled by the traditional system as well, even though this system failed to provide security. Because of other system, specification requirement and power consumption lead to the system failure [7]. Based on cryptography, several techniques have been introduced by the researchers which are designed for IoT and IoMT systems. The techniques include keyless noncryptographic techniques, asymmetric cryptography, and symmetric cryptography.

The remainder of the paper is arranged as follows: Section 2 discussed about Literature review; Section 3 introduced Applicability of ML-IoT (Machine Learning &

978-1-6654-8035-2/22/$31.00 ©2022 IEEE



2022 International Conference on Computer Communication and Informatics (ICCCI) | 978-1-6654-8035-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCCI54379.2022.9740983

***Paper 2:*** Sharma, N., & Shambharkar, P. G. (2024). A systematic literature review of the emerging technologies used in securing healthcare data. *2024 12th International Conference on Internet of Everything, Microwave, Embedded, Communication and Networks (IEMECON)*, 1–12. https://doi.org/10.1109/iemecon62401.2024.10846068 ***(Published) (Scopus-Indexed)***

IEMECON 2024

# CERTIFICATE

**OF APPRECIATION**

THIS CERTIFICATE IS PRESENTED TO

Nikhil Sharma

*For the*

**Presentation of his/her Research Paper entitled**

*A Systematic Literature Review of the Emerging Technologies used in securing healthcare data*

*In*

*12th International Conference on Internet of Everything, Microwave, Embedded, Communication & Networks (IEMECON 2024)*

*Organized by*

*Department of Electronics & Communication Engineering, University of Engineering & Management, Jaipur, Rajasthan, India*

***24th — 26th October, 2024***

**Venue:** Rajasthan International Centre (RIC), Jaipur.

*Prof. (Dr.) Biswajoy Chatterjee* — Patron

*Prof. (Dr.) Pradeep Kumar Sharma* — Finance Chair - IEMECON 2024

*Prof. (Dr.) Prashant Ranjan* — Convenor - IEMECON 2024

12th IEEE International conference on Internet of Everything, Microwave, Embedded, Communication & Networks (IEMECON-2024)

## A Systematic Literature Review of the Emerging Technologies used in Securing Healthcare Data

Nikhil Sharma[1*], Prashant Giridhar Shambharkar[2]
[1,2]Department of Computer Science & Engineering
[1,2]Delhi Technological University
[1,2]Delhi, India
nikhilsharma1694@gmail.com[1]
prashant.shambharkar@dtu.ac.in[2]

***Abstract*— This study investigates the unification of blockchain technology and artificial intelligence (AI) models to improve the security and privacy of healthcare records. As healthcare systems increasingly depend on digital technologies, the protection of sensitive patient information has become a critical challenge. Blockchain's decentralized, immutable ledger offers robust data integrity and transparency, while AI techniques provide powerful tools for predictive analytics and decision-making. However, combining these two technologies introduces substantial challenges, including scalability issues, privacy concerns, and the requirement for regulatory compliance. This study systematically reviews existing literature to identify key challenges and promise solutions in the amalgamation of blockchain and AI within healthcare. We evaluate the applicability, objectives, techniques, and security measures of selected studies to assess their relevance and contribution to the field. Our findings reveal that while significant progress has been made, gaps remain in areas such as the alignment of these technologies with regulatory frameworks, the development of privacy-preserving AI methods, and the protection of AI models from adversarial attacks. The paper concludes by proposing a set of research questions and future directions to address these challenges, with the aim of advancing the secure and ethical integration of blockchain and AI in healthcare.**

***Keywords—Artificial Intelligence, Blockchain, Smart Healthcare System, Cyber-attacks, Internet of Things (IoT).***

### I. INTRODUCTION

The healthcare area is experiencing a substantial digital change, led by the extensive use of sophisticated technologies aimed at effectively controlling and analyzing large sizes of complex records. This information, encompassing electronic health records (EHRs), diagnostic images, genetic sequences, and other essential patient information, which is pivotal to contemporary healthcare systems [1]. It is essential for increasing care quality, expanding medical research, facilitating precision medicine, and ultimately enhancing patient outcomes. However, the swift digitalization of healthcare presents significant concerns, especially in guaranteeing the security and privacy of this rapidly digitized and interconnected information.

