An IoMT Cybersecurity Vulnerability Study:

Medjacking Attacks on Artificial Pancreas Devices

Fahad Nazir

Thesis submitted for the degree of Doctor of Philosophy

College of Science and Engineering

James Cook University

July 2024

Acknowledgements

I am grateful to everyone who has helped me in completing my research. First and foremost, I want to express my deepest gratitude to my supervisors, Professor Trina Myers, Dr Kristin Wicking, and Dr Art Suwanwiwat, for their guidance and unwavering support from our first meeting in a café to our most recent meetings. Their expert guidance, mentoring, and encouragement have helped me improve my academic writing and experimental design skills. Thank you, Trina, Kristin, and Art, for your valuable advice and sincere mentorship.

I am also extremely grateful to James Cook University for awarding me the James Cook University Postgraduate Research Scholarship (JCUPRS), which not only minimised my financial burden but also gave me access to outstanding resources, facilities, and opportunities for professional growth. I also want to thank learning advisers Kellie Johns, Dr Melanie Smith, and Mr Lyle Cleeland for their assistance with academic writing. Many others have been instrumental in this process, and I would like to acknowledge their role in the completion of my work. Thank you, Professor Elizabeth Tynan, for providing the research skills training.

I also would like to thank my peers for collaborating with me, sharing thoughts, and offering feedback. Their friendship and collaboration have enhanced my research experience and made my endeavour more rewarding.

Finally, I am grateful to my parents and siblings for their continuous support, understanding, prayers, and encouragement throughout this academic pursuit.

Thank you to everyone who has been involved in this journey, whether directly or indirectly. Your support and encouragement have been important to me.

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given. Every reasonable effort has been made to gain permission from and to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Fahad Nazir Signature: _____

Date:

Statement of Contribution of Others

This thesis was completed by Fahad Nazir under the supervision of Professor Trina Myers, Dr Kristin Wicking, and Dr Art Suwanwiwat.

Nature of assistance	Contribution	Co-Contributor
Intellectual support	Supervision, editorial assistance, survey design, conceptual guidance	Professor Trina Myers, Dr Kristin Wicking, and Dr Art Suwanwiwat
Editing support	Initial writing, editing, or formatting assistance before thesis was finalised	Kellie Johns, Dr Melanie Smith, and Dr Eileen Siddins
Financial assistance	Stipend and fee waiver	James Cook University Research Scholarship
Infrastructure assistance	Use of high-performance computers	James Cook University

Abstract

The Internet of Things (IoT) is used in medical devices, and the terminology is referred to as the Internet of Medical Things (IoMT). Linked devices and systems collect and evaluate health data to enhance patient care, operational efficiency, and healthcare costs. The IoMT refers to interconnected medical devices (wearable and non-wearable), applications, and systems that can send and receive patient health data through the internet. In 2022, 58,000 people accessed Continuous Glucose Monitoring (CGM) in Australia, and the use of insulin pumps to manage diabetes has since grown. The use of technology to manage diabetes has accelerated the development of the Artificial Pancreas System (APS). The APS combines a CGM sensor and an insulin pump sensor to automatically inject insulin when blood glucose levels rise.

Statistics indicate that the use of digital devices for diabetes management highlights the critical need for effective cybersecurity to protect these devices. For example, Tandem Diabetes released a recall for their t:connect mobile application designed for Apple iOS devices. Over 200 people experienced injuries due to a technological malfunction that resulted in the unexpected shutdown their insulin pumps. The software malfunction caused the recall of over 85,000 versions of a mobile application, designated t:connect and created by Tandem Diabetes Care. The United States Food and Drug Administration (FDA) published a warning to both patients and healthcare practitioners concerning the prospective cybersecurity vulnerabilities linked to Medtronic insulin devices. Johnson & Johnson found that cyber attackers could hack insulin pumps to deliver incorrect insulin doses into the patient's body, which poses significant life-threatening risks. These reports demonstrate the relevance of ongoing diabetes devices and their application to investigate software defects and cybersecurity.

The IoMT revolutionises effective healthcare delivery by lowering healthcare expenditures and boosting real-time outcomes. While the IoMT offers significant benefits (e.g.,

increased access to real-time data), weaknesses are introduced, such as incompatibility of protocols, unreliability of third-party services, insufficient regulation of the technologies, and importantly, weakness in security. A crucial security weakness is medical hijacking ("medjacking"), which is the act of seizing control of the IoMT for malicious reasons. Medjacking poses a serious threat to patients with the potential to cause harm or fatalities and significantly impact patient safety. Rao et al., conducted simulation based eight malware types, encompassing Fuzz20 and Fuzz100, File Manipulation, Information Leakage, Synthetic Malware 1, Synthetic Malware 2, Synthetic Malware 3, and Synthetic Malware 4, targeted smart connected insulin pumps. The researchers in their study classified malware into three distinct categories: (1) Malware that compromises health, direct threat to a patient's health (2) Malware that undermines data sensitivity, secretly disseminating private data and (3) Synthetic threats to security, compromise the operational integrity of the targeted system.

To examine the challenges of medjacking, the experimental methodology consisted of four stages: 1) APS prototype development; 2) cybersecurity attacks exemplification; 3) dataset collection; and 4) Hybrid Intrusion Detection System (H-IDS) development. This study included brute force, Distributed Denial-of-Service (DDoS), Man-In-The-Middle (MITM), and Trojan attacks. Within the general domain of cybersecurity, these attacks specifically aim at various components of the CIA triad, namely Confidentiality, Integrity, and Availability. Within the context of IoMT hijacking, Brute Force Attacks specifically aim at credential hijacking, DDoS attack target recourse, interception of communication via MITM attacks, and the trojan attack aimed at malware-based hijacking.

APS development on Raspberry Pi B and ESP32-WROOM using B. Bergman's model. The Raspberry Pi incorporated the DHB actor. The Raspberry Pi controls glucose-insulin dynamics according to the Bergman model. The ESP32-WROOM is a peripheral device for glucose and insulin administration. The Raspberry Pi estimates insulin dosage and sends ESP32

commands to manage insulin administration. The testbed was developed, the security attacks were executed using a modular design. This study used Kali Linux and Metasploit for brute force attacks, used hping3 on Kali Linux in a virtual machine for DDoS attacks. Ettercap and a virtual machine for MITM attack.

The H-IDS was constructed, a neural network autoencoder was used during anomaly detection. Signature detection, Multi-Layer Perceptron (MLP), Decision Tree (DT), Support Vectors Machine (SVM), and Extreme Gradient Boosting (XGBoost) models were trained and tested to detect medjacking attacks. This study computed both the accuracy and loss performance matrices for the training and validation of anomaly detection. Training accuracy examined the training data, whereas validation accuracy measured its ability to differentiate normal and attack data. The difference between original training data and reconstructions during training was termed training loss. Validation loss was compared to normal and attack patterns. Low validation loss occurred when the model correctly separated normal and attack input. This study computed performance metrics to access the model's performance, the MLP, SVM, DT, and XGBoost models without dimensionality reduction and with three dimensionality reduction approaches, including Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP).

This study found that H-IDS can detect the medjacking patterns in the APS. One of the benefits of automated detection is its capacity to rapidly detect attacks, thereby reducing the potential for harm within a short period during medjacking occurrences. Automated detection improves the defence against medjacking and decreases the need for human supervision.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Address Resolution Protocol (ARP)

Artificial Pancreas System (APS)

Artificial Intelligence (AI)

Automated Cyber Defence (ACD)

Bluetooth Low Energy (BLE)

Constrained Application Protocol (COAP)

Continuous Glucose Monitoring (CGM)

Decision Tree (DT)

Deep Learning (DL)

Denial Of Service (Dos)

Diabetic Human Body (DBH)

Distributed Denial of Service (DDoS)

Domain Name System (DNS)

Exploratory Data Analysis (EDA)

Extreme Gradient Boosting (XGBoost)

Federal Communications Commission Identity (FCC ID)

Food And Drug Administration (FDA)

General Practitioners (GP)

Hybrid Intrusion Detection System (H-IDS)

Internet Control Message Protocol (ICMP)

Internet Of Things (IoT)

Internet Of Medical Things (IoMT)

Internet Protocol (IP)

Juvenile Diabetes Research Foundation (JDRF)

Long Short-Term Memory (LSTM)

Machine Learning (ML)

Man In the Middle (MITM)

Medical Hijacking (Medjacking)

Message Queuing Telemetry Transport (MQTT)

Multi-Layer Perceptron (MLP)

National Diabetes Services Scheme (NDSS)

Principal Component Analysis (PCA)

Recursive Feature Elimination with Cross Validation (RFECV)

Rest Tremor Velocity (RTV)

Secure Shell (SSH)

Secure Sockets Layer (SSL)

Support Vectors Machine (SVM)

Synchronise (SYN)

SYN–Acknowledgement (SYN–ACK)

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Transmission Control Protocol (TCP)

Uniform Manifold Approximation and Projection (UMAP)

User Datagram Protocol (UDP)

Chapter 1. Introduction

The healthcare sector is experiencing a significant transformation with advancements in digital technology. This digital transformation yields numerous benefits, including an increased emphasis on improving patient care, enhancing service accessibility, and reducing costs [1, 2]. Historically, the healthcare domain has been characterised by face-to-face engagements, reliance on physical documentation, and a primarily reactive approach to treatment methodologies [2-4]. Nonetheless, the increasing cases of chronic diseases [5, 6], coupled with an ageing population type and the growing need to enhance healthcare effectiveness, have encouraged the integration of digital technologies [7-10]. An important component of this digital transformation is the development and incorporation of the Internet of Medical Things (IoMT) [11-13]. The Internet of Things (IoT) is used in medical devices, and the terminology is referred to as the IoMT [14]. Linked devices and systems collect and evaluate health data to enhance patient care, operational efficiency, and healthcare costs.

The IoMT contributes to a significant transformation in healthcare, evolving from conventional reactive techniques to proactive [15, 16], predictive, and personalised care [17-22]. This transformation has its foundation in the capacity of IoMT to facilitate real-time, continuous monitoring of patients via interconnected devices, thereby allowing for rapid responses and enhanced decision-making [11, 23]. A prime instance is using remote patient monitoring devices, including blood pressure cuffs and Electrocardiogram (ECG) patches [24-29]. These devices offer real-time data to healthcare providers, facilitating the early identification of hypertension or cardiac anomalies prior to the onset of severe complications [30, 31]. Within healthcare settings, IoMT-enabled infusion pumps facilitate accurate medication administration [32], whereas intelligent beds monitor patient mobility, thereby averting pressure ulcers and enhancing overall care [33, 34]. In the context of medication management, intelligent pill bottles serve to remind patients of their prescribed doses, thereby

enhancing compliance [35, 36]. Continuous temperature monitoring tracks the body temperature of patients undergoing chemotherapy, which allows for the early identification of infection indicators [37, 38].

Implantable devices enabled by the IoMT, including pacemakers and cochlear implants, signify a new frontier in medical technology [39, 40]. Modern pacemakers integrate telemetry systems to monitor cardiac function and carefully relay data for continuous oversight. In contrast, cochlear implants enhanced by IoT technology adaptively refine auditory perception in response to varying environmental acoustics. A further significant application lies in the field of smart prosthetics and orthotics, which utilise IoMT technologies to dynamically adjust in response to users' movements. These devices incorporate sensors to deliver tailored feedback and enhance mobility [41, 42]. For individuals with diabetes, smart footwear integrated with pressure sensors identifies hotspots or ulcers at an early stage, thereby minimising the risk of serious complications [43-45].

Robotic-assisted surgical systems, such as the da Vinci, facilitate minimally invasive procedures augmented by IoMT, wherein real-time data analytics significantly enhance surgical precision [46-48]. In the field of neurological, in individuals suffering from neurological disorders, wearable seizure monitors observe electrical activity within the brain, identifying seizure patterns and promptly transmitting alerts to services [49-54]. Similarly, monitoring systems for Parkinson's disease, including motion sensors and wearable devices, evaluate the severity of tremors and irregularities in gait, thereby assisting clinicians in making therapies to meet the specific needs of each patient [55-58]. The IoMT technologies, including remote patient monitoring devices, wearable health monitors, intelligent prosthetics, and implantable devices, are transforming the healthcare landscape by facilitating early detection, tailoring treatment approaches, and improving patient engagement. These advancements collectively facilitate a more efficient, proactive, and patient-centred healthcare ecosystem.

The IoMT is a network of interconnected medical devices, software applications, and health systems allowing instant data exchange, fostering a more tailored and efficient approach to healthcare delivery [59-66]. This interconnected IoMT encompasses wearable devices, remote patient monitoring systems, smart inhalers, integrated medical imaging, and implantable devices. These technologies function collaboratively to enhance diagnostic capabilities, therapeutic interventions, and all aspects of patient care. Using the IoMT ecosystem improves the quality of digitally managed treatment while decreasing the need for hospitalisation, resulting in cost savings for patients and healthcare corporations. Additionally, data generated from connected devices provides insights to make appropriate real-time decisions [14]. IoMT concepts have the potential to evolve healthcare operations by enhancing diagnosis accuracy, personalising treatment regimens, and forecasting disease outbreaks.

