5-22-2024

A Distributed and Hybrid AI-Based Security Framework for 5G Real-time Applications

Ali Ghubaish
Washington University – McKelvey School of Engineering

WASHINGTON UNIVERSITY IN ST. LOUIS

McKelvey School of Engineering
Department of Computer Science & Engineering

Dissertation Examination Committee:
Roger Chamberlain, Chair
Ashutosh Dutta
Raj Jain
Alvitta Ottley
Ning Zhang

A Distributed and Hybrid AI-Based Security Framework for 5G Real-time Applications

by
Ali Hussain A Ghubaish

A dissertation presented to
the McKelvey School of Engineering
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

August 2024
St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgments

Completing this PhD has been a profound journey, one enriched and made possible by the support and encouragement of many. I am deeply grateful to each person who has contributed to my academic and personal growth during this pivotal phase of my life.

First and foremost, my heartfelt thanks go to my mother, Khadijah, whose unwavering support and belief in me have been my constant source of motivation throughout my school years and up to this PhD degree. Her emotional and financial backing has been crucial, fueling my pursuit of knowledge and academic success. Her passion for education and desire for it to be a vital part of our lives have truly shaped my path.

To my loving wife, Khadijah, my soulmate, who has shown an extraordinary level of dedication and support. She postponed her own graduate studies until the last semester of my PhD to provide emotional and physical support and to care for our daughters, Ola and Hala, ensuring that I could focus wholly on completing this degree. We embarked on this journey together, getting engaged just one month before I started the PhD program and married by the end of the first year. Your sacrifices and love have been the bedrock of my success.

To my grandmother, Saadah, whose gentle encouragement and wisdom have been a guiding light throughout this journey. To my aunt Fatimah, an assistant professor whose own academic achievements and nurturing support have inspired me since my undergraduate days. Fatimah has not only ignited my ambition to achieve a PhD but has also treated me as her own son, constantly pushing me toward this academic milestone.

Special thanks are also due to my aunt, Moluk, and my uncle, Alawi. Your enduring support and belief in my capabilities have been a comfort and motivation throughout the ups and downs of this PhD journey.

To my siblings—Saja, Khaled, Fawaz, Abdullah, Faisal, and Marwah—each of you has contributed to this journey in your own unique way, providing encouragement and a sense of home. I extend special thanks to my elder sister, Saja, and her daughters, Alreem, Alma, and Aljazi, and to my younger brother, Khaled, and his son, Abdulaziz, for their love and cheer.

I owe a profound debt of gratitude to my advisor, Professor Raj Jain, who has been a guiding light since my Master's degree. Over the past eight years, you have not only taught me invaluable lessons in research but in life as well. Your mentorship has transcended the academic realm, providing me with support in both difficult personal times and academic challenges. You have truly been an academic role model for me, and the lessons I have learned from you are immeasurable and something I could never repay. Your nomination of me as one of the top five students for this PhD program was a pivotal moment in my academic career.

To my dissertation committee, your insights and feedback have been indispensable. Professor Zhang, you have been a mentor and a supporter from the very start of my PhD program, always there when I needed guidance. Professor Chamberlain, having known you since my Master’s degree, your courses and mentorship have profoundly shaped my academic path. Professor Ottley, your readiness to step into the committee last minute was a testament to your commitment to education and your students. Professor Dutta, your work at JHU and our collaboration on the NSF proposal have been incredibly enriching experiences for me.

To my lab friends, each of you has contributed to my PhD experience in significant ways. Lav Gupta, a friend despite our age difference, has guided me through many challenges. Tara Salman, a rival and advisor all at once, pushed me to excel in every endeavor, from coursework to manuscript writing. Her tough critiques ensured my work was always polished, and our late-night sessions were both rigorous and rewarding. Maede Zolanvari, a keen editor and a supportive friend, was invaluable in refining my papers. Zebo Yang, although newer to the program, offered critiques with respect and supported me through many challenges, continuing the legacy of friendship and guidance I valued with Tara. Additional gratitude goes to friends like Irfan Alahi, Hannen Alfauri, Modhi Bin Kulaib, Arghya Datta, Behrooz Farkiani, Gustavo Gratacos, Yin Li, and especially Yousef Alshehri, a brother from a different mother with whom I shared the entirety of my academic journey from undergraduate to PhD. His friendship and solidarity, despite us attending different universities, have been a cornerstone of my academic and personal growth. Saeed Almutlaq, a steadfast friend since our undergraduate days, has remained a constant source of support and camaraderie, keeping in touch and encouraging me throughout my academic journey.

I extend my deepest appreciation to my government, Saudi Arabia, and my employer, Prince Sattam Bin Abdulaziz University, for sponsoring me with a full scholarship to pursue this degree. Their generous support has covered everything we needed or might have needed, allowing me to focus solely on my academic and research endeavors without financial concerns. This sponsorship has been instrumental in my success, and I am truly grateful for this opportunity.

A special thank you to the staff and faculty of the department. Professor Cytron, your course on quantum computing opened new vistas of knowledge for me. Myrna Harbison, Monét Demming, Kelli Eckman, Sharon Matlock, Lia Garofolo, Cheryl Newman, and Cleopatra Benos—your support and assistance behind the scenes have been essential to my success.

I would also like to extend my thanks to everyone who has supported me throughout this journey, whether mentioned by name or not. I regret any omissions and deeply appreciate every contribution that has helped guide me to this point. Your collective support, wisdom, and encouragement have been invaluable, and I am truly thankful for every gesture of kindness and support.

Each of you has left a mark on my journey, and I am forever grateful for your presence in my life. Thank you for the wisdom, the laughter, and even the challenges, as each was instrumental in shaping my path to this achievement.

Ali Hussain A Ghubaish

Washington University in St. Louis

August 2024

Dedicated to my family.

ABSTRACT OF THE DISSERTATION

A Distributed and Hybrid AI-Based Security

for 5G Real-time Applications

by

Ali Hussain A Ghubaish

Doctor of Philosophy in Computer Engineering

Washington University in St. Louis, 2024

Professor Roger Chamberlain, Chair

This dissertation develops a multifaceted security framework tailored for 5G-enabled real-time Internet of medical things (IoMT) systems to significantly enhance the security infrastructure within healthcare environments. The framework pivots around three core technological advancements: the development of the light feature engineering based on the mean decrease in accuracy (LEMDA), the construction of a 5G testbed that serves as a distributed intrusion detection system (IDS), and the implementation of a hybrid deep reinforcement learning (HDRL) method.

LEMDA represents a breakthrough in data processing for IoMT systems. By intelligently reducing data complexity, LEMDA enhances the speed and accuracy of threat detection mechanisms, which is crucial for handling the immense volumes of data generated in healthcare settings. This method speeds up the detection process and ensures that essential data nuances are not lost, thereby maintaining high precision in threat identification.

Establishing the 5G testbed introduces a novel approach to distributed IDS. This testbed leverages the latest in 5G and multi-access edge computing (MEC) technologies to distribute the processing

load, thereby enhancing the overall resilience and efficiency of the network. This strategic distribution also helps overcome traditional challenges associated with centralized systems, such as scalability issues and vulnerability to single points of failure. Furthermore, this initiative has led to creating a new dataset specifically designed to support the development of IDS methodologies congruent with the architectures of 5G and MEC. This dataset is a valuable resource for researchers across both academic and industrial spheres, facilitating the advancement of tailored intrusion detection strategies.

Lastly, the HDRL method integrates deep learning and reinforcement learning techniques tailored to harness network and host data for improved threat detection. This innovative approach dynamically adapts to evolving threat landscapes, reducing the need for constant human supervision and frequent retraining. The HDRL method showcases a significant enhancement in threat detection efficacy, setting new benchmarks in the field.

In addition to these primary contributions, the dissertation delves into creating comprehensive datasets through the EHMS testbed and reviews current IoMT security measures and attack techniques. These endeavors provide a holistic view of the security landscape and inform the development of the proposed security framework.

Chapter 1:Introduction

The Internet of things (IoT) has been transformative across various sectors, especially healthcare, through the Internet of medical things (IoMT). Integrating these advanced technologies within healthcare frameworks presents significant opportunities and complex challenges. By the end of 2024, there are estimated to be over 207 billion IoT devices globally, indicating a continued rapid growth trajectory [1]. In healthcare, IoMT devices are projected to significantly expand their presence within the IoT sector, with the market expected to reach a value of USD 169.99 billion by 2030 [2]. This growth is driven by the devices' potential to enhance efficiencies in managing chronic diseases and expanding telehealth services. The rapidly growing sector underscores its vast economic impact and highlights potential vulnerabilities, particularly in cybersecurity.

The widespread adoption of IoT solutions has profoundly impacted daily life, revolutionizing the healthcare industry with the development of IoMT. These compact, versatile devices are crucial for healthcare applications, offering significant cost reductions and improvements in care delivery. However, securing these devices remains a considerable challenge as they process and store critical health data that, if compromised, could threaten patient safety and privacy. Since 2020, healthcare statistics breach costs have increased by 53.3%. The extraordinarily regulated healthcare enterprise has visible a considerable upward thrust in information breach charges since 2020. For the 13th year in a row, the healthcare enterprise said the maximum high priced records breaches, at an average fee of USD 10.93 million [3].

IoT's expansion into healthcare has enabled the development of sophisticated, low-cost, low-power monitoring systems that enhance patient care through continuous health monitoring and real-time

data processing. These systems support a range of applications from early diagnosis to emergency management, significantly reducing the need for constant physical healthcare provider presence. Machine learning (ML) plays a crucial role in this context by enhancing the security of these systems, enabling the detection of both known and previously unseen cybersecurity threats, thus ensuring the protection of sensitive health data, crucial for maintaining patient confidentiality and system integrity.

With IoT expected to reach over 25 billion devices by 2030 [4], the role of advanced technologies like 5G becomes increasingly essential. These technologies are pivotal in managing the vast data produced by IoT devices, particularly in healthcare settings, where data sensitivity and the need for rapid processing are paramount. The advent of 5G technology offers transformative potential for healthcare applications, significantly enhancing IoMT and mobile healthcare capabilities. Integrating multi-access edge computing (MEC) within 5G infrastructures optimizes processing extensive data loads closer to their source, which is crucial for applications requiring low latency, such as remote surgeries. However, the distributed nature of these new technologies introduces additional security challenges, necessitating the development of dynamic and intelligent intrusion detection systems (IDS) capable of adapting to evolving threats with minimal human intervention.

This dissertation highlights the various cyber threats that can compromise the integrated IoT and IoMT systems, particularly within a 5G framework. These include:

- **Network Communication (Man-in-the-middle (MitM) attacks)**: Data between user equipment (UE) and MEC servers in the 5G network is vulnerable to interception and alteration, compromising data integrity and confidentiality.

- **Server Availability (Distributed denial of service (DDoS) attacks)**: MEC servers can be overwhelmed by excessive traffic originating from within the network, disrupting services and affecting the normal functioning of network services.

- **Endpoint Security (Ransomware attacks)**: UE operating in the 5G network is susceptible to ransomware attacks that encrypt data, demand a ransom for decryption, and threaten data availability and financial stability.

- **Application and System Integrity (Buffer overflow attacks)**: Applications and operating systems on UE are at risk from buffer overflow vulnerabilities, which allow arbitrary code execution, undermining system and data security.

The profound integration of IoT, particularly through IoMT in healthcare, underscores a pivotal transformation in how medical services are delivered and monitored. However, this transformation is accompanied by an escalating complexity in cybersecurity threats that exploit the vulnerabilities inherent in these rapidly evolving technologies. As detailed in the preceding chapters, while the advancements in IoT and related technologies such as 5G and MEC bring considerable benefits, they also introduce significant risks that must be meticulously managed. This transition highlights the challenges and sets the stage for discussing this dissertation's innovative contributions. The subsequent sections will delve into specific challenges these technological integrations present, particularly focusing on data complexity, system vulnerabilities, and the dynamic nature of cyber threats. Following this, we will explore the strategic contributions made by this research in addressing these challenges, detailing the development of robust security frameworks and advanced detection systems designed to fortify the integrity and reliability of IoMT systems.

1.1 Challenges

The primary challenges addressed in this dissertation reflect the complex nature of modern cybersecurity in healthcare environments, where technology integration continuously evolves, and the potential for cyber threats escalates. Healthcare systems increasingly depend on IoMT technologies, which enhance patient care and expose sensitive data and critical operations to cyber risks. These systems must handle massive amounts of confidential data, making them attractive cyberattack targets. Furthermore, the need for real-time data processing in medical settings amplifies the challenges, as any delay or disruption in data handling can have dire consequences. Addressing these issues requires a robust cybersecurity framework that defends against a broad spectrum of cyber threats and ensures compliance with stringent regulatory requirements for data protection and patient privacy. This dissertation aims to tackle these challenges by introducing innovative security solutions tailored to healthcare cybersecurity's dynamic and complex landscape.

1.1.1 Data Complexity and Volume

The IoMT devices used in modern healthcare environments produce a staggering volume of data, encompassing everything from patient vital signs to operational telemetry of medical devices. This vast data challenges traditional systems' storage capacities and strains the processing capabilities necessary for timely and effective security analysis. The complexity is further compounded by the diverse nature of the data, which ranges from structured numerical data to unstructured video feeds and images, each requiring different handling and security protocols. Effective management of this data is crucial, as any compromise in data integrity or a delay in its processing could directly impact patient care and safety.

1.1.2 Centralized IDS Vulnerabilities

Relying on centralized systems for managing and securing IoMT data poses significant risks, such as becoming single points of failure during cyberattacks like DDoS or MitM attacks, often facilitated by insiders with authorized access. These systems' susceptibility to attacks leads to potential disruptions in healthcare services, causing critical delays in patient care and data breaches. Additionally, as the number of connected devices increases, centralized systems face scalability issues that affect performance and complicate timely updates crucial for security. These challenges underscore the need for more robust, adaptive security frameworks and the development of decentralized or distributed mechanisms to enhance the resilience and reliability of healthcare cybersecurity infrastructures.

1.1.3 Dynamic and Evolving Threat Landscape

Cyber threats in the healthcare sector are exceptionally dynamic, with new vulnerabilities and attack vectors emerging continually. Traditional IDS often struggle to keep pace with the rapid development of sophisticated cyberattack methods, such as polymorphic malware or advanced social engineering tactics targeting the healthcare sector. Moreover, the stakes are incredibly high in healthcare, where a successful attack can result in more than just financial losses or data breaches—it can directly endanger lives. This necessitates a security system that responds to known threats and can predict and mitigate new threats as they emerge, ensuring continuous protection of critical healthcare infrastructure.

1.2 Contributions

The research undertaken in this dissertation follows a structured sequence of investigations and implementations that build upon each other to advance the field of IoMT security for healthcare systems. The progression of the contributions is as follows:

1) State-of-the-Art IoMT Security Survey and Framework Development:

We began our research by conducting a comprehensive survey of the current state-of-the-art security approaches in IoMT for healthcare systems. This survey helped us identify gaps and opportunities for enhancement, creating a secure framework tailored to address these specific needs.

2) EHMS Testbed Construction:

Recognizing the scarcity of healthcare datasets available to academic and industrial researchers and the need to demonstrate the benefits of integrating network and biometric data, we constructed the enhanced healthcare monitoring system (EHMS) testbed. This testbed not only facilitates the collection of rich datasets but also allows us to compare the effectiveness of using combined data types against each individually, highlighting the advantages of a multi-modal approach.

3) Implementation of LEMDA for Feature Reduction:

Building on the insights from the EHMS testbed, we implemented the light feature engineering based on the mean decrease in accuracy (LEMDA). This method significantly reduces the number of features required to train the IDS model, speeding up attack detection and enhancing overall IDS performance. LEMDA utilizes the diverse dataset generated from the EHMS testbed, proving its efficacy in real-world scenarios.

4) Development of a 5G-Based Testbed:

Developing a 5G-based testbed is key to addressing the challenges posed by new network technologies and specific threats like MitM and ransomware attacks outlined in the threat model. This testbed offers a controlled environment for simulating these attacks, enabling the study and enhancement of defense mechanisms against insider threats. It supports the creation of dynamic, intelligent IDS that adapt to evolving threats with minimal human oversight. This section provides a rationale for the proposed cybersecurity strategies by incorporating the threat model. It highlights the practical application of these findings to develop advanced, tailored solutions that enhance IoMT security in healthcare settings.

5) Design of a Hybrid Deep Reinforcement Learning Method:

Finally, we designed a hybrid deep reinforcement learning (HDRL) method that integrates both network and host data features to improve the threat detection efficacy of the IDS. This innovative approach reduces the need for human supervision or frequent retraining, as it dynamically adapts to new threats and evolving attack patterns using a hybrid dataset collected from our advanced 5G testbed.

Together, these contributions form a coherent and structured approach to tackling the security challenges in IoMT for healthcare systems, demonstrating significant advancements in IDS' efficiency, reliability, and robustness.

1.3 Dissertation Structure

The rest of the dissertation has the following structure:

Chapter 2 – Development of Secure IoMT Framework [5]: Details the survey findings on existing security measures and attacks and discusses the development of a new security framework based on these insights.

[5] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the internet-of-medical-things (IoMT) systems security," *IEEE Internet of Things Journal,* vol. 8, no. 11, pp. 8707-8718, 2020.

Chapter 3 – EHMS Testbed Implementation[6]: Describes the creation of the EHMS testbed, its operational setup, and its role in generating a valuable dataset for intrusion detection research.

[6] A. A. Hady*, A. Ghubaish*, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access,* vol. 8, pp. 106576-106584, 2020. (*Equal Contribution).

Chapter 4 – Feature Reduction with LEMDA [7]: Explores the design and implementation of LEMDA, showcasing its effectiveness in enhancing IDS performance through feature reduction.

[7] A. Ghubaish, Z. Yang, A. Erbad, and R. Jain, "LEMDA: A Novel Feature Engineering Method for Intrusion Detection in IoT Systems," *IEEE Internet of Things Journal,* 2023.

Chapter 5 – Advancements with 5G Testbed and HDRL Method [8]: This chapter discusses significant advancements in 5G technology and the HDRL method, detailing their response to the threats identified in the threat model. It highlights how these technologies improve IDS efficiency and network resilience by adapting to evolving cybersecurity challenges, particularly insider attacks and 5G vulnerabilities.

[8] A. Ghubaish, Z. Yang, and R. Jain, "HDRL-IDS: A Hybrid Deep Reinforcement Learning Intrusion Detection System for Enhancing the Security of Medical Applications in 5G Networks," in *2024 IEEE International Conference on Smart Applications, Communications and Networking (SmartNets)*, Harrisonburg/Washington DC, VA, USA, 2024.

Chapter 6 – Conclusion and Future Work: Summarizes the research contributions, discusses the practical implications, and suggests directions for future research in IoMT security.

Chapter 2: Recent State-of-the-Art Security Approaches

Building upon the transformative impact of IoT in healthcare, this chapter focuses on analyzing the security challenges specific to IoMT systems. In this chapter, we present state-of-the-art techniques to secure IoMT systems' data during collection, transmission, and storage [5]. We comprehensively overview IoMT systems' potential physical and network attacks. Our findings reveal that most security techniques do not consider various types of attacks. Hence, we propose a security framework that combines several security techniques. The framework covers IoMT security requirements and can mitigate most known attacks.

[5] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the internet-of-medical-things (IoMT) systems security," *IEEE Internet of Things Journal,* vol. 8, no. 11, pp. 8707-8718, 2020.

2.1 Introduction and Motivation

The healthcare industry's increasing reliance on IoMT systems introduces complex security challenges. The healthcare data involved in IoMT systems requires protection at various stages, including data collection, transmission, and storage. Breaches have a significant impact, as nearly half of IoMT devices are vulnerable to exploits [9]. These systems differ from others as they directly affect patient safety and privacy [10]. Furthermore, healthcare data's high value makes it a lucrative target on the black market.

To ensure the success of IoMT systems, robust security measures are essential. These systems must ensure data confidentiality, integrity, availability, non-repudiation, and authentication (CIANA) [11]. While traditional security approaches offer some protection, power consumption

and other constraints on IoMT devices may limit their effectiveness [11]. Researchers have therefore focused on developing techniques specifically designed for IoMT and IoT environments. These techniques broadly fall into symmetric cryptography, asymmetric cryptography, and keyless non-cryptographic approaches.

Existing literature extensively reviews the limitations, security issues, and potential solutions within IoMT systems. Yaacoub et al., for example, classify these techniques as cryptographic and non-cryptographic, further categorizing countermeasures into authorization, availability, IDS, and awareness [12]. Vyas and Pal address open issues such as flexibility, single point of failure, and emergency handling [13]. Additionally, Bhushan and Agrawal explore securing patient data in the cloud within IoMT systems [14].

Advanced technologies like ML, artificial intelligence (AI), and blockchain offer promising potential to enhance IoMT security [15, 16]. These techniques can improve system performance, provide tolerance against specific attacks like denial-of-service (DoS), and address issues like single points of failure. ML, in particular, can significantly reduce physical layer authentication errors compared to traditional methods [17, 18].

The rest of the chapter is organized as follows: A brief background of the IoMT system types and architecture is provided in Section 2.2. In Section 2.3, we present IoMT threats at different stages, along with security requirements and different types of security techniques. State-of-the-art security techniques, including symmetric, asymmetric, and keyless, are discussed in Sections 2.4, 2.5, and 2.6, respectively. The IoMT attack surface is described in Section 2.7, while our proposed security framework is presented in Section 2.8. Finally, we summarize the chapter in Section 2.9.

2.2 Background

This section provides a background on IoMT systems as well as their architecture. This helps to understand the later sections, where we present IoMT systems' security requirements, attacks, and countermeasures.

2.2.1 IoMT Types

IoMT systems provide the necessary or improved assistance for many medical conditions. The necessary devices are implantable for particular medical conditions, e.g., pacemakers for heart conditions. On the other hand, the assisting devices are mostly wearables for improved healthcare experience, e.g., smartwatches. These differences put the IoMT systems into two categories: implantable medical devices (IMDs) and Internet of wearable devices (IoWDs).

1) Implantable Medical Devices (IMDs)

Any device implanted to replace, support, or enhance a biological structure is an IMD. For example, a pacemaker is an IMD that helps control abnormal heart rhythms, i.e., by promoting the heart to beat at a normal rate if it is beating too fast or too slow [19]. Figure 2.1 shows several popular IMDs and their placement locations in the human body. Wireless IMDs have been proposed to solve problems associated with wired IMDs, e.g., infection and cable breakage [20]. IMDs are mostly very small and have very long battery life. Hence, low power consumption, small storage space, and small batteries that last long are essential for these devices to stay inside a human body for a long time. For instance, pacemaker implants last 5 to 15 years [21].

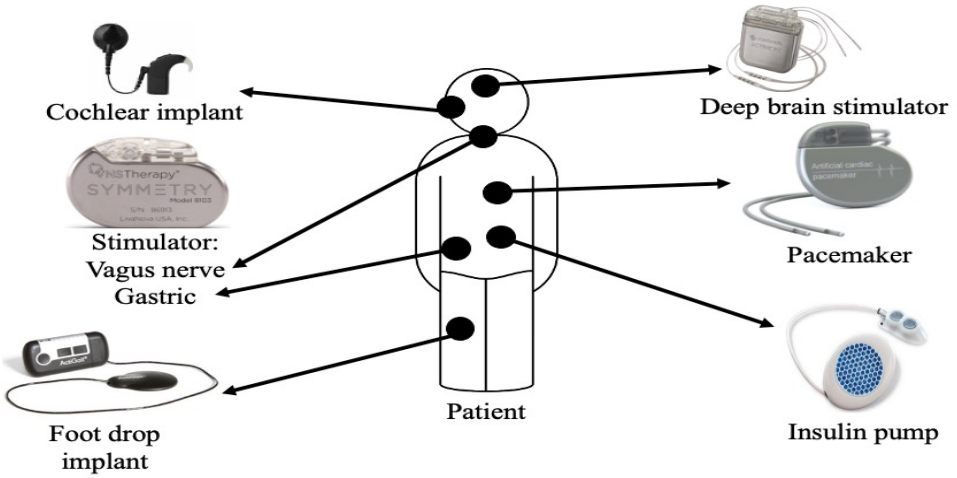


Figure 2.1 Examples of IMDs and their locations in the human body.

2) Internet of Wearable Devices (IoWDs)

Individuals wear these devices to monitor their biometrics, e.g., heart rate, which may help improve their overall health. Examples include smart watches, fall detection bands, electrocardiogram (ECG), and blood pressure monitors [22]. Smartwatches are currently one of the most well-known forms of IoWDs for monitoring biometrics such as heart rate and movement. The monitoring can be used to detect slow and fast heartbeats when the individual is not active. The new watches also support fall detection and ECG readings to detect atrial fibrillation (irregular heartbeat) medical conditions. They are currently widely used for non-critical patient monitoring [23]. However, these devices have sensor accuracy and battery life limitations; thus, they are not likely to replace IMDs in critical conditions [24].

2.2.2 IoMT Systems Architecture

Most current IoMT systems are typically divided into four layers, as shown in Figure 2.2 [25]. These layers include all data stages, starting from the individual's biometric collection stage and ending in data storage and subsequent visualization by a physician for analysis. Moreover, the patient can also visualize their overall health status from the cloud. The current advances in IMDs,

IoWDs, and IMDs mostly share the same architecture, given that IMDs can communicate with the gateways, as exemplified by Medtronic peacemaker [26].

1) Sensor Layer

This layer consists of small implanted or worn sensors that collect the patient's biometrics. The data are transmitted to the second layer over wireless protocols such as Wi-Fi, Bluetooth, or the MedRadio frequency spectrum reserved for IMDs [27].

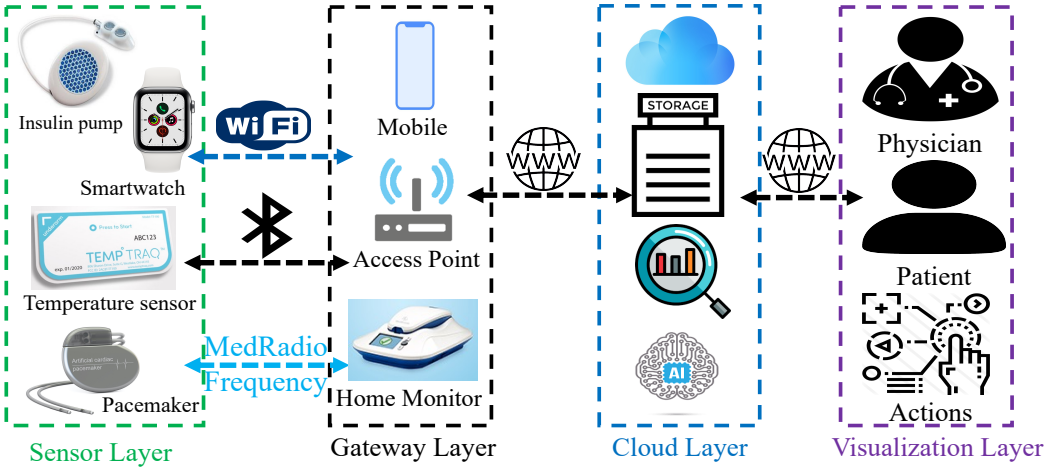


Figure 2.2 IoMT system architecture.

2) Gateway Layer

Due to the processing and storage limitations of IoMT sensors, the data are transferred without processing to the second layer, i.e., the gateway layer. The devices in this layer can be the patient's smartphone or a dedicated access point (AP), which are generally more potent than sensors. They can perform preprocessing operations like validation, short-term data storage, and simple AI-based analysis. In addition, they send the sensor data to the cloud over the Internet.