Healthcare records are both vulnerable and significant, making it an ideal target for cyberattacks. Breach of healthcare records can have disastrous effects, including identity theft, financial loss, compromised patient safety, and a loss of faith in healthcare organizations [2]. As the size and intricacy of healthcare records increase, protecting its security, integrity, and availability has become a top priority. The growing number and sophistication of cyber-attacks highlight the critical demand for strong security measures to defend healthcare records from unwanted attackers. In response to these growing issues, recent technical breakthroughs have created new tools and frameworks for securing healthcare data. Among these, blockchain technology is a particularly viable solution due to its decentralized and immutability. Blockchain offers a transparent, tamper-resistant architecture for data management, with each transaction cryptographically secured and recorded on a distributed ledger. This technology assures that healthcare data cannot be edited or destroyed without detection, therefore preserving the integrity and security of patient information.

Alongside blockchain, Deep Learning (DL) and Machine Learning (ML) techniques have also shown significant potential in enhancing healthcare data security [3]. These technologies excel at processing and analyzing large, complex datasets, making them well-suited for detecting anomalies, identifying patterns indicative of cyber threats, and predicting potential security breaches [4]. By employing advanced algorithms, DL and ML models can constantly realise from new information, alter to developing threats, and provide real-time insights that strengthen the overall security posture of healthcare systems.

Despite the promising capabilities of blockchain, DL, and ML, their integration into healthcare systems presents several challenges [5]. Blockchain, while offering unparalleled security benefits, faces issues related to computational complexity, scalability, and energy consumption. These challenges are particularly pronounced when integrating blockchain with resource-constrained Internet of Medical Things (IoMT) devices, which are becoming progressively prevalent in modern healthcare environments [6]. Moreover, the immutability of blockchain, while advantageous in many contexts, raises concerns about managing erroneous or outdated information in healthcare, where data accuracy is vital. Similarly, the application of DL and ML in healthcare security is still in its early stages. These technologies offer powerful tools for data analysis and threat detection, but several issues remain unresolved [7]. These include the need for large, high-quality labelled datasets for training models, challenges related to the interpretability of complex models (often referred to as the "black box" problem), and difficulties in handling imbalanced datasets where certain types of attacks may be underrepresented [8]. Additionally, the deployment of these models in real-world healthcare settings must account for factors such as computational resource availability, model robustness, and the potential for adversarial attacks.

Considering these challenges, a comprehensive review of the state-of-the-art research on using DL, ML, and blockchain in healthcare data security is essential. Through a critical analysis of the literature work already done in this area, this systematic literature review aims to fill this gap [9]. The review seeks to verify the benefits and drawbacks of existing methods, evaluate the efficacy of various strategies, and bring

979-8-3503-8731-5/24/$31.00 ©2024 IEEE



2024 12th International Conference on Internet of Everything, Microwave, Embedded, Communication and Networks (IEMECON) | 979-8-3503-8731-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/IEMECON62401.2024.10846068

# Author's Biography

**Nikhil Sharma** received his B.Tech. Degree from Guru Gobind Singh Indraprastha University, Delhi, India, in 2016. He completed his M.Tech. in Information Security from Ambedkar Institute of Advanced Communication Technologies & Research, Government of NCT of Delhi (currently Netaji Subhash University of Technology East Campus), Delhi, India, in 2018. He is pursuing his PhD in Computer Science and Engineering at Delhi Technological University (DTU), Delhi, India.

His research interests include Machine Learning, Deep Learning, Blockchain, Internet of Things (IoT), and Information Security, focusing on developing intelligent, secure, and scalable solutions for healthcare systems. His research centres on designing a comprehensive security framework for the Internet of Medical Things (IoMT) by integrating Deep Learning-based Intrusion Detection and Blockchain-enabled privacy preservation mechanisms.

He has published extensively in peer-reviewed international journals and conferences and actively contributes to the academic community as a reviewer for several high-impact Journals and Conferences in Artificial Intelligence, Cybersecurity, and next-generation networks.