Wearable devices in the IoMT setting, such as smartwatches and fitness trackers, have become indispensable tools for monitoring critical health variables, including heart rate, oxygen saturation, and glucose amounts [11, 62, 63, 67]. These devices provide critical data that aids healthcare professionals in remote monitoring of patient health, allowing for timely intervention in the event of irregularities. In the context of chronic disease, insulin pumps and continuous glucose monitoring (CGMs) have emerged as critical tools for managing conditions like diabetes. Insulin pumps are compact; computerised devices affixed to the body that delivers a continuous, precisely regulated dose of insulin tailored to the individual patient's needs. Comparably, CGM sensors are implanted beneath the skin's surface to assess glucose levels in real-time, transmitting data to interconnected devices such as smartphones or smartwatches [68-73]. These implantable devices, including pacemakers, insulin pumps, and neurostimulators, constantly track and regulate medical conditions such as cardiac arrhythmias, diabetes, and neurological disorders [62, 74-77].

The advancement of the IoMT has made a substantial impact on the healthcare sector by facilitating uninterrupted connectivity and the integration of medical devices, applications, and systems. However, these innovations introduce security risks and vulnerabilities that put the patients' lives at risk of medical complications, and these risks also may open disputes between healthcare professionals and service providers [78, 79]. IoMT systems and devices are vulnerable to several security concerns, such as medical hijacking (medjacking) [80-82], unauthorised access [83], data breaches, and device function manipulation [84, 85]. Medjacking, also known as medical hijacking, denotes the unauthorised control of IoMT for malicious objectives. With the growing interconnectivity of medical devices, medjacking is emerging as a serious risk to patient safety, possessing the capacity to compromise health services and cause harm. With the advent of interconnected medical devices such as insulin pumps [81, 82, 86-89], CGM [81, 90-92], pacemakers [93-98], and infusion pumps [99], these devices have concurrently emerged as targets for hijacking and other cyber-attacks in literature. In 2015, an important instance emerged: the United States Food and Drug Administration (FDA) released a warning notice regarding the Hospira infusion pump. Investigators have determined that the pump is susceptible to remote access via its network, thereby enabling unauthorised individuals to modify the dosage of medication administered to patients [82, 100].

In the current literature, changes or modifications related to the functionalities or operations of medical devices are documented; however, there have been no reported fatalities or significant adverse implications, as all studies serve basically as a proof of concept. For instance, in one study [98], the researchers indicated that the FDA warns against unauthorised users gaining access to the device (an infusion pump) and altering dosages remotely. In a comparable investigation, another study [101] indicated that an unauthorised individual possesses the capability to manipulate the device and alter the dosage administered by the pump in a malicious manner. In [102], researchers introduced a new class of "process-aware" cyber-

attacks targeting medical devices, particularly insulin pumps utilised to regulate blood glucose levels in individuals with Type-I diabetes. These attacks are designed to fraudulently modify the drug administration process, thereby inflicting harm on patients over an extended duration while remaining within operational parameters. In another investigation [103], the scholars articulated that throughout data modification attacks, an attacker attempts to alter the vital signs of a patient by breaking the device or by intercepting and modifying the communication packets exchanged between the healthcare and the programming device. According to literature, a bad actor (attacker) exploits security flaws in an IoMT system for different purposes, including financial gain (ransomware). This bad actor harms others on purpose sometimes in response to intense emotional pain or revenge. For example, a bad actor may hijack the medical device of any targeted person to change the operation of the medical device.

The vulnerabilities associated with real-time medical devices present considerable threats to patient safety, privacy, and the integrity of the healthcare system [78, 80, 103-109]. The research unit of Forescout Vedere Labs, as stated in its report titled 'Unveiling the Persistent Risks of Connected Medical Devices,' has identified 162 vulnerabilities that impact IoMT devices. Cybercriminal attacks on interconnected medical devices primarily seek to appropriate sensitive patient information, encompassing personal details and medical and treatment histories. In extreme cases, attacks can disrupt healthcare operations and compromise patient safety. The IoMT poses a considerable threat to healthcare delivery organisations [110]. This report indicates that the exploitation of vulnerabilities has the potential to cause direct harm to patients. Furthermore, the interference with real-time monitoring devices could delay timely treatment.

The potential consequences of dosage modifications are severe and multifaceted. For instance, an overdose of insulin could result in hypoglycemia, causing seizures or unconsciousness. In contrast, an overdose may lead to hyperglycemia and long-term

complications such as kidney damage or cardiovascular disease. Similarly, incorrect dosages of antibiotics can promote drug resistance, reducing their effectiveness in future treatments and posing a public health risk. While current studies are largely proof-of-concept, and no fatalities have been reported, the risks remain significant if these vulnerabilities were exploited. Long-term harm to patients could result from sustained device manipulation, while systemic healthcare failures could occur if attacks disrupted hospital operations or delayed treatments. For example, interference with real-time monitoring devices could prevent timely interventions, increasing the likelihood of patient harm. Additionally, the exploitation of IoMT vulnerabilities may erode trust in medical technologies, hampering their adoption and limiting future advancements in healthcare delivery.

Following the current literature, the risk of medjacking remains a significant concern. As technological advancements progress and an increasing number of devices achieve interconnectivity, the prospective attack surface grows. To investigate APS medjacking and mitigation solutions, the primary goal of this research was to undertake a thorough examination aimed at illustrating the complexity of APS in the context of medjacking attacks and the proactive strategies for their mitigation. The secondary goal involved the development of a cost-effective ASP prototype and a cybersecurity testbed to collect datasets. The fourth goal included the development of preventative strategies, referred to as hybrid intrusion detection systems (H-IDS), aimed at examining the possibilities of employing Artificial Intelligence (AI) methodologies in the identification of anomalous patterns. The cyberattack cases include brute force, Distributed Denial-of-Service (DDoS), Man-in-the-middle (MITM), and Trojan attacks.

AI has fundamentally transformed the landscape of cybersecurity by facilitating proactive defence strategies that detect, predict, and mitigate cyber threats [111-114]. According to Talukder et al., ML algorithms can comprehend information from historical

datasets, thereby identifying anomalies or patterns that may indicate potential intrusions. Mahboubi et al., stated that the implementation of ML significantly enhances cybersecurity, thereby fostering more analytical and efficient defence mechanisms that decrease dependence on temporal resources and human effort. Data serve an indispensable function in facilitating informed decision-making and formulating strategic techniques that enhance the effectiveness of ML techniques. ML is integrated into threat hunting, consisting of solutions into five categories: (i) Supervised Machine Learning, (ii) Unsupervised Machine Learning, (iii) Reasoning techniques, (iv) Graph-based approaches, and (v) Rule-based approaches [114]. These ML techniques constantly examine patterns or application logs to identify anomalous activities or trends. For instance, ML models can identify anomalous patterns by comparing conventional patterns with unusual ones that may fail human analysts.

Predictive analytics is another tool of ML in the realm of proactive defence [115]. Through the statistical examination of historical data and trends, ML models possess the capacity to predict prospective attacks. For example, supervised learning can be used when a target is specified to be achieved from a specified number of inputs, thereby exemplifying a task-oriented technique. Within the domain of ML, the primary supervised learning techniques are identified as classification and regression techniques. These techniques are commonly used to categorise or forecast future occurrences pertaining to a specific security issue [116].

Moreover, behavioural analytics is another proactive technique that uses ML to systematically observe user and entity behaviours over time, constructing profiles that facilitate the identification of cyber threats. According to Sarker, the practical application of cybersecurity serves as the fundamental platform for employing the CyberLearning model, which systematically examines network behaviour to identify security patterns that identify normal behaviour, thereby facilitating the detection of anomalies or related attacks [8]. Similarly, Sarker stated that ML can analyse vast datasets promptly and efficiently, making it

significantly faster than human-led threat detection efforts. ML uses behavioural analysis alongside rapidly evolving parameters to identify anomalies that may indicate the presence of an attack [115].

In the context of threat hunting, the researchers in a study implemented machine learning-based models to proactively threat hunting [117]. The models encompassed within their system include random forest (RF), support vector machine (SVM), multi-layer perceptron (MLP), and AdaBoost, all of which are employed for proactive threat hunting. Their system consisted of data pre-processing and developing machine learning models to identify anomalies [117]. A further study of threat hunting in [118] illustrated that it involves the analysis of system logs to identify malicious activities that might have bypassed current security measures. The researcher proposed the continuous bag-of-terms-and-time (CBoTT) framework, wherein data preprocessing was conducted, and a neural network model was employed to train the dataset, their framework was developed for anomaly detection [118].

In [119], the researchers proposed threat hunting for industrial IoT edge devices using a machine learning-based approach. Their study employed Decision Tree (DT), Support Vector Machine (SVM), Logistic Regression (LR), and Random Forest (RF) models for threat (attacks/anomalies/malicious/malware) hunting (detection/identify/find/) in industrial IoT edge devices [119]. Similarly, in threat hunting for industrial IoT, the researchers in [120] proposed an ensemble of deep federated learning techniques aimed at threat hunting within the context of industrial IoT. This approach encompassed data pre-processing, and a model was developed to hunt the threats systematically [120].

In this research, proactive mitigation denotes systematic attack hunting using machine learning models and algorithms. The H-IDS developed in this study is presented as a proof of concept, enabling an evaluation of its performance and a comparative analysis of the results.

This thesis discusses the challenges associated with medjacking vulnerabilities, presents concrete evidence of medjacking, and explores the potential consequences. Further, this thesis offers a comprehensive examination of medjacking experiments and their results. This thesis reviews the medjacking vulnerabilities, methods, and proactive mitigation solution. The thesis proposed H-IDS and provides essential suggestions for protecting APS devices and minimising medjacking by identifying anomalies. This thesis substantially contributes to APS security by providing a possible solution as a Hybrid Intrusion Detection System, including synthesised medjacking literature as a guide for the research community addressing medjacking attacks and solutions. This research synthesises existing information, provides significant perspectives, and proposes actionable measures for APS security.

1.1 Overview of Technology

The Internet of Things (IoT) connects and exchanges data between devices, automobiles, buildings, and other objects, utilising sensors, software, and other technologies [121]. The IoT devices utilise network protocols such as Wi-Fi, Bluetooth, Zigbee, and cellular communication [122]. Edge layer or cloud computing can evaluate data locally or transfer it to centralise platforms for additional analysis. The Edge Layer is considered a decentralised computing architecture wherein data processing and decision-making occur in proximity to the data sources, including sensors, devices, or other endpoints [123]. The Edge layer within an IoMT architecture functions as an essential link between medical devices and centralised systems, including cloud infrastructures [11, 124]. It facilitates the processing, storage, and transmission of data produced by IoMT devices in proximity to the source, thereby ensuring minimal latency, enhanced security, and optimal resource utilisation [125-128]. To facilitate uninterrupted data transmission and ensure interoperability, the Edge layer depends on a multitude of network protocols, such as Wi-Fi, Bluetooth, Zigbee, and cellular connectivity [129]. Zigbee, characterised by its low power consumption and limited bandwidth, is

specifically engineered for highly dense IoT networks, providing it particularly suitable for healthcare settings where numerous devices engage in effective communication [130]. Bluetooth, characterised by its short-range and low-power capabilities, is particularly advantageous for portable IoT devices [131]. Cellular networks, encompassing 3G, 4G, and 5G technologies, facilitate the connectivity of IoMT devices in remote or mobile environments where alternative networking solutions are inaccessible [132-134]. The advent of 5G significantly enhances the Edge layer by providing ultra-low latency and elevated bandwidth, thereby facilitating the deployment of advanced applications [135]. Wi-Fi is extensively employed within the Edge layer for devices necessitating elevated data throughput, especially in settings characterised by pre-existing Wi-Fi infrastructure, such as hospitals, clinics, and residential environments [136, 137]. The IoT enables remote detection and operation of goods, integrating the real world with computer-based systems for competence and economic benefits. The IoT includes devices, sensors, networking, data processing, and user interface. Environmental data is sent by sensors and devices to centralise platforms. Examples include industrial equipment, home appliances, and healthcare gadgets.

In the context of industrial equipment, the IoT facilitates predictive maintenance and enhances operational efficiency. For example, sensors integrated within manufacturing equipment can incessantly observe parameters such as temperature, pressure, and vibration [138, 139]. These sensors relay information to a centralised system wherein sophisticated analytics forecast potential failures prior to their development, thus mitigating downtime and decreasing maintenance expenditures [139, 140]. Within the home appliances domain, the IoT has resulted in an epoch characterised by the intelligent or smart home wherein devices are interlinked and subject to remote administration [141, 142]. For example, intelligent thermostats incorporate the inhabitants' preferences and autonomously modulate the temperature to optimise comfort while concurrently conserving energy resources [143]. These

smart home devices are accessible to control through smartphone applications [144, 145], thereby enabling users to regulate their domestic environment from mobile applications.

The integration of IoT technology has significantly enhanced healthcare devices, thereby improving patient care and optimising medical outcomes. Wearable devices, including smartwatches and fitness trackers, facilitate the monitoring of essential physiological parameters, encompassing heart rate, blood pressure, and sleep patterns [11, 12, 14, 66, 146]. These medical devices can notify users regarding potential health concerns in real-time, thereby facilitating prompt intervention. For individuals afflicted with chronic diseases, the IoT-enabled medical devices, exemplified by glucose monitoring devices for diabetic patients or remote monitors for those with cardiac conditions, facilitate continuous health monitoring [147, 148]. These continuous monitoring devices allow the more efficacious management of medical conditions and decrease the necessity for frequent hospital visits [14, 61], thereby enhancing the overall quality of care and patient outcomes.