3) Cloud Layer

The cloud layer gets the data from the gateway for storage, analysis, and secure access. The analysis may include data processing to find changes in the patient's health and presenting them to the physicians or patients for further action. The key generation server (KGS) generates IDs and keys for various system nodes. The access to the sensors can be remotely managed and controlled from this layer.

4) Visualization/Action Layer

In this layer, the data are presented to the physicians and the patients to track their health. This layer also includes the actions recommended by the physician based on the patient's health conditions. Examples of actions include prescribing or adjusting the dosage for various medicines.

2.3 IoMT Security Model

In this section, we discuss the threats to the IoMT systems' data at three different stages. Also, we present the IoMT systems' security requirements and generally categorize countermeasure techniques. In subsequent sections, each countermeasure category will be further detailed with its associated techniques and use in IoMT systems.

2.3.1 IoMT Threats at Different Stages

IoMT systems must protect the patients' data at all stages, including collection, transmission, and storage. As shown in Figure 2.2, these stages combine the four architecture layers.

1) Data Collection

Collecting the patient's data in the sensor layer is the first stage of an IoMT system. Attacks at this stage can be software (i.e., data tampering) or hardware (i.e., sensor hardware manipulation) attacks. These attacks can threaten patients' lives if the sensor hardware or software is affected. Thus, protecting the data against these attacks is vital to keep the system running.

2) Data in Transit

This stage includes communications between the devices in all four layers, e.g., between the IoMT sensors in the sensor layer and the AP in the gateway layers. Attacks here can manipulate or block the sensor data being transmitted. Thus, securing against these attacks would prevent the data from being affected while being transferred among the four layers.

3) Data in Storage

After the patient's data are collected and transmitted from the sensor and gateway layers, they are stored in the cloud. Attacks in this layer vary from stealing account credentials to DoS or distributed DoS (DDoS) attacks. Protecting the data in this layer and the visualization layer from unauthorized access is essential. This is critical since, in this layer, most of the data are resting; hence, they are at more risk than any other stage.

2.3.2 IoMT Security Requirements

Due to the patient data's sensitivity and safety, a set of requirements is needed to ensure IoMT systems' security at all layers. The set has been derived from CIANA considerations and consists of the following 11 security requirements [28, 29]:

4) Confidentiality/Privacy

The ability to keep the data private while being gathered, transmitted, or stored. In addition, they must only be accessible to authorized users. The most common techniques to fulfill this requirement are data encryption and access control lists, which will be discussed further in the next section.

1) Integrity

This is related to protecting the data from unauthorized tampering during the collection, transmission, and storage stages.

2) Availability

The ability to correctly keep the IoMT systems continuously running. This can be done by keeping the system up to date, monitoring any changes in its performance, providing redundant data storage or transmission routes in case of DoS attacks, and fixing any problem as soon as possible.

3) Non-Repudiation

The ability to make each authorized user responsible for their actions. In other words, this requirement guarantees that any interaction in the system cannot be denied. This can be achieved using digital signature techniques, as discussed later in the chapter.

4) Authentication

The capability to validate the identity of a user accessing the system. Mutual authentication is the most secure form where the server and the client authenticate each other before any secure data/key exchange.

5) Authorization

The ability to allow authenticated users only to execute commands to which they are authorized. Similar to confidentiality, authorization can be achieved using proper data encryption and access control techniques.

6) Anonymity

The capability to keep the patients'/physicians' identities hidden from unauthorized users when interacting with the system. Using smart cards can fulfill the anonymity requirement.

7) Forward/Backward Secrecy

Forward secrecy provides the ability to keep future transmitted data/keys safe even if old data/keys are compromised. Backward secrecy ensures the opposite, where old data/keys are safe even if an attack has successfully affected current data/keys. Forward/Backward secrecy can be achieved by time-based authentication parameters, e.g., time-based keys that can be generated and used only when the clock time at both nodes match.

8) Secure Key Exchange

The ability to securely share the keys between the nodes in the system. Diffie-Hellman key exchange is an example of a secure key exchange.

9) Key-Escrow Resilience

The system administrator cannot impersonate any authorized user in the system. This protects against internal threats. Using asymmetric keys with a cryptographic hash function (CHF) can fulfill this requirement.

10) Session Key Agreement

The nodes in the system must use session keys after the authentication process. Like key-escrow resilience, symmetric/asymmetric keys with CHF can fulfill this requirement.

2.3.3 IoMT Systems Security Techniques

There are several different techniques to secure IoMT systems. These techniques can be divided into three main categories: symmetric, asymmetric, and keyless, as shown in Figure 2.3. Symmetric and asymmetric techniques rely on cryptographic algorithms, while keyless techniques are non-cryptographic. The cryptographic techniques explained in the following three sections include one-factor and two-factor authentication methods. One-factor authentication uses only one authentication technique to protect the system. In contrast, two-factor authentication adds a second authentication technique (factor), such as biometrics, to protect the system if one of the two factors is compromised.

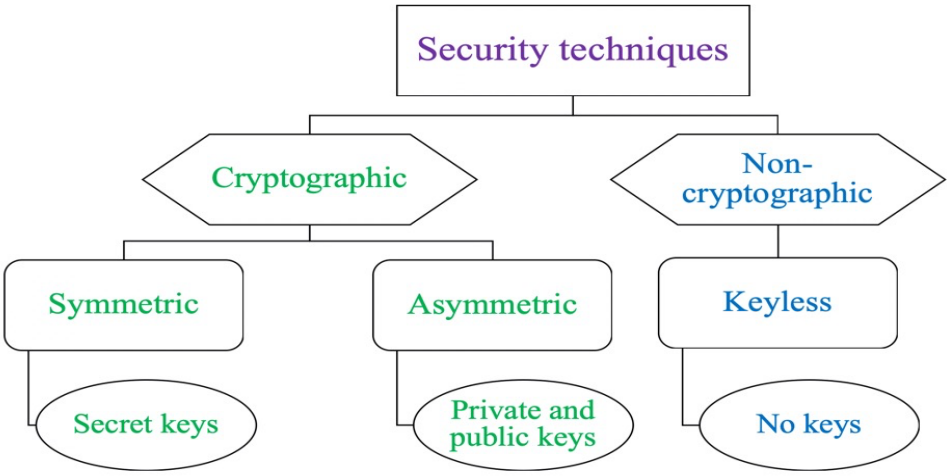


Figure 2.3 Security techniques.

2.4 Symmetric-Key Algorithms

As shown in Figure 2.4, symmetric cryptography includes any cryptographic algorithm based on a secret/shared key between two or more nodes wanting to communicate. The key is to be generated and distributed before using asymmetric cryptography or a prior communication stage.

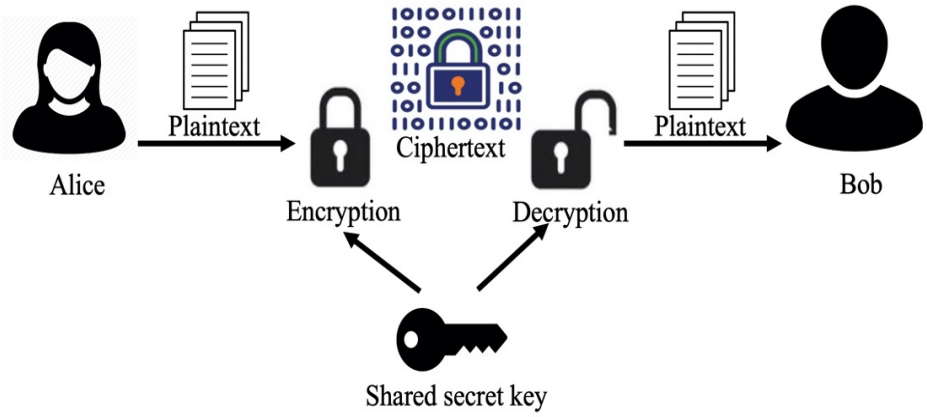


Figure 2.4 Symmetric cryptography.

In this section, we discuss integrating symmetric cryptographic algorithms in IoMT systems. These algorithms can be used for IoMT systems to allow hierarchical access to the patient's data and initiate secure connections without prior setup. Further, they can also be used in two-factor authentication, where they act as a first factor while other techniques, such as facial recognition and pattern-based, act as a second factor.

2.4.1 Hierarchical Access

This technique allows hierarchical access control to patients' data stored in the cloud layer. One approach utilizes a hierarchical role-based model and provides authorization based on the user's role [29]. For example, all authenticated nurses can administer medicines, but prescribing a new medication requires a person authenticated as a doctor. The model supports a relatively low complex hierarchical security scheme that encrypts the patients' data and only decrypts that part of the data to which the user is authorized. Belkhouja et al. [29] used the Chinese remainder theorem (CRT) to support this hierarchal access where the user with a higher privilege can access any patient's data. In contrast, the user with a lower privilege can access part of the data related to their roles [30].

2.4.2 Wireless Signal Characteristics

This technique utilizes wireless signal characteristics to secure IoMT systems by generating keys without prior connections. Radio signal strength (RSS) is one of these characteristics, and it measures the received signal power, which varies based on the medium it passes through [31]. IMDs can be excellent candidates for this technique since the RSS value variation inside the human body differs from outside the body [32]. The proposed technique uses the randomness in RSS values to generate a shared key. This key can secure communication between a headless cardiac pacemaker and a subcutaneous (under-the-skin) implant without prior knowledge of the keys. In this technique, two bits can be extracted from a single cardiac cycle (a beat) with a 128-bit key in 60 seconds if we consider the average human heart rate of 64 beats per minute (bpm).

2.4.3 CHF with XOR

CHF is a one-way mathematical function that converts arbitrary data sizes to fixed ones [33]. Exclusive-OR (XOR) can be used to check if one of its operands differs. In a medical setting, initial parameters (i.e., a sensor ID and a shared key) can be XORed together and then hashed. Then, these hashed parameters are shared from the key generation server to the sensor and gateway nodes. These nodes can generate their keys with the help of these parameters [34]. Combining the CHF, a symmetric key, and the XOR operator can secure the IoMT systems' communications using new authenticated key agreement protocols, as illustrated by Alzahrani et al. [35] and Xu et al. [35, 36]. Using the hash function, this technique also supports unique identification parameters for the system's nodes. However, the system administrator must manually add initial parameters to all the nodes in the system's initialization step.

2.4.4 Gait-Based Technique

This technique uses the human walking pattern to generate unique symmetric keys. A system proposed by Sun and Lo can generate a symmetric key using a set of IoMT sensors attached to the individual's body in just a matter of 10 gait-cycles. They claim their system can generate three times the number of bits per gait cycle than those generated by similar state-of-the-art techniques [37]. The gait cycle is one movement cycle between two repetitive events while walking. This system employs an artificial neural network (ANN) model to generate a 13 b/gait-cycle, generating a 128-bit key in just ten gait cycles. This key can be used later to secure the communications between the IoMT sensors and the AP or mobile in the gateway layer. It outperforms finger-based systems by generating binary keys at different times, which provides randomness to the keys without direct user interaction with the system.

2.4.5 Facial Recognition

This technology is one way that IoMT systems can rely on authenticating users by scanning their faces. Using shared keys as a first factor, facial recognition can be used as a second factor in continuous role-based authentication [38]. This helps secure the connection between the sensor and the medical controller in the gateway layer based on each authorized user's privilege. Since this technique continuously scans the user's face while using the system, it can secure the system in a medical setting. For example, this technique can prevent medical staff with lower privileges from accessing patients' data without a higher privilege, such as a medical staff member who has authenticated themselves but has not logged out of the system.

2.4.6 Pattern-Based Technique

This technique is similar to the facial recognition system but uses a pattern-based technique as a second factor [39]. This technique uses a tab pattern the patient generates to control the sensor. After successfully passing the first factor with the medical controller in the gateway layer, the controller sends a random tab pattern as a second factor to the user before executing a sensitive command. The technique can also keep the sensor communication turned off until a specific pattern is performed, preserving the sensor's battery power in case of IMDs.

2.5 Asymmetric-Key Algorithms

Asymmetric cryptography includes cryptographic algorithms that use two keys, a public and a private, with one of them for encryption/validation and the other used for decryption/signature. Asymmetric cryptography is also known as public-key cryptography. The public key is known to everyone, while the private key is only known to its owner. An example of how encryption and decryption can be used is shown in Figure 2.5. Some of the known algorithms in this category include Rivest–Shamir–Adleman (RSA) and elliptic-curve cryptography (ECC) [40, 41]. ECC is the most common encryption technique for securing IoMT systems due to its lightweight characteristics. An ECC key of 160 bits is as good as a 1024-bit RSA key and is 15 times faster [42].

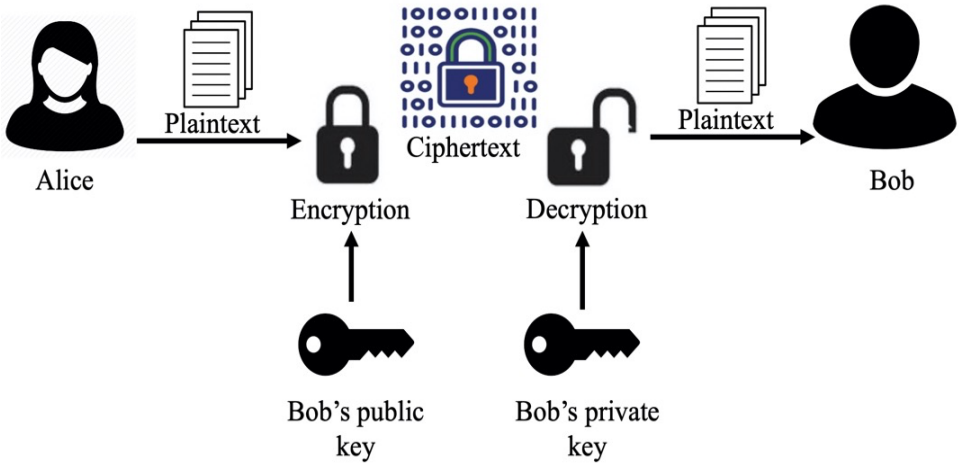


Figure 2.5 Asymmetric cryptography.

This section discusses the integration of asymmetric cryptographic algorithms in IoMT systems. This includes asymmetric keys with CHF, homographic encryption (HE), or digital signatures. Also, similar to symmetric keys, asymmetric keys can be used for two-factor authentication. They act as a first factor for authentication with other techniques, such as smart cards as a second factor. Smart cards are extensively used in hospitals nowadays.

2.5.1 CHF with ECC

CHF function and ECC keys can be used as a secure certificateless channel between patients and their medical doctors [28]. The idea of combining the ECC and the CHF is to allow a secure way for sharing keys between the key generation server in the cloud layer and the nodes in the IoMT sensor and gateway layers, respectively. The ECC public key of the KGS and initial parameters, such as a node ID, are hashed together using CHF; then, they are sent to the nodes in the IoMT sensor and gateway layers. The nodes can generate their asymmetric keys with the help of the received hashed values. As a result, this system solves the problem of sharing the secret keys as in the symmetric cryptographic techniques.

It can also overcome the overhead in cloud certificate management for data storage and sharing [43]. The IoMT data sizes are substantial, and they are increasing. We are in the zettabyte era. The zettabyte is one billion terabytes, whereas a terabyte is a typical hard disk size nowadays [44]. By dividing the patient's data into subsets and converting them using ECC keys and CHF, they can be securely shared among the system's entities. The average energy consumption in this technique is around 30% less than similar techniques.

2.5.2 Homomorphic Encryption (HE)

HE is an encryption technique that preserves data confidentiality and allows limited mathematical operations on encrypted data [45]. This technique protects the patient's data privacy and stores them as ciphertext in the cloud layer for mathematical operations, such as data integrity. However, this technique differs from other techniques since it allows only the patient to see their data, not the medical staff, except during emergencies. In other words, this is useful for some IoMT sensors, such as a smartwatch, which allows the data to be encrypted at all times and only seen by the patient except in emergencies where the patient's data can be sent to the medical staff to make correct diagnostics.

There are three different schemes for HE: partial HE (PHE), somewhat HE (SHE), and fully HE (FHE). PHE supports one mathematical operation an unlimited number of times, while SHE supports only a limited number of operations. FHE supports an unlimited number of operations; therefore, it can be suitable for fast data aggregation without compromising data confidentiality [42]. Hence, it is ideal for healthcare monitoring systems in hospitals. Jariwala and Jinwala claim that their AdaptableSDA HE framework consumes only 10% more power with the privacy requirement than without it [42].

Optimal HE (OHE) is a modification of FHE. It differs from FHE in that it is based on the step-size firefly optimization (SFFO) algorithm, in which the key with the maximum breaking time is selected [46]. This technique reduces the computation time and increases the breaking time by 2% to 8% compared to other HE and non-HE techniques.

2.5.3 Digital Signatures

Digital signature techniques can be used even in a small IoMT system. Generally, they can verify the data/command authenticity using the sender's (Alice) private and public keys for signature and verification, respectively [47]. In IoMT systems, digital signatures can be integrated into the sensor's firmware with an add-on software shim, intercepting and validating the sensor's wireless communications [48]. These techniques require storing a list of authorized users' (i.e., medical staff's) public keys in the sensor's firmware to validate them.

2.5.4 Smart Cards

This technique differs from the first three since it relies on physical keys [49]. These keys are utilized as a second factor, with the ECC keys as the first factor for authentication. In IoMT settings, the medical staff must enter a key and use their smart cards to access the system. This technique helps the system be resistant to cyber-breaks if one of the factors is stolen or lost. This has made them quite common nowadays.

2.6 Keyless Algorithms

This section discusses keyless techniques that provide security without pre-shared keys. The techniques in this category can be based on biometrics, token-based security, or proxy-based techniques. Cutting-edge technologies such as blockchain technology and AI also fall in this category since they can be used for security without pre-shared keys.

2.6.1 Biometrics

Since they are easy to use, the biometric sensors used to identify users' physical characteristics are the most common technique to provide security for IoMT systems. In a medical setting, medical staff or patients can access the medical records only using their biometrics. Biometric factors include fingerprint and ECG-based sensors that are handy in emergencies. The fingerprint sensors are based on reading the fingerprint image, while the ECG-based sensors record the heartbeat activities to encrypt the data. Fingerprint sensors reduce the messages' size during transmission and the computational overhead compared to the ECG-based techniques [50].

The performance of the fingerprint sensors is based on the extraction algorithm used. Popular algorithms used in these sensors are Delaunay Triangulation-based feature representation, Pair-polar coordinate-based feature representation, and Minutia Cylinder-Code-based feature representation [51]. According to Zheng et al., Delaunay performs better and is less complicated than the other techniques. The advantages of using fingerprint biometrics include their long history and credibility than face recognition-based systems.

2.6.2 Token-Based Security

User authentication can be done using software or hardware tokens. For instance, the x-auth-token field in the hypertext transfer protocol (HTTP) header can be used as a software token embedded in the user web browsers [52]. Cloud data analytics companies use these tokens, e.g., IoT Ubidots [53], to secure the connection between the cloud layer and the nodes in the IoMT sensor and gateway layers. Likewise, RFID can be used as a hardware token for secure logistic management of sensors in a hospital information system (HIS) [54].

2.6.3 Proxy-Based and Light-Based Systems

Proxy-based systems are made of a middleware device that controls the communication between the sensors and any device communicating with them, such as medical controllers. Besides, they can provide full-duplex secure communications between these devices, where they can simultaneously communicate. These middleware devices can be a set of microprocessors inside a jacket or a belt to be worn by the patient [55, 56].

Light-based communication technologies, such as light-fidelity (Li-Fi), can be used to secure the monitoring capabilities for HIS, as presented by Mosaif and Rakrak [57]. Since Li-Fi does not use wireless communications, it has no interference with the hospital network, substantial free operation frequency, and short coverage range for enhanced security.

2.6.4 Blockchain Technology and AI

These are new techniques for use in IoMT systems due to their success in providing security in other fields, such as finance [58-60]. Blockchain technology is typically used in IoMT systems as a security management sharing technique for data sharing between patients and other parties, such as doctors and insurance companies. On the other hand, AI systems can detect anomaly behaviors (leading to attacks) in network flows and patients' data. However, there are some challenges to IoMT systems adopting these techniques. For example, blockchain technology may suffer from latency, storage issues, and communications overhead, given the data sizes and communication requirements in IoMT systems. High latency is typical for public blockchain technology due to its decentralized nature. Therefore, private blockchains may be considered for real-time systems. AI systems require a large amount of data; hence, they may not be ideal for detecting rare attacks.

Blockchain technology and AI are being adopted in IoMT systems, mainly in the cloud layer [6, 61].

2.7 IoMT Systems Risks and List of Attacks

In this section, we explore the attack surface of IoMT systems. We discuss possible attacks targeting such systems, including physical and network attacks. Table 2.1 summarizes the security requirements for IoMT systems, possible attacks, and countermeasures [16]. As shown in the table, the countermeasures for 11 of the 14 attacks are based on keyless methods, and more than half are based on two-factor authentication methods. The popularity of these methods is due to their simplicity during system implementation and management.

Table 2.1 List of attacks and countermeasures.

No.	Attack	Effects	Countermeasure	Reference
1	○ Physical security token loss	– Authentication – Authorization	– Asymmetric (two-factor)	[49]
2	○ Impersonation	– Anonymity – Forward secrecy	– Asymmetric – Keyless	[28, 35, 50, 51]
3	○ Tampering	– Data confidentiality – Data Integrity	– Symmetric (two-factor) – Keyless	[35, 38, 43, 50]
4	○ Side channel		– Keyless	[39, 58]
5	○ RF jamming	– Availability		[59]
6	○ DoS/DDoS			
7	○ Sniffing	– Data confidentiality	– Symmetric /asymmetric (two-factor) – Keyless	[58, 62]
8	○ MITM	– Data confidentiality – Authorization		[28, 29]
9	○ Relay	– Authorization		[29, 59]
10	○ Replay			[36, 38, 49, 58]

11	○ Clock synchronization	– Secure key exchange	– Asymmetric (two-factor)	[49]
12	○ Parallel session	– Authentication – Authorization		
13	○ Brute force		– Keyless	[37]
14	○ Stepping stone			-

2.7.1 Physical Attacks

These attacks target the physical components (e.g., sensors, physical keys) of the IoMT systems to extract patient data or security keys. They require some component of the IoMT systems to be physically accessible to the attacker. These attacks can be summarized as follows:

1) Physical Security Token Loss

This includes any attack where the attacker steals an authorized user's physical security token, such as a smart card, to access the system. The violated security requirements here are authentication, authorization, anonymity, and forward secrecy. Kumari *et al.* showed that integrating asymmetric keys, such as ECC, with smart cards can mitigate such attacks since stealing the smart card is insufficient to hijack the system [49].

2) Impersonation/Presentation

In this attack, the attacker impersonates an authorized user's identity, e.g., by replicating the fingerprint or face print. This can target any node in the IoMT system. The attack violates authentication, authorization, anonymity, and forward secrecy security requirements. It can be avoided using symmetric/asymmetric techniques, such as CHF, or keyless techniques, such as biometrics [28, 35, 50, 51].

3) Tampering

Any modification to the IoMT systems' data at the collection, transit, or storage stage is considered a tampering attack. This may include attaching external devices to alter the data and attack sensors during emergencies. It violates data confidentiality and integrity and can be mitigated by combining symmetric keys with facial recognition or using keyless methods [35, 38, 43, 50].

4) Side Channel

These attacks occur during the communications among devices in the IoMT system. They are based on leaked information about the cryptographic operation in the communications. These attacks violate data confidentiality and privacy requirements and can be alleviated using keyless cryptography. Maji *et al.* suggest using the datagram transport layer security (DTLS) protocol to avoid them. Blockchain technology and AI can act as other detection and mitigation strategies, as shown by Saif *et al.* [39, 58].

5) Radio Frequency Jamming/Desynchronization

Radio Frequency Jamming attacks target the system's availability, which is dangerous for critical systems such as IoMT. Also, they can cause battery depletion, knowing that IoMT sensors are battery-power-constrained. Blockchain technology and AI can reduce the effects of such intrusions by finding alternative routes or terminating the channel connection with the attacker [59].

2.7.2 Network Attacks

Other attacks may target communication between different layers of the IoMT system, such as Bluetooth or Internet links, as presented in Figure 2.2. These attacks usually aim to steal or fabricate patients' data or block the connections between the IoMT systems' layers.

6) DoS/ DDoS

These attacks load the system's communication links with undesirable connections, making regular connections unavailable. They may also cause network fragmentation. Thus, a fragmentation attack is a particular type of DDoS [63]. These attacks usually target the cloud layer in the IoMT systems to prevent the system from being available to the users (i.e., patients and medical staff); hence, it violates the availability requirement. Blockchain technology and AI can reduce the effects of such intrusions by finding alternative routes or terminating the channel connection with the attacker [59], similar to those mentioned in the RF jamming attacks.

7) Sniffing

A sniffing attack passively intercepts the data transmitted between two nodes, resulting in patient data confidentiality violation. In a medical setting, an attacker can see the data transmitted between the layers in the IoMT system architecture, which violates the data confidentiality security requirement. Any encryption algorithm, i.e., symmetric, asymmetric, or keyless, can mitigate these attacks [58, 62].

8) Man-in-the-Middle

MitM attack is a type of eavesdropping attack. After a successful sniffing attack, the attacker can alter the intercepted data before sending them to the original destination. For example, the attacker can change the patient's biometric data transmitted from any two layers in the IoMT system (i.e., from the sensor layer to the gateway layer). This can be done using unmanned aerial vehicles (UAV), resulting in a drone-in-the-middle (DitM) attack, as discussed by Sethuraman *et al.* [62]. To make this attack more powerful, the UAV can be connected to a cloud to perform more intensive computation quickly. This attack violates authorization in addition to data confidentiality requirements and can be mitigated using encryption or two-factor authentication techniques [28, 29].

9) Relay

After a successful sniffing attack, the attacker can relay the intercepted data to a third node without altering them, for instance, sending the patient's data after intercepting them (i.e., from the sensor layer) to the attacker's computer before sending them to the intended layer (i.e., gateway layer). This attack breaches the authorization requirement and can be mitigated using asymmetric keys, such as hierarchal access, supporting secure session keys [29, 59].

10) Replay

After a successful sniffing attack, the attacker can resend the intercepted data later to the original destination without altering them. By repeating this process, this attack may also result in a DoS/DDoS attack. This attack violates the authorization requirement, similar to the replay attack. It can be mitigated using a timestamp, part of some symmetric, asymmetric, and keyless techniques [36, 38, 49, 58].

11) Clock Synchronization

This type of attack targets the clock synchronization protocol, which is necessary for real-time systems, such as IoMT systems. The attack violates the secure key exchange requirements. The attacker successfully initiating this attack can make relay, replay, and MitM attacks not easily detectable. This attack can be mitigated using two-factor techniques like ECC with smart cards [49].

12) Parallel Session

These attacks break one-way authentication protocols that use asymmetric keys. The effects of such attacks are authentication and authorization violations, which can be avoided using two-factor techniques, such as ECC with smart cards [49].

13) Brute Force

The attacker in this type of attack tries many credentials until it is successful. One way is the dictionary attack, which relies on known passwords or words in dictionaries. These attacks can also be performed in the off-line phase after capturing the encrypted data decrypted with powerful machines. A dictionary attack is one of the significant problems for IoT devices since their short, simple, or factory-set default passwords can be guessed using a simple python script, making them easier to find online [64]; therefore, IoMT systems can be affected. These attacks have violated authentication and authorization security requirements as the parallel session attacks but can be alleviated using keyless methods, such as biometrics [37].

14) Stepping Stone

Instead of relying on one computer/host to attack the IoMT system, a chain of hosts can be used to attack the system. Sethuraman *et al.* perform this attack using a series of UAVs to extend the communication link between the UAVs and the attacker's computer. Hence, The attacker can launch an attack in restricted areas (i.e., in a hospital) that are not directly accessible by the attacker [62]. This attack violates authentication and authorization security requirements but can be avoided using keyless methods like AI.