Different terms have been used to describe the connection and amalgamation of IoMT devices. For instance, the IoMT is used among the research community to describe the connection of all medical devices to a healthcare provider's computer system [65, 66, 78, 149-151]. Alternatively, the term healthcare IoT is commonly used to describe smart wearable medical devices (e.g., smartwatches, step trackers, and wristbands), remote patient monitoring, home care treatment, and mobile health. Android applications often deploy healthcare IoT technology in contrast to IoMT technology [105, 152-156]. Despite the widespread usage of IoT technology in the medical field, no consistent terminology or keywords have been identified in the literature. Wearable IoT has been used to describe medical devices that apply IoT technology [157]. The term IoMT is used in this thesis as synonymous with healthcare IoT, wearable IoT, and other terms describing the use of IoT technology in healthcare settings.

The IoMT includes medical gadgets, sensors, connectivity, analytics, and interfaces. Fitness trackers, implantable gadgets, and remote monitoring systems collect health data. These devices provide data to General Practitioners (GP) and centralise platforms. Data analytics analyse enormous health data sets to give tailored therapy, early identification, and surveillance. Patients and clinicians may access health data live via mobile applications and web platforms. Remote patient monitoring improves chronic disease treatment and decreases hospitalisations. IoMT devices and associated equipment also improve hospital procedures and asset management, ensuring the availability of resources. Although IoMT offers numerous benefits, it has security, privacy, interoperability, data management, and scalability issues.

1.2 Research Motivation

IoMT devices have contributed to managing diabetes by continuously tracking their diabetes condition. In 2023, the International Diabetes Federation found that 10.5% of adults (aged 20–79 years old) have diabetes, and nearly half of this percentage do not know they have the disease [158]. The Australian National Diabetes Services Scheme and Australasian Paediatric Endocrine Group registered 5.1% (or over 1.3 million) of Australians with diagnosed Type 1 and Type 2 diabetes in 2021. According to the National Diabetes Services Scheme and Australasian Paediatric Endocrine Group, diabetes cases increased with age, and nearly one in five Australians aged 80–84 were living with diabetes in 2021, which is over 30 times higher than the incidence of the disease among people aged under 40. After adjusting for age, males had a higher prevalence of diabetes, approximately 1.3 times higher as compared to females [159]. In 2022, a report from the Department of Health and Aged Care, stated that 1.8 million Australians are living with diabetes, and over 120,000 have been diagnosed with diabetes in the previous year. Of those diagnosed, approximately 58,000 Australians with Type 1 diabetes were given a glucose monitoring device [160]. The analysis of statistical trends and patient demographics highlights the essential need for robust cybersecurity solutions to protect these

devices. The IoMT devices for diabetes management adopt digital connectivity, which may open cybersecurity vulnerabilities.

In 2024, Tandem Diabetes released a recall for their t:connect mobile application designed for Apple iOS devices. The recall is caused by a software malfunction that resulted in the application crashing, and subsequently relaunching in a continuous loop. This conduct ended in accelerated rapid battery depletion of the insulin pump, thereby posing the risk of an unexpected shutdown pump. Such interruptions in insulin delivery may precipitate hyperglycemia or diabetic ketoacidosis, both of which are serious medical conditions [161]. Theoretically, shutdown pump operation will interrupt insulin delivery, potentially resulting in inadequate insulin administration. This scenario may engender hyperglycemia or, in more severe instances, diabetic ketoacidosis, a condition that poses significant life-threatening risks due to high blood sugar levels and insufficient insulin availability. Tandem Diabetes Care is a medical device enterprise in the United States that focuses on advancing technologies for diabetes management, specifically in insulin infusion medical care. Tandem Diabetes Care sets up a network of international distribution partners, encompassing Australia, the Bahamas, Belgium, the Czech Republic, Denmark, Finland, and France, and these countries use Tandem products.

In a real-world scenario, due to battery depletion, according to [162], over 200 people experienced injuries due to a technological malfunction that resulted in the unexpected shutdown of their insulin pumps. The software malfunction caused the recall of over 85,000 versions of a mobile application, designated t:connect and created by Tandem Diabetes Care. This software malfunction has shown the consequences of this failure indicating that over 200 individuals with diabetes experienced injuries. According to Diabetes Australia [163], there have been no reported cases of individuals in Australia being adversely affected by the t:slim X2 insulin pump, they highlighted that the mobile connectivity for t:connect was made disabled

for Australian devices by its manufacturer, Tandem Diabetes Care, prior to the occurrence of the malfunction. These reports demonstrate the importance of diabetes devices and their application to investigate software defects and cybersecurity. According to [164], certain manufacturers and service providers within the IoT sector develop devices without adequate security measures, thereby making them susceptible to cyberattacks. Cyberattacks that may be perpetrated against an IoT device encompass identity theft, eavesdropping, MITM, and energy depletion attacks. A typical target for malicious attackers within an IoT device is the small, limited-capacity battery that serves as its power source, commonly called an energy depletion attack.

Other than Tandem Diabetes Care, Medtronic provides an extensive variety of devices designed to manage diabetes, encompassing insulin pumps, continuous glucose monitoring, and software solutions. Medtronic diabetes devices are used in many countries, including Australia, New Zealand, and other Asian countries. In 2019, the FDA published a warning to both patients and healthcare practitioners concerning the prospective cybersecurity vulnerabilities linked to Medtronic insulin devices. The principal rationale for this advisory was the inability of Medtronic devices to perform software updates [165]. According to this report, Medtronic insulin devices are theoretically vulnerable to cyber-attacks.

The rapid development of digital technology and the ubiquitous adoption of the IoMT revolutionised treatment management and operations using advanced technology. The healthcare industry transformed during 2010–2023, as stated in [166], another study reviewed digital transformation from 2014 to 2022 [167], and investment in the IoMT has increased [168]. IoMT device sales are expected to reach $136.8 billion by 2025 [169]. Moreover, investment in the IoMT has risen due to sensor technology acceleration, wearable device necessity, and demand for remote patient monitoring. Competitive innovations in the field of healthcare have introduced security risks that leave end users vulnerable to potential threats

that compromise their treatment. In [170], related to the healthcare domain, the researchers stated that many vendors manufacture various healthcare products and devices, and new ones keep joining IoT's fascinating technological race. Manufacturers have not followed device-compatibility interface and protocol standards. In another study [171], the researchers highlighted that the worldwide IoT device market is projected to expand from 249 billion dollars in 2018 to 457 billion dollars by 2020, prompting numerous healthcare and medical device manufacturers to engage in a race endeavour to develop smart devices.

The researchers in [172] stated that embedded medical devices typically lack robust security defence mechanisms as strong as conventional computers. The incorporation of robust hardware security measures into medical devices, such as the Home Monitoring Unit, experiences significant financial implications. Manufacturers might face a dilemma between prioritising security and managing costs, as the funds allocated for enhancing the security of the Home Monitoring Unit are diverted from the development of treatment functionalities that are crucial for saving lives. Furthermore, there exists a race a rush to penetrate the market, coupled with a rigors certification protocol that limits the simple modification of an already approved design. According to [173], manufacturers of IoT devices are aggressively releasing new products in a mad race to secure their position within the IoT market. The increased pressure to distinguish based on usability-centric functionalities has led to the development of products and features that lack adequate security examination. According to the literature [172, 173], the development of cutting-edge IoMT devices has often prioritised speed and functionality over comprehensive security measures. As a result, medjacking and other vulnerability exploits have occurred widely. The vulnerabilities associated with medjacking in the IoMT are of extreme significance, given their potential to significantly impact patient safety [146]. In conventional healthcare cybersecurity, breaches primarily include data theft, wherein patient records or financial information are compromised [174]. Nonetheless, cyberattacks

associated with the IoMT may extend beyond mere data theft to interfere with or alter the operational integrity of medical devices, thereby posing serious implications [103]. One of the principal distinctions between hacking within the IoMT and other cybersecurity threats in the healthcare sector lies in the tangible connection with patients. For instance, medjacking attacks on insulin pumps [81, 82, 86-89] or pacemakers [93-98] may enable attackers to manipulate device configurations, resulting in the administration of incorrect dosages of a device's operational capabilities. Moreover, In [175], Kwarteng and Cebe studied security issues in implantable medical devices, including insulin pumps, cardiac implants, cochlear implants, ocular implants, and foot drop implants. The researchers investigated eavesdropping, Denial of Service (DoS), replay, Man in the Middle (MITM), software injection, and Side Channel attacks. The researchers concluded that a cyber attacker could infiltrate these devices to remotely manipulate their functions [175].

Some case studies elaborated further on how eavesdropping, DoS, replay, and MITM attacks are related to medjacking; for example, in [172], the researchers elucidated that, upon obtaining physical access to the devices, an adversary was capable of integrating a physical device equipped with a wireless communication interface within it, thereby gained remote access to the device. This allowed an attacker to intercept all communications between the home monitoring unit and the backend server and perform an MITM, given the proprietary protocol was identifiable. The researchers also stated that an adversary can access the information transmitted by the pacemaker to the home monitoring unit [172]. In another study [176], the researchers presented a proof of concept demonstrating that IoMT devices and their associated smartphone applications are susceptible to hijacking attacks. The researchers determined that medical data derived from the glucometer and oximeter devices was retrievable on the attacker's smartphone. The researchers outlined in their proof-of-concept experiment that the DoS attack could be launched against similar devices. Similarly, in [177],

the researchers performed MITM, replay, connection hijacking, DoS, and Denial of Sleep attacks on health IoT devices. The attacks targeted disrupting the communication between health IoT devices and their corresponding applications [177]. In 2019, the FDA warned patients and medical professionals about potential cybersecurity risks associated with Medtronic insulin devices [165]. The primary reason for the warning was that Medtronic devices are unable to update their software.

IoMT devices help people manage diabetes. However, medjacking threats remain an open research issue. As mentioned earlier in the chapter, cyberattacks on IoMT devices pose a life-threatening risk. The IoMT is transforming the healthcare landscape through the integration of medical devices, sensors, and healthcare systems, thereby facilitating remote patient monitoring, diagnostics, and data-informed decision-making processes. Nevertheless, this increased interconnectivity makes IoMT devices vulnerable to medjacking attacks. Consequently, IoMT devices, including insulin pumps, pacemakers, wearable health monitors, and implantable devices, are under rigorous investigation to mitigate medjacking vulnerabilities. The main motivation for this study involves medjacking attacks, syntheses the medjacking issues, and developing solutions in the APS devices, as these devices are life-critical devices. Medjacking attacks may target such devices, and result in life-threatening consequences, these consequences highlighting the need for security measures that prioritise security.

Current intrusion detection systems for the IoMT represent an important area of research aimed at effectively mitigating medjacking attacks. These techniques demonstrate the ability to detect intrusions and further investigate a proof-of-concept study within the context of the IoMT and medjacking required to identify which technique or solution is feasible to mitigate medjacking attacks. This research proposes a Hybrid Intrusion Detection System (H-IDS) to mitigate medjacking attacks in APS devices and solve these challenges. In this study,

the H-IDS employing ML and DL strategies as a proof of concept, wherein the anomaly detection module was trained to discern deviations from normal behaviour. Conversely, the signature-based component functions primarily as a preliminary filter for identified medjacking.

To comprehensively demonstrate the primary risks, the following section will discuss diabetes, the significance of utilising diabetes management devices, their vulnerabilities, and the prospective consequences if these devices are hijacked. Diabetes management devices, including insulin pumps and continuous glucose monitors, are fundamental to managing diabetes. Their ongoing improvement in digital connectivity and functionality improved overall benefits to patients. Nevertheless, the wireless connectivity facilitating remote monitoring and control of these devices simultaneously introduced considerable security vulnerabilities. The occurrence of a medjacking attack on a diabetes management device might have serious consequences, including unauthorised control over the device and disruption of essential medical functionalities, which may lead to serious health impacts or, in extreme cases, fatalities. Consequently, safeguarding these devices against unauthorised access becomes essential. The subsequent sections provide a detailed discussion of these risks, the technological aspects of the IoMT, and the consequences associated with medjacking attacks.

1.2.1 Justification of Why Medjacking is Critical in IoMT

Cybersecurity vulnerabilities, such as medjacking, present a direct and immediate threat to patient safety [78, 98, 178, 179], significantly impacting patient safety. Medical devices, including insulin pumps [81, 82, 86-89], CGM [81, 90-92], pacemakers [93-98], and infusion pumps [99], have simultaneously shown as subjects of medjacking and various cyber-attacks within the literature. Many IoMT devices, including pacemakers and insulin pumps, fulfil critical life-saving functions. These devices can be exploited upon compromise to cause patient safety [82, 94, 178, 180-182]. Other than safety, a significant concern pertains to the potential disruption of device functionality. For example, as stated in [183], physical IoMT devices are

vulnerable to tampering, enabling attackers to alter their operational parameters or render them entirely or partially inoperative. Attackers may further leverage certain firmware vulnerabilities, enabling them to deliver ransomware within the IoMT device and seize control [183]. Furthermore, according to [184], attacks on implantable and wearable devices may be deliberately designed to drain the battery life of such devices, thereby posing a potential threat to patients who depend on them for monitoring or therapeutic purposes. The phenomenon of battery drain is complexly associated with various forms of cyberattacks, particularly those initiated by malware infiltrating a device, which subsequently leads to excessive consumption of energy [184].

The operational disruptions caused by the jamming attack are serious. According to [185], most wireless devices communicate through radio frequency signals. In this context, an attacker may disturb and obstruct the communication channels between sensors and receivers, resulting in complete operational discontinuation. Cyber-attacks have an increased influence on the operational and financial dimensions of infrastructure and the ethical and social considerations involved [185]. According to [186], the consequences of inadequate security within IoMT healthcare systems can be the breach of patient privacy due to eavesdropping, as well as the delayed identification of life-threatening incidents attributable to the disruption of normal functioning of IoMT devices caused by DoS attacks [186].

1.3 Overview of Diabetes

A chronic condition of high glucose levels in the blood is known as diabetes [187]. Diabetes is classified into two types. Type 1 diabetes is associated with insulin injections and is a condition defined by inadequate insulin production. Type 2 diabetes is associated with insufficient or resistant insulin activity that is typically associated with alterations to lifestyle. Another less common kind of diabetes is gestational diabetes, which can develop during pregnancy and normally fades after delivery [188].

Diabetes is caused by gene mutations or underlying medical conditions, such as monogenic and secondary diabetes. Medicine, food, exercise, and blood glucose monitoring all relate to health management, and several experts recommend lifestyle modification with proper diet control to manage glucose levels [189-191]. Patients generally go through a treatment cycle and, depending on the medical expert's screening experiments and recommendations, the patients will undergo individualised care under the direction of a GP to effectively manage their diabetes.

1.3.1 Artificial Pancreas System

An Artificial Pancreas System (APS) mimics the role of the pancreas by administering insulin and controlling glucose levels for patients diagnosed with Type 1 diabetes. The APS consists of three primary subsystems: the Continuous Glucose Monitoring (CGM) sensor, insulin pump, and control algorithm or controller subsystem. A CGM subsystem continuously monitors the glucose levels in an individual's body often by utilising a minuscule sensor inserted beneath the skin. The CGM subsystem transmits glucose readings in real time for the APS to monitor and adapt to changes. Depending on the CGM readings, insulin pumps may then inject insulin to normalise the glucose level. This process is illustrated in Figure 1.1.

Figure 1.1

Working Mechanism of the Artificial Pancreas System.

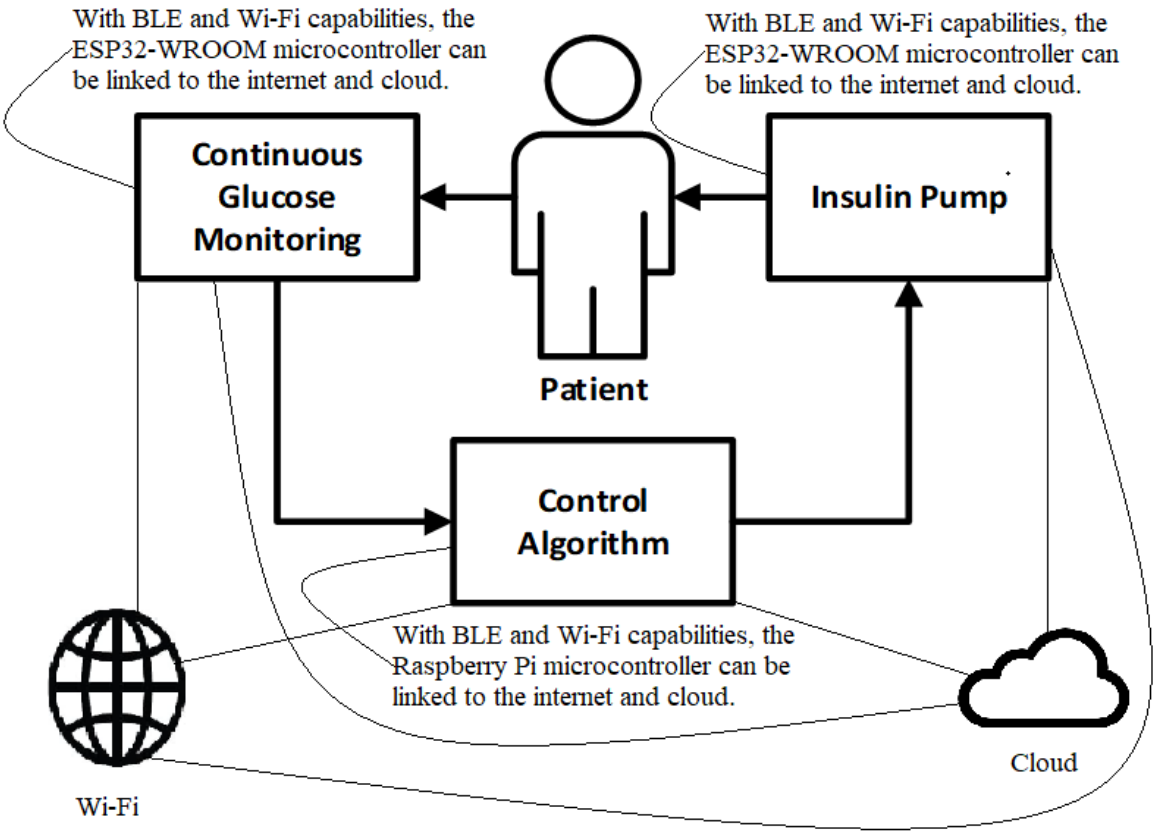


Figure 1.1 depicts a standard APS workflow where the CGM sensor checks glucose levels, and the insulin pump injects insulin. The control algorithm provides information that enables a patient's decision making, namely in the calculation of the appropriate dose. An insulin pump is a device designed to deliver insulin, which aids in regulating blood glucose levels, while a CGM monitors glucose levels in real-time. Both devices can be enhanced by incorporating Wi-Fi and Bluetooth functionalities, enabling seamless communication, cloud connectivity, and real-time data transmission. These features significantly improve diabetes management for individuals living with the condition. Prototypes of these devices can be developed using cost-effective microcontrollers like the ESP32, which come with integrated Wi-Fi and Bluetooth modules. The control algorithm is crucial for automating insulin delivery,

serving as a decision-making system that adjusts insulin administration based on deviations from target glucose levels. Various decision-making algorithms have been developed, such as the Proportional Integral Derivative (PID) controller, model predictive control, and event-triggering variations [192-195].

Maintaining normal glucose levels and preventing hypoglycaemia and hyperglycaemia are primary objectives of an APS. Currently, APSs are standard tools for managing diabetes in the long term to reduce hyperglycaemia, improve glycaemic control, and mitigate the risk of complications caused by diabetes. The APS is a promising system with the potential to transform diabetes management, and continuing research is necessary to improve its functionality and accessibility to more patients who experience this chronic condition.

1.3.2 Artificial Pancreas System Architectural Components

APS devices improve the quality of communication and connectivity with the internet and other devices. Different communication protocols can be used to enhance the communication and connectivity of APS devices. According to current literature, several protocols are used to achieve bidirectional IoT communication. The following communication protocols are well known for demonstrating architectural functions: Constrained Application Protocol (CoAP), devices profile for web services, data distribution service, extensible messaging and presence protocol, Message Queuing Telemetry Protocol (MQTT), and Bluetooth Low Energy (BLE) [121]. Although CoAP, MQTT, and BLE are all lightweight protocols, they are widely implemented in IoMT devices [196, 197]. The MQTT is expected to be the standard protocol of IoT communication in the future [198].

1.3.3 Artificial Pancreas System Security Vulnerabilities

The security vulnerabilities of the IoMT in general, and specifically the APS, are rising at an alarming level. These vulnerabilities are well documented in research literature and

informal publications, such as those published by security organisations. A crucial security weakness is medjacking, [81, 89, 199]. Medjacking has serious life-and-death implications for patient safety. The potential of medjacking assaults is rising due to the expanding number of interconnected IoMT devices.

In 2015, the FDA issued the first cybersecurity warning regarding security flaws in Hospira infusion pumps and stated the pumps could be attacked by remotely accessing a hospital's network [82, 100]. According to the FDA, the Medtronic company identified that approximately 4,000 patients who use insulin pumps could be compromised. As a result, the FDA issued the first recall of an insulin pump due to cybersecurity risks [200].

Similarly, MiniMed insulin pumps were recalled due to cybersecurity threats [201] preventing the software patch from updating. The FDA issued a notification for MiniMed insulin pumps, stating that the "use of these devices may cause serious injuries or death." This evidence of security weakness allows the research community to develop potential solutions for diabetes management devices to mitigate or prevent cybersecurity threats.

In 2016, Johnson & Johnson warned diabetic patients about their insulin pump's vulnerability to hacking [202]. The CGM sensor and insulin pump have proprietary wireless communication methods that might be reverse engineered. A study examining the actions of a person who hacked a wi-fi-enabled thermostat [203] reported that any control device linked to the internet, including insulin pumps and CGM sensors, is hackable. Consequently, users might be seriously harmed or die because they are not able to disconnect from their surgically embedded devices in real time.

In 2018 and 2019, a researcher concluded that security breaches can occur between insulin pumps and CGM sensors due to wireless communication [87]. Some conceptual and experimental studies that relate to medjacking identify security flaws in APSs. For instance, [81] reported that insulin pumps and CGM sensors are hackable due to wireless technology.

Similar security problems are possible in electronic heart implantable devices and other IoMT devices.

In 2019 and 2020, a detailed survey of cybersecurity in IoMT devices clearly showed that IoMT devices are hackable [199]. Although another study analysed a simulated replay attack on an APS [204] and showed the insulin pump's hackability, the authors were unable to propose any mitigation solutions. Another recent study concluded that IoMT devices can be hacked because of wireless connectivity [205].

Patients who use IoMT devices may believe or assume that the devices are sufficiently secured; therefore, they may not be aware of security flaws that allow medjacking to occur. As patients may not be aware of the problem, no further steps are taken to deter, prevent, detect, or mitigate medjacking. To mitigate medjacking, an in-depth understanding of the associated security flaws is crucial. This thesis consists of three research domains: cybersecurity, medical devices (i.e., diabetes devices), and the IoT as shown in Figure 1.2.

Figure 1.2

Three Research Domains: Security, Medical Devices, and Internet of Things.

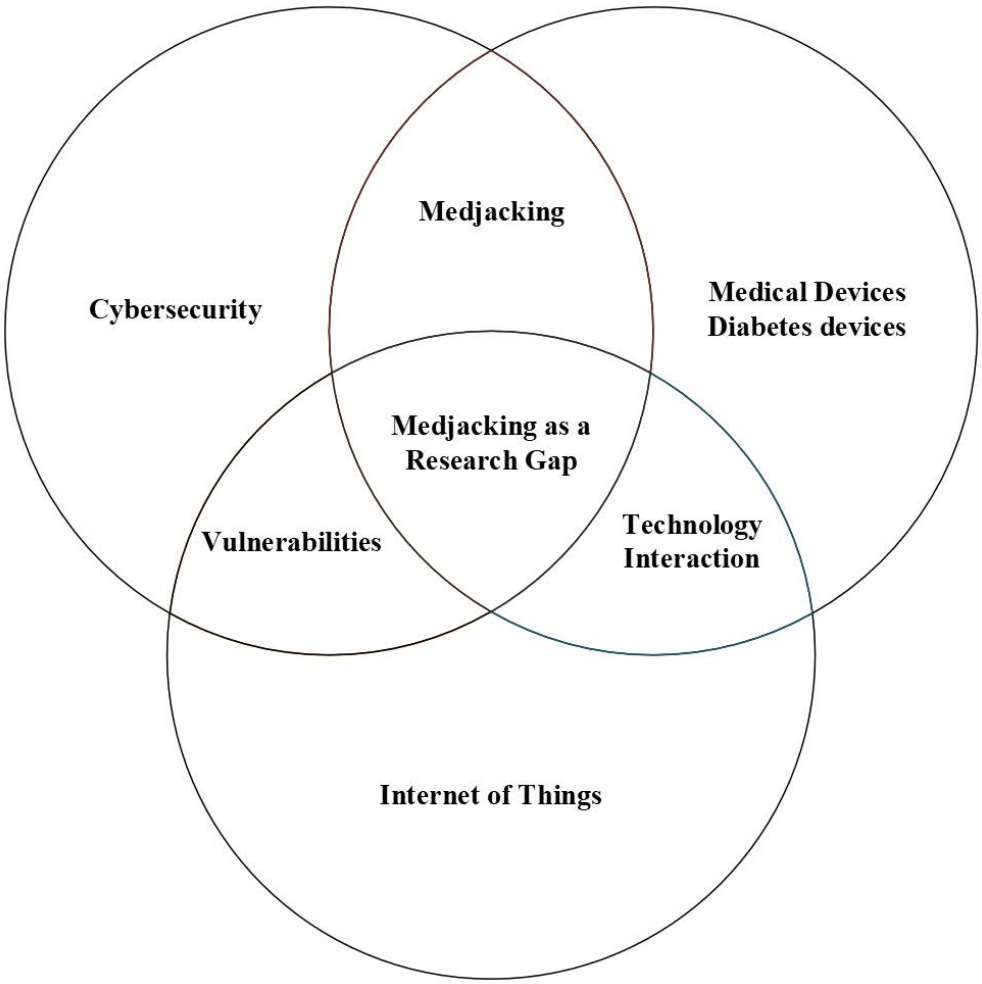


Figure 1 outlines a Venn diagram illustrating the intersection of the Internet of Things, cybersecurity, and medical devices. Owing to limitations in IoT, it introduced significant challenges in managing cybersecurity. These devices are essential for patients to manage or monitor their disease using IoT; however, IoT medical devices exhibit insufficient cybersecurity measures and other security weaknesses, rendering them susceptible to cybersecurity attacks. Inadequate defences against attacks are further exacerbated by the inconsistency of security standards related to IoMT devices. The proliferation of IoMT devices creates significant concerns regarding unauthorised access, thereby enhancing the security challenges inherent within this ecosystem.