2.8 Proposed Security Framework for IoMT

As the previous section shows, no single technique can provide a secure environment for IoMT systems. Hence, we propose a framework that protects IoMT systems from the 14 attacks mentioned in the previous section. The framework also fulfills all the security requirements required by IoMT systems. Three parts of the framework are based on the IoMT security model stages mentioned in Section 3.1, as shown in Figure 2.6.

2.8.1 Securing Data Collection

The first step in securing IoMT systems is to secure how other systems interact, protecting the patient's data collection stage. Two-way factor authentication techniques are good options to provide such security and resistance to some attacks mentioned. If one of the two factors is compromised, the other can still provide essential overall security. ECC keys are commonly used techniques for first-factor authentication due to their lightweight keys and reliable protection [28, 43, 49].

Adopting the hierarchical access technique with ECC is a perfect way to secure data sharing with other medical staff based on their role, which has been used for other fields like smart homes [65]. This technique requires KGS, located in the cloud layer, as shown in Figure 2.7. Biometric sensors are considered the most common way nowadays as a second factor due to their convenience for everyday use and emergencies [50]. These sensors authenticate the patient to access the sensor layer nodes, as shown in Figure 2.7. As explained in Section 6.3, proxy-based techniques can be used to provide security to existing unsecured sensors at the sensor layer [55, 56].

ECC and Biometrics can protect the system in case of a software attack during the data collection stage. However, in the case of a hardware attack, the system needs another technique to alert the patient and the medical staff to reduce or eliminate the effects of such an attack. The AP or a similar device in the gateway layer should alert the user and the physician if they cannot connect to the IoMT sensor for a specific period (i.e., one hour).

Edge computing (EC) has recently gained attention in IoMT systems since it reduces latency and provides powerful resources for these systems' sensors [66, 67]. EC, which is usually located in the gateway layer, as shown in Figure 2.7, can act as the gateway to the IoMT sensors or as a main

gateway for a set of secondary gateways. It can also be used here to utilize an AI model, which will be detailed in Subsection 2.8.3. This model can be used to track the changes in sensor readings as an initial analysis to fulfill the patient's data confidentiality and integrity security requirements. If either requirement is violated, the EC can warn the patient early about this violation. The system can immediately alert the physician if the patient does not respond.

These techniques can provide the system with confidentiality, integrity, authentication, authorization, anonymity, forward/backward secrecy, key-escrow resilience, and session-key agreement. The system can be resilient to attacks by guaranteeing these requirements, including physical-security token, impersonation, tampering, side channel, sniffing, MITM, relay, replay, clock synchronization, parallel session, and brute force.

However, the techniques in this subsection assume pre-shared keys or initial parameters, which may lead to the following challenges:

- An initial manual setup is required to prepare the KGS for the hierarchical access technique.
- Unusable if the second factor is lost or inaccessible, especially during emergencies.

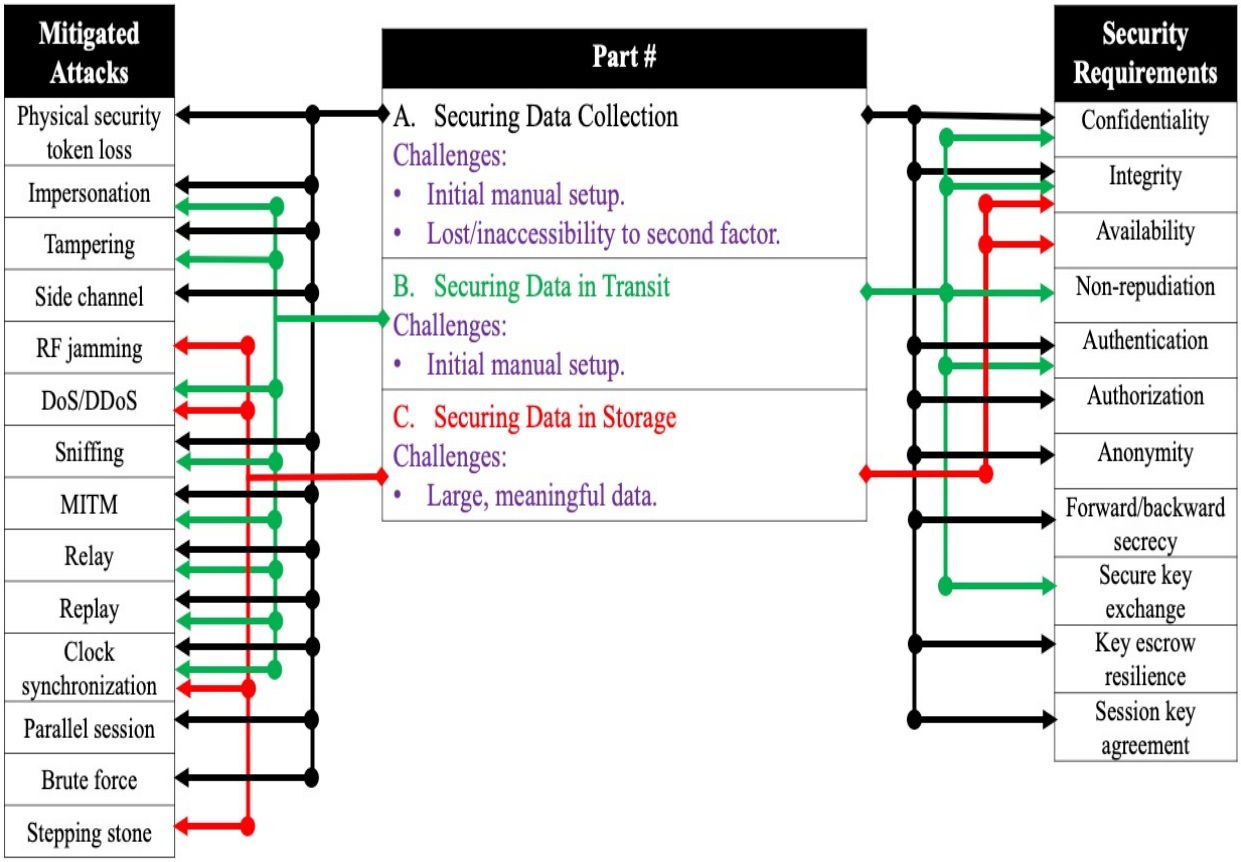


Figure 2.6 Proposed framework security features.

2.8.2 Securing Data in Transit

To enhance the IoMT systems' security when connected to other devices over the network, we advise utilizing some security protocols, such as constrained application protocol (CoAP) [68]. CoAP is an application protocol designed explicitly for resource-constrained IoT applications, such as IoMT systems, for communications between the sensor and gateway layers, as shown in Figure 2.7. The rest of the layers can be linked using secure HTTP (HTTPs) or transport layer security (TLS) version 1.3 [69]. Thus, it is convenient for use in IoMT systems. To reduce the certificate management overhead in the cloud layer, certificateless cryptography, ID-based cryptography (IBC) branch, can be used, as shown in Figure 2.7 [70, 71]. The key generation process in certificateless cryptography uses the KGS public key with some initial parameters to help the IoMT systems' nodes generate their keys. Then, certificate-less authenticated encryption (CLAE),

which does not require central key management, is used for authentication [72]. CoAP protocol and IBC help protect the system against impersonation, tampering, sniffing, MITM, relay, replay, and clock-synchronization attacks. The protection from these attacks provides the system with confidentiality, integrity, non-repudiation, authentication, and secure key exchange. However, the system requires an initial manual setup similar to that described in Subsection 8.1.

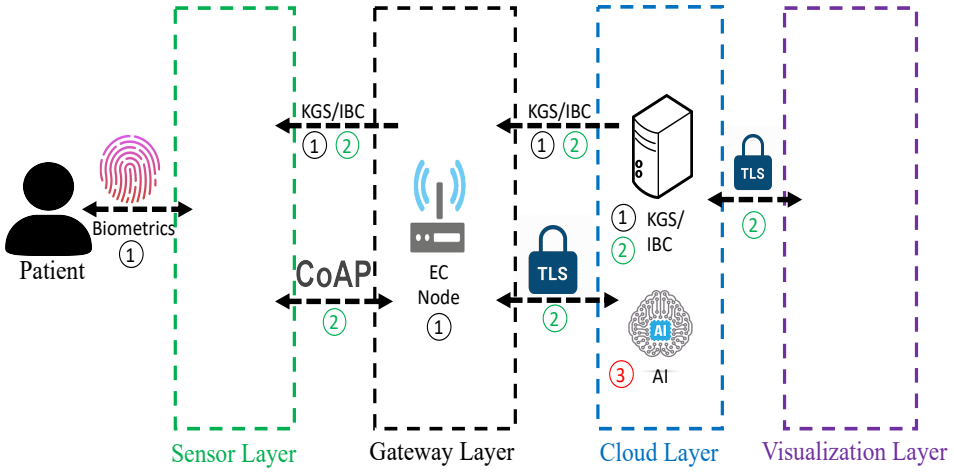


Figure 2.7 Proposed IoMT secure system architecture.

2.8.3 Securing Data in Storage

Some of the attacks in IoMT systems target the availability and integrity of the system, such as DoS/DDoS, RF jamming, and stepping-stone attacks. These attacks can be detected using AI techniques. AI techniques can be used to build detection models with mitigation techniques imposed on these models. For example, deep neural networks (DNN) can be used to build intrusion detection models. Once this model detects suspicious activity, the compromised connection is terminated to mitigate the attack. Adopting these intrusion detection models in the cloud layer, as shown in Figure 2.7, can warn the system administrator when such attacks occur, which can verify early warnings (if they exist) from the EC nodes in the gateway layer. Collecting enough and meaningful data is very critical for AI techniques. This is a challenging step in reducing the error rate with these techniques. The cloud can detect any compromise by keeping logs of the presence of the

connected gateways or ECs. It can also find alternative routes to IoMT sensors by providing a backup gateway in case of attacks, breaks, or loss of the original gateway.

2.9 Summary

Due to the demand for using IoMT sensors to reduce healthcare spending and provide better care for patients, securing these devices has become extremely important. However, IoMT sensors tend to have constrained resources, and some already implanted require external devices to secure them. This chapter discussed an overview of the security requirements, state-of-the-art security techniques, and new types of attacks. Since no technique could satisfy these systems' security requirements and mitigate most attacks, we proposed a framework that combines these techniques to meet all security requirements. This framework covered all data and device security stages, from data collection to storage and sharing.

Chapter 3: IDS for Healthcare Systems

Introducing IoT systems to healthcare applications has made it possible to remotely monitor patients' information and provide proper diagnostics whenever needed. However, providing high-security features that guarantee the correctness and confidentiality of patients' data is a significant challenge. Any alteration to the data could affect the patients' treatment, leading to human casualties in emergency conditions. Due to the high dimensionality and prominent dynamicity of the data involved in such systems, machine learning promises to provide an effective solution for intrusion detection. However, most healthcare IDS use network flow metrics or patients' biometric data to build their datasets. This chapter aims to show that combining network and biometric metrics as features perform better than using only one of the two features. We have built a real-time EHMS testbed that monitors the patients' biometrics and collects network flow metrics [73]. The monitored data is sent to a remote server for further diagnostic and treatment decisions. Man-in-the-middle cyber-attacks have been used, and a dataset of more than 16 thousand records of normal and attack healthcare data has been created. The system then applies different machine learning methods for training and testing the dataset against these attacks. Results prove that the performance has improved by 7% to 25% in some cases, showing the robustness of the proposed system in providing proper intrusion detection.

[6] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study," *IEEE Access,* vol. 8, pp. 106576-106584, 2020.

3.1 Introduction and Motivation

The potential of IoT-enabled healthcare systems for early diagnosis, real-time monitoring, and improved patient outcomes is undeniable. Continuous monitoring of vital signs through wearable

sensors is essential to achieve these benefits, enabling remote healthcare communication between providers and patients. However, the security of these systems is paramount, as breaches can lead to severe privacy violations, incorrect diagnoses, or delayed treatment with potentially fatal consequences [74].

Traditional security models often struggle with cyberattacks' evolving, complex nature, especially in the healthcare context. MitM attacks, where packet alteration occurs in-transit, are just one example [75]. ML offers a promising solution. While not a panacea, ML excels in analyzing complex data for pattern detection and classification, making it well-suited for identifying potential security threats [76].

To investigate this approach, we have built an EHMS testbed that utilizes ML for managing security issues. This testbed includes a gateway for data gathering, an IDS computer for monitoring network traffic and detecting abnormal behaviors, an attacker component to imitate real-world threats, and a server to store and provide access to healthcare data. ML models detect data alteration and spoofing threats by analyzing patients' biometric data and network traffic characteristics. The system reports a threat alert to system managers if traffic metric or biometric data is detected as abnormal. For attack detection, we have chosen four ML methods: random forest (RF), k-nearest neighbor (KNN), support vector machine (SVM), and ANN [77-80]. Our research builds upon the understanding that ML methods require representative and reliable training data [81].

The rest of the chapter is organized as follows. The related work is presented in Section 3.2. Section 3.3 discusses the proposed framework architecture. Section 3.4 describes the results gathered from the experiments on the system. Finally, Section 3.5 summarizes the chapter and provides future work.

3.2 Related Work

In recent years, numerous approaches have been proposed for building health monitoring systems, and the following are some examples. Fotouhi et al. propose a general framework for a healthcare monitoring system [74]. The system consists of three components: a coordinator, access points, and a gateway. The coordinator is a node that lies on the human body to gather information from the sensors. The access points (APs) are static nodes attached to the walls in the room that use the same communication protocol as the one used by the sensors (i.e., ZigBee, 6LoWPAN, or BLE). These APs forward the data to a gateway, which forwards the data to the cloud through the Internet. In this system, some general approaches have been proposed for securing data without concrete description and testing. Also, the authors have not proposed a solution for discovering successful attack scenarios.

ML has been used in healthcare for many purposes, such as managing and controlling false alerts while reporting severe health threats, as Clifton et al. explain, where a wearable health monitoring system has been described [82]. In their approach, the generated data is collaborated with the clinical observations of a specific patient to provide early alerts of any expected emergencies. The experimental work has been tested at Oxford University Hospital. This approach has not tackled security problems in such a system.

In [83], a cloud-based healthcare system has been proposed by Rani et al., where data is accessed only by authorized users. The system uses the SVM method to predict patients' conditions and expected diseases. This system uses an ML approach for data mining and not to attack discoveries in data like our system.

Chakraborty et al. [84] propose a healthcare system design framework using blockchain technology. Blockchain technology is known to assure security, but the authors have not investigated the framework or tested it to present any benchmark results.

Alabdulatif et al. implement a system that provides a privacy-preserving cloud-based real-time change detection and abnormality prediction framework for multiple patient vital signs [85]. The system comprises three main blocks. The first is the Smart Community Resident, where data is collected and aggregated to be sent to Cloud Storage, stored in an encrypted format. The last and main block is the Smart Prediction model, which works mathematical models on the data without decryption to detect abnormal changes and thus expect attacks. This approach focuses on conventional methods for securing data but does not consider new methods, such as ML, for predicting security violations.

A hardware approach is proposed by Tao et al. in [86], where KATAN Hardware approaches for the security of IoT-based healthcare monitoring systems have been introduced. A secret cipher algorithm is implemented and optimized on the Field Programmable Gate Array hardware platform for data collection with security. This approach has the complications of hardware approaches and problems in [13].

Zhang et al. propose a security framework that detects anomaly traffic using the RF method on the KDD 1999 dataset [87]. The accuracy of the RF method as an anomaly detector is 95%, with a 1% false-positive rate. Note that the knowledge discovery and data mining (KDD) dataset is a generic dataset used in competitions since 1999 [88]. It is not specific to healthcare and is very old. Although one of the methods in our system uses the same ML method, we have implemented a testbed

to collect a dataset that closely resembles real healthcare monitoring system applications. Furthermore, our proposed system uses network flow metrics and biometrics as anomaly detection features.

The authors of [89, 90] use the KNN method as a basis for their cybersecurity methods. In [89], Rao et al. use indexed partial distance search KNN (IKPDS) to test different types of attacks, resulting in an accuracy of 99.6%. Shapoorifard and Shamsinejad [90] focus on reducing the false alarm rate and show an accuracy of 85.2%. These two approaches use an enhanced version of the KDD dataset but still suffer from the same problems and differences we mentioned earlier with the original KDD dataset.

3.3 EHMS Testbed

As shown in Figure 3.1, our testbed has been built using a health monitoring sensor board that collects data from several healthcare sensors placed on the patient's body. The board is attached to a Windows-based computer using a USB port. C++-based software has been developed to capture the sensed data. The computer is the gateway to transfer data to a server through Wi-Fi using transmission control protocol (TCP). All the machines are connected to a switch using Ethernet cables except the gateway computer. The switch is connected to the Internet through a router, and the gateway is connected via Wi-Fi. Securing transferred data in the testbed mainly relies on using ML to help the healthcare monitoring system detect any tampering in the transmitted data between the nodes in the network in real-time. If detected, the system reports a threat alert to the system managers. In addition to these flow packets, the sensed data from the sensors attached to the pa-

tient's body are collected to help train the model. We have assumed that the data is being transmitted in plain text since the other methods, like TLS certificates, require more processing power, which is generally not feasible with low-cost sensors.

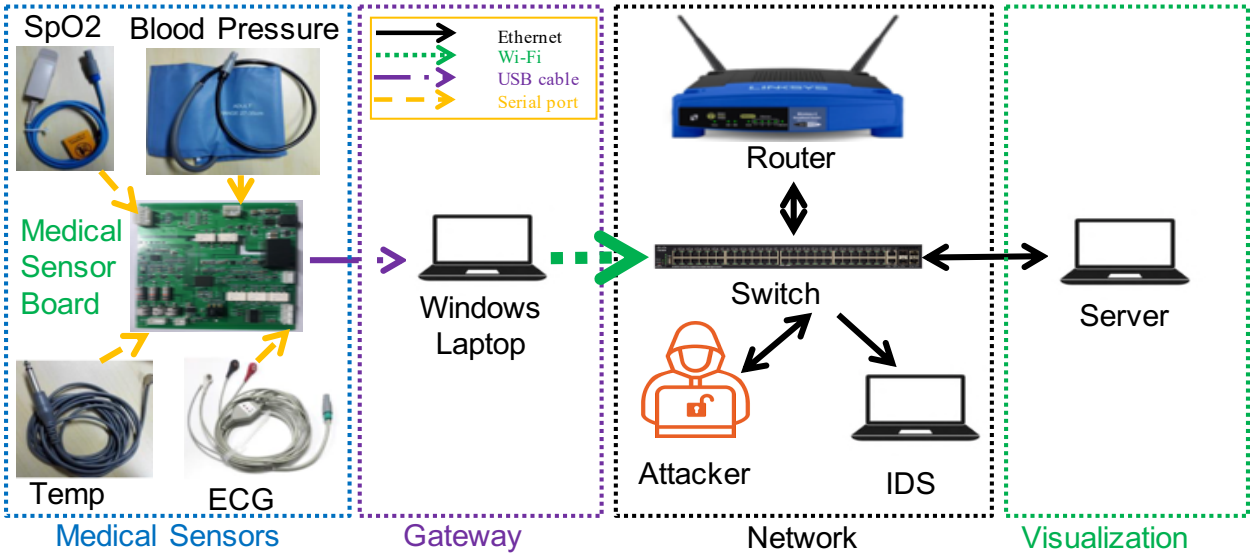


Figure 3.1 EHMS testbed.

Our EHMS testbed system works as shown in Figure 3.2, and data flows across the system from sensors attached to the patient's body through the sensor board to the gateway to the switch and, finally, to the display screen of the server. On the journey of the data from the switch to the server, an attacker may intrude to spoof or alter data before it arrives at the server. Meanwhile, network and patient data metrics are captured on the IDS computer. Data is processed at the IDS for training and testing the machine learning methods as well as real-time detection of any abnormalities.

Our system uses Argus to collect all network traffic flows and patient data between the gateway and the server. Argus is open-source software that monitors the real-time network flow traffic [91].

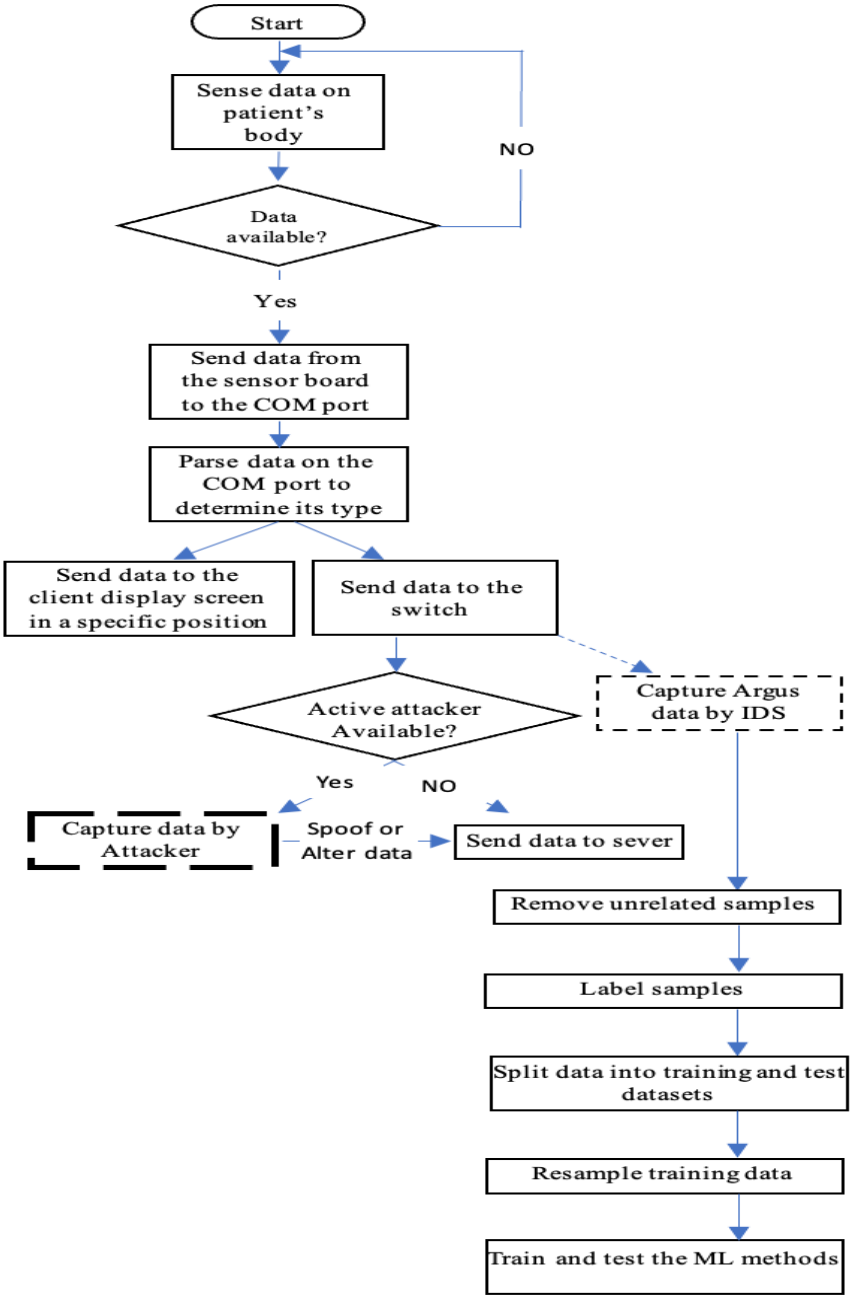


Figure 3.2 EHMS flowchart.

3.3.1 Model Architecture

The system consists of six building blocks: a multi-sensor board, a gateway, a server, an IDS, an attacker, and a network. The functionality of each block is summarized below:

1) **PM4100 six pe multi-sensor board:**

A product of Medical Expo that is used for sensing the patient's biometric data using a set of sensors attached to the patient's body [92]. The board has four sensors, as shown in Figure 3.3:

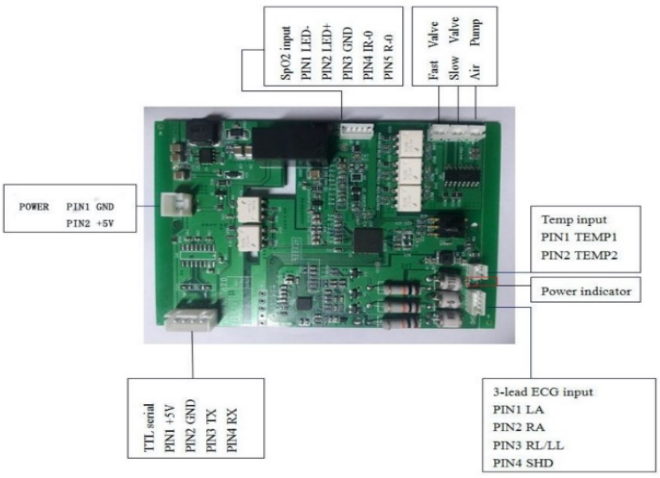


Figure 3.3 PM4100 six pe multi-sensor board.

a. **The electrocardiogram (ECG or EKG) sensor** consists of three-electrode pads attached to the patient's body to measure the patient's heart electricity.
b. **The blood oxygen saturation (SpO2) sensor** measures the oxygen level in the patient's blood and the heart rate. A value of 95-100 percent is considered normal. While a level below 90 percent results in hypoxemia, levels below 80 percent may compromise brain and heart functions and may lead to respiratory or cardiac arrest.
c. **The temperature sensor** is used to measure the patient's body temperature.
d. **The blood pressure sensor** is a step-wise gassing method that measures the patient's systolic and diastolic arterial pressure.

2) **The Gateway:**

A Windows-based laptop to which the multi-sensor board is connected via a USB port. The data received from the board is presented on the graphical user interface (GUI) to monitor the

patient's biometric data. The gateway sends this real-time data to the server for processing. All this process is done via a C++ program. This gateway is connected to the switch with an Ethernet cable. The GUI, as shown in Figure 3.4, shows the following:

a. ***HR:*** Heart rate in beats per minute (BPM)
b. ***RR:*** Respiration rate in BPM
c. ***ST:*** Electrically neutral area between ventricular depolarization (QRS complex) and repolarization (T wave) in millivolts (mv).
d. ***SYS:*** Systolic blood pressure.
e. ***DIA:*** Diastolic blood pressure.
f. ***SPO2:*** Blood oxygen.
g. ***PR:*** Pulse rate in BPM.
h. ***TEMP:*** Temperature in celsius degrees.

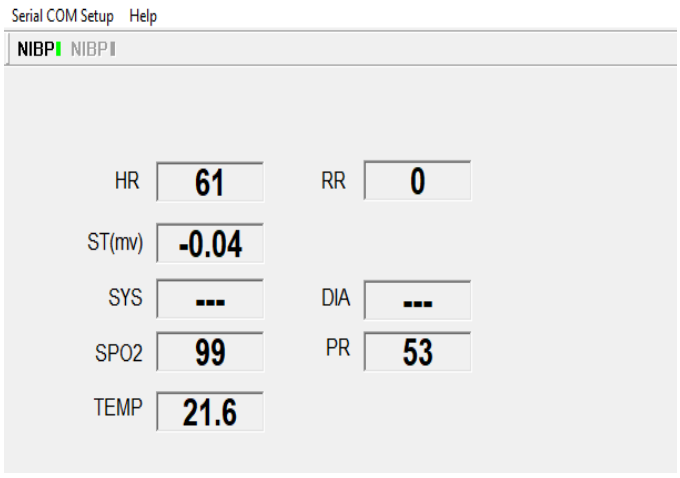


Figure 3.4 Gateway's GUI.