Medjacking attacks on medical devices imperil cybersecurity. The unauthorised manipulation of devices or systems enables adversaries to exploit vulnerabilities caused by network communication protocols. Cybercriminals exploit vulnerable devices to appropriate sensitive information, undermine operational integrity, or gain unauthorised access. The consequences are severe when such attacks impact critical systems, such as medical devices. Any act of medjacking or other cybersecurity attacks could compromise patient safety [82, 94, 178, 180-182], as these devices are required to operate continuously. The limitations of the IoT, challenges associated with cybersecurity, and the integration of implanted medical devices present a significant challenge. Medical devices are increasingly integrating into the IoT ecosystem, facilitating continuous monitoring and data interchange yet simultaneously increasing its vulnerability to cyber intrusions.

1.4 Research Questions

This research sought to answer the following research questions:

1. What vulnerabilities and attack methods exist within the Bluetooth and Internet of Things communication protocols in the APS?
2. What is the outcome on an APS in a simulated attack environment?
    a. How can an appropriate dataset be developed to effectively study medjacking vulnerabilities in APS?
    b. What exploratory data analysis technique can be used to explore data patterns of medjacking methods, including brute force, DDoS, MITM, and Trojan attacks?
3. What machine learning and deep learning techniques as proactive defences techniques can be used to detect medjacking vulnerabilities in an APS?
4. What dimensionality reduction technique can be performed optimally in the development of a machine learning model for medjacking detection in the APS?

a. What differences can be experienced with and without dimensionality reduction techniques in the development of a machine learning model for medjacking detection in the APS?

1.4.1. Research Aims/Objectives

The main objective of this thesis is to develop and examine a proposed hybrid intrusion detection system for APSs to effectively detect medjacking attacks, focusing on addressing brute force, DDoS, MITM, and trojan attacks. This study's objective is to conduct an exploratory data analysis to explore data patterns, providing insights into the different attacks and normal patterns. This study acknowledges the complex and dynamic characteristics of medjacking attacks; therefore, this study concedes that it addresses a limited component of the overarching issue of medjacking, rather than providing a comprehensive elimination of a variety of threats associated with medjacking in IoMT devices. As discussed in Section 1.3.3, a heterogeneous nature in an APS creates a new vulnerability that can potentially lead to significant consequences. Therefore, a medjacking study to investigate the security of APS devices, and to develop a medjacking detection strategy, is essential embedding a reliable and secure APS device in a diabetes patient is crucial. The development of the APS devices and medjacking attacks exemplification to collect dataset is equally important. This allows for the development of medjacking attacks detection system. To meet the main research objective, the following aims have been formulated:

- The first research aim was to analyse existing literature on technologies used in the APS, such as Bluetooth and IoT communication protocols, focusing on known vulnerabilities and potential attack methods.

- The second research aim was to simulate medjacking attacks on APS devices to determine the vulnerabilities within Bluetooth and IoT communication protocols, aiming to identify potential successful attack vectors.
- The third research aim was to gather dataset comprising normal and attack data to conduct the exploratory data analysis. This dataset was obtained during attacks to APS in a simulated attack environment. The dataset was required for exploratory data analysis to identify the correlations with one another data features.
- The fourth aim was to examine existing techniques in literature to identify appropriate technique to analyse the medjacking methods, including brute force, DDoS, MITM, and Trojan attacks.
- The fifth research aim was to examine existing proactive defence mitigation techniques, with a focus on intrusion detection and Machine Learning (ML) methods that could be applied to the APS, and to identify the limitations of these existing techniques.
- The sixth research aim was to systematically investigate and develop proactive techniques to automatically detect intrusions in an APS, thereby enhancing security and identifying potential threats.
- The seventh aim was to conduct a performance evaluation of the developed H-IDS to validate its effectiveness in detecting intrusions.

1.4.2. Justification of How the Current Work Contributes to a Deeper Conceptual Understanding of Medjacking Attacks

In line with the first research question, the primary objective of this study was to consolidate the current challenges related to medjacking within the APS, identify potential strategies for minimising medjacking vulnerabilities, and examine and evaluate ML techniques significant to medjacking. Despite the progress accomplished in the field of diabetes digital devices, led by the emergence of innovative technologies, the investigation into security,

especially in the context of medjacking attacks, continues to be significantly limited. Furthermore, the significance of medjacking within the context of the IoMT is elaborated in Section 1.2.1. The formulation of the first research question aims to investigate medjacking attacks on APS devices, which is considered an important step in strengthening APS systems. This research established a correlation between theoretical perspectives and guided the identification of potential mitigation strategies. This investigation lays the foundation for future progress in APS security by amalgamating research enquiry with empirical experimentation. A proposed solution in this study for detecting medjacking aims to enhance the security of APS devices. This study identifies and analyses existing attacks and synthesises theoretical insights derived from this study and current literature. Through synthesising insights derived from prior research, this study offers a substantial contribution to the understanding of medjacking attacks, establishing an essential basis for mitigating such threats in APS devices.

In line with the second research question, this study systematically synthesised the existing literature and conducted a comparative analysis of the various ML techniques used to detect medjacking attacks. The current study involves the development of the APS prototype and the demonstration of attacks within a controlled simulated environment. In line with the second research sub-question (b), The formulation of a dataset was essential to this research, as it directly assisted in the identification and proposal of techniques aimed at mitigating medjacking. This investigation also provides a thorough rationale for the datasets outlined in Chapter 2, Section 2.7, which examines the constraints of related datasets, in addition to a detailed justification for dataset sizes in Section 6.3.3, entitled Justification of Dataset Sizes. This investigation further provided a comparative examination of relevant datasets in section 2.7.1, titled Related Datasets Comparison. This study required the dataset to conduct an exploratory data analysis to explain data patterns, thereby yielding insights into various attacks and normal behaviours. Furthermore, an intrusion detection system was developed, this was

essential for evaluating the effectiveness of the study's proposed mitigation strategies. This study evaluates various Machine Learning techniques within the framework of the proposed solution for APS devices. This research critically assesses a range of Machine Learning methodologies within the context of the proposed solution for APS devices. This research, via the addition of a sub-question (b), aims to conduct an analysis of data patterns relevant to medjacking, thereby promoting a more nuanced understanding within this domain.

In line with the third research question, this study contributes novel insights to the research community, facilitating the selection of the most effective machine learning model for the detection of medjacking attacks in the APS. This research addresses the medjacking detection in the APS through systematically evaluating ML and DL techniques. This study highlights the importance of proactive defence strategies, emphasising the identification and detection of threats (attacks) in sections. All these explanations are provided in sections: 2.7.1 Justification of How ACD Security Measures Apply to IoMT Devices, 2.7.2 Justification of How Al-Driven Security Helps Prevent an Attacker from Tampering with an IoMT Device, and 2.7.3 Justification of How ACD Systems Monitor Communication between IoMT Devices to Detect Anomalies that might Indicate a Medjacking Attempt. Chapter Five outlines and evaluates a comprehensive framework for the detection of medjacking in the APS. The effectiveness of the proposed detection system is outlined in Chapter Six. This research investigates ML and DL techniques to assess the performance relevant to medjacking. The performance insights yield detection performance matrices and the theoretical discussion of each machine learning model is presented for the APS medjacking. The findings establish a fundamental framework for the implementation of detection systems in APS devices.

In line with a fourth research question, this study substantially examined the understanding of dimensionality reduction techniques in the development of ML and DL models aimed at identifying APS medjacking attacks. An essential advancement of this

study lies in its comparative technique, which examines three dimensionality reduction techniques (such as PCA, t-SNE, and UMAP) in conjunction with trained models and models examined without dimensionality reduction. This dual perspective (with and without dimensionality reduction) offers a profound comprehension of the influence of dimensionality reduction on the effectiveness of medjacking detection performance. This research presents the statistical analysis and evaluation of performance outcomes with and without dimensionality reduction. This contributes to the existing body of knowledge regarding optimising machine learning pipelines for detecting medjacking. The study enhances theoretical understanding and provides practical proposals for the application of dimensionality reduction in the detection of APS medjacking. This contribution can impact forthcoming research endeavours aimed at developing effective solutions for detecting medjacking.

In line with the fourth research sub-question (b), this study provides substantial insights into optimising medjacking detection mechanisms for APS. By demonstrating the extensive implications of dimensionality reduction findings, the research significantly enhances the body of knowledge pertaining to both practical applications and prospective future research. The use of dimensionality reduction may significantly enhance the effectiveness of medjacking detection in the APS. The comparison of scenarios "with and without" dimensionality reduction illustrates how ML and DL models identify anomalies. For APS, understanding these distinctions is essential for the development of robust detection mechanisms designed to protect against medjacking. The findings lay a groundwork for future enquiries to explore the relationship between specific dimensionality reduction techniques, including PCA, t-SNE, and UMAP, and the unique data patterns of APS. This study enhances both empirical and theoretical understanding by identifying and quantifying the variances in model behaviour dependent upon implementing dimensionality reduction methodologies.

1.4.3. Research Questions and Research Objectives Limitations

This study's research questions and objectives acknowledge several limitations that require additional investigation. The first limitation is related to the need for formal mathematical models, which is a great research direction for mathematicians in the domain of developing formal mathematical models. The current study primarily utilises simulations to assess the effectiveness of attack detection mechanisms within APS. Although this approach yields significant insights into the patterns of attacks, the lack of formal mathematical models for APS communication protocols and the dynamics of attack lifecycles refers to a limitation. Future work will focus on developing formal mathematical models to establish a more robust and systematic foundation for comprehending the complex dynamics of APS communication and the potential possibilities for attack through formal mathematical models.

Secondly, this study has not examined the attack lifecycle and its various stages related to APS systems. In the context of future research endeavours, this study recommends establishing a systematic formal mathematical model for comprehending the attack lifecycle, encompassing critical stages such as reconnaissance, exploitation, and persistence.

Third, the utilisation of a simulated environment in this research serves as a robust mechanism for the identification of potential attacks within APS, thereby enhancing practical applicability. The principal limitation on real-world attack scenarios and proprietary devices for the purpose of research investigation lies in their necessity for substantial ethical approval, and research budget. Nonetheless, simulations cannot fully replicate the complexities of real-world environments, including real-world attacks aimed at proprietary devices for research purposes. These attacks could potentially pose significant damaging or life-threatening risks to both patients and systems, as well as the financial risks of conducting real-world research on real devices. This study contributes substantially to comprehending medjacking attacks within controlled environments; however, subsequent investigations will seek to validate these results

in real-world settings and incorporate formal mathematical models and attack lifecycles to enhance their practical significance.

1.5 Research Methodology Overview

This study's research methodology is segmented into stages, and comprehensive information about each stage is provided in Chapter 3. These stages encompass creating the research design and research approach to develop the APS prototype and a security attacks exemplification, collecting a dataset, and developing a H-IDS, ultimately, evaluate the performance of the H-IDS. Standard tools and services are required to develop an APS prototype and illustrate security attacks. For example, many IoMT devices use low-cost microcontrollers [138, 179, 206]. Thus, the development of the prototype for this research involved the use of microcontrollers. The APS prototype utilised cost-effective microcontrollers and a simulation approach to examine medjacking attacks. This study compiled a comprehensive list of research activities utilised to attain the research objectives; Table 1.2 lists these activities.

Table 1.1

The Overall Research Activities Associated with Outcomes and Tasks.

No.	Activities	Outcome	Tasks
1	Conduct a comprehensive literature review	Select a device or application	1. Conduct literature review on the IoMT, APS, cybersecurity risks, and defence mechanisms
			2. Synthesise and classify the APS
			3. Synthesise and classify cybersecurity risks
			4. Synthesise and classify defence mechanisms
			5. Systematically arrange a decision matrix

No.	Activities	Outcome	Tasks
2	Finalise the research direction	Design the research questions and objectives	1. Identify the research background and gap 2. Formulate research questions and objectives 3. Choose a research methodology 4. Define research outcome
3	Present the research methodology	Create research design	1. Define Research design 2. Research approach
4	Develop the APS prototype	Develop the device or application	1. Simulate an APS device 2. Execute the APS device simulation
5	Apply architectural components	Apply de facto protocol	1. Apply de facto protocol 2. Execute protocol
6	Create and implement security attacks on the APS prototype	Choose and implement security attacks	1. Use different types of infiltration
7	Collection of a dataset	Collection of normal and attack data	1. Extract packet-based features from the packet capture data and transform into the (CSV) file format
8	Exploratory Data Analysis	Identify the correlation of different attacks	1. Apply Pearson Correlation Coefficient
9	Develop potential solutions for the H-IDS to detect medjacking	Develop intrusion detection	1. Develop and implement intrusion detection mechanisms to detect medjacking attacks 2. Validate intrusion detection
10	Validate outcome	Present results and discussion	1. Analyse H-IDS performance 2. Determine future research directions

Chapters 1 and 2 present the research background and finalise the research questions and objectives displayed in Table 1.2. The research methodology is presented in Chapter 3. Chapter 4 discusses APS development, security attacks, data collection, and exploratory data

analysis. Chapter 5 presents the development of a potential solution. Chapter 6 presents the results and discussion. Conclusion and future research are present in Chapter 7.

1.6 Research Significance and Contributions

This research investigates APS device medjacking attacks, a crucial step in strengthening APS systems. This study correlated theoretical perspectives and proposed mitigation strategies. This research-experimentation study lays the groundwork for APS security advancement and proposes improvements to the security of APS devices to detect medjacking. This study analyses existing attacks and synthesises theoretical insights from this study and current literature. This study contributes to understanding medjacking attacks by synthesising prior research and laying the groundwork for mitigating such threats in APS devices. This research's quantitative and theoretical contribution is to assess the quantitative performance of different classification models' effectiveness using the 2022APS dataset. Based on quantitative performance, this study presents the theoretical findings of each model. The significance of this study aimed to examine the existing medjacking vulnerabilities and attack methods, contribute to the building of security defences, strengthen security, and protect end-users. This research provides practical insights into the risks associated with APS medjacking. The process of gathering a dataset comprising normal and attack data is crucial for effectively conducting Exploratory Data Analysis (EDA). This study contributes to providing the EDA of the APS devices. This thesis also provided a detailed justification for datasets in Chapter 2, section 2.7, Related Datasets Limitations, and dataset size justifications in Section 6.3.3, Justification of Dataset Sizes. This study also provided a comparison of related datasets in section 2.7.1 Related Datasets comparison.

EDA, particularly through the application of Pearson correlation, illustrates the comprehension and representation of the complex patterns associated with medjacking attacks within the context of the APS. The findings from EDA elucidate the relationships among

various data patterns and provide insights into the fundamental structure of the data patterns, which is essential for the development of effective security measures. EDA technique assists with data-informed decision-making and establishes a foundation for the development of robust intrusion detection. Through the application of EDA, this study fills research gaps in the literature by demonstrating and examining the complex data patterns linked to APS medjacking attacks. EDA offers an enhanced comprehension of attack dynamics, illustrates feature significance for predictive models, and assists in advancing more resilient, data-informed models in future research.

The investigation and development of proactive techniques to automatically detect intrusions in APS devices hold significant potential to protect against attacks. Automated techniques minimise the risks of medjacking and promote evidence-based detection using ML. Systematically developing these techniques could enhance security measures and identify threats more effectively, aligning with the overarching goal of safeguarding APS devices from intrusions. The positive results from the experiments conducted in this study on the proposed H-IDS demonstrate that the system can effectively detect attacks, further highlighting its importance in advancing APS cybersecurity.

The novelty of this research lies in identifying medjacking attacks within APS devices through developing and testing a robust detection system. The study incorporates an advanced selection of features to improve the detection process and systematically examines the effectiveness of various machine learning models, encompassing MLP, SVM, DT, and XGBoost in the context of signature detection. This study employs an autoencoder for anomaly detection. The comprehensive evaluation of these models provides valuable insights into their effectiveness in detecting attack scenarios involving brute force, man-in-the-middle, distributed denial-of-service, and trojan attacks.

This study makes a significant contribution by addressing the gaps in examining the medjacking attacks, both with and without dimensionality reduction techniques. This study uses dimensionality reduction techniques, including PCA, t-SNE, and UMAP, to examine the impact of feature space reduction on the effectiveness of detection systems. This analysis offers an enhanced comprehension of the limitations between different model effectiveness and different dimensionality reduction techniques, thereby advancing the formulation of more efficient and effective intrusion detection strategies for APS devices. Moreover, this study fills significant gaps in the existing literature by assessing dimensionality reduction techniques to improve the identification of medjacking attacks in APS devices.

The use of dimensionality reduction may significantly enhance the effectiveness of medjacking detection in the APS. The comparison of scenarios "with and without" dimensionality reduction illustrates how ML and DL models identify anomalies. For APS, understanding these distinctions is essential for the development of robust detection mechanisms designed to protect against medjacking. The findings lay a groundwork for future enquiries to explore the relationship between specific dimensionality reduction techniques, including PCA, t-SNE, and UMAP, and the unique data patterns of APS. This study analyses model behaviour variations due to dimensionality reduction methods, improving empirical and theoretical understanding. This research offers theoretical and practical perspectives on enhancing the security of APS devices. For future research, it is recommended to consider the application of generative AI models to significantly enhance medjacking detection. In addition, the incorporation of formal models is also recommended for subsequent research aims.

1.7 Thesis Organisation

This thesis focuses on describing the medjacking challenges and feasible cybersecurity solutions to detect these challenges in the APS. The study examines medjacking infiltrations using an APS prototype and presents a detection solution that strengthens security measures.

Seven chapters are included in this thesis. The introduction and background information in this chapter (Chapter 1) highlight the research rationale and goals. The research questions, objectives, focus, and motivation were also discussed.

In Chapter 2, the literature review comprehensively analyses existing research on cybersecurity and threats in the APS. This review presents a thorough examination of current frameworks for intrusion detection systems. Chapter 3 discusses the research design and research approach, including the APS prototype and cybersecurity attacks, data collection, and analysis.

Chapter 4 discusses the APS prototype's deployment and test with the medjacking attacks. This chapter also includes the dataset collection and evaluation outlined in Table 1.2. Chapter 5 presents the proposed H-IDS. This proposed solution was thoroughly analysed in Chapter 6 and verified by consulting existing literature. The conclusion and discussion are provided in Chapter 7 by encompassing a concise overview of the findings, an in-depth exploration of their implications, and recommendations for future research.

1.8 Summary

This chapter presented a comprehensive strategy to enhance cybersecurity measures and detect medjacking vulnerabilities that are becoming more prevalent in APS devices. A formal study was introduced to combine the latest techniques of medjacking detection in an APS. Research questions and objectives were established as part of this study. The chapter provided an overview of the comprehensive examination and verification research activities that were carried out in this study, including the analysis of relevant literature and the simulation of medjacking scenarios to gather the dataset, ascertain correlations with the typical APS operations, and use the H-IDS technique to identify and detect medjacking. Further, the chapter discussed how advanced ML techniques and quantitative analysis were used to differentiate between legitimate activities and medjacking attempts. The knowledge acquired

from this research is valuable for ongoing assessment and enhancement of each APS's ability to identify and respond to threats.

Chapter 2. Literature Review

This chapter presents a comprehensive examination of the literature, specifically addressing the cybersecurity of Artificial Pancreas Systems (APS), and automated defence mechanisms. The main objective of this Chapter is to examine and consolidate current research, identify areas that need further investigation, and bring attention to new developments in APS security. Following that, the existing countermeasures are then presented, demonstrating the latest advancements in the field. Ultimately, the research has gathered the significant instances of medjacking offences in the APS. Exploratory data analysis techniques are crucial for detecting patterns that indicate trends and direction in the data is also present in this Chapter. The security of APS ecosystems presents a complex challenge that requires continuous improvement and constant monitoring. This chapter addresses the first, second B, and third research questions:

1. What vulnerabilities and attack methods exist within the Bluetooth and Internet of Things communication protocols in the APS?
2. (b) What exploratory data analysis technique can be used to explore data patterns of medjacking methods, including brute force, DDoS, MITM, and Trojan attacks?
3. What machine learning and deep learning techniques as proactive defences techniques can be used to mitigate medjacking vulnerabilities in an APS?

2.1 Research Technical Background

The Internet of Medical Things (IoMT) has revolutionised healthcare by providing various benefits for patients, healthcare professionals, and service providers. From a technological perspective, the IoMT has improved the ability of General Practitioner (GPs), patients, and third parties to self-manage the delivery of healthcare services. The most-used IoMT devices are wearable, and devices such as smartwatches, fitness trackers, and biosensors

are beneficial. Wearable devices may improve patient outcomes, illness management, and healthcare expenditures, and patients' hospital stay time and costs have been reduced [155, 207].

Machine Learning (ML) models and data analytics can be added to discover trends, patterns, and predictive insights that can guide patient-specific treatment plans or customised treatment methods [155, 208]. A data analytics approach to healthcare can reduce chronic illness burden and improve quality of life through early treatments. ML applications in diagnostics, operational efficiency, and security are crucial to the IoMT [209-212]. Predictive analytics [18, 19, 213] uses ML to predict health outcomes from IoMT data streams [18, 213]. ML models can predict patient health, including heart failure, by assessing heart rate, oxygen saturation, and blood pressure data. For instance, some advanced ML algorithms, such as Long Short-Term Memory (LSTM), are adept at managing the temporal characteristics inherent in IoMT data, thereby facilitating precise predictions of blood glucose levels [214]. Wearable sensors can also track Parkinson's illness by analysing gait and movement patterns with ML classifiers like SVM [215, 216]. ML-enhanced IoMT systems can detect diabetic retinopathy in wearable ophthalmic devices, enabling timely treatment [217-221].

The integration of ML within the IoMT presents various opportunities; however, it concurrently introduces a range of substantial challenges. IoMT devices generate large amounts of real-time data. However, their dependability and consistency may present significant challenges [209]. For example, IoMT sensors provide noisy data due to environmental variables, poor device use, or sensor degradation [20, 209, 222]. The presence of such noise may result in inaccuracies within ML model predictions, thereby influencing pivotal healthcare decisions. Various methods are being developed for IoMT data quality, given that research in this area is ongoing.

Interoperability challenges in IoMT ecosystems present a significant concern. The IoMT often includes devices from different manufacturers that use different data formats and communication protocols [223-225]. The absence of standardisation obstructs the rapid integration of ML models across various devices. Furthermore, the diverse data characteristics produced by these devices cause the complexities associated with preprocessing, which is crucial for the effectiveness of ML training and inference [226-228]. ML integration into the IoMT is crucial. Hence, the research community is actively developing and accessing several strategies to enhance ML in this domain, including lightweight ML models and security and privacy-preserving methods.

Wearable devices allow patients to manage their health, for example, an implanted glucose monitoring system can autonomously regulate blood glucose levels by administering insulin to the patient [229]. Further, the advancement of energy-efficient wearable biosensors and ultra-low power body area networks [74, 230], implantable medical devices [231], and a variety of lightweight communication protocols [232] have enabled the development of compact healthcare devices.

IoMT devices and applications, including implanted and wearable biosensors, have improved smart healthcare infrastructure. Wearable devices, including smartwatches [49], fitness trackers [44, 47, 233], and health monitoring systems [14, 64, 67], have experienced substantial development, attributable to their capacity for simple incorporation into daily life. Their diverse applications render them vital tools across various domains, including healthcare and personal fitness. The capacity of wearable devices to facilitate the real-time monitoring of health metrics constitutes one of their most significant transformative advantages [11, 47, 61]. Devices equipped with sensors can monitor various parameters, including heart rate, blood oxygen level, blood pressure, sleep patterns [11, 62, 63, 234], and even blood glucose levels. The constant monitoring facilitates the prompt identification of health complications, as

anomalies in cardiac rhythms, levels of oxygen, or elevated levels of glucose can be discerned prior to their progression into severe medical conditions. Furthermore, wearable devices facilitate the management of chronic diseases in patients diagnosed with conditions such as diabetes [73].

The development of networked medical devices, including insulin pumps [81, 82, 86-89], CGM [81, 90-92], pacemakers [93-98], and infusion pumps [99], has made them vulnerable to hijacking and software failure. Tandem Diabetes recalled its iOS t:connect app due to a technical glitch that caused it to crash and reload in a loop, resulting in an abrupt pump shutdown [161]. According to [162], nearly 200 patients were injured when their insulin pumps stopped down unexpectedly due to battery depletion. Tandem Diabetes Care's t:connect mobile app was recalled over 85,000 times due to a software error. Due to this software error, approximately 200 diabetics were injured.