3) Server:

An Ubuntu-based laptop to which the data is transmitted from the gateway for further saving and analysis to make suitable medical decisions. The data is collected using a C++ program.

4) Network:

A regular Ethernet switch connects the server, IDS, and attacker computer in one network. A router has been connected to this switch to dynamically assign IP addresses for all computers. The gateway is attached to this router via Wi-Fi.

5) IDS:

The switch makes a copy of (i.e., mirrors) all packets going to the server and sends it to the IDS computer. This computer runs Argus network flow monitoring software, collecting network flow metrics and the patient's biometric data. This computer also makes an online decision for any new traffic packet using the four methods.

6) Attacker:

A Kali-Linux-based computer is used to initiate attacks on the system and mimic a dangerous scenario in healthcare monitoring systems. These attacks include spoofing and altering a patient's biometric data during its transmission over the network. A Python script with a Scapy library has initiated these attacks [93]. This library features sniffing of live connections, spoofing packets, and packet alteration on the fly. It supports active and passive protocol dissection and includes network and host insecurity analysis features.

3.3.2 Types of Attacks

The system uses a MitM attack where the attacker pretends to be a router and gets the packets first. It spoofs/alters the packets and redirects them to the server, as shown in Figure 3.5 and discussed below:

7) Spoofing attacks:

In this attack, the attacker gets a copy of each packet in the network. This attack violates the confidentiality and privacy legally required in healthcare systems.

8) Data alteration:

In this attack, the attacker alters some parts of the data and redirects it to the attacker's computer from the gateway computer. The alterations may be random or according to a rule. It then redirects the packet back to the server. This attack may cause severe harm to the patients as they may get the wrong treatment based on the false diagnostics resulting from the modifications made by the attacker.

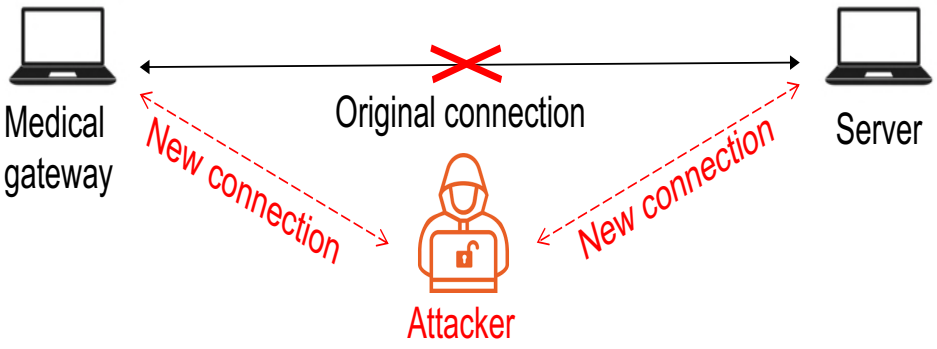


Figure 3.5 MitM attack.

3.3.3 Dataset Collection

The data features used for training and testing are presented in Table 3.1. Sixteen thousand data samples were collected and labeled as 0 for normal (non-attack) traffic, and 1 for the attack traffic. The source media access control (MAC) address is used to label the data, where the samples with the attacker computer MAC addresses are labeled as one while the rest are labeled as zero. In addition, unrelated samples to the gateway, attacker, and server MAC addresses are removed.

Table 3.1 Machine learning features.

Metric	Description	Type
SrcBytes	Source bytes	Flow metric
DstBytes	Destination bytes	Flow metric
SrcLoad	Source load	Flow metric
DstLoad	Destination load	Flow metric
SrcGap	Source missing bytes	Flow metric
DstGap	Destination missing bytes	Flow metric
SIntPkt	Source inter-packet	Flow metric
DIntPkt	Destination inter-packet	Flow metric
SIntPktAct	Source active inter-packet	Flow metric
DIntPktAct	Destination active inter-packet	Flow metric
SrcJitter	Source jitter	Flow metric
DstJitter	Destination jitter	Flow metric
sMaxPktSz	Source Maximum Transmitted Packet size	Flow metric
dMaxPktSz	Destination Maximum Transmitted Packet size	Flow metric
sMinPktSz	Source Minimum Transmitted Packet size	Flow metric
dMinPktSz	Destination Minimum Transmitted Packet size	Flow metric
Dur	Duration	Flow metric
Trans	Aggregated packets count	Flow metric
TotPkts	Total packets count	Flow metric
TotBytes	Total packets bytes	Flow metric
Loss	Retransmitted or dropped packets	Flow metric

pLoss	Percentage of retransmitted or dropped packets	Flow metric
pSrcLoss	Percentage of source retransmitted or dropped packets	Flow metric
pDstLoss	Percentage of destination retransmitted or dropped packets	Flow metric
Rate	Number of packets per second	Flow metric
Load	Load	Flow metric
Temp	Temperature	Biometric
SpO2	Peripheral oxygen saturation	Biometric
Pulse_Rate	Pulse rate	Biometric
SYS	Systolic blood pressure	Biometric
DIA	Diastolic blood pressure	Biometric
Heart_Rate	Heart rate	Biometric
Resp_Rate	Respiration rate	Biometric
ST	ECG ST segment	Biometric

3.3.4 ML Models

We used four ML methods for training and testing the system against attacks. RF, KNN, SVM, and ANN are used to build the attack detection models. The following will highlight these methods to give the reader a brief overview of their concepts, but extensive details can be found in [77-80]:

1) RF:

This model is based on a set of decision trees from a random subset of the dataset. It then collects all the votes from these decision trees to determine the suitable class for the test objects. In this method, the maximum number of features for the best split in the trees can be assigned. We set the maximum number of features at 18 for the network-only and combined set of features since it achieves the highest performance for both. Since only eight biometric features are involved in the bio-related features, we set the maximum number of features to three features.

2) KNN:

This non-parametric model classifies the test object by a plurality vote of its neighbors, with the object assigned to the class most common among its k-nearest neighbors. The hyperparameters used for all types of features (Net-only, Bio-only, combined) are as follows:

- The number of neighbors equals two, the best out of a range of 1 to 100.
- The power parameter equals four, the best out of a range from 1 to 100.

3) SVM:

The SVM method used in this chapter is linear SVM, which is a parametric method. It classifies the test object by separating the objects using a hyperplane.

4) ANN:

This multi-layer network is fully connected, a brain-like system used to find patterns in data with input, hidden, and output layers. We have set the layers as follows: 40, 40, 20, 10, 10, 10, 10, 1, where 40 is the dimension of the input layer, 1 is the dimension of the output layer, and the rest are for hidden layers. The initial settings of this setup have been taken from [94].

Our dataset comprises 14k normal samples and 2k attack samples, making up 16k samples. We used 80% of these for training and the rest for testing.

3.4 Results

This section presents our analysis and results using the abovementioned dataset and ML methods. First, we discuss the dataset preprocessing stage, including the cleaning and resampling techniques. Then, we evaluate the ML methods using the Accuracy and Area-under the receiver operating characteristic (ROC) curve (AUC) metrics.

3.4.1 Data Preprocessing

In any ML application, preprocessing the data is essential since the ML method results are as good as the data used. Hence, the traffic flow metrics and biometrics are first preprocessed using the following steps:

1) **Splitting data into train and test datasets:**

   To correctly measure the performance of the ML models, we split the dataset into training and testing datasets with a distribution of 80% and 20%, respectively.

2) **K-Fold:**

   The K-fold method with ten folds was applied only on the training dataset to show the variety of the performance among the folds [95].

3) **Resampling:**

   The collected dataset was unbalanced; normal samples constituted about 88% of the data. This issue can result in bad models that cannot classify attacks [96]. Therefore, we used an over-sampling technique, the synthetic minority over-sampling technique (SMOTE), to balance the dataset at the training stage [97].

3.4.2 Models' Evaluation

We used four ML methods to check the validity of using ML to differentiate between normal and attack biometric data. We compared them based on their performances using accuracy and AUC metrics. Accuracy is the ratio of the number of samples correctly predicted to the total number. At the same time, AUC summarizes the area under the ROC curve into a float number ranging from 0 to 1. ROC is an excellent evaluation metric for sensitivity and specificity trade-offs [98]. K-fold

cross-validation with 10-fold is used to validate the training dataset results statistically. For this, the dataset is divided into ten subsets; in each fold, nine subsets are used for training and one for testing [95].

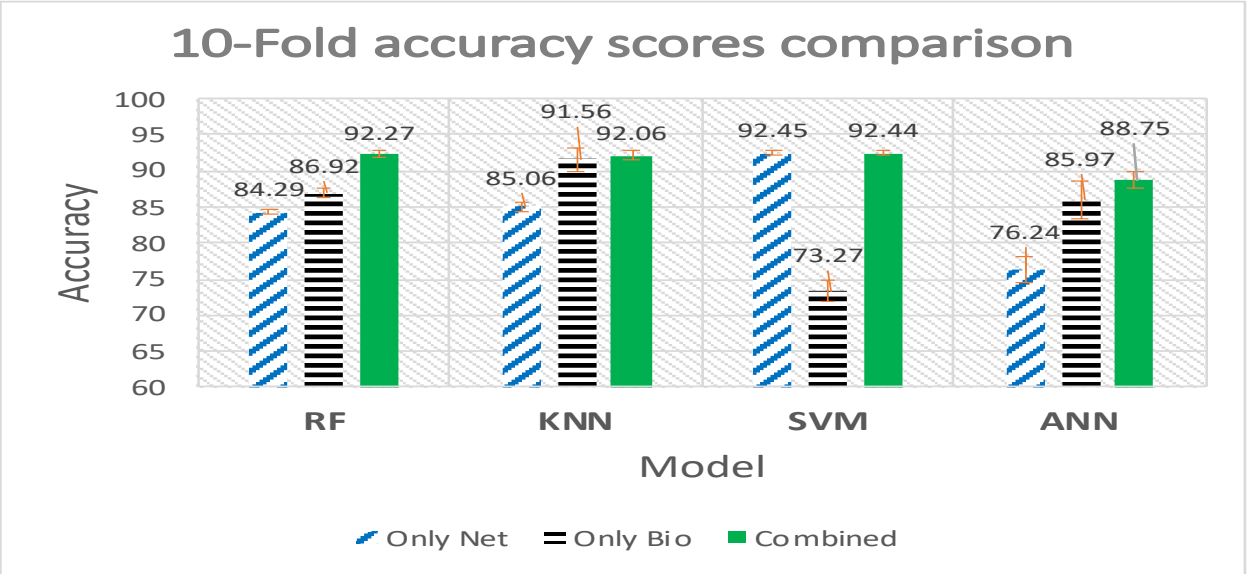


Figure 3.6 10-Fold accuracy scores comparison.

Figure 3.6 shows the accuracy results for all four models with only biometrics, network, and combined features. As can be seen, all models perform better with combined features compared to only biometrics features. Compared to only network features, RF, KNN, and ANN show significantly better results, while SVM performance is similar. These results indicate that combining features provides better results than using only one of the two types of features. However, some confidence intervals of the accuracy results over the ten K-fold runs overlap. This overlap indicates that accuracy is invariant in these overlapping cases or that the performance is not statistically different.

Given the previous invariant results and the fact that accuracy is not a good measure for security applications [99], we also used the AUC metric to show the validity of the accuracy results. As shown in Figure 3.7, the AUC scores confirm the advantage of using combined features with no overlap.

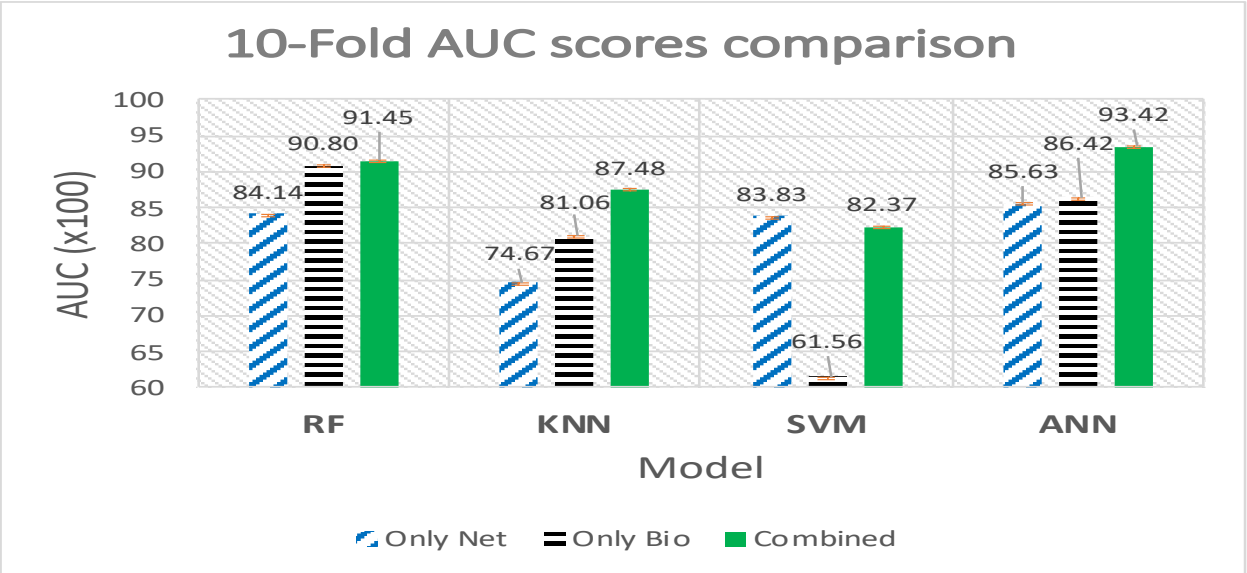
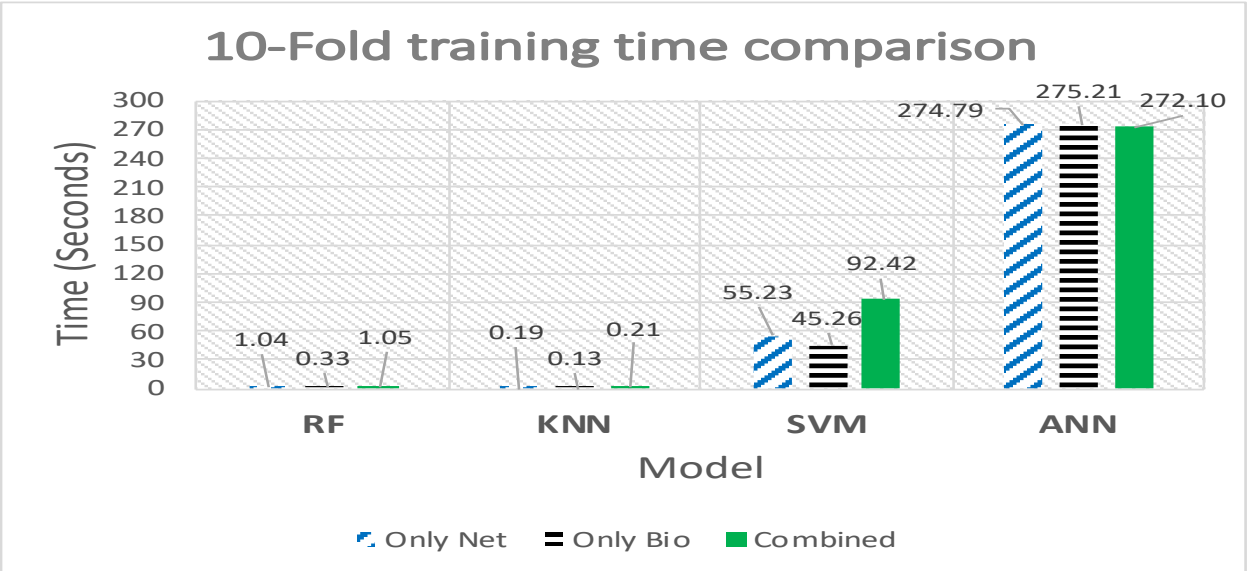
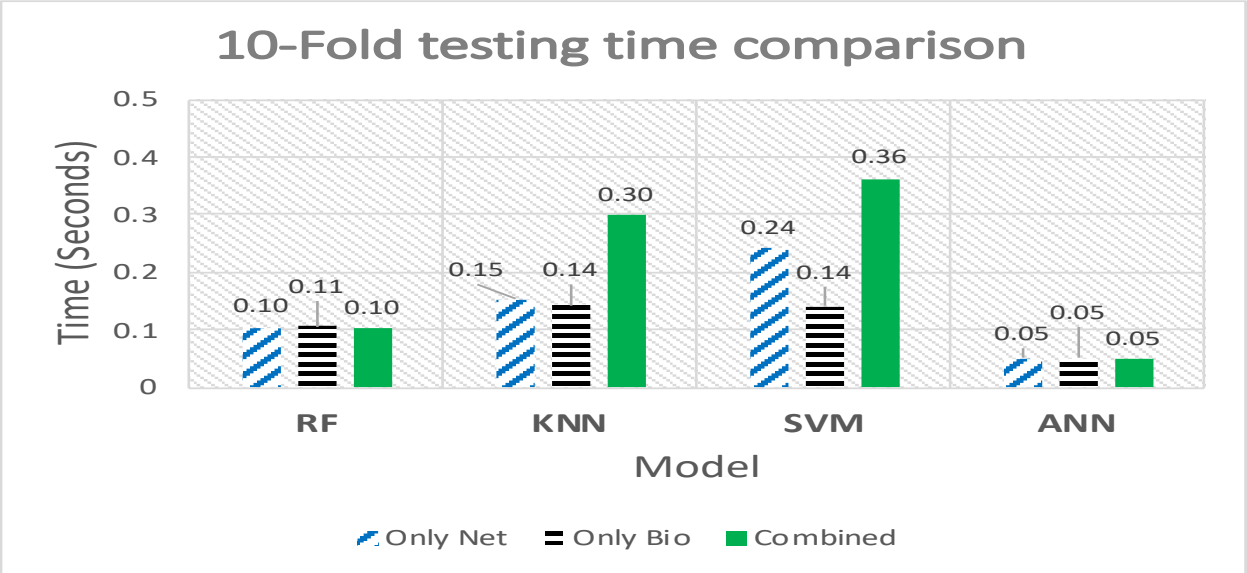


Figure 3.7 10-Fold AUC scores comparison.

Finding the optimal model is essential in healthcare systems, but the time spent training and predicting the samples is as essential. As a result, the average training time and prediction time using the K-fold method for all four ML methods are shown in Figures 3.8(a) and 3.8(b), respectively.



a.10-Fold training time comparison



b.10-Fold testing time comparison

Figure 3.8 Time comparison for all the models.

As shown in Figure 3.8, the training times for RF, KNN, and SVM are less than 1.5 minutes across different types of features, compared to ANN, which is around five minutes. Also, the training time increases as the number of features increases in the first three methods. However, the training time is during offline mode. On the other hand, prediction time is crucial since it is during the online mode, and every second is essential for these systems. All the models have taken 300 milliseconds in the worst-case scenario. However, this time is still high in such systems, considering the system's real-time requirements. ANN shows the lowest prediction time and the highest AUC compared to the other three models. Thus, this model is the best for these systems.

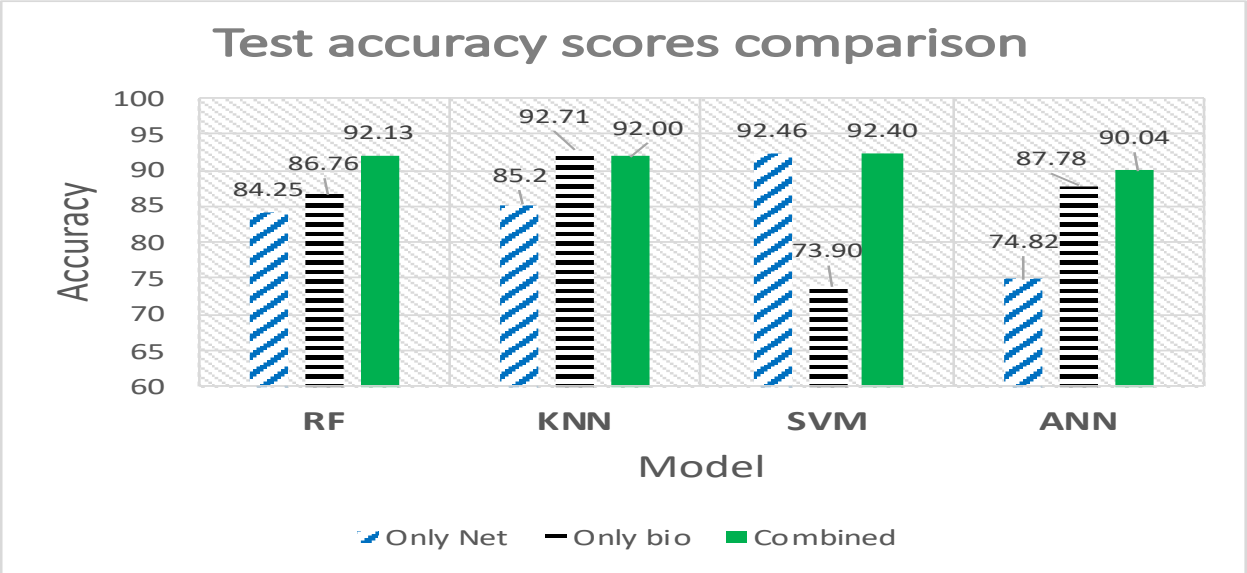


Figure 3.9 Test accuracy scores comparison.

As shown in Figures 3.9 and 3.10, applying the same models to the test dataset shows that all the models perform similarly or better using the combined features. These results are similar to the K-fold results, where AUC distinguishes their performance from accuracy. The improvement in AUC scores reaches up to 25% (in the SVM model.) In addition, ANN shows the highest performance compared to other methods, with an AUC score of 92.98%. We do not show their figures because all models' training and prediction times are similar to the average timing in the K-fold experiment.

These results conclude that using network flow metrics with patients' biometrics enhanced the ML methods for securing health monitoring systems. Also, these results have shown that not all ML methods are suitable for health monitoring systems, especially regarding prediction time. ANN requires the lowest time for prediction compared to the other methods.

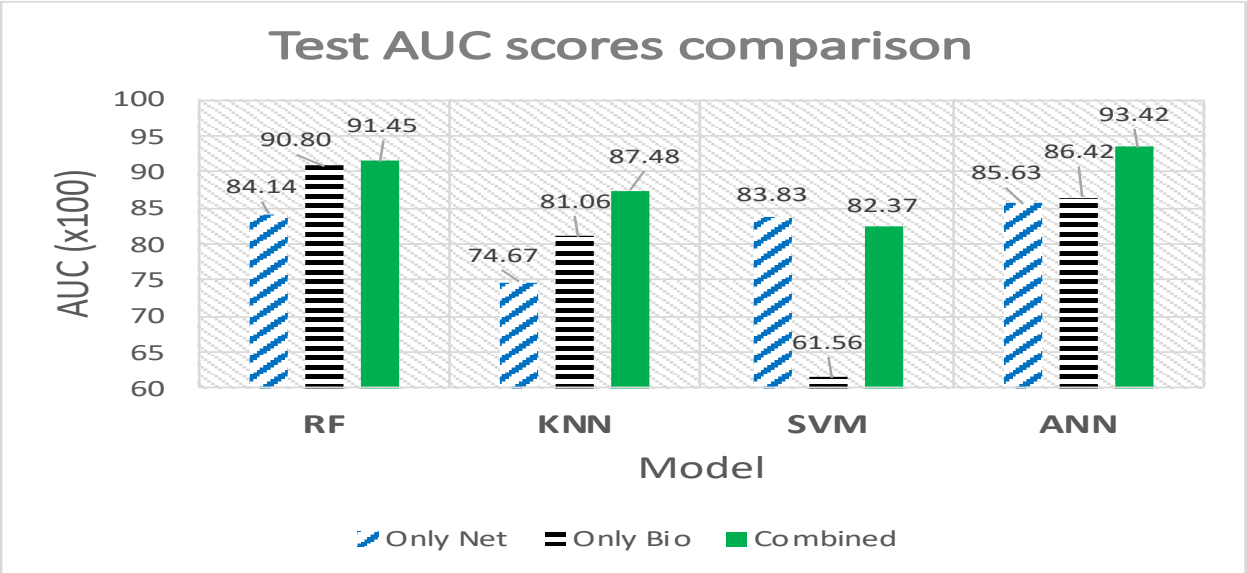


Figure 3.10 Test AUC scores comparison.

3.5 Summary

Due to the high demand for remote healthcare monitoring systems nowadays, a secure system that guarantees the integrity and confidentiality of the data is required. Several small sensors are attached to a patient's body to record the biometric data and track the patient's health. To achieve the full advantages of these sensors, their ability to communicate with remote servers is essential. However, their physical constraints, such as low processing power and limited battery power, may prevent them from providing the required security and privacy for the patient's data. One of the solutions to such constraints is using IDSs to ensure the security requirements of such systems.

Nevertheless, most available healthcare IDSs use network flow metrics or patients' biometric data to build their datasets. In this chapter, we presented the design of an EHMS testbed, where several small sensors were attached to a patient's body. We created a realistic healthcare dataset of more than 16 thousand normal and MitM attack packet records. To build an efficient IDS, we proposed combining the network flow metrics and the patient's biometrics as features to enhance the system

performance. We used four different ML methods: RF, KNN, SVM, and ANN. Then, we compared their performance using three different types of features to train them. Results showed that the AUC could be enhanced by up to 25% by combining the flow metrics and biometrics data. Furthermore, these features had minimal effect on the testing prediction time for the best-performing model.

However, the results show that the system performance is not optimal, which requires further investigation. We plan to enhance the methods' performance for future work by choosing optimal hyperparameters, reducing feature space, and launching more sophisticated attacks.

Chapter 4: Feature Engineering Method

IDS for IoT systems can use AI-based models to ensure secure communications. IoT systems tend to have many connected devices producing massive amounts of data with high dimensionality, which requires complex models. Complex models have notorious problems such as overfitting, low interpretability, and high computational complexity. Adding model complexity penalty (i.e., regularization) can ease overfitting, but it barely helps interpretability and computational efficiency. Feature engineering can solve these issues; hence, it has become critical for IDS in large-scale IoT systems to reduce the size and dimensionality of data, resulting in less complex models with excellent performance, smaller data storage, and fast detection. This chapter proposes a new feature engineering method called LEMDA (light feature engineering based on the mean decrease in accuracy) [7]. LEMDA applies exponential decay and an optional sensitivity factor to select and create the most informative features. The proposed method has been evaluated and compared to other feature engineering methods using three IoT datasets and four AI/ML models. The results show that LEMDA improves the F_1 score performance of all the IDS models by an average of 34% and reduces the average training and detection times in most cases.

[7] A. Ghubaish, Z. Yang, A. Erbad, and R. Jain, "LEMDA: A Novel Feature Engineering Method for Intrusion Detection in IoT Systems," *IEEE Internet of Things Journal,* 2023.

4.1 Introduction and Motivation

The 5G era accelerates large-scale IoT deployment, leading to a surge in high-dimensional data and the associated challenge of "feature explosion" [100]. The sensitive nature of IoT data, especially in domains like IoMT, makes robust security crucial. Traditional IDS approaches often

struggle to efficiently process and analyze this complex data, leading to potential overfitting, interpretability issues, and performance bottlenecks [101, 102]. Feature engineering offers a promising solution by reducing dataset dimensionality, simplifying models, and improving IDS efficiency [103].

Feature engineering is one popular method of such techniques. Feature engineering methods help select the best features for these models, expediting the processes of finding the optimal hyperparameters for the IDS models. We use "feature engineering" and "feature reduction" interchangeably in the rest of this chapter to describe the methods that reduce the datasets' dimensionality.