The FDA warned consumers and healthcare providers about Medtronic insulin device cybersecurity vulnerabilities [165]. The FDA has recalled 627 medical devices [235] due to software defects, also known as coding defects or bugs in code, software, or firmware, that allow attackers to exploit security weaknesses [236]. Types of software defects include buffer overflow, structured query language injection, and Cross-site Scripting, also known as XSS [237]. IoMT security weaknesses have been discussed both by the media and academia [103, 237-239]. For example, the United States former Vice President, Dick Cheney’s pacemaker story became popular in mainstream media when his doctor deactivated the wireless connectivity of the pacemaker for fear of it being hacked. Revisiting this story, researchers verified the hackability of pacemakers [93]. Another study argued that the Distributed Denial-of-Service (DDoS) attacks and hijacking attacks are severe challenges to IoMT device security, yet the researchers were unable to propose appropriate mitigation solutions for these challenges [240]. In addition, researchers have identified multiple cyber assaults on implantable cardiac

defibrillators, including remotely deactivating and reprogramming defibrillator therapy [241]. Implantable devices are more vulnerable to medjacking attacks, and researchers have demonstrated security threats for a diabetes treatment system by hijacking an insulin pump [242]. According to [164], some IoT manufacturers and service providers create devices without sufficient security, making them vulnerable to cyber attackers. Identity theft, eavesdropping, MITM, and energy depletion may attack IoT devices. In another study [103], researchers stated that data modification attacks involve breaking the device or intercepting and modifying healthcare-programming device communication packets to change a patient's vital signs. Real-time medical device vulnerabilities threaten patient safety, privacy, and healthcare system integrity [78, 80, 103-109].

Industries have limited standardised protocols for efficiently updating security. Due to profound and deleterious health implications, a proactive approach to tackling any security concerns associated with healthcare systems can be progressive. Unfortunately, current studies present limited comprehensive security measures to successfully counter the increasing number of cyberattacks on wearable and implanted devices. Several countermeasures, such as privacy-preserving communication protocols and encrypted databases, have been suggested by scholars. However, it is important to note that these measures are currently insufficient in mitigating cyberattacks.

2.2 Internet of Things

The Internet of Things (IoT) refers to interconnected physical objects, such as home appliances, wearable equipment, and other objects using software and network connections that are linked over the internet [121]. The IoT allows people and objects to connect with the digital world and improve digital lifestyles. The IoT devices send and receive data from a single device or multiple devices that are networked to generate or measure any activity and perform a task. The data are then delivered to the linked devices within a connected network [121]. The

smart home is one of the most significant examples of the IoT. A user may gather, analyse, and share data to use certain equipment, such as a smart lock, light, refrigerator, or air conditioning, with a central computing system or a smartphone [142, 243].

Smartwatches, fitness trackers, smart thermostats, and home security objects (e.g., camera or other security device) are other examples of objects used in IoT ecosystems. These smart objects can connect and communicate with other systems for additional processing and analysis using application domains including web portals and software. A recent study of IoT technologies and applications clearly showed that the development of environmental monitoring, healthcare monitoring, public safety surveillance, smart city technology, smart manufacturing systems, and intelligent transportation systems has begun [122].

The IoT denotes a system of interlinked physical devices, equipped with sensors, software, and communication technologies [244-246], which facilitate the collection and exchange of data via the internet or other communication networks [246-249]. These devices, frequently referred to as "things," encompass a diverse array of entities, from smart home things such as smart thermostats [250] and refrigeration units to industrial machinery [251-253], healthcare devices [11, 37, 64, 133, 254], and autonomous vehicles [255-261]. The IoT may not confined to the parameters of a Personal Area Network (PAN); rather, it can effectively utilise the technologies associated with PANs [262]. PAN technologies exhibit utility for localised, low-power connections [263, 264]; however, they lack the capacity to operate effectively over extensive distances [265]. In solving this inconsistency, IoT systems frequently utilise protocols such as MQTT across Wide Area Networks (WANs) to facilitate data transmission from PAN devices to cloud platforms or remote systems [121, 223].

Similarly, the IoT is not fundamentally classified as a Wide Area Network (WAN); however, it can leverage WAN technologies to facilitate communication over extensive distances [266-268]. WANs are indispensable in IoT architectures by facilitating data

transmission from geographically dispersed devices to centralised servers or cloud platforms [269-271]. For instance, MQTT is a protocol that has garnered extensive adoption in IoT WAN contexts, attributable to its lightweight characteristics, and message transmission [272]. The IoT isn't limited to a singular network or communication framework category. The IoT constitutes an ecosystem that dynamically integrates diverse technologies to address particular use cases [273]. The IoT additionally utilises WAN to enable the remote monitoring and management of devices over extensive distances. In these contexts, the MQTT protocol assumes a pivotal function in the transmission of data [121, 223].

Smart devices allow humans to easily make decisions or complete tasks since IoT devices are connected to the network, and these devices interact and exchange information with each other. As a result, the IoT ecosystem has become a need in everyday life, and these ecosystems are continuously and rapidly developing with cutting-edge technologies. Short-range radio technologies such as Bluetooth low energy (BLE), Near Field Communication (NFC), Radio Frequency Identification (RFID), and Zonal Intercommunication Global-Standard (Zigbee) are widely used to enable connectivity and communication between devices [122]. In general, BLE, long-term evolution, wi-fi, and Zigbee technologies are commonly used for connecting devices to an IoT network [274].

IoT software and hardware creators are at the forefront of this development and are having a significant impact on several industries, including healthcare, transportation, construction, and agriculture. Manufacturers are progressively producing new ubiquitous gadgets [140, 275]. Unfortunately, connecting IoT devices over the internet is a primary concern, as it poses major cybersecurity vulnerabilities related to security, privacy, trust, confidentiality, integrity, and safety [276-278].

2.2.1 Internet of Medical Things

The IoT facilitates the effective communication of health data, enables remote monitoring of patients, enhances healthcare service delivery, and empowers individuals to take control of their medical conditions using digital devices [14]. The term IoMT describes the widespread use of interconnected hardware and software in healthcare systems to create, transmit, and analyse patient health data. Wearable devices, remote patient monitoring systems, networked medical equipment, health applications, and data analytics platforms are among the items and technologies covered by the IoMT. IoMT technology has enhanced the delivery of in-house, in-hospital, and remote healthcare services. The IoMT provides easy access and real-time data, thereby providing a platform for integrating services with emerging embedded technology such as telecare, which is a term used for remotely monitoring real-time emergencies. The risks of living alone with medical conditions can be better managed with the help of telecare. Telecare can be described as mobile health (mHealth) or electronic health (eHealth); these two terms are more general and apply to healthcare services that are accessible through mobile devices [279].

The delivery and quality of medical services are now reliant on IoT technology. This technology enables GPs, patients and third parties to one-way communication and monitor (disease, patient, time series real-time data etc.) remotely and in real-time. Although direct communication between parties may be limited, IoT devices enable unidirectional communication by relaying patient health information to GP for the purposes of efficient monitoring or agreed services. According to Hood et al., Remote Patient Monitoring (RPM) acts as a form of telehealth wherein healthcare providers monitor patients beyond the parameters of conventional care environments through the utilisation of digital medical devices, including weight scales, blood pressure monitors, pulse oximeters, and blood glucose metres. The data collected from these devices are subsequently transmitted electronically to providers

for the purpose of care management. Automated feedback mechanisms and workflows can be integrated into the data collection process, allowing for the identification, and flagging of out-of-range values or anomalous readings of concern. Historically, RPM has served as a pivotal tool in the assessment of symptoms associated with chronic conditions, including cardiovascular diseases, diabetes mellitus, and asthma. Patients may have encountered this phenomenon via wearable devices, such as Holter monitors, which can measure cardiac rhythms to facilitate the remote detection of cardiac conditions and the monitoring of cardiovascular diseases [280].

Pradeesh et al. proposed an IoT-based smart E-Inhaler designed for asthma patients, which simultaneously monitors symptoms and geographical location in real-time, thereby facilitating immediate adjustments in medication at the time specified. The researcher stated that the proposed technique facilitates the remote monitoring of patient health conditions at any time and from any location, thereby enabling healthcare professionals to access, analyse, track, and monitor patient health status in real-time. The developed application provides a notification when the user either misses or exceeds the dosage. Subsequently, the doctor's prompt attention is taken in instances of serious medical conditions. An alarm has been set for the user to be used as a reminder to administer their inhalation in accordance with the doctor's recommendations [281]. The IoMT delivers error-free treatment management and health assistance to patients while improving end users' self-management and satisfaction [65, 66, 154]. IoMT devices, services, and applications are used remotely or within inpatient and outpatient settings [155]. IoMT technology used in treatments is age-friendly [282], and devices can be wearable, implanted, or based in the patient's home (e.g., smart kits).

Research has shown that IoMT technology has many potential uses, particularly for the long-term management of chronic conditions such as obesity and diabetes [283]. Diabetes involves long-term treatment, and IoMT devices have also been recognised as promising for

supporting the self-management of glucose levels and insulin calculation [284]. Table 2.1 summarises some of the benefits and limitations of IoMT technology.

Table 2.1

Benefits and Limitations in Medical Devices, Services, and Applications Using Internet of Medical Things Technology.

Category	Benefits	Limitations
IoMT services	Cost-efficiency, real-time diagnosis, self-evaluation, early diagnostics, early emergency prevention, patient assistance, and satisfaction.	Development of cross-platform software and scalability, security, and privacy limitations.
IoMT embedded devices	Real-time communication with wearable medical devices and hospital equipment and monitoring of hospital equipment and devices.	Computational, energy and memory constraints, and accuracy, scalability, security, and privacy limitations.
IoMT wearable devices	Real-time tracking of information; activity monitoring (e.g., heart rate and continuous glucose monitoring); reduction of costs and hospital stays; and understanding of sleep cycles, heart rate, and steps.	Body injury risks; computational, energy, and memory constraints; and reliability, accuracy, scalability, security, and privacy limitations.
IoMT applications	Leverage real-time control, health situational awareness, process optimisation, and connected healthcare appliances in the house and hospital.	Embedded software and dynamic security update constraints; and scalability, security, and privacy limitations.

According to Ardito et al., Situation Awareness (SA) involves constantly monitoring and using knowledge to reduce risk. SA approach is growing in eHealth, where clinics, IT firms, and AI researchers comprehend the importance of awareness and clinical decision-making. Telemedicine is a subfield of eHealth that is essential for remote patient monitoring at home. The eHealth and ageing society adopt Ambient Assisted Living (AAL) through “domiciliary hospitalisation” to increase patient support at home. AI, IoMT, and mobile technologies can help patients at home by monitoring vital parameters, assessing health conditions, and

requesting clinical staff in emergencies. SA in eHealth can be done by using the "new 3-d paradigm" devices, data, and diagnoses (predictive). AI algorithms can process all the data from devices, to develop predictive diagnoses that promote, validate, and adapt to patient home activities [285].

Lofu et al., proposed a situation-awareness computational intelligence model for metabolic syndrome management. The researchers stated that the IoT has experienced extensive adoption in home and clinical care settings. Consequently, it is feasible to set up a contextually aware IoT smart home and health environment through the following techniques: (i) provisioning sensed data to facilitate the monitoring of environments, (ii) detecting situations predicated on documented event logs, and (iii) initiating an action in response to the recognised situations. The researcher stated that a situation-aware system in eHealth facilitates the personalisation of therapeutic pathways for each patient, taking into account the biological characteristics of the pathology, specifics of the clinical history, and the contextual factors of the living environment [286].

Chen et al., define scalability as computational efficiency, lower communication overhead (e.g., how often a device needs to communicate with the remote cloud centre), and compact information needed (e.g., what type of information a device needs before making sensible decisions) [287]. Similarly, Pradyumna et al., demonstrated that significant growth in real-time data interchange and remote monitoring introduced scalability challenges in healthcare network infrastructure. Scalability refers to creating networks that can handle more IoMT devices and their data. Scalability enables healthcare to extend its IoMT ecosystems without network congestion or performance deterioration [288].

According to the literature, health situational awareness within the context of the IoMT refers to an ability to acquire, analyse, and respond to real-time data regarding a patient's health condition through the utilisation of interconnected medical devices and data systems. This

notion illustrates the significance of continued monitoring, the synthesis of data, and the contextual examination to improve decision-making processes within the healthcare sector. Furthermore, according to the literature, scalability within the context of the IoMT refers to the system's capacity to effectively contribute to its capabilities, functionalities, and scope in response to growing requirements while maintaining optimal performance and reliability.

2.2.1.1 Internet of Medical Things Embedded Devices

An embedded device is defined as an object connected to another object. IoMT technology encourages the development of embedded IoMT devices ranging from tiny to large sizes. Different tasks can be assigned to tiny sensors, RFIDs, and wearable devices depending on their purpose. The embedded devices then provide information or data to other interconnected devices (e.g., a smartphone, smartwatch, laptop, and iPad). Embedded devices are classified as implantable or wearable devices that send and receive data. In certain circumstances, these embedded devices send real-time health alerts to the patient and subsequently send them to appropriate healthcare facilities [289]. Therefore, both GPs and patients can access or monitor real-time data.

Smart diabetic socks are an example of embedded devices. These socks are a wearable IoMT system designed to prevent diabetic foot disease [290]. In another example, a research study proposed embedded wearable devices to manage diabetic foot ulcers [291]. Aside from patient health monitoring, in-hospital embedded devices can perform a variety of real-time communication tasks, such as locating hospital staff to allocate medical equipment (e.g, a defibrillator, nebulizer, oxygen, or pump) or locating hospital equipment (e.g, a wheelchair, ambulance, or monitoring devices).