Most informative features can be selected using feature engineering or dimension reduction techniques, such as feature selection and feature extraction. Feature selection techniques help simplify complex models by reducing the dimensionality of the dataset (number of features), which avoids over-fitting and results in less training time and storage space. Only the most essential features are retained after feature selection. On the other hand, feature extraction techniques, such as principal component analysis (PCA), create new features that preserve the data's variance based on existing features.

Feature selection techniques are divided into four categories: filter, wrapper, embedded, and hybrid [104]. Filter methods are fast but may fail to select the most informative features, leading to low accuracy in ML models. Wrapper methods, like recursive feature elimination (RFE) and forward feature selection (FFS), are effective in selecting informative features but are slow and susceptible to overfitting. Embedded methods, such as the mean decrease in impurity (MDI) and the mean decrease in accuracy (MDA), provide a tradeoff between accuracy and speed, thus providing balanced results between filter and wrapper methods. Finally, hybrid methods are a mix of two or

more of these methods (e.g.,[105, 106]) and feature extraction such as PCA (e.g., [107]). However, these methods are usually designed for specific datasets or models. More about hybrid methods can be found in [108].

To address these limitations, we propose LEMDA, a novel feature engineering method that utilizes the mean decrease in accuracy (MDA) to select informative features, and a weighted exponential decay formula (WEDF) to create a new, informative feature. This approach aims to achieve a superior balance between accuracy, speed, and generalizability for IDS in IoT systems.

Our method consists of two parts: 1) creating a list of the most informative features using the MDA method and 2) creating a new feature from the first feature (the most informative one) in that list. The new feature is created using the weighted exponential decay formula (WEDF) technique. In addition, in cases where the most informative feature is categorical, we utilize the sensitivity factor (SF) to complement the WEDF method for creating a new feature. This case happens, for example, when most attacks are passive, e.g., sniffing. WEDF and SF optimize the relationship between the values in the most informative feature and the samples' classes, as shown in the Evaluation section of this chapter.

LEMDA is a general feature engineering method using AI-based models for the supervised ML-based IDS in IoT systems. We demonstrate the effectiveness of our method by using three different datasets and comparing three different ML models using three different metrics. The evaluation results show the outstanding performance of LEMDA in IDS, with high accuracy and low detection time.

The remainder of this chapter is organized as follows. A brief background of the commonly used feature engineering methods and the related work is provided in Sections 4.2 and 4.3, respectively.

In Section 4.4, we present our proposed method. The experimental methodology and results are shown in Sections 4.5 and 4.6. Finally, we summarize this chapter in Section 4.7.

4.2 Background

This section presents a background of the most commonly used feature engineering methods.

4.2.1 Categories of Feature Selection Techniques

The difference between the four common feature selection techniques – filter, wrapper, embedded, and hybrid – is briefly explained in this subsection. More detailed information can be found in [108] and [109].

1) Filter:

In this technique, features are sorted based on their relevance. Then, a threshold is applied to select the features that have strong relevance. This results in a fast selection but may lead to low accuracy if the dataset distribution is not uniform and the features are highly correlated. The correlation coefficient method is an example of this technique. It measures the linear relationship between the features and selects the features with a correlation below a specific threshold.

2) Wrapper:

In this method, the features are selected by measuring the performance improvement for an ML model using a subset of the features. The subset with the highest improvement in the ML model is selected. This technique effectively selects informative features but is very slow since

it is computationally expensive, and the complexity increases as the number of features increases. For example, the RFE method uses all the initial features, recursively removes them, and then sorts them by their incremental improvement.

3) Embedded:

Embedded techniques combine the advantages of filter and wrapper techniques by embedding feature selection within the ML model. However, this makes it less generic than filter and wrapper techniques. MDI and MDA methods are examples of this technique, which will be explained in detail in the following subsection.

4) Hybrid:

In this technique, two or more filter and wrapper methods (e.g., [105]) are combined to select a subset of the features to take advantage of each method and avoid their disadvantages. It is similar to the embedded technique but is more generic.

4.2.2 Existing Feature Engineering Methods

We chose the embedded technique for comparison with our method among the four feature selection techniques, considering their balance between accuracy and selection time. Specifically, we delve into two embedded feature selection methods, MDI and MDA. Additionally, we introduce the PCA feature extraction technique as part of our comparison, as it is commonly used in similar studies.

By introducing the three techniques, this subsection aims to clarify the differences between our method and existing methods, which will be highlighted and compared in the results section.

5) MDI method:

MDI, also called Gini importance, is based on RF and is used to calculate the importance of each feature based on the weighted sum of the actual decrease in impurity for each feature across all trees [110]. The larger the MDI score, the more important the feature. Since IDS uses a binary classification model, labeling with normal and attack, the decrease in impurity (I) can be calculated using Eq. 4.1:

$$I = G_{PE} - P_{LS} - G_{LS} - P_{RS} - G_{RS} \tag{4.1}$$

Here, G_{PE} is the parent Gini (G) impurity index, as shown in Eq. 4.2. G_{LS} and G_{RS} are G indices for the left and right splits from the parent node in the tree, and P_{LS} and P_{RS} are the proportions for each split from their parent node (i.e., $P_{LS} + P_{RS} = 1$).

$$G = \sum_{i=1}^{n_c} p_i(1 - p_i) \tag{4.2}$$

Here, n_c is the number of classes, which in our case is 2, and p_i is the ratio for the i^{th} class. G equals 0.5 if the number of samples for each class is the same and 0 if only one class is found in the dataset. However, this method is known to be biased toward high cardinality features [111].

6) MDA method:

MDA is also called permutation importance (Perm) and is similar to MDI as both are based on RF [111]. This method requires a validation set to calculate the importance score for each feature (*f*). This score is the weighted difference between the model's prediction error rate for the validation set before and after the permutation of each feature *f* across all the trees, as shown in Eq. 4.3:

$$MDA\ \text{score}_f = \frac{1}{n_t}\sum_{i=1}^{n_c}(sa_{jf} - sb_{jf}) \tag{4.3}$$

Here n t is the number of trees, sa_{jf} and sb_{jf} are the scores after and before permutating feature f in the j^{th} tree, respectively. Similar to MDI, the larger the score, the more important the feature. In general, MDA can result in ignoring more irrelevant features than the MDI method.

7) PCA method:

PCA differs from the previous two methods since it creates new features different from the original ones. These new features are called principal components (PCs) that are uncorrelated and represented by a set of eigenvectors [108]. These eigenvectors and their corresponding eigenvalues are calculated using a covariance matrix. The PCs are sorted in descending order based on their explained variance, where the first PC has the highest explained variance among all features. The explained variance for each PC (var_PC_i) is the ratio of that PC's eigenvalue (λ_i) to the sum of all eigenvalues, as shown in Eq. 4.4.

$$var\_PC_i = \frac{\lambda_i}{\sum_{i=1}^{n_{pc}}\lambda_i} \tag{4.4}$$

The easiest and most effective way to set the required number of PCs (n_{pc}) with good performance is by setting a threshold to calculate the necessary number of PCs to get 95%-99% explained variance [112]. PCA improves model performance and is versatile to most ML models, but it is laborious to tune the threshold.

4.3 Related Work

The classification of IDS can be divided into five categories: network IDS (NIDS), host IDS (HIDS), protocol-based IDS (PIDS), application protocol-based IDS (APIDS), and hybrid IDS [113]. NIDS [114, 115] are designed to monitor the network traffic of all network communications and are usually centralized in one point of the system, such as the cloud. On the other hand, HIDS [116] only monitors the traffic of only one device. PIDS[117, 118] and APIDS are set up to monitor specific protocol connections, e.g., hypertext transfer protocol secure (HTTPS), and application-specific protocols, e.g., structured query language (SQL), respectively. As mentioned above, hybrid IDS integrates multiple IDSs to leverage each IDS type's strengths.

Most of the IDS in the IoMT systems are NIDS since the extensive infrastructure of IoMT systems requires IDS that can monitor the whole network. While various types of IDS exist, our method generically applies to all of them. *To prevent any confusion, it is essential to clarify that the main focus of this chapter is on feature engineering techniques for IDS in IoT systems. The intention is not to introduce a new IDS for IoT but to propose a generic feature engineering approach that can be applied effectively in IoT environments.*

Different prior works have shown the importance of feature engineering in improving the IDS's performance [119] in the context of IIoT security, such as supervisory control and data acquisition (SCADA) systems [120] and cloud security [121]. According to Hakim et al. [119], feature engineering has improved some of the tested models' accuracies from 51% to 97%. Also, the required training time in all models has been almost reduced by half. Thus, developing a feature reduction or feature extraction approach to enhance ML models' performance is commonly recommended [122].

Improving IDS's performance can be achieved by using one or more feature reduction methods (i.e., the hybrid technique introduced in Subsection 4.2.1), as discussed in [106, 123-125]. Ravindranath et al. [123] propose a feature reduction method that utilizes the whale Pearson hybrid wrapper. This method is based on the binary Whale optimization algorithm, a swarm intelligence algorithm. It reduces the data features from 42 features in the HackerEarth network attack prediction dataset [126] to only 8, with an 8% accuracy improvement compared to the original dataset using the k-nearest neighbors algorithm. Padmashree and Krishnamoorthi [124] propose a decision tree-based Pearson correlation recursive feature elimination (DT-PCRFE) model to select a subset of the features to detect various attacks via an optimized DNN model using the BOT-IoT dataset. This model reduces the number of features in the BOT-IoT dataset to only nine with 99.20% accuracy.

Kamarudin et al. [106] combine filter and wrapper methods as a single hybrid feature reduction method. This method reduced the number of features from 41 and 33 to 12 and 5 for the KDD CUP'99 and DARPA 1999 datasets, respectively. Also, it enhanced the IDS performance by 9%.

Another feature reduction method for IDS developed by Pawar et al. [125] selects a subset of features based on a voting scheme from a list of feature selection methods. This scheme reduced the number of features from 41 to 14 for the NSL-KDD dataset and 47 to 18 for the UNSW-NB15 dataset. Nevertheless, none of these feature engineering methods are designed to work on IoT systems.

Another way to design a hybrid feature reduction method is by combining a feature reduction method with some specific ANN algorithm. Jingyi et al. [127] implement a method based on su-

pervised locality-preserving projections and use a backpropagation neural network called an extreme learning machine. Madanan et al. [128], Abdul Lateef et al. [129], Fatani et al. [130], and Dahou et al. [131] also design similar methods using intelligent water drops, crow swarm optimization algorithms, Aquila optimizer (AQU), and reptile search algorithm (RSA), respectively. While these methods use the KDD CUP'99 dataset, the work of Fatani et al. included three other datasets, including NSL-KDD, BOT-IoT, and CIC2017. Using the KDD CUP'99 dataset, the Fatani et al. method performed the best with an accuracy score of 99.92% compared to 99.56%, 92.34%, 98.58%, and 98.34% for Madanan et al., Dahou et al., Jingyi et al., and Abdul Lateef et al., respectively. However, these methods must work with ANN models, which require significant computing power and only work on powerful devices.

Hybrid feature reduction methods are also used to improve the detection rate for medical diagnostics, such as [105, 107]. Shaban et al. [105], similar to [106], employ filter and wrapper methods to improve the performance of a KNN model, which is used as a new COVID-19 detection strategy. On the other hand, Li et al. [107] illustrate that using multiple feature reduction methods, including PCA in a support vector machine model, can enhance the detection rate for sleep apnea. Nimbalkar et al. [132] propose a hybrid feature selection method based on the information gain and gain ratio methods to detect DoS and DDoS attacks in IoT systems using the BOT-IoT and KDD Cup 1999 datasets.

In general, our method stands out from other approaches as it significantly improves the performance of IDS in IoT systems. Also, it takes advantage of both feature selection and extraction methods and reduces their drawbacks. LEMDA is based on embedded methods and achieves a better tradeoff between performance and speed. It supports various attacks without needing a

specific ML or complex ANN model. Additionally, it often leads to faster IDS models compared to alternative methods.

4.4 Our Proposed Method

Designing a new feature reduction method is essential to enhance the prediction for IDS, especially for IoT systems, since they require fast detection. This method makes high accuracy and fast execution indispensable for IDS models.

Our method, LEMDA, is based on MDA and consists of two techniques to satisfy the high accuracy and fast speed requirements of IoT-oriented IDS. The primary technique is WEDF, which runs after MDA, where the list of the most informative features is selected. The second one, SF, is an add-on technique to handle the datasets with a categorical feature as the most important feature for the cases when there are a majority of passive attacks like sniffing. In this section, we explain these two techniques in detail. For the rest of the chapter, we will use f_m to represent the most informative feature in the list, selected by the MDA method, and f_{nm} to represent a new feature created by the WEDF method.

4.4.1 Weighted Exponential Decay Formula (WEDF)

WEDF creates a new feature f_{nm} based on a predefined dictionary (WEDF). This dictionary is constructed from f_m by transforming its samples' values into weights using the exponential decay formula (Eq. 4.5).

$$f(x) = ab^x \tag{4.5}$$

Here, *f(x)* is the output value (after the decay) in the exponential decay formula, a is the initial value (before the decay), b is the decay factor (a static fraction, $0 < b < 1$, that needs to be set

before running the WEDF method, e.g., b = 0.5), and x is the period (during which a has been decayed). Since a is a static parameter, it can be removed in the WEDF method (i.e., considering a = 1).

$$WEDF_u = f(p)w_u = b^p w_u \qquad \text{where} \quad w_u = \frac{z_u}{n_u} \tag{4.6}$$

Eq. 4.6 calculates $WEDF_u$, the WEDF score for each u that constitutes WEDF. u is a specific unique value from all data instances of the f_m feature. Each u corresponds to a unique data value. In the context of the WUSTL-EHMS dataset, for instance, u can be "TCP," which is a value of the f_m feature. A more detailed example is provided in the next paragraph. We add a new weight parameter. w_u for each unique value u in f_m. z_u represents the number of attack samples in the training dataset for each u in f_m, and n_u is the total number of samples for each u in f_m. Hence, z_u divided by n_u will result in w_u for each u. All the weights are sorted in descending order based on n_u. Let us denote each u's index as p (i.e., p ranges from 1 to the number of unique values in f_m).

For instance, let us assume that 100 out of 1000 samples in the training dataset have u = TCP as their unique value, 10 of which are attack samples; then, z_{TCP} = 10, n_{TCP} = 100, and w_{TCP} = 10/100 = 0.1, which will be stored in the w dictionary. Then, assuming TCP is the first unique value in the w (i.e., p = 1) and setting b = 0.5, we can calculate $WEDF_{TCP}$ by $b^p w_{TCP} = 0.5^1 \times 0.1 = 0.05$. Hence, all the samples with u = TCP will have $WEDF_{TCP}$ = 0.05. Other unique values in f_m with zero attack samples will have a WEDF score of zero in f_{nm}.

Finally, the WEDF scores for the u values in the f_m feature using the training dataset are stored in a dictionary (WEDF). This dictionary creates the f_{nm} feature in the training and testing datasets. Then, the f_m feature is deleted from both datasets. Algorithm 1 shows the step-by-step implementation of generating the dictionary.

Upon generating the dictionary (assigning a weight to each u based on the proportion of attack samples associated with each value, i.e., w_u to attack samples and 0 to normal samples), the feature distribution for the f_{nm} feature will become roughly a bimodal distribution. Consider an example of a f_m with a standard normal distribution $N(0, 1)$. After applying WEDF, the normal samples will tend to cluster around the distribution's left side (0.1% region), and the attack samples will cluster around the right side, resulting in a gap in the distribution between the normal and attack samples. This gap helps the ML model easily separate the normal and attack samples. It increases the importance of the f_{nm} feature compared to the original f_m feature, resulting in better IDS performance.

Algorithm 1 WEDF Method

Input: f_m feature from the Training Dataset (D_T)
Output: $WEDF$ for all $WEDF_u$ in f_m to create f_{nm}
$WEDF, z, w, w_{attack}, w_{normal} = \{\}, \{\}, \{\}, \{\}, \{\}$
$n \leftarrow value\_counts(D_T[f_m])$
for u in n **do**
    $z[u] \leftarrow length(D_T[f_m][u], label = attack)$
    if $z_u >= 1$ **then**
        $w_{attack}[u] \leftarrow z[u]/n[u]$
    else
        $w_{normal}[u] \leftarrow 0$
    end if
end for
$w \leftarrow concatenate(w_{attack}, w_{normal})$
$p = 1$
for u in w **do**
    $WEDF[u] \leftarrow b^p \times w[u]$
    $p = p + 1$
end for

4.4.2 Sensitivity Factor (SF)

SF has been added as an add-on besides using the WEDF technique in case the f_m feature is a categorial feature like the Flags feature in networking, and most attacks are passive attacks like sniffing. This add-on requires the training and testing datasets, individually, to be arranged in a sequential order, typically based on the timestamps associated with the samples.

SF is also based on the exponential decay formula without multiplying weights w. As shown in Eq. 4.7, we use d, an index of the current sample (s) based on the last seen suspicious sample, as the input to the exponential decay formula (Eq. 4.5). Similar to the WEDF method, a new feature f_{snm} is created in the training and testing datasets using the f_m feature. Using the *Flags* feature in networking as an example, any item in Flags that differs from the common values of *Flags*, such as duplicate MACs (M), is considered a suspicious sample. This add-on is used in one of the three datasets, and its results are promising, as shown in the results section.

$$SF_s = b^d \tag{4.7}$$

In the case of suspicious samples, the SF score reaches its peak (i.e., a higher SF score indicates a greater likelihood of being an attack sample). Then, the score exponentially decreases for each sample after the suspicious sample(s) until the score reaches zero, as shown in Algorithm 2. This decrease is because cyber-attacks usually exhibit intensive behaviors over a continuous period, and the network traffic returns to a normal state after a certain duration.

Algorithm 2 SF Method

Input: f_m feature from the Training Dataset (D_T)
Output: f_{snm} feature
$common\_value \leftarrow$ the most common value in the normal samples
$b, d = 0, 1$
for s in f_m **do**
  if s is not a $common\_value$ in f_m **then**
    $b, d = 0.5, 1$
  end if
  $s_{new} \leftarrow b^d$
  $d = d + 1$
end for

4.5 Experimental Methodology

In this section, we will demonstrate the evaluation results of the proposed method using three datasets and three models. The datasets are WUSTL-EHMS [6], MQTT-IoT [133], and BOT-IoT [94]. Two are collected from general IoT systems (MQTT-IOT and BOT-IoT), and one from an IoMT system (WUSTL-EHMS), which is presented in Chapter 3. Our ML models include decision trees (DT), RF, and two ANN models. We use the F_1 score, the Safety score [134], and the accuracy metrics to compare the performance of feature engineering methods applied to these models.

Using these datasets and models, we compare our methods to two widely recognized feature reduction techniques, PCA and MDA, and the scenario where no feature reduction (Base) is applied.

4.5.1 Datasets

The three IoT datasets used in our experiments have different sizes starting from 16k to 10M samples and similar numbers of features, as shown in Table 4.1:

Table 4.1 Datasets statistics.

Dataset	Number of Samples	Number of Features
WUSTL-EHMS	16,317	44
MQTT-IoT	2,000,000	31
BOT-IoT	10,000,000	35

8) WUSTL-EHMS:

This dataset was collected from a real-time EHMS testbed, presented in Chapter 3 at Washington University in St. Louis, representing a real IDS for the IoMT systems [6]. The types of attacks in this dataset are based on MitM attacks, such as sniffing and injection attacks. Hence, this dataset has passive (sniffing only) and active (injection) attacks. This dataset is explained in [6] and [135].

9) MQTT-IoT:

This dataset uses message queuing telemetry transport (MQTT) protocol for IoT systems [133]. The types of attacks in this dataset are as follows: user datagram protocol (UDP) scan, aggressive scan, MQTT brute-force Sparta, and secure shell protocol (SSH) brute-force. The number of samples in this dataset is 20M, but we have randomly selected 2M samples containing all attacks in the original dataset. More about this dataset is available in [133, 136].

10) BOT-IoT:

This well-known dataset was created using IDS for IoT systems in the Cyber Range Lab of UNSW Canberra. It has different types of attacks, including theft, reconnaissance, DoS, and DDoS attacks. We selected 10M out of 73M samples to test our method. More about this dataset is available in [94, 137].

The number of selected features for each feature reduction method using these datasets is presented in Table 4.2. The Base method represents the method where we use all the features (without feature

engineering). It is worth noting that we have removed the identifier features, such as IP addresses and port numbers, from all the methods, including the Base method.

Table 4.2 Number of features per reduction method.

Dataset	Base	PCA*	MDA	LEMDA
WUSTL-EHMS	35	14	5	5+1**
MQTT-IoT	25	9	5	5
BOT-IoT	23	10	5	5

* Explained variance = 95%
** Additional feature using the SF method

4.5.2 Models

We have used DT, RF, and two ANN models. The scikit-learn package has been utilized for DT and RF models with default hyperparameters [138, 139]. A simple multi-layer perceptron (MLP) model with two layers is used as a simple ANN model. To show the effect of complex ANN models, we added a convolutional neural network (CNN) model with five layers and Max pooling layers. The Keras package has been utilized for the two ANN models [140]. More about the hyperparameters are shown in Table 4.3.

Our objective is not to develop optimal machine learning models with optimized hyperparameters but to demonstrate our method's robustness and maintain consistency in experimental comparisons with other methods; we use these simple models with identical parameters across all three datasets.

Table 4.3 ANN models hyperparameters.

Parameter	MLP Typical Value(s)	CNN Typical Value(s)
Number of Layers	2	5

Number of neurons per layer	20, 1	32, 128, 512, 1024, 1
Number of epochs	20	2
Kernel Size	None	2
Pooling	None	Max Pooling (size=2)
Batch Size	1000	
Loss Function	*Binary cross-entropy*	
Optimizer	*Adam*	
Activation Function	*tanh, sigmoid*	

The k-fold cross-validation method with ten folds has been utilized on all three models to analyze the models' performance. In each fold, the dataset is divided into ten subsets; nine of them are used for training and one for testing.

4.5.3 Metrics

We evaluate the models' performances using the accuracy, F_1, and Safety scores. All the metrics are calculated based on the following four categories: true negative (TN), true positive (TP), false positive (FP), and false negative (FN). Label attack is defined as positive, and label normal is defined as negative. TN and TP are the cases when an IDS model correctly predicts a normal sample as normal and an attack sample as an attack, respectively. FP is when the model mistakenly predicts a normal sample as an attack, while FN is when an attack sample is predicted as normal.

1) Accuracy:

This metric represents the correct and total prediction ratio, as illustrated in Eq. 4.8.

$$Accuracy = \frac{TN + TP}{FP + FN + TN + TP} \tag{4.8}$$

2) F_1 score:

This metric is popular for security applications such as IDS. It is the harmonic mean between recall and precision, as illustrated in Eq. 4.9.

$$F_1\text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{4.9}$$

3) Safety score:

The safety score is designed by Salman et al. [134] specifically for security applications to fulfill the shortcomings of other metrics in these types of applications. This metric adds weights (Eq. 4.10) to the four primary model outcome categories (TP, TN, FP, and FN).

$$\text{Safety score} = \frac{w_{TN}TN + w_{TP}TP}{w_{FP}FP + w_{FN}FN + w_{TN}TN + w_{TP}TP} \tag{4.10}$$

For generality, we assign the following weights assuming that both FN and FP have the same importance, as explained in [6] and [134]:

$$w_{TN} = \frac{1}{19}, w_{TP} = \frac{2}{19}, w_{FP} = \frac{8}{19}, w_{FN} = \frac{8}{19}$$

Table 4.4 List of selected features.

	Dataset		
No.	WUSTL-EHMS	MQTT-IoT	BOT-IoT
1	***Flgs***	***protocol***	***state***
2	*DIntPkt*	*mqtt_messagetype*	*sbytes*
3	*DstJitter*	*ttl*	*bytes*
4	*Rate*	*mqtt_messagelength*	*proto*
5	*DstLoad*	*ip_len*	*srate*

4.6 Experimental Results

The evaluation results are categorized into five subsections. The first three subsections present the results for each dataset individually, considering all the methods. The fourth subsection compares the training and detection times across all datasets and models, employing all the methods. Lastly, the fifth subsection provides a comprehensive comparison with previous works.

For the metrics and time comparisons, we use the average of the k-fold cross-validation with ten folds. We set the b decay factor for WEDF and SF to be 0.5 as a balanced tradeoff between FN and FP values. Further investigation involved conducting multiple trials within the range of 0.1 to 0.9. Through these trials, we found that the FP value increased, the FN value decreased as b increased, and vice versa. Since we have set both values to be equally important, as explained in the previous section, the *b* value is set to be 0.5 for WEDF and SF. For other applications, the *b* value should be set based on the importance of the FN value over the FP value or vice versa. The selected features using the MDA method and our method for all three datasets are shown in Table 4.4. Note that our method creates an f_{nm} feature for each dataset using their f_m features, shown in the first row of Table 4.4.

4.6.1 WUSTL-EHMS

The WUSTL-EHMS dataset, as shown in Table 4.4, has the *Flgs* feature as the most informative feature f_m, and most attacks are passive attacks (sniffing). As a result, among the three datasets, it is the only one suitable for using the SF add-on with the WEDF method. Across the three models in Table 4.5, our method shows average values for accuracy, F_1 score, and Safety score of approximately 95%, 78%, and 73%, respectively. These outcomes show that our method has an average

improvement of about 28%, 52%, and 61% in accuracy, F_1 score, and Safety score, respectively, compared to the other methods.

DT, RF, and ANN models show similar performances across the Base, PCA, and MDA methods, while our method outperforms all. As seen in Table 4.5, our method performs almost twice better with the security-oriented metric Safety score than the other methods in the three models. With the accuracy and the F_1 score, our method still significantly shows improved results compared to other methods.

Table 4.5 Methods results for the three datasets.

WUSTL-EHMS Dataset										
Model	Method	TP	FN	FP	TN	Accuracy	F_1	Safety	Training Time	Detection Time
DT	Base	127	78	358	1069	73.284%	36.812%	27.499%	0.247	0.000219
	PCA*	109	95	244	1184	79.228%	39.138%	34.079%	0.336	0.000173
	MDA	125	80	288	1139	77.451%	40.453%	32.056%	0.082	0.000194
	LEMDA	173	32	43	1384	**95.404%**	**82.185%**	**74.249%**	**0.069**	**0.000132**
RF	Base	115	90	125	1302	86.826%	51.685%	47.109%	0.451	0.020998
	PCA*	105	100	139	1288	85.355%	46.771%	43.930%	0.661	0.020724
	MDA	122	83	189	1238	83.333%	47.287%	40.514%	0.386	0.021060
	LEMDA	183	22	37	1390	**96.385%**	**86.118%**	**78.815%**	**0.287**	**0.020516**
MLP	Base	104	101	179	1248	82.843%	42.623%	39.394%	1.716	0.067576
	PCA*	105	100	221	1206	80.331%	39.548%	35.542%	1.238	0.071022
	MDA	150	55	730	697	51.900%	27.650%	13.701%	1.229	**0.063846**
	LEMDA	116	89	15	1412	**93.627%**	**69.048%**	**66.397%**	**1.213**	0.064459
CNN	Base	134	71	546	881	62.194%	30.282%	18.882%	2.369	0.142412
	PCA*	159	46	1018	409	34.804%	23.010%	7.869%	1.498	0.128004
	MDA	180	25	1142	285	28.493%	23.576%	6.462%	**1.067**	**0.106574**
	LEMDA	126	79	7	1420	**94.730%**	**74.556%**	**70.847%**	1.068	0.120796
MQTT-IoT Dataset										
Model	Method	TP	FN	FP	TN	Accuracy	F_1	Safety	Training Time	Detection Time

DT	Base	19596	160651	2383	17370	18.483%	19.380%	4.156%	4.251	0.008660
	PCA*	113441	66806	6508	13245	63.343%	75.578%	29.049%	3.221	0.006749
	MDA	20361	159886	2463	17290	18.826%	20.053%	4.276%	**1.038**	0.004655
	LEMDA	160890	19357	9984	9769	**85.330%**	**91.644%**	**58.549%**	1.077	**0.004508**
RF	Base	55720	124527	2061	17692	36.706%	46.818%	11.309%	58.882	0.124873
	PCA*	57690	122558	2093	17659	37.675%	48.069%	11.771%	60.652	0.119515
	MDA	20234	160013	2433	17320	18.777%	19.943%	4.257%	31.612	0.109121
	LEMDA	178417	1830	12052	7701	**93.059%**	**96.255%**	**76.649%**	**24.630**	**0.107833**
MLP	Base	64395	115852	1088	18665	41.530%	52.411%	13.616%	40.425	1.752910
	PCA*	112778	67469	4794	14959	63.869%	75.736%	29.381%	38.434	1.727500
	MDA	16252	163995	629	19124	17.688%	16.489%	3.772%	**37.485**	1.737760
	LEMDA	178972	1276	12061	7691	**93.332%**	**96.408%**	**77.411%**	37.505	**1.708496**
CNN	Base	115844	64404	4715	15037	65.441%	77.022%	30.853%	181.872	6.220485
	PCA*	94843	85404	8631	11122	52.983%	66.857%	21.069%	108.087	5.084061
	MDA	72885	107362	4843	14910	43.898%	56.505%	15.183%	**72.776**	**4.669276**
	LEMDA	179002	1246	12071	7681	**93.341%**	**96.414%**	**77.439%**	72.949	4.673046

BOT-IoT Dataset

Model	Method	TP	FN	FP	TN	Accuracy	F_1	Safety	Training Time	Detection Time
DT	Base	917734	82136	5	125	91.786%	95.716%	73.638%	175.926	0.050922
	PCA*	965000	34870	42	88	96.509%	98.223%	87.359%	392.076	0.059293
	MDA	950423	49447	5	125	95.055%	97.464%	82.774%	38.299	0.038469
	LEMDA	999854	16	7	123	**99.998%**	**99.999%**	**99.991%**	**36.079**	**0.036017**
RF	Base	981079	18791	3	127	98.121%	99.051%	92.883%	445.560	0.713076
	PCA*	984417	15453	42	88	98.451%	99.219%	94.077%	982.130	0.829192
	MDA	962375	37495	2	128	96.250%	98.089%	86.517%	311.985	0.635074
	LEMDA	999857	13	11	119	**99.998%**	**99.999%**	**99.990%**	**152.270**	**0.611572**
MLP	Base	938838	61032	7	123	93.896%	96.852%	79.362%	223.130	8.510574
	PCA*	936317	63553	28	102	93.642%	96.716%	78.641%	**202.012**	8.656834
	MDA	499925	499945	22	108	50.003%	66.665%	20.000%	206.871	8.560095
	LEMDA	998828	1042	47	83	**99.891%**	**99.946%**	**99.566%**	207.164	**8.422081**
CNN	Base	964703	35167	26	104	96.481%	98.209%	87.267%	964.367	41.930902
	PCA*	930038	69832	27	103	93.014%	96.380%	76.897%	619.433	25.352587

	MDA	487379	512491	10	120	48.750%	65.541%	19.210%	410.698	23.708180
	LEMDA	977230	22640	27	103	**97.733%**	**98.854%**	**91.510%**	**403.539**	**23.344931**

* **Explained variance = 95%**

4.6.2 MQTT-IoT

Similar to the WUSTL-EHMS dataset results, across the three models, our method showcases average accuracy, F_1 score, and Safety score of approximately 91%, 95%, and 73% across the three models. MQTT-IoT results showed that our method has outperformed other methods by at least 50% using the F_1 score, as illustrated in Table 4.5. Furthermore, even when the MDA method uses the same 4 out of 5 features as our method, the difference in performance between them reached almost 70% on average. The average improvements of our method using the accuracy and Safety scores are 56% and 79%, respectively.

4.6.3 BOT-IoT

This dataset has more attacks, such as DoS and DDoS attacks, making it more general for IoT systems than the others. Our method demonstrates an average performance of approximately 99% for accuracy, 99% for the F_1 score, and 98% for the Safety score across the three models. Even here, our method shows clear performance improvement in the DT and RF models compared to the other methods in terms of accuracy, F_1 score, and Safety score, as shown in Table 4.5.

Given these results and the varying attacks in each dataset, our method demonstrates superior performance, rendering it more suitable as an IDS for IoT systems using AI-based models than other methods.

In particular, by comparing the results of the MLP and CNN models across all three datasets, the CNN exhibits superior performance over the MLP in the two large datasets, MQTT-IoT and BOT-

IoT. These results suggest that a more complex ANN model benefits large datasets without feature engineering. Additionally, LEMDA exhibits enhancement (from MLP to CNN) in two datasets, WUSTL-EHMS and MQTT-IoT. However, this increase in complexity leads to very high training and detection times.

These findings confirm that feature engineering methods are essential to reduce the computational complexity with simpler models. While more complex models may not necessarily enhance performance, they still contribute to this reduction of computing time.

4.6.4 Training and Detection Time Comparison

Improving model performance is not the only requirement for IDS in IoT systems since the models' training and detection time are also critical. As presented in Table 4.5, our method achieves the lowest or very close to the lowest training time compared to other methods, with an average of 0.66s, 34.04s, and 199.76s in WUSTL-EHMS, MQTT-IoT, and BOT-IoT datasets, respectively.

The detection times are also shown in Table 4.5. Similar to the average training time, our model detection times are the lowest in almost all the cases, with an average of 0.05s, 1.62s, and 8.10s in WUSTL-EHMS, MQTT-IoT, and BOT-IoT datasets, respectively. These results let us conclude that our method enhances the IDS performance and takes less time to train and detect attacks using different models in most cases. Hence, it makes the IDS models very accurate and fast to train ML models and detect attacks.

4.6.5 Related Work Comparison

In addition to comparing our work with PCA and MDA, we further assess its performance against four related works [124] and [130-132] using the BOT-IoT dataset, as presented in Table 4.6. As mentioned in Section 4.3, [130-132] used only one ML classifier model to report their results,

while the [124] method uses DNN classifier. In contrast, we have tested our method with multiple ML models, including DT and RF, in addition to the two ANN models, including MLP and CNN, to show the versatility of our method.

All the attacks were included in [124] and [130, 131] results, while only DoS/DDoS attacks were included in [132]. The methods proposed by [124] and [130, 131] require an ANN model as part of their implementation. The methods in [130, 131] were built to transform the feature space before selecting the best set of features. On the other hand, our method will only transform the most informative feature after the MDA method completes the selection process. Compared to these methods, our work shows comparable or better results than these works with up to 85% feature reduction rate using ML and ANN models.

Table 4.6 Related work comparison using BOT-IoT dataset.

	LEMDA	[124]	[130]	[131]	[132]
Category	Supervised	Supervised	Supervised	Supervised	Supervised
Type of attacks	All	All	All	All	DoS/DDoS
Model	DT/RF/ANN	DNN	KNN	KNN	JRip
Require ANN	No	Yes	Yes	Yes	No
Reduction Approach	Selection+ Extraction	Only selection	Selection+ Extraction	Selection+ Extraction	Only selection
Reduction Method	MDA+WEDF	DT-PCRFE	AQU+CNN	RSA+CNN	IG+GR
No. of Samples	10M	13.9M	3.6M	3.6M	0.7M
Feature Reduction Rate	85.7%	74.3%	—	—	54.3%
No. of Features	5	9	—	—	16
Accuracy	99.998% [DT/RF]	99.200%	99.994%	99.993%	99.999%
*F*1	99.999% [DT/RF]	98.910%	99.992%	99.992%	—

4.7 Summary

IDS models for IoT systems require faster training and detecting time along with high performance. Therefore, these require specialized feature reduction methods. This chapter presented a new feature reduction method called LEMDA. Our proposed method uses two new WEDF and SF techniques to generate a representative feature based on the most informative feature from the MDA method. We used three different datasets with different sizes, three different ML models, and three different metrics. We compare our method with other methods, including MDA, PCA, and a base method without feature reduction methods as the ground truth of our experimental results. Our results show that LEMDA performs better than the other methods in all three datasets and ML models by an average of 34%, 57%, and 56% using the F_1, the Safety scores, and the accuracy scores, respectively. Furthermore, the proposed method achieved the lowest required training and detection times in most cases, making it run faster than other methods.

For future work, we plan to investigate the improvement of our method using best-optimized models and then compare the results with the plain models. We will examine the potential of our method for semi-supervised and unsupervised ML-based IDS. Moreover, applying our method to applications (other than IoT systems) can help determine its limits.

Chapter 5: Hybrid IDS using 5G Network

The evolution of next-generation wireless networks, particularly the integration of MEC in 5G, is set to revolutionize the infrastructure of secure systems. This evolution is exemplified in the IoMT field, where benefits such as remote surgeries and diagnostics become increasingly common, especially in pandemic scenarios. However, incorporating MEC services into the 5G framework significantly enlarges the network's vulnerability to traditional security breaches and introduces new, sophisticated types of attacks. These emerging threats, often undetectable by conventional methods, necessitate the development of an adaptive IDS capable of identifying such complex security challenges with minimal human intervention. To tackle these issues, this chapter introduces a novel HDRL IDS [8] rooted in an actor-critic methodology designed to detect more complex security threats adaptively and autonomously with limited human oversight. Our HDRL IDS combines network and host features analysis, effectively leveraging the advantages of both NIDS and HIDS. Empirical results indicate that our HDRL IDS outperforms traditional NIDS and HIDS in threat detection efficacy. Furthermore, we present a novel dataset derived from an emulated 5G testbed with integrated MEC services to facilitate advanced research and development in the field, specifically for applications designed to address intricate attacks and scenarios demanding high reliability.

[8] A. Ghubaish, Z. Yang, and R. Jain, "HDRL-IDS: A Hybrid Deep Reinforcement Learning Intrusion Detection System for Enhancing the Security of Medical Applications in 5G Networks," in *2024 IEEE International Conference on Smart Applications, Communications and Networking (SmartNets)*, Harrisonburg/Washington DC, VA, USA, 2024.

5.1 Introduction and Motivation

The integration of next-generation wireless networks, particularly 5G and MEC, promises to revolutionize healthcare applications within the IoMT. However, the distributed nature of edge clouds

introduces new security vulnerabilities and makes traditional IDS less effective [141]. Against increasingly sophisticated attacks, there is a pressing need for IDS that can intelligently adapt and detect new threats with minimal human intervention [142].

Deep reinforcement learning (DRL) has emerged as a promising candidate for such dynamic IDS, offering solutions with little need for human oversight to detect new attacks (zero-day attacks) without retraining the IDS model. DRL combines the strengths of reinforcement learning (RL) and deep learning (DL), allowing the RL agent to learn effectively from a low-dimensional feature space [142, 143]. To address the increasing prevalence of zero-day attacks, the primary focus of DRL for IDS has been on NIDS. However, these methods often rely on centralized architectures, which are not readily applicable to the MEC services in 5G.

Most studies prioritize NIDS over HIDS because of the suitability of NIDS for implementing RL algorithms like Q-learning and deep Q-learning (DQL). These methods effectively analyze and learn from network traffic patterns but can be biased and inefficient for large states or action spaces [144]. Additionally, many studies rely on datasets that do not adequately reflect the characteristics of 5G networks, with the NSL-KDD dataset being a common example [145].

To address these gaps, we integrate the advantages of both NIDS and HIDS within an actor-critic DRL framework. We utilize the deep deterministic policy gradient (DDPG), known for its model-free and online capabilities. Moreover, we have developed an emulated 5G testbed with MEC services designed explicitly for healthcare systems. The testbed produced a dataset that more closely aligns with the most recent developments in 5G network technology. Built using Simu5G [146], our testbed replicates a 5G network environment with multiple MEC servers and UEs based on the OMNet++ framework [147].

The rest of the chapter is organized as follows. Section 5.2 provides a background on the DRL model and 5G infrastructure. Sections 5.3 and 5.4 discuss related work and our proposed HDRL testbed. Section 5.5 presents our testbed results. The summary and future work are detailed in Section 5.6.

5.2 Background

This section introduces the typical DRL algorithms and the 5G infrastructure to provide readers with fundamental insights into our approaches.

5.2.1 DRL

RL is a subset of ML techniques in which an agent learns decision-making by engaging with its environment, akin to how humans acquire knowledge through trial and error. RL comprises five key components: agent, environment, state, action, and reward. The agent is the decision-maker, while the environment is what the agent interacts with. The state represents the current situation in the environment. The action is the agent's decision based on the state, determining whether the agent receives a reward for a correct action or a penalty for an incorrect one.

DRL extends RL by integrating DNN to handle complex, high-dimensional environments. This advantage has led to an increased interest in DRL in recent years. The main categories of RL are as follows (as shown in Figure 5.1 [148]):

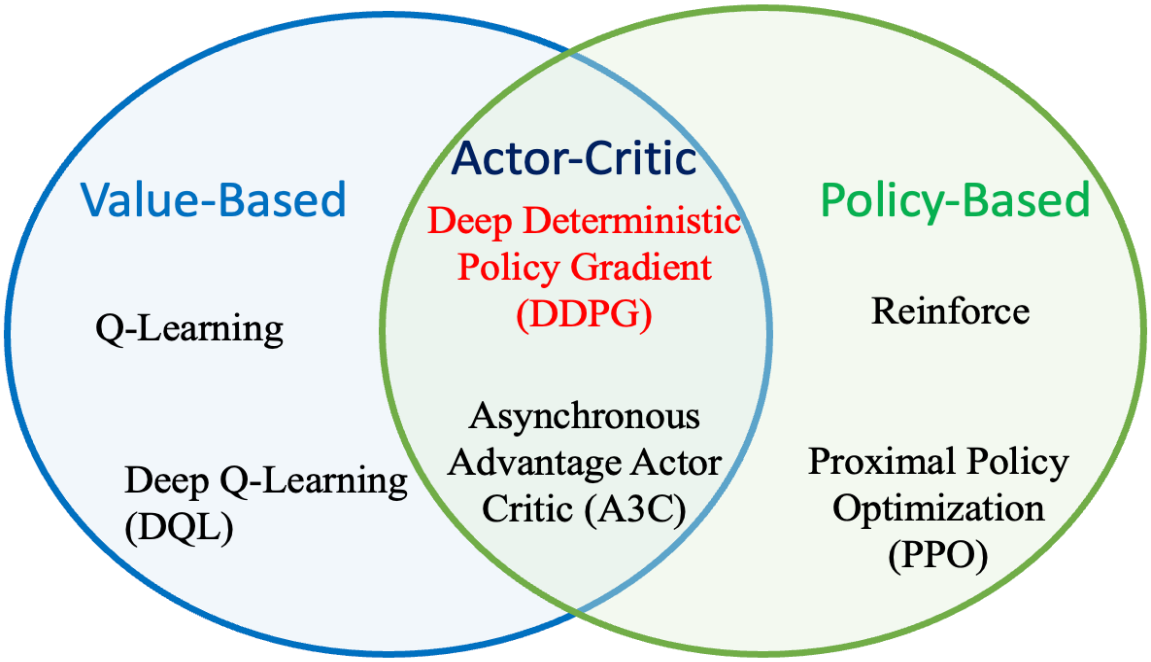


Figure 5.1 Classifications of RL categories.

1) **Value-based methods:**

These methods aim to estimate the value of actions in the current state to find the optimal action that maximizes rewards. Examples include Q-learning and DQL. However, they are prone to high bias [149].

2) **Policy-based methods:**

Instead of estimating action values, these methods directly learn a policy that maps states to actions. Examples are reinforce and proximal policy optimization (PPO). However, they can suffer from high variance and large gradient noise [149].

3) **Actor-critic methods:**

Combining the strengths of value-based and policy-based methods, they consist of an actor who makes decisions and a critic who evaluates these decisions to help the actor adjust its

policy for better performance. Examples include DDPG and asynchronous advantage actor-critic (A3C). In this chapter, we employ DDPG, an actor-critic method, for our HDRL model.

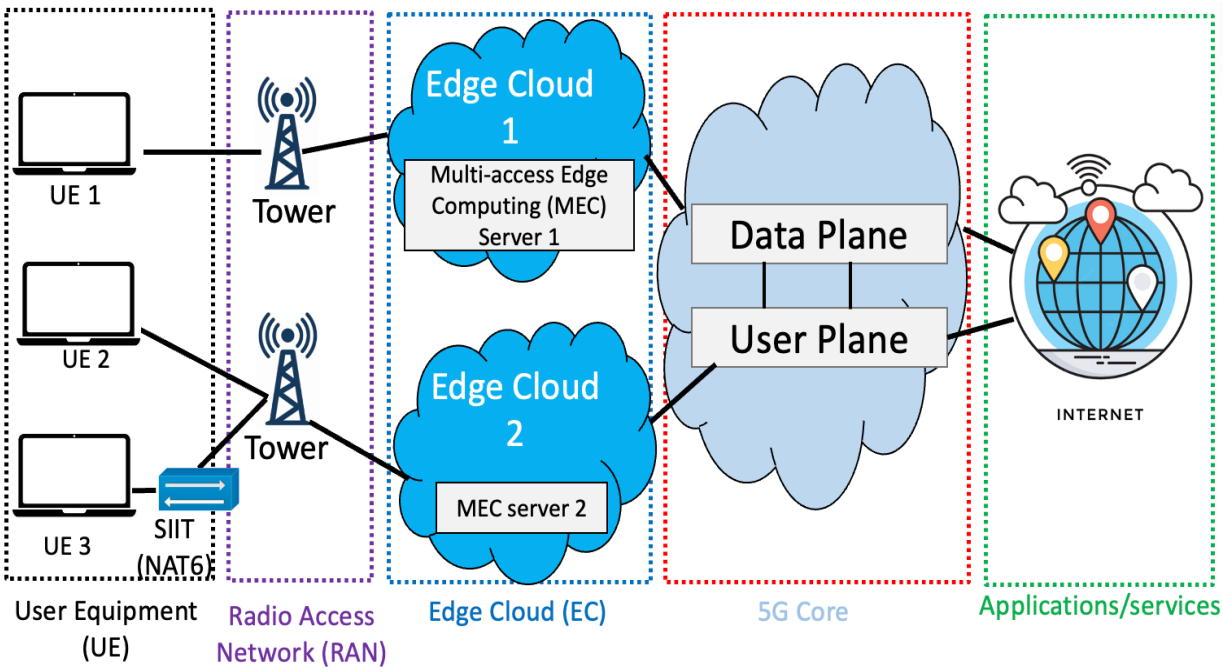


Figure 5.2 5G End-to-End system infrastructure.

5.2.2 5G Infrastructure

Figure 5.2 illustrates the system-level components of the 5G infrastructure, including End Users, the Radio Access Network (RAN), the Edge Cloud, and the Core Cloud. We focus on the URLLC applications, for which the Edge Cloud is a crucial enabler. UEs include two types of users: direct users and indirect users. Direct users are directly connected to the 5G network, while indirect users are inside a private network to benefit from 5G services while not being directly connected to the 5G network. All UEs significantly benefit from MECs in 5G networks by experiencing lower latency and faster data processing. MECs allow data processing closer to the data source at the network's edge.

While 5G has overcome many limitations of 4G networks, it has an increased attack surface. The 5G infrastructure is susceptible to various attack vectors, such as DDoS attacks targeting the edge cloud's network infrastructure. Other potential threats include ransomware, buffer overflow attacks, and MitM attacks, which can involve data manipulation during transmission.

5.3 Related Work

IDS are pivotal in network security and are classified primarily into two types: signature-based IDS (SIDS) and anomaly-based IDS (AIDS) [150]. SIDSs are highly effective in recognizing known attacks leveraging predefined patterns. However, they fall short in identifying novel, unseen threats. AIDSs, conversely, offer a solution to this limitation by focusing on deviations from normative behavior, making them apt for detecting zero-day attacks. Zero-day attacks, such as the infamous Stuxnet worm [151] that targeted industrial control systems, are cyber threats that exploit previously unknown software or hardware vulnerabilities before developers can issue a fix or patch. Consequently, AIDS has seen increasing focus in recent literature, particularly those employing DRL.

Among the notable works, Benaddi et al. [142] proposed a DRL-IDS specifically for the IoT and wireless sensor networks (WSN), utilizing a DQL model. Their system demonstrated remarkable detection rates and low false negatives when benchmarked against standard RL and KNN using the NSL-KDD dataset.

Similarly, Hsu and Matsuoka [143] designed a DRL-based IDS aligned with Benaddi et al.'s approach. Still, they extended their comparison to include machine learning models like RF and SVM, using both NSL-KDD and UNSW-NB15 datasets. Their results indicated superior accuracy of the DRL-based IDS in detecting intrusions across these datasets.

Expanding on these foundations, Alavizadeh et al. [152] developed a DQL-based IDS, focusing on efficiency with a training duration of just 250 episodes. The model achieved a notable accuracy rate, highlighting the effectiveness of DRL in IDS.

Ren et al. [153] introduced an innovative approach with their ID-RDRL, a NIDS that combines DQL with recursive feature elimination (RFE) to reduce redundant features significantly. This strategy not only streamlined the feature set but also enhanced the NIDS's performance, as evidenced by their accuracy and F_1 scores of 96.20% and 94.90%, respectively, on the CSE-CIC-IDS2018 dataset.

In the context of 5G networks, Moudoud and Cherkaoui [145] constructed a DRL-based IDS to enhance the security of IoT systems. Utilizing a DQL model and training on the NSL-KDD dataset, their system showed promising results, achieving a high F_1 score of 94% with limited training episodes.

Our research diverges from these existing studies by introducing a unique dataset tailored for 5G networks with MEC services, employing hybrid data sources from both network and host, and leveraging an actor-critic DRL method. This approach combines the advantages of both NIDS and HIDS, making it a novel contribution to the field of anomaly-based, DRL-based IDS. To the best of our knowledge, this is the first instance of an IDS that integrates these specific features for enhanced intrusion detection in 5G networks. Additionally, it marks the first dataset generated by an emulated 5G testbed with MEC services.

5.4 HDLR Testbed

HDRL consists of IDS distributed components in the edge clouds (i.e., MEC) of the 5G network to provide IDS for both direct and indirect users. In other words, an HDRL component is integrated

into every edge cloud, allowing for the dynamic updating of IDS models that combine the benefits of using host data and network-based IDS in each edge cloud. Furthermore, using an IDS in the edge clouds eliminates the single point of failure issues in centralized IDSs, minimizes the effects of an attack on the whole system, and increases the system's resilience.

The IDS can be enhanced and adapted to different types of attacks (including zero-day attacks) with the help of DRL algorithms with two innovations. DRL enhances and adapts IDS for new attacks by learning and updating policies from environmental interactions. Hence, it enables continuous adaptation to emerging threats without the extensive retraining required by other ML and DNN methods. First, to avoid the drawbacks of value-based and policy-based algorithms [149], we use one of the actor-critic algorithms, DDPG. Second, to take advantage of both types of IDS, we create a hybrid DRL model in the MECs. We combine the advantages of host data and network-based IDS.

We have developed a testbed to emulate user and attacker interactions to the 5G MEC and core. First, the edge clouds monitor and collect the network metadata for the links between the edge clouds and the edge nodes. Next, the UEs send the host metadata (i.e., CPU usage and running processes) to the edge clouds as part of the transmitted data used by the hybrid DRL model. Finally, our algorithm executes the IDS procedures, processing the network and host metadata in combination as a hybrid approach.

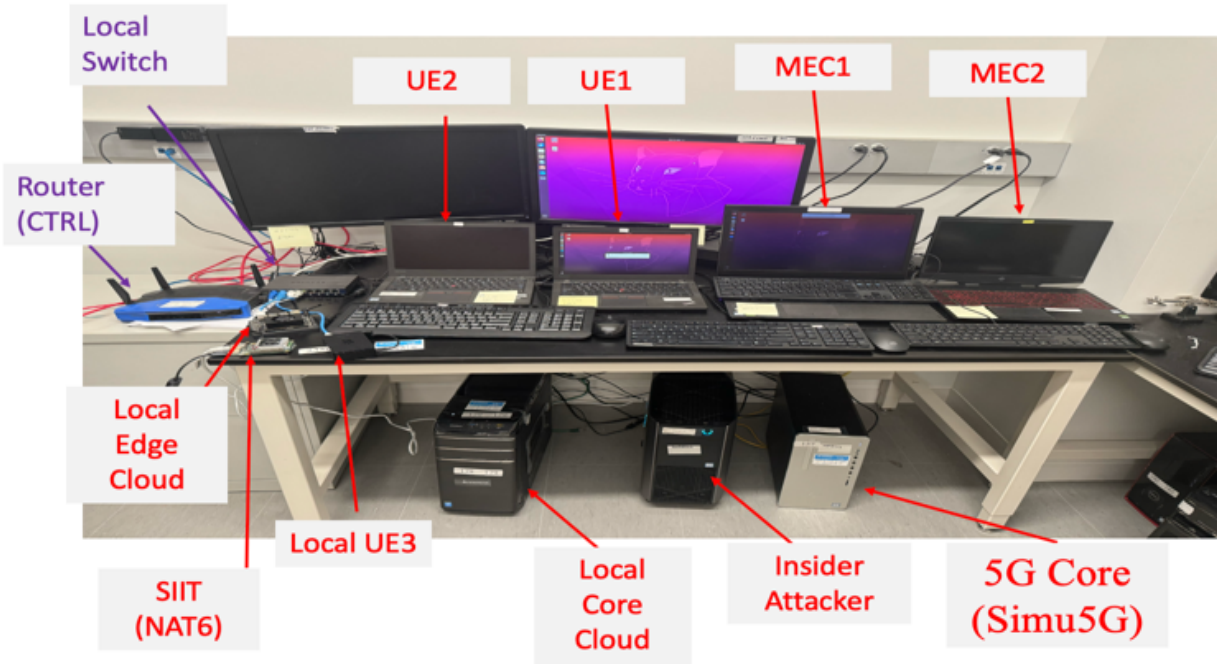


Figure 5.3 Our testbed.

5.4.1 Testbed Architecture

The testbed comprises six main components, as depicted in Figure 5.3, based on the structure in Figure 5.2: a 5G core, a local network, two MECs, three UEs, an insider attacker, and a router.

1) 5G Core:

At the core of the testbed, the Ubuntu 20.04 computer emulates the behaviors of the 5G infrastructure using Simu5G. The emulation setup is based on the "extUeAppMecApp" instance from Simu5G v1.2.2, initially facilitating the emulation of one MEC and one UE. We have extended the system to include multiple MECs and external hosts, providing a more complex and representative emulation of a 5G network. The core host is the command center for all communications within the testbed's 5G infrastructure.

2) Local Network:

This component replicates a typical local network linked to the broader 5G network. It includes edge clouds with a local core, a local switch, and a stateless IP/ICMP translation (SIIT) system implemented using the Jool package [154]. The SIIT component, particularly crucial, addresses the network address translation (NAT) issues, enabling local devices to use IPv6 for individual identification within OMNET-emulated 5G networks, which predominantly utilize IPv4. Given the current limitations of OMNet++ version 6.0 in terms of IPv6 support, this solution is a workaround, which is necessary for complete compatibility with Simu5G.

3) MECs:

Running on Ubuntu 20.04 OS, the PCs hosting the MEC services are integral to the testbed. Here, the HDRL IDS model is stationed, monitoring network data via Argus [91] and gathering host data using the psutil library [155]. The collected data is crucial for functioning the hybrid IDS model within the testbed.

4) UEs:

UE1 and UE2, based on Ubuntu 20.04 OS, are directly connected to the 5G network. In contrast, UE3, operating on Raspberry Pi OS, is situated within the local network, representing a diverse range of end-user scenarios.

5) Insider Attacker:

This Ubuntu 20.04 computer is configured to generate various attacks, including MitM, DDoS, ransomware, and buffer overflow attacks, as elaborated in Section 5.3.2. This component is essential for testing the resilience and effectiveness of the IDS against a range of security threats.

6) **Router:**

The router's role is to manage the entire testbed from a central control PC. Its function is critical in ensuring unbiased data collection for the new dataset. Randomizing the timing and duration of attacks helps create a diverse and unpredictable set of scenarios for robust IDS testing.

Each component creates a dynamic and realistic 5G network environment, enabling thorough testing and evaluation of the proposed HDRL IDS model in various scenarios and attack conditions.

5.4.2 Types of Attacks

In this dataset, four types of attacks are implemented, offering a broad spectrum of applications in various domains. All attacks are developed using Python, except for the buffer overflow attack, executed using a Bash script. The descriptions of these attacks are as follows:

1) **MitM:**

This attack involves data alteration, where host data sent from a UE to a MEC server is altered. It primarily targets UE1 and UE3. The Scapy [156] and NetfilterQueue [157] libraries intercept and modify the packets before redirecting them to the MEC server.

2) **DDoS:**

Executed by multiple compromised PCs, this attack floods MEC1 and MEC2 with overwhelming requests, disrupting services for regular users like UE1, UE2, and UE3. The attacker's PC simulates a DDoS attack from multiple devices by generating numerous sessions with unique IPs. The Scapy library is again used to create and dispatch these packets.

3) **Ransomware:**

In this scenario, files on UE1's PC are encrypted by an attacker, who then demands payment for decryption. The attack encrypts a folder using the cryptography library [158] and follows and implements a Python-based ransomware attack outlined in [159].

4) Buffer overflow:

This attack manipulates a program's execution flow to execute arbitrary code, potentially damaging the UE's OS. The method for this attack is based on instructions from [160].

5.4.3 5G Edge Threat Model

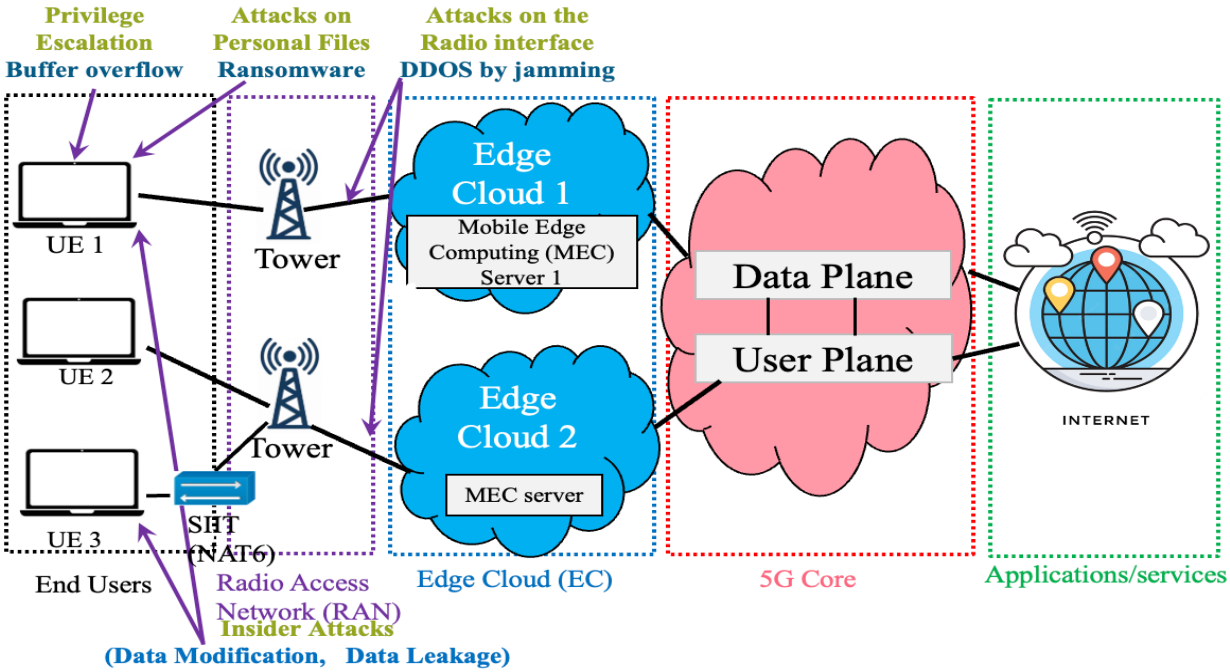


Figure 5.4 5G edge threat model

1) Surface and Vector:

- **Network Communication (MitM Attacks):** Data between UE and MEC servers in the 5G network is vulnerable to interception and alteration.
- **Server Availability (DDoS Attacks):** MEC servers can be overwhelmed by excessive traffic originating from within the network, disrupting services.

- **Endpoint Security (Ransomware Attacks):** UE operating in the 5G network is susceptible to ransomware attacks that encrypt data, demanding a ransom for decryption.

- **Application and System Integrity (Buffer Overflow Attacks):** Applications and operating systems on UE are at risk from buffer overflow vulnerabilities, which allow arbitrary code execution.

2) Threat Agents:

- **Insider Attacker:**

  a) Has authorized access and potentially extensive knowledge of the 5G network and associated systems.

  b) Capable of intercepting and altering data in transit between UEs and MEC servers (MitM).

  c) Can initiate DDoS attacks by manipulating network traffic to overload MEC servers.

  d) May deploy ransomware on vulnerable UEs within the network.

  e) Could exploit system vulnerabilities to execute buffer overflow attacks on applications and operating systems of UEs.

2) Potential Impacts:

- **Data Integrity and Confidentiality:** Compromised in MitM attacks, leading to potential information leakage or corruption.

- **Service Availability:** Disrupted during DDoS attacks, affecting the normal functioning of network services and impacting user access.

- **Data Availability and Financial Loss:** Threatened by ransomware attacks, which can lead to significant operational and financial consequences.

- **System and Data Security:** Undermined by buffer overflow attacks, which can result in unauthorized access or damage to the system.

3) **Security Measures**:

- **Encryption and Secure Communication Protocols:** To safeguard against MitM attacks by ensuring that data in transit is secure and cannot be easily intercepted or altered.

- **Traffic Monitoring and Management:** To detect and mitigate DDoS attacks by identifying abnormal traffic patterns and implementing rate limiting or blocking of malicious sources.

- **Endpoint Protection and Backup Strategies:** To prevent ransomware infections through antivirus defenses and regular backups that can restore encrypted data without paying a ransom.

- **Software Development Best Practices:** To avoid buffer overflow vulnerabilities by using secure coding techniques and regular security testing of applications.

These security measures have been discussed in detail in Chapter 2 of the dissertation, providing a comprehensive strategy for mitigating the identified risks.

5.4.4 5G Dataset

The datasets collected from MEC1 and MEC2 using our 5G testbed have been combined for ease of result analysis and feature comparison. The dataset comprises 77 features, categorized as follows: 35 network features as defined by Hady et al. [6], 40 host features (detailed in Table 5.1), and two label features (one for multi-label classification and the other for binary classification). Overall, the dataset includes 145k samples, consisting of 132k normal samples and 13k attack samples [161].

Table 5.1 Dataset host features.

Metric	Description
IMEI	International Mobile Equipment Identity
RTime	Packet sending time
Packet_num	Packet number
scputimes_user	Normal processes' CPU time
scputimes_nice	Prioritized processes' CPU time
scputimes_system	Kernel processes' CPU time
scputimes_idle	Idle CPU time
scputimes_iowait	CPU time waiting for I/O
scputimes_irq	Hardware interrupts CPU time
scputimes_softirq	Software interrupts CPU time
scputimes_steal	Virtualized CPU time
scputimes_guest	Normal virtual CPU time
scputimes_guest_nice	Prioritized virtual CPU time
scpustats_ctx_switches	Number of context switches
scpustats_interrupts	Number of interrupts
scpustats_soft_interrupts	Number of software interrupts
scpustats_syscalls	Number of system calls
svmem_total	Total physical memory size
svmem_available	Available memory size
svmem_percent	Percentage of used memory
svmem_used	Broadly consumed memory
svmem_free	Free memory size

svmem_active	Currently in-use memory
svmem_inactive	Currently unused memory
svmem_buffers	System metadata cache
svmem_cached	Other things cache
svmem_shared	Shared memory
svmem_slab	In-kernel data structures cache
ram_usage_warning	RAM usage warnings
sswap_total	Total swap memory in bytes
sswap_used	Used swap memory in bytes
sswap_free	Free swap memory in bytes
sswap_percent	Swap memory percentage usage
sswap_sin	Number of bytes swapped in from disk
sswap_sout	Number of bytes swapped out from disk
sdiskusage_total	Total disk size
sdiskusage_used	Used disk size
sdiskusage_free	Free disk size
sdiskusage_percent	Percent disk used
Boot_Time_with_date	Boot time and date
DTime	Packet delivery delta time

5.4.5 DRL Model

The DRL model implements the DDPG algorithm designed for continuous action spaces. This model was developed using the Keras library from the Tensorflow package [162]. It comprises two main components: the actor and the critic. The actor is tasked with selecting actions, while the critic evaluates these actions by estimating their value functions. Details of the actor and critic models, including their hyperparameters, are presented in Table 5.2. The architecture includes three hidden layers: the first consists of 32 gated recurrent unit (GRU) neurons, while the other two layers each have 48 fully connected neurons, following the setup outlined in [163]. An epsilon-greedy strategy is employed to adjust the action policies. Additionally, the learning rate dynamically changes based on the rewards received during the model's training phase.

Table 5.2 DRL model hyperparameter.

Parameter	Typical Value(s)
Number of layers	4
Number of neurons per layer	32, 48, 48, 1
Number of episodes	10
Actor's initial learning rate	0.0001
Critic's initial learning rate	0.001
Discount factor	0.99
epsilon	0.1
Anomaly rate range	(-0.1, 0.0)
Optimizer	*Adam*
Activation function	*relu, tanh*

5.5 Experimental Results

In this section, we discuss the analysis and evaluation of the dataset using the newly developed HDRL model. First, we explain the preprocessing steps applied to the dataset. Subsequently, we present the initial results obtained from the HDRL model, utilizing the collected dataset. These results include a comprehensive comparison between the usage of hybrid features, network-only features, and host-only features.

5.5.1 Dataset Preprocessing

Preprocessing the dataset is crucial for any ML method, including DRL models, to ensure accurate results. The preprocessing steps consist of dataset splitting, data encoding, normalization, and resampling, as follows:

1) Dataset splitting:

The dataset is divided into three parts for training and evaluating the DRL model: 75% for training (10% of this portion is allocated for validation) and 25% for testing.

2) Data encoding and normalization:

Categorical features are encoded, and the dataset is scaled using the label encoder and Standard Scaler from the scikit-learn package [164], respectively.

3) Divide the features based on their types:

We have prepared three separate datasets to compare hybrid, network-only, and host-only features. The first two consist of a single feature type, while the third contains both types. We then run the same model on these datasets and present their results in Section 5.5.2.

5.5.2 Results

We compared the hybrid HDRL model to those using only network or host features. This comparison utilized two different metrics: the F_1 score and accuracy. The F_1 score, a harmonic mean of precision and recall, is widely used in IDS results. Accuracy is defined as the ratio of correct predictions to total predictions. As these are initial results, the anomaly rate range (-0.1, 0.0) was determined by taking the average action values from the training samples and adjusting it by $\pm$ 0.05 based on the average action values in each training episode.

Figure 4 shows the accuracy scores for all three types of features. As shown in the figure, the HDRL model achieved 58.17%. In contrast, the features from the net-only dataset achieved 91.76%, compared to a mere 24.99% for the host-only features. Despite the net-only model's high accuracy score, these results underscore that accuracy alone may not be the most reliable measure of IDS effectiveness. This is because the net-only IDS failed to identify nearly all attack samples, rendering it as effective as a model with a 0% detection rate, which still achieves a 91.42% accuracy score. However, combining both yields significantly better results than using either type individually.

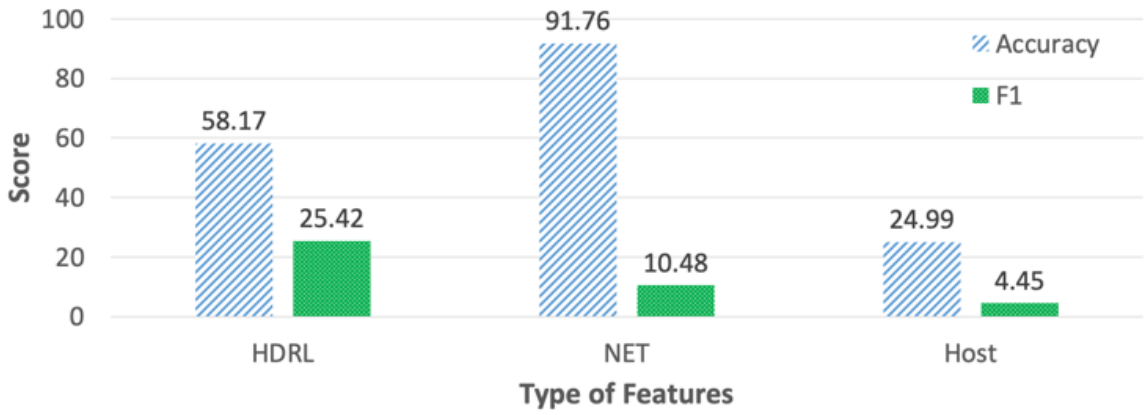


Figure 5.5 Accuracy and F_1 scores comparison.

Similarly, Figure 5.4 presents the F_1 scores for all three types of features. Given that the F_1 score is more suitable for security applications, a clear improvement in IDS performance is observed when combining network and host features. The HDRL method's score was more than two times higher than the other two feature types. Nonetheless, based on predefined parameters from other research, these initial results may not fully showcase the HDRL model's potential. Future work would explore the impact of training episode numbers and the use of feature-reduction methods to decrease training time.

5.6 Summary

With advancements in cybersecurity and next-generation infrastructures like 5G, there is a growing need for dynamic IDS that can adapt quickly to evolving attack patterns with minimal human intervention. This chapter introduces a novel dataset for IoMT systems, created using an emulated 5G network that includes direct and indirect end users and MEC services. This dataset, a first of its kind, addresses the gap in 5G IDS datasets, facilitating the detection of zero-day attacks. We collected 145k samples, comprising normal samples and four types of attacks: MitM, DDoS, ransomware, and buffer overflow. A hybrid DRL model for IDS, combining the strengths of Network IDS and Host IDS, was developed to evaluate this dataset. The analysis demonstrates the benefits

of combining features compared to using only network or host features. Our results show that the HDRL model outperforms the F_1 score and accuracy metrics by 25.42% and 58.17%, respectively. Future work will focus on increasing the training episodes for the HDRL model to optimize IDS performance further.

Chapter 6: Conclusion and Future Work

As this dissertation concludes, it is crucial to reflect on the progress made in enhancing the security of IoMT systems. Integrating IoMT technologies into healthcare has ushered in transformative improvements in patient monitoring and data management but has also introduced significant cybersecurity challenges. These challenges are pivotal as they threaten healthcare services' integrity and operational continuity. Addressing these challenges was the core focus of this dissertation, aiming to fortify the security frameworks that protect sensitive medical data and ensure the reliability of healthcare operations.

The research began with a thorough review of existing IoMT security measures, which are essential for identifying the current vulnerabilities and setting the stage for subsequent developments. This foundational work was critical in shaping the research direction, highlighting the need for innovative solutions that could adapt to the evolving landscape of cyber threats.

The introduction of the EHMS testbed marked a key phase in the research, serving as a practical platform for testing security measures and as a source of valuable data on IoMT security dynamics. This testbed was instrumental in understanding how various security strategies performed under real-world conditions.

Central to addressing the cybersecurity challenges was exploring how advanced technologies, like machine learning and 5G networks, could be leveraged to improve the security of IoMT systems. The research investigated how these technologies could be integrated into existing healthcare infrastructures to enhance threat detection and response capabilities, thus safeguarding against known and emerging cyber threats.

As the dissertation shifts focus from the detailed contributions to the broader challenges and future directions, it is evident that the journey to secure IoMT systems is ongoing. The subsequent sections will explore the outcomes achieved and outline future research avenues to ensure continuous improvement in IoMT security as healthcare technology evolves. This reflection sets the stage for a deeper discussion on "What Have We Achieved in This Dissertation?" and the "Future Work" necessary to build on the foundational efforts laid out in this research.

6.1 What Have We Achieved in This Dissertation?

Integrating IoT technologies in healthcare, particularly through the IoMT, has presented unparalleled opportunities for enhancing patient care and reducing healthcare expenditures. However, the proliferation of these technologies also brings forth significant cybersecurity challenges that must be addressed to safeguard sensitive medical data and ensure the reliability of medical services. This dissertation has developed a robust framework that leverages advanced security techniques to protect IoMT devices from emerging threats, ensuring the integrity, confidentiality, and availability of healthcare data across various stages, from data collection to storage and sharing.

In Chapter 2, the dissertation proposed a security framework that integrates various state-of-the-art techniques to comprehensively fulfill the security requirements of IoMT systems. This holistic approach is crucial for ensuring that all aspects of data and device security are covered, addressing the vulnerabilities inherent in the current systems that single methods cannot mitigate.

Chapter 3 demonstrated the importance of IDS in remote healthcare monitoring systems, particularly in ensuring data integrity and confidentiality. By designing the EHMS testbed, this research provided a realistic dataset and highlighted the effectiveness of combining network flow metrics with biometric data to enhance IDS performance. Despite the successes, there remains room for

optimization, particularly in fine-tuning the machine learning models to improve their efficacy further.

Chapter 4 introduced a novel feature reduction method, LEMDA, which significantly enhances the performance of IDS models by focusing on the most informative features. This approach not only improves the accuracy of the models but also reduces the training and detection times, making the systems more efficient. Future work will explore the application of LEMDA in other domains beyond IoT, potentially broadening its applicability and effectiveness.

Chapter 5 explored the dynamic capabilities of IDS within the context of advanced 5G infrastructures, presenting a novel dataset specifically designed for IoMT systems in a 5G environment. Developing an HDRL model demonstrated superior performance in detecting threats, particularly zero-day attacks, which are notoriously difficult to identify with traditional methods.

In conclusion, the research conducted throughout this dissertation has systematically addressed several key facets of IoMT security within 5G environments, ultimately enhancing the robustness and responsiveness of IDS. By integrating a novel dataset tailored for IoMT systems and leveraging the capabilities of advanced 5G infrastructure, we have established a comprehensive approach that mitigates risks and enhances operational efficacy. The development of the HDRL model represents a significant breakthrough in the field, showcasing the ability to effectively detect complex threats like zero-day attacks, which pose immense risks due to their unpredictability and novelty. This achievement marks a substantial advancement in IoMT security, setting a new standard for future developments in the field and underscoring the critical importance of adaptive, intelligent cybersecurity measures in modern healthcare environments.

6.2 Future Work

Looking ahead, the dissertation sets the stage for several promising research areas. Firstly, there is an ongoing need to enhance the machine learning models used in IDS, particularly through hyperparameter optimization and advanced attack simulations. This effort will help refine the models to handle more sophisticated threats and reduce false positives, which is critical for maintaining trust in automated healthcare systems.

Secondly, applying the LEMDA method will be extended beyond IoMT to see if its benefits hold in other contexts, such as industrial IoT or consumer IoT environments. This exploration will help in understanding the method's versatility and limitations.

Finally, for the 5G-enabled IDS models, future research will focus on extending the training episodes and incorporating more diverse attack scenarios to improve the models' robustness and adaptability continuously. Additionally, as 5G technology evolves, so will the requirements for IDS in these environments, necessitating ongoing updates and innovations in the developed models.

By addressing these areas, future research based on this dissertation's findings will continue to advance the field of cybersecurity in IoMT, ensuring that healthcare systems are efficient and cost-effective but also secure and reliable in the face of an ever-evolving cyber threat landscape.

References

[1] B. Marr. "2024 IoT And Smart Device Trends: What You Need To Know For The Future." [Online]. Available: https://www.forbes.com/sites/bernardmarr/2023/10/19/2024-iot-and-smart-device-trends-what-you-need-to-know-for-the-future/?sh=18cac8ae7f34 [Accessed: April 17, 2024].

[2] Grand View Research. "Internet Of Things In Healthcare Market To Reach $169.99 Billion By 2030." [Online]. Available: https://www.grandviewresearch.com/press-release/global-iot-in-healthcare-market [Accessed: April 18, 2024].

[3] IBM Security, "Cost of a Data Breach Report 2023," 2023. [Online]. Available: https://www.ibm.com/reports/data-breach?utm_content=SRCWW&p1=Search&p4=43700077724063991&p5=e&p9=587000085236194 12&gclid=CjwKCAjwouexBhAuEiwAtW_Zx92teKfFKXuBhdzKSqhb16G5oHVEmxwZDYv1VTQ9sbrJ2jjnR8Q7ExoCf0IQAvD_BwE&gclsrc=aw.ds

[4] A. Jay. "Number of Internet of Things (IoT) Connected Devices Worldwide 2024: Breakdowns, Growth & Predictions." [Online]. Available: https://financesonline.com/number-of-internet-of-things-connected-devices/ [Accessed: April 18, 2024].

[5] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the internet-of-medical-things (IoMT) systems security," *IEEE Internet of Things Journal,* vol. 8, no. 11, pp. 8707-8718, 2020.

[6] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study," *IEEE Access,* vol. 8, pp. 106576-106584, 2020.

[7] A. Ghubaish, Z. Yang, A. Erbad, and R. Jain, "LEMDA: A Novel Feature Engineering Method for Intrusion Detection in IoT Systems," *IEEE Internet of Things Journal,* 2023.

[8] A. Ghubaish, Z. Yang, and R. Jain, "HDRL-IDS: A Hybrid Deep Reinforcement Learning Intrusion Detection System for Enhancing the Security of Medical Applications in 5G Networks," in *2024 IEEE International Conference on Smart Applications, Communications and Networking (SmartNets)*, Harrisonburg/Washington DC, VA, USA, 2024.

[9] CyberMDX. "2020 Vision: A Review of Major IT & Cyber Security Issues Affecting Healthcare." [Online]. Available: https://www.healthcareinfosecurity.com/whitepapers/2020-vision-review-major-cybersecurity-issues-affecting-healthcare-w-6017 [Accessed: May 1, 2024].

[10] W. MADDOX. "Why Medical Data is 50 Times More Valuable Than a Credit Card." [Online]. Available: https://www.dmagazine.com/healthcare-business/2019/10/why-medical-data-is-50-times-more-valuable-than-a-credit-card/ [Accessed: May 1, 2024].

[11] United States Naval Academy. "Information Assurance." [Online]. Available: https://www.usna.edu/Users/cs/wcbrown/courses/si110AY13S/lec/l21/lec.html [Accessed: May 1, 2024].

[12] J.-P. A. Yaacoub, M. Noura, H. N. Noura, O. Salman, E. Yaacoub, R. Couturier, and A. Chehaba, "Securing Internet of Medical Things Systems: Limitations, Issues and Recommendations," *Future Generation Computer Systems,* vol. 105, pp. 581-606, 2020.

[13] A. Vyas and S. Pal, "Preventing Security and Privacy Attacks in WBAN," in *Handbook of Computer Networks and Cyber Security*, B. Gupta, G. Perez, D. Agrawal, and D. Gupta Eds.: Springer, 2020, pp. 201-225.

[14] D. Bhushan and R. Agrawal, "Security Challenges for Designing Wearable and IoT Solutions," in *A Handbook of Internet of Things in Biomedical and Cyber Physical System*, vol. 165, V. Balas, V. Solanki, R. Kumar, and M. Ahad Eds. Intelligent Systems Reference Library: Springer, 2020, pp. 109-138.

[15] F. Pesapane, M. B. Suter, M. Codari, F. Patella, C. Volonté, and F. Sardanelli, "Chapter 52 - Regulatory issues for artificial intelligence in radiology," in *Precision Medicine for Investigators, Practitioners and Providers*, J. Faintuch and S. Faintuch Eds.: Academic Press, 2020, pp. 533-543.

[16] J. Sengupta, S. Ruj, and S. D. Bit, "A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT," *Journal of Network and Computer Applications,* vol. 149, pp. 1-20, 2020.

[17] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *Journal of Network and Computer Applications,* vol. 161, pp. 1-18, 2020.

[18] L. Xiao, Y. Li, G. Han, G. Liu, and W. Zhuang, "PHY-Layer Spoofing Detection With Reinforcement Learning in Wireless Networks," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 12, pp. 10037-10047, 2016.

[19] Wikipedia. "Implant (medicine)." [Online]. Available: https://en.wikipedia.org/wiki/Implant_(medicine) [Accessed: May 1, 2024].

[20] J. E. Ferguson and A. D. Redish, "Wireless communication with implanted medical devices using the conductive properties of the body," (in eng), *Expert Rev Med Devices,* vol. 8, no. 4, pp. 427-433, 2011.

[21] Mayo Clinic. "Pacemaker." [Online]. Available: https://www.mayoclinic.org/tests-procedures/pacemaker/about/pac-20384689 [Accessed: May 1, 2024].

[22] A. Phaneuf. "Latest trends in medical monitoring devices and wearable health technology." [Online]. Available: https://www.businessinsider.in/science/news/latest-trends-in-medical-monitoring-devices-and-wearable-health-technology/articleshow/71970305.cms [Accessed: May 1, 2024].

[23] Apple. "Heart health notifications on your Apple Watch." [Online]. Available: https://support.apple.com/en-us/HT208931 [Accessed: May 1, 2024].

[24] A. Kos, V. Milutinović, and A. Umek, "Challenges in wireless communication for connected sensors and wearable devices used in sport biofeedback applications," *Future Generation Computer Systems,* vol. 92, pp. 582-592, 2019.

[25] H. Jahankhani and J. Ibarra, "Digital Forensic Investigation for the Internet of Medical Things (IoMT)," *Journal of Forensic Legal & Investigative Sciences,* vol. 5, no. 029, 2019.

[26] Medtronic. "Pacing Systems - Azure | Medtronic." [Online]. Available: https://europe.medtronic.com/xd-en/healthcare-professionals/products/cardiac-rhythm/pacemakers/azure.html [Accessed: May 1, 2024].

[27] Federal Communications Commission (FCC). "Medical Device Radiocommunications Service (MedRadio)." [Online]. Available: https://www.fcc.gov/medical-device-radiocommunications-service-medradio [Accessed: May 1, 2024].

[28] P. Kasyoka, M. Kimwele, and S. Mbandu Angolo, "Certificateless pairing-free authentication scheme for wireless body area network in healthcare management system," *Journal of Medical Engineering & Technology,* vol. 44, no. 1, pp. 12-19, 2020.

[29] T. Belkhouja, S. Sorour, and M. S. Hefeida, "Role-Based Hierarchical Medical Data Encryption for Implantable Medical Devices," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 9-13 Dec. 2019 2019, pp. 1-6.

[30] Wikipedia. "Chinese remainder theorem." [Online]. Available: https://en.wikipedia.org/wiki/Chinese_remainder_theorem [Accessed: May 1, 2024].

[31] Wikipedia. "Received signal strength indication." [Online]. Available: https://en.wikipedia.org/wiki/Received_signal_strength_indication [Accessed: May 1, 2024].

[32] M. F. Awan, K. Kansanen, S. Perez-Simbor, C. Garcia-Pardo, S. Castelló-Palacios, and N. Cardona, "RSS-Based Secret Key Generation in Wireless In-body Networks," in *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2019, pp. 1-6.

[33] Wikipedia. "Cryptographic hash function." [Online]. Available: https://en.wikipedia.org/wiki/Cryptographic_hash_function [Accessed: May 1, 2024].

[34] Wikipedia. "XOR gate." [Online]. Available: https://en.wikipedia.org/wiki/XOR_gate [Accessed: May 1, 2024].

[35] B. A. Alzahrani, A. Irshad, A. Albeshri, and K. Alsubhi, "A Provably Secure and Lightweight Patient-Healthcare Authentication Protocol in Wireless Body Area Networks," *Wireless Personal Communications,* pp. 1-23, 2020.

[36] Z. Xu, C. Xu, W. Liang, J. Xu, and H. Chen, "A Lightweight Mutual Authentication and Key Agreement Scheme for Medical Internet of Things," *IEEE Access,* vol. 7, pp. 53922-53931, 2019.

[37] Y. Sun and B. Lo, "An Artificial Neural Network Framework for Gait-Based Biometrics," *IEEE Journal of Biomedical and Health Informatics,* vol. 23, no. 3, pp. 987-998, 2019.

[38] V. H. Tutari, B. Das, and D. R. Chowdhury, "A Continuous Role-Based Authentication Scheme and Data Transmission Protocol for Implantable Medical Devices," in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, 25-28 Feb. 2019 2019, pp. 1-6.

[39] S. Maji, U. Banerjee, S. H. Fuller, M. R. Abdelhamid, P. M. Nadeau, R. T. Yazicigil, and A. P. Chandrakasan, "A Low-Power Dual-Factor Authentication Unit for Secure Implantable Devices," in *2020 IEEE Custom Integrated Circuits Conference (CICC)*, 2020, pp. 1-4.

[40] Wikipedia. "RSA (cryptosystem)." [Online]. Available: https://en.wikipedia.org/wiki/RSA_(cryptosystem) [Accessed: May 1, 2024].

[41] Wikipedia. "Elliptic-curve cryptography." [Online]. Available: https://en.wikipedia.org/wiki/Elliptic-curve_cryptography [Accessed: May 1, 2024].

[42] V. J. Jariwala and D. C. Jinwala, "Chapter 4 - AdaptableSDA: secure data aggregation framework in wireless body area networks," in *Wearable and Implantable Medical Devices*, vol. 7, N. Dey, A. S. Ashour, S. James Fong, and C. Bhatt Eds.: Academic Press, 2020, pp. 79-114.

[43] T. Bhatia, A. K. Verma, and G. Sharma, "Towards a secure incremental proxy re-encryption for e-healthcare data sharing in mobile cloud computing," *Concurrency and Computation: Practice and Experience,* vol. 32, no. 5, pp. 1-16, 2020.

[44] Wikipedia. "Zettabyte." [Online]. Available: https://en.wikipedia.org/wiki/Zettabyte [Accessed: May 1, 2024].

[45] Wikipedia. "Homomorphic encryption." [Online]. Available: https://en.wikipedia.org/wiki/Homomorphic_encryption [Accessed: May 1, 2024].

[46] G. Kalyani and S. Chaudhari, "An efficient approach for enhancing security in Internet of Things using the optimum authentication key," *International Journal of Computers and Applications,* vol. 42, no. 3, pp. 306-314, 2020.

[47] Wikipedia. "Digital signature." [Online]. Available: https://en.wikipedia.org/wiki/Digital_signature [Accessed: May 1, 2024].

[48] C. Easttom and N. Mei, "Mitigating Implanted Medical Device Cybersecurity Risks," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2019, pp. 0145-0148.

[49] A. Kumari, S. Jangirala, M. Y. Abbasi, V. Kumar, and M. Alam, "ESEAP: ECC based secure and efficient mutual authentication protocol using smart card," *Journal of Information Security and Applications,* vol. 51, pp. 1-12, 2020.

[50] G. Zheng, W. Yang, C. Valli, L. Qiao, R. Shankaran, M. A. Orgun, and S. C. Mukhopadhyay, "Finger-to-Heart (F2H): Authentication for Wireless Implantable Medical Devices," *IEEE Journal of Biomedical and Health Informatics,* vol. 23, no. 4, pp. 1546-1557, 2019.

[51] G. Zheng, W. Yang, M. Johnstone, R. Shankaran, and C. Valli, "3 - Securing the elderly in cyberspace with fingerprints," in *Assistive Technology for the Elderly*, N. K. Suryadevara and S. C. Mukhopadhyay Eds.: Academic Press, 2020, pp. 59-79.

[52] N. Ibtihel and S. M. Hadj, "Smart ECG Monitoring Through IoT," in *Smart Medical Data Sensing and IoT Systems Design in Healthcare*, C. Chinmay Ed. Hershey, PA, USA: IGI Global, 2020, pp. 224-246.

[53] Ubidots. "Industrial IoT Platform for

Energy Management." [Online]. Available: https://ubidots.com [Accessed: May 1, 2024].

[54] W. Youssef, A. O. Zaid, M. S. Mourali, and M. H. Kammoun, "RFID-based System for Secure Logistic Management of Implantable Medical Devices in Tunisian Health Centres," in *2019 IEEE International Smart Cities Conference (ISC2)*, 2019, pp. 83-86.

[55] S. Kulaç, "A New Externally Worn Proxy-Based Protector for Non-Secure Wireless Implantable Medical Devices: Security Jacket," *IEEE Access,* vol. 7, pp. 55358-55366, 2019.

[56] S. Kulaç, "Security Belt for Wireless Implantable Medical Devices," *Journal of Medical Systems,* vol. 41, no. 11, pp. 1-9, 2017.

[57] A. Mosaif and S. Rakrak, "A Li-Fi based wireless system for surveillance in hospitals," *Biomedical Spectroscopy and Imaging,* vol. 8, pp. 81-92, 2019.

[58] S. Saif, S. Biswas, and S. Chattopadhyay, "Intelligent, Secure Big Health Data Management Using Deep Learning and Blockchain Technology: An Overview," in *Deep Learning Techniques for Biomedical and Health Informatics*, S. Dash, B. R. Acharya, M. Mittal, A. Abraham, and A. Kelemen Eds. Cham: Springer International Publishing, 2020, pp. 187-209.

[59] X. Chen, H. Zhu, D. Geng, W. Liu, R. Yang, and S. Li, "Merging RFID and Blockchain Technologies to Accelerate Big Data Medical Research Based on Physiological Signals," *Journal of Healthcare Engineering,* vol. 2020, pp. 1-17, 2020.

[60] V. Manjula and R. Thalapathi Rajasekaran, "Security Vulnerabilities in Traditional Wireless Sensor Networks by an Intern in IoT, Blockchain Technology for Data Sharing in IoT," in *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*, S.-L. Peng, S. Pal, and L. Huang Eds. Cham: Springer International Publishing, 2020, pp. 579-597.

[61] L. Gupta, T. Salman, M. Zolanvari, A. Erbad, and R. Jain, "Fault and performance management in multi-cloud virtual network services using AI: A tutorial and a case study," *Computer Networks,* vol. 165, 2019, Art no. 106950.

[62] S. C. Sethuraman, V. Vijayakumar, and S. Walczak, "Cyber Attacks on Healthcare Devices Using Unmanned Aerial Vehicles," *Journal of Medical Systems,* vol. 44, no. 1, pp. 1-10, 2019.

[63] Wikipedia. "IP fragmentation attack." [Online]. Available: https://en.wikipedia.org/wiki/IP_fragmentation_attack [Accessed: May 1, 2024].

[64] O. Shwartz, Y. Mathov, M. Bohadana, Y. Elovici, and Y. Oren, "Opening Pandora's Box: Effective Techniques for Reverse Engineering IoT Devices," Cham, 2018: Springer International Publishing, in Smart Card Research and Advanced Applications, pp. 1-21.

[65] Y. Jiang, Y. Shen, and Q. Zhu, "A Lightweight Key Agreement Protocol Based on Chinese Remainder Theorem and ECDH for Smart Homes," (in eng), *Sensors (Basel),* vol. 20, no. 5, pp. 1-13, 2020.

[66] S. Almajali, H. B. Salameh, M. Ayyash, and H. Elgala, "A framework for efficient and secured mobility of IoT devices in mobile edge computing," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, 23-26 April 2018 2018, pp. 58-62.

[67] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, "The Role of Edge Computing in Internet of Things," *IEEE Communications Magazine,* vol. 56, no. 11, pp. 110-115, 2018.

[68] Wikipedia. "Constrained Application Protocol." [Online]. Available: https://en.wikipedia.org/wiki/Constrained_Application_Protocol [Accessed: May 1, 2024].

[69] J. A. Salowey, S. Turner, and C. A. Wood. "TLS 1.3." [Online]. Available: https://www.ietf.org/blog/tls13/ [Accessed: May 1, 2024].

[70] Wikipedia. "ID-based cryptography." [Online]. Available: https://en.wikipedia.org/wiki/ID-based_cryptography [Accessed: May 1, 2024].

[71] Wikipedia. "Certificateless cryptography." [Online]. Available: https://en.wikipedia.org/wiki/Certificateless_cryptography [Accessed: May 1, 2024].

[72] S. S. Gaur, A. Kumar, and B. Mohapatra, "An optimal lightweight cryptographic approach for WSN and its energy consumption analysis," *Int. J. Intell. Eng. Syst,* pp. 59-68, 2017.

[73] A. A. Hady*, A. Ghubaish*, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access,* vol. 8, pp. 106576-106584, 2020. (*Equal Contribution).

[74] H. Fotouhi, A. Causevic, K. Lundqvist, and M. Björkman, "Communication and Security in Health Monitoring Systems--A Review," in *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*, 2016, vol. 1: IEEE, pp. 545-554.

[75] Q. Zhang, P. Wang, D. S. Reeves, and P. Ning, "Defending against sybil attacks in sensor networks," in *25th ieee international conference on distributed computing systems workshops*, 2005: IEEE, pp. 185-191.

[76] A. Mathews, "What can machine learning do for information security?," *Network Security,* vol. 2019, no. 4, pp. 15-17, 2019.

[77] L. Breiman, "Random forest, vol. 45," *Mach Learn,* vol. 1, 2001.

[78] Wikipedia. "k-nearest neighbors algorithm." Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm [Accessed: May 1, 2024].

[79] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning,* vol. 20, pp. 273-297, 1995.

[80] Wikipedia. "Neural network (machine learning)." Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Neural_network_(machine_learning) [Accessed: May 1, 2024].

[81] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *Ieee access,* vol. 6, pp. 35365-35381, 2018.

[82] L. Clifton, D. A. Clifton, M. A. Pimentel, P. J. Watkinson, and L. Tarassenko, "Predictive monitoring of mobile patients by combining clinical observations with data from wearable sensors," *IEEE journal of biomedical and health informatics,* vol. 18, no. 3, pp. 722-730, 2013.

[83] A. A. V. Rani and E. Baburaj, "Secure and intelligent architecture for cloud-based healthcare applications in wireless body sensor networks," *International Journal of Biomedical Engineering and Technology,* vol. 29, no. 2, pp. 186-199, 2019.

[84] S. Chakraborty, S. Aich, and H.-C. Kim, "A secure healthcare system design framework using blockchain technology," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, 2019: IEEE, pp. 260-264.

[85] A. Alabdulatif, I. Khalil, A. R. M. Forkan, and M. Atiquzzaman, "Real-time secure health surveillance for smarter health communities," *IEEE Communications Magazine,* vol. 57, no. 1, pp. 122-129, 2018.

[86] H. Tao, M. Z. A. Bhuiyan, A. N. Abdalla, M. M. Hassan, J. M. Zain, and T. Hayajneh, "Secured data collection with hardware-based ciphers for IoT-based healthcare," *IEEE Internet of Things Journal,* vol. 6, no. 1, pp. 410-420, 2018.

[87] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 38, no. 5, pp. 649-659, 2008.

[88] The UCI KDD Archive. "KDD Cup 1999 Data." [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [Accessed: May 1, 2024].

[89] B. B. Rao, K. V. Krishna, and K. Swathi, "A Fast KNN Based Intrusion Detection System For Cloud Environment," *Jour of Adv Research in Dynamical & Control Systems,* vol. 10, no. 7, pp. 1509-1515, 2018.

[90] H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved KNN," *Int. J. Comput. Appl,* vol. 173, no. 1, pp. 5-9, 2017.

[91] QOSIENT LLC "ARGUS + ML." [Online]. Available: https://openargus.org [Accessed: May 1, 2024].

[92] Berry. "ECG module for multi-parameter monitor PM4100." [Online]. Available: https://www.medicalexpo.com/prod/shanghai-berry-electronic-tech-co-ltd/product-122578-866837.html [Accessed: May 1, 2024].

[93] Scapy. "Scapy Project." [Online]. Available: https://scapy.net [Accessed: May 1, 2024].

[94] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems,* vol. 100, pp. 779-796, 2019.

[95] Wikipedia. "Cross-validation (statistics)." [Online]. Available: https://en.wikipedia.org/wiki/Cross-validation_(statistics) [Accessed: May 1, 2024].

[96] S. Liu, Y. Wang, J. Zhang, C. Chen, and Y. Xiang, "Addressing the class imbalance problem in twitter spam detection using ensemble learning," *Computers & Security,* vol. 69, pp. 35-49, 2017.

[97] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *J. Artif. Int. Res.,* vol. 16, no. 1, pp. 321–357, 2002.

[98] Wikipedia. "Receiver operating characteristic." [Online]. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic [Accessed: May 1, 2024].

[99] J. Brownlee. "Failure of Classification Accuracy for Imbalanced Class Distributions." Machine Learning Mastery. [Online]. Available: https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/ [Accessed: May 1, 2024].

[100] F. Li, R. Xie, Z. Wang, L. Guo, J. Ye, P. Ma, and W. Song, "Online distributed IoT security monitoring with multidimensional streaming big data," *IEEE Internet of Things Journal,* vol. 7, no. 5, pp. 4387-4394, 2019.

[101] A. Thakkar and R. Lohiya, "A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges," *Archives of Computational Methods in Engineering,* vol. 28, no. 4, pp. 3211-3243, 2021/06/01 2021, doi: 10.1007/s11831-020-09496-0.

[102] R. Vijayanand and D. Devaraj, "A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network," *IEEE Access,* vol. 8, pp. 56847-56854, 2020.

[103] J. Cui, L. Zong, J. Xie, and M. Tang, "A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data," *Applied Intelligence,* vol. 53, no. 1, pp. 272-288, 2023/01/01 2023, doi: 10.1007/s10489-022-03361-2.

[104] P. Yang, H. Huang, and C. Liu, "Feature selection revisited in the single-cell era," *Genome Biology,* vol. 22, pp. 1-17, 2021.

[105] W. M. Shaban, A. H. Rabie, A. I. Saleh, and M. Abo-Elsoud, "A new COVID-19 Patients Detection Strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier," *Knowledge-Based Systems,* vol. 205, p. 106270, 2020.

[106] M. H. Kamarudin, C. Maple, and T. Watson, "Hybrid feature selection technique for intrusion detection system," *International Journal of High Performance Computing and Networking,* vol. 13, no. 2, pp. 232-240, 2019/01/01 2019, doi: 10.1504/IJHPCN.2019.097503.

[107] X. Li, S. H. Ling, and S. Su, "A hybrid feature selection and extraction methods for sleep apnea detection using bio-signals," *Sensors,* vol. 20, no. 15, p. 4323, 2020.

[108] S. S. Shekhawat, H. Sharma, S. Kumar, A. Nayyar, and B. Qureshi, "bSSA: binary salp swarm algorithm with hybrid data transformation for feature selection," *Ieee Access,* vol. 9, pp. 14867-14882, 2021.

[109] P. Yang, H. Huang, and C. Liu, "Feature selection revisited in the single-cell era," *Genome Biology,* vol. 22, no. 1, p. 321, 2021/12/01 2021, doi: 10.1186/s13059-021-02544-3.

[110] Y. A. Farrukh, Z. Ahmad, I. Khan, and R. M. Elavarasan, "A sequential supervised machine learning approach for cyber attack detection in a smart grid system," in *2021 North American Power Symposium (NAPS)*, 2021: IEEE, pp. 1-6.

[111] I. R. Ward, L. Wang, J. Lu, M. Bennamoun, G. Dwivedi, and F. M. Sanfilippo, "Explainable artificial intelligence for pharmacovigilance: What features are important when predicting adverse outcomes?," *Computer Methods and Programs in Biomedicine,* vol. 212, p. 106415, 2021/11/01/ 2021, doi: https://doi.org/10.1016/j.cmpb.2021.106415.

[112] S. B. Vilsen and D.-I. Stroe, "Battery state-of-health modelling by multiple linear regression," *Journal of Cleaner Production,* vol. 290, p. 125700, 2021/03/25/ 2021, doi: https://doi.org/10.1016/j.jclepro.2020.125700.

[113] D. Swain, N. Chillur, S. Patel, and A. Bhilare, "Intelligent system for detecting intrusion with feature bagging," in *2021 international conference on artificial intelligence and machine vision (AIMV)*, 2021: IEEE, pp. 1-4.

[114] X.-H. Nguyen, X.-D. Nguyen, H.-H. Huynh, and K.-H. Le, "Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways," *Sensors,* vol. 22, no. 2, p. 432, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/2/432.

[115] J. Yu, X. Ye, and H. Li, "A high precision intrusion detection system for network security communication based on multi-scale convolutional neural network," *Future Generation Computer Systems,* vol. 129, pp. 399-406, 2022/04/01/ 2022, doi: https://doi.org/10.1016/j.future.2021.10.018.

[116] D. Lightbody, D.-M. Ngo, A. Temko, C. Murphy, and E. Popovici, "Host-based intrusion detection system for iot using convolutional neural networks," in *2022 33rd Irish Signals and Systems Conference (ISSC)*, 2022: IEEE, pp. 1-7.

[117] S. Mandal, A. Sai Sabitha, and D. Mehrotra, "Analysis on Protocol-Based Intrusion Detection System Using Artificial Intelligence," in *Machine Intelligence and Smart Systems*, Singapore, S. Agrawal, K. Kumar Gupta, J. H. Chan, J. Agrawal, and M. Gupta, Eds., 2021// 2021: Springer Nature Singapore, pp. 131-143.

[118] M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, H. Jabeen, S. A. Haider, and A. Rahim, "Protocol-based deep intrusion detection for dos and ddos attacks using unsw-nb15 and bot-iot data-sets," *IEEE Access,* vol. 10, pp. 2269-2283, 2021.

[119] L. Hakim and R. Fatma, "Influence analysis of feature selection to network intrusion detection system performance using nsl-kdd dataset," in *2019 International conference on computer science, information technology, and electrical engineering (ICOMITEE)*, 2019: IEEE, pp. 217-220.

[120] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids," *IEEE Transactions on Network and Service Management,* vol. 18, no. 1, pp. 1104-1116, 2020.

[121] P. Mishra, V. Varadharajan, E. S. Pilli, and U. Tupakula, "VMGuard: A VMI-based security architecture for intrusion detection in cloud environment," *IEEE Transactions on Cloud computing,* vol. 8, no. 3, pp. 957-971, 2018.

[122] T. Parlar and E. Sarac, "IWD based feature selection algorithm for sentiment analysis," *Elektronika ir Elektrotechnika,* vol. 25, no. 1, pp. 54-58, 2019.

[123] V. Ravindranath, S. Ramasamy, R. Somula, K. S. Sahoo, and A. H. Gandomi, "Swarm intelligence based feature selection for intrusion and detection system in cloud

infrastructure," in *2020 IEEE congress on evolutionary computation (CEC)*, 2020: IEEE, pp. 1-6.

[124] A. Padmashree and M. Krishnamoorthi, "Decision Tree with Pearson Correlation-based Recursive Feature Elimination Model for Attack Detection in IoT Environment," *Information Technology and Control,* vol. 51, no. 4, pp. 771-785, 2022.

[125] Y. Pawar, N. Zamzami, and N. Bouguila, "An effective hybrid anomaly detection system based on mixture models," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, 2020: IEEE, pp. 1-6.

[126] HackerEarth. "Predict Network Attacks." [Online]. Available: https://www.hackerearth.com/problem/machine-learning/sample/ [Accessed: May 1, 2024].

[127] W. Jingyi, G. Xusheng, H. Jieli, and L. Shenghou, "ELM Network Intrusion Detection Model Based on SLPP Feature Extraction," in *2021 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 2021: IEEE, pp. 46-49.

[128] M. Madanan, A. Venugopal, and N. C. Velayudhan, "A hybrid anomaly based intrusion detection methodology using IWD for LSTM classification," in *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2020: IEEE, pp. 1-5.

[129] A. A. A. Lateef, S. T. F. Al-Janabi, and B. Al-Khateeb, "Hybrid intrusion detection system based on deep learning," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, 2020: IEEE, pp. 1-5.

[130] A. Fatani, A. Dahou, M. A. Al-Qaness, S. Lu, and M. A. Elaziz, "Advanced feature extraction and selection approach using deep learning and Aquila optimizer for IoT intrusion detection system," *Sensors,* vol. 22, no. 1, p. 140, 2021.

[131] A. Dahou, M. Abd Elaziz, S. A. Chelloug, M. A. Awadallah, M. A. Al-Betar, M. A. Al-Qaness, and A. Forestiero, "Intrusion detection system for IoT based on deep learning and modified reptile search algorithm," *Computational Intelligence and Neuroscience,* vol. 2022, 2022.

[132] P. Nimbalkar and D. Kshirsagar, "Feature selection for intrusion detection system in Internet-of-Things (IoT)," *ICT Express,* vol. 7, no. 2, pp. 177-181, 2021.

[133] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset)," in *International Networking Conference*, 2020: Springer, pp. 73-84.

[134] T. Salman, A. Ghubaish, D. Unal, and R. Jain, "Safety Score as an Evaluation Metric for Machine Learning Models of Security Applications," *IEEE Networking Letters,* vol. 2, no. 4, pp. 207-211, 2020, doi: 10.1109/LNET.2020.3016583.

[135] Raj Jain. "WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) Cybersecurity Research." [Online]. Available: https://www.cse.wustl.edu/~jain/ehms/index.html [Accessed: May 1, 2024].

[136] H. Hindy, C. Tachtatzis, R. Atkinson, E. Bayne, and X. Bellekens, "Mqtt-iot-ids2020: Mqtt internet of things intrusion detection dataset," *IEEE Dataport,* 2020.

[137] N. Moustafa, "The bot-iot dataset," *IEEE Dataport,* vol. 5, 2019.

[138] scikit-learn. "DecisionTreeClassifier." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html [Accessed: May 1, 2024].

[139] scikit-learn. "RandomForestClassifier." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html [Accessed: May 1, 2024].

[140] Keras. "The Sequential model." [Online]. Available: https://keras.io/guides/sequential_model/ [Accessed: April 18, 2024].

[141] Intel. "Understanding the Advantages of 5G." [Online]. Available: https://www.intel.com/content/www/us/en/wireless-network/5g-benefits-features.html [Accessed: May 1, 2024].

[142] H. Benaddi, K. Ibrahimi, A. Benslimane, and J. Qadir, "A Deep Reinforcement Learning Based Intrusion Detection System (DRL-IDS) for Securing Wireless Sensor Networks and Internet of Things," Cham, 2020: Springer International Publishing, in Wireless Internet, pp. 73-87.

[143] Y. F. Hsu and M. Matsuoka, "A Deep Reinforcement Learning Approach for Anomaly Network Intrusion Detection System," in *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*, 9-11 Nov. 2020 2020, pp. 1-6, doi: 10.1109/CloudNet51028.2020.9335796.

[144] J. Qi, Q. Zhou, L. Lei, and K. Zheng, "Federated Reinforcement Learning: Techniques, Applications, and Open Challenges," *ArXiv,* vol. abs/2108.11887, 2021.

[145] H. Moudoud and S. Cherkaoui, "Empowering Security and Trust in 5G and Beyond: A Deep Reinforcement Learning Approach," *IEEE Open Journal of the Communications Society,* vol. 4, pp. 2410-2420, 2023, doi: 10.1109/OJCOMS.2023.3313352.

[146] A. Virdis, G. Nardini, G. Stea, and D. Sabella, "End-to-end performance evaluation of MEC deployments in 5G scenarios," *Journal of Sensor and Actuator Networks,* vol. 9, no. 4, p. 57, 2020.

[147] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 35-59.

[148] A. T. D. Perera and P. Kamalaruban, "Applications of reinforcement learning in energy systems," *Renewable and Sustainable Energy Reviews,* vol. 137, p. 110618, 03/01 2021, doi: 10.1016/j.rser.2020.110618.

[149] Q. Jiaju, Z. Qihao, L. Lei, and Z. Kan, "Federated reinforcement learning: techniques, applications, and open challenges," *Intelligence & Robotics,* vol. 1, no. 1, pp. 18-57, 2021, doi: 10.20517/ir.2021.02.

[150] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity,* vol. 2, no. 1, p. 20, 2019/07/17 2019, doi: 10.1186/s42400-019-0038-7.

[151] J. P. Farwell and R. Rohozinski, "Stuxnet and the Future of Cyber War," *Survival,* vol. 53, no. 1, pp. 23-40, 2011/02/01 2011, doi: 10.1080/00396338.2011.555586.

[152] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep Q-Learning Based Reinforcement Learning Approach for Network Intrusion Detection," *Computers,* vol. 11, no. 3, p. 41, 2022. [Online]. Available: https://www.mdpi.com/2073-431X/11/3/41.

[153] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, "ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model," *Scientific Reports,* vol. 12, no. 1, p. 15370, 2022/09/13 2022, doi: 10.1038/s41598-022-19366-3.

[154] NIC Mexico. "Jool." [Online]. Available: https://nicmx.github.io/Jool/en/index.html [Accessed: May 1, 2024].

[155] G. Rodola. "psutil." GitHub repository. [Online]. Available: https://github.com/giampaolo/psutil [Accessed: May 1, 2024].

[156] Scapy community. "Scapy." [Online]. Available: https://scapy.net [Accessed: Jan 25, 2024].

[157] J. Oreman. "python-netfilterqueue." GitHub repository. [Online]. Available: https://github.com/oremanj/python-netfilterqueue [Accessed: May 1, 2024].

[158] Python Cryptographic Authority. "Cryptography." [Online]. Available: https://cryptography.io/en/latest/ [Accessed: May 1, 2024].

[159] A. Fadheli. "How to Make a Ransomware in Python." [Online]. Available: https://thepythoncode.com/article/make-a-ransomware-in-python [Accessed: May 1, 2024].

[160] J. Erickson, *Hacking: the art of exploitation, 2nd edition*. No Starch Press, 2008.

[161] *WUSTL-HDRL-2024 Dataset for Cybersecurity Research on Medical Applications in 5G Networks*. [Online]. Available: https://www.cse.wustl.edu/~jain/hdrl/index.html

[162] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "{TensorFlow}: a system for {Large-Scale} machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265-283.

[163] L. Zhang, Z. Pan, Y. Pan, S. Guo, Y. Liu, S. Xia, Q. Zheng, H. Li, and W. Bai, "A Hidden Attack Sequences Detection Method Based on Dynamic Reward Deep Deterministic Policy Gradient," *Security and Communication Networks,* vol. 2022, p. 1488344, 2022/01/28 2022, doi: 10.1155/2022/1488344.

[164] P. Fabian, "Scikit-learn: Machine learning in Python," *Journal of machine learning research 12,* p. 2825, 2011.