2.2.1.2 Internet of Medical Things Wearable Devices

Wearable smart medical devices can track and analyse a wide range of vital signs, such as a person's blood pressure, glucose level, heart rate, pulse rate, respiration rate, and body temperature. Smart fitness trackers, smartwatches, and smart heart rate monitors are some examples of smart wearable medical devices [62]. Wearable devices can provide real-time alerts [292]. As a part of a daily routine, smart fitness devices, such as TomTom Spark [293] and Moov Now [294], are used to calculate exercise sessions and step counts. The Omron EVOLV [295], iHealth FEEL Blood Pressure Monitor, and Philips Upper Arm Blood Pressure Monitor [296] are also smart sensors that monitor blood pressure. Smart sensors that monitor glucose, including GlucoWise [297] and iBGStar Blood Glucose Meter [298], are used to monitor the blood sugar levels of diabetic patients. Various IoMT wearable devices minimise expenditures and hospital stay due to their monitoring and assessing capabilities.

2.2.1.3 Internet of Medical Things Applications

IoMT applications provide several in-house and in-hospital benefits, including real-time monitoring and communication with patients, GPs, and third parties [64]. IoMT applications can also reduce GP visits and medical costs while increasing patient satisfaction. Remote monitoring is now in high demand for in-house telehealth environments [299, 300] and telecare environments [301]. Almost every wearable device, including fitness trackers and monitoring (glucose, heart rate, pulse, depression, and asthma) devices use Android applications [61, 233].

Paramedics use smart medical devices with Android applications to provide medical services and assistance to patients. Drone deployment has also been proposed to respond immediately to cardiac arrest patients [302]. Virtual reality simulators [303] including emergency training simulators are now used to train students and staff [304].

2.3 Diabetes Technology

The initial insulin closed-loop administration system was developed in the 1960s by Dr Arnold Kadish. Kadish implemented an intravenous catheter and a pair of syringe pumps containing insulin and glucose or glucagon. Kadish successfully developed a servomechanism for blood glucose control that enabled rigorous monitoring of blood glucose levels. According to Templer, the first APS was invented in the early 1970s. The researchers developed similar glucose monitor and pump devices and reported their findings in 1974. Both devices used a computer to monitor venous glucose and provide insulin or dextrose intravenously [305]. According to Alsaleh et al., the first commercial APS was developed in 1977 and is known as Biostator [306]. In 1980, researchers led by Motoaki Shichiri invented the first wearable artificial pancreas device, which fitted into the user's jacket pocket [305, 307].

Automated insulin delivery innovators have progressed rapidly in diabetes management. These diabetes care pioneer have improved patients' lives by enhancing diabetes care and quality of life using innovative technology. From design to implementation, automatic insulin delivery has been remarkable; programmable insulin pumps started automatically administering insulin in the 1970s. According to Kesavadev et al., automated diabetes management accelerated in the 2000s with rapid progress in CGM sensor and insulin pump development [72].

Diabetes management technology has advanced rapidly and significantly to improve the general well-being of diabetic patients. While major advancements have been made in current devices, many remain prohibitively expensive. In addition to this accessibility barrier, the major cybersecurity problems for using digital devices must be understood.

2.3.1 Artificial Pancreas Systems

Embedded computing and IoT technology have enabled the development of advanced solutions for chronic conditions, such as Type 1 diabetes. The healthcare sector has derived

considerable advantages from the integration of digital technology due to the interconnectivity of devices that facilitate the generation and dissemination of information [153]. The use of embedded computers and IoT technology facilitates the creation of communication systems and control algorithms that enable the automated delivery of insulin from insulin pumps to the patient's body [308]. Various APS platforms have been proposed and created using Android or iOS smartphones, mini-computers, and laptops.

In 2013, a community known as the “Do-It-Yourself Artificial Pancreas System” movement was formed online using the hashtag “#WeAreNotWaiting.” This community consisted of families and carers who collaborated to develop do-it-yourself APSs. Various studies and sources have documented the existence and impact of this movement. The do-it-yourself APS was a hybrid closed-loop system representing a significant advancement in diabetes technology [72, 309-314]. Following this, in 2014, Dana and colleagues with diabetes began to create a DIY APS; Dana Lewis and colleagues introduced the DIY APS open-source platform. The objective of OpenAPS was to enhance the advancement of APS technology. OpenAPS facilitated connectivity with an insulin pump and CGM sensor [315].

In 2014, the Bionic Pancreas platform was developed using a sophisticated control algorithm and smartphone device. The Bionic Pancreas included an automated insulin dosage calculator that relied on the assessed glucose levels obtained from the CGM system [316]. The Bionic Pancreas platform utilised an insulin pump with a touchscreen interface and a smartphone application to establish a seamless connection between the insulin and CGM sensor. Similarly, in 2014, Lane Desborough and JDRF initiated building an APS called Bigfoot Biomedical SmartLoop. Bigfoot utilised FreeStyle Libre CGM sensors from Abbott [69]. The iLet Bionic pancreas was first proposed in 2015 and received a prestigious FDA breakthrough designation the following year [317, 318].

In 2016, the first commercial APS, known as Medtronic's MiniMed 670G, was introduced [319]. The MiniMed 670G combined the 670G insulin pump with MiniMed's Guardian 3 CGM sensor. In 2018, a new APS called Diabeloop was developed in Europe [320]. Diabeloop incorporated the Kaleido insulin pump and Dexcom G6 CGM sensor, and it utilised Bluetooth for wireless connectivity with a smartphone terminal for remote monitoring. Diabeloop has been available for commercial use in Europe; however, it has not yet been introduced to the market in the United States of America. Currently, both Minimed 670G and Diabeloop are commercially available, yet high prices and some technological obstacles limit global access [72].

In 2019, Medtronic's MiniMed 780G started the clinical trial phase [321]. MiniMed 780G includes Bluetooth-enabled communication, connectivity, and remote monitoring. The MiniMed 780G utilises Medtronic's SmartGuard algorithm for making decisions [322]. However, in 2019, the FDA issued a warning to patients and healthcare professionals regarding possible cybersecurity issues associated with Medtronic MiniMed insulin pumps, leading to a recall of all MiniMed variants (see Section 1.1). The FDA has reported that MiniMed insulin pumps were unable to update or patch the latest software [88, 323]. As of 2024 the current market provides a wide range of commercially accessible APSs, such as the Medtronic MiniMed 670G, Tandem Diabetes Care Control-IQ, Insulet Omnipod 5, Beta Bionics iLet Bionic Pancreas, and Roche Diabeloop. Table 2.2 presents the current development of APSs and their components.

Table 2.2

Features of Artificial Pancreas Systems for Diabetes Management.

APS	Insulin pump	CGM sensor	Communication
Medtronic MiniMed 670G	MiniMed 670G	Guardian 3 sensor	Bluetooth

APS	Insulin pump	CGM sensor	Communication
Medtronic MiniMed 780G	MiniMed 780G	Guardian 4 sensor	Bluetooth, Android application, and remote monitoring
Insulet Omnipod Horizon	Omnipod insulin pump	Dexcom and Abbott	Bluetooth and Android application
Diabeloop	Kaleido's insulin pump	Dexcom G6	Bluetooth and Android application
DIY-Loop	Medtronic's pumps and OmniPod	Dexcom and MiniMed CGM sensor	RileyLink, iPhone and Apple smartwatch
OpenAPS	Medtronic's pumps	Dexcom and FreeStyle Libre	Bluetooth, smartwatch, and Linux microcomputer
AndroidAPS	Accu-Chek Insight Pump, Dana R and RS Pumps, and Medtronic's pumps	Dexcom, FreeStyle Libre	Android smartphone and smartwatches
Bionic Pancreas	Touchscreen pump	Dexcom G5	smartphone application
Bigfoot Biomedical Smartloop	Lilly Diabetes tubed Insulin pump smart pen	FreeStyle Libre	iOS and smartphone application

According to a report, the market for diabetes management devices has experienced rapid growth due to growing global demand and significant technological advances. The global market for diabetes devices is projected to reach a valuation of USD 50.33 billion by the year 2030, demonstrating a compound annual growth rate (CAGR) of 7.45% from 2024 to 2030. The market is primarily influenced by variables, including the rising prevalence of diabetes, which is associated with technological progressions and the introduction of novel products [324]. Different organisations have contributed to automated insulin delivery technology. A description of these pioneers and their work is highlighted in the following subsections.

2.3.1.1 Beta Bionics

Beta Bionics was developed by the University of Boston. In 2005, Beta Bionics initially conducted an experimental evaluation on animals. In 2008, human trials were conducted after the successful completion of the animal trial phase. After successful human testing, the research team at Beta Bionics announced their decision to move forward with the development of the iLet Bionic Pancreas in 2016. Lilly Diabetes, an American pharmaceutical company (see Section 2.1.8), allocated a substantial sum of $5 million towards the development of the iLet Bionic Pancreas medical devices [325, 326].

2.3.1.2 Bigfoot Biomedical

The Juvenile Diabetes Research Foundation (JDRF), in collaboration with Lane Desborough, founded SmartLoop Laboratory to develop an artificial pancreas in 2014. This company is known as Bigfoot Biomedical (Bigfoot). Bigfoot also purchased Asante Solutions Company, which develops insulin pumps [327]. Furthermore, Bigfoot established a strategic alliance with Abbott Diabetes Care and consequently leveraged their CGM technology known as FreeStyle Libre. In 2019, Bigfoot announced its cooperation with Lilly Diabetes to develop a connected insulin injection system using AI. According to the contractual arrangement between Bigfoot and Ypsomed Group [328], Bigfoot is currently utilising insulin infusion sets provided by Ypsomed. In 2020, the FDA granted clearance to Bigfoot products, thereby authorising the launch of their commercial offerings in the market. In a noteworthy development, Bigfoot also disclosed securing substantial funding of $55 million [329].

2.3.1.3 Diabeloop

Diabeloop developed a semi-closed-loop APS [330]. In 2019, Diabeloop announced a €31 million investment round to create AI-based diabetes automation and personalisation

solutions. In 2020, Roche, a global healthcare business based in Switzerland, partnered with Diabeloop to collaborate on the advancement of insulin pump management [331].

2.3.1.4 Dexcom

The Dexcom corporation has produced and globally distributed CGM systems. In 2014, Dexcom received approval from the FDA to implement their proprietary software into the Dexcom G4 CGM device [332]. In 2020, Dexcom formally made strategic alliances with Insulet Corporation for their product platform, Omnipod. Dexcom also announced that CGM models (G6 and G7) will be used to develop a closed-loop APS using the t:slim X2 Insulin Pump developed by Tandem Diabetes and Omnipod 5 [333]. Moreover, in 2020, Dexcom announced its collaboration with the University of Virginia [334].

2.3.1.5 DreaMed Diabetes

DreaMed Diabetes was founded in 2014 with the primary objective of commercialising APSs through the utilisation of Glucositter software. As a tool for managing diabetes, Glucositter tracks blood sugar levels and the insulin's dynamic modulation in real time. Medtronic secured authorisation to employ the Glucositter technology following a substantial financial commitment of $2 million in collaboration with DreaMed Diabetes [335]. The Pro therapy software, which incorporates AI, was developed by DreaMed Diabetes [336] and has been granted breakthrough designation by the FDA [337].

2.3.1.6 EoFlow

The EoPancreas system is currently under development by EoFlow, a prominent South Korean enterprise. The EoPancreas system incorporates a CGM sensor, developed by POCTech, alongside the utilisation of a patch pump [338]. In 2017, JDRF announced a partnership with EoFlow with a $2 billion investment for research funding [339]. In 2019,

EoFlow secured a substantial funding amount of $28 million [340] to produce CGM sensors. EoFlow was also granted a prestigious breakthrough designation by the FDA in 2019.

2.3.1.7 Insulet Corporation

Insulet Corporation (Insulet) developed the Tubeless Omnipod insulin pump [341]. The corporation formed a strategic alliance with Dexcom in 2014 and a contractual agreement with Automated Glucose Control LLC to collaboratively develop an advanced automated glucose control system. Insulet has obtained $69.5 million in funding to develop APSs [342]. Dexcom and Insulet have jointly declared their intention to integrate the Dexcom CGM sensors (G6 and G7) into Insulet's Omnipod Horizon platform [343]. The corporation successfully acquired regulatory approval from the FDA in 2019.

2.3.1.8 Lilly Diabetes

The pharmaceutical corporation, Lilly Diabetes, developed an innovative insulin delivery system with a tubed pump for insulin and a smart pen [344-346]. Lilly Diabetes has a substantial investment of $400 million [347]. Lilly Diabetes and Ypsomed Group engaged in collaborative efforts to advance the development of an interconnected pen using Dexcom CGM sensors and YpsoPump [348]. Lilly Diabetes and Ypsomed Group are currently developing AI-based diabetes systems [349]. Lilly Diabetes obtained authorisation from the FDA in 2020.

2.3.1.9 Medtronic

Insulin pumps and CGM sensors are extensively produced by the Medtronic medical device company [350, 351]. Medtronic announced its receipt of a substantial $337 million investment [352]. The Medtronic insulin pumps (MiniMed 770G and MiniMed 670G) and CGM sensors have received approval from the FDA. However, some Medtronic devices are still under review for regulatory approval [353].

2.3.1.10 Senseonics

Senseonics is a pioneering company that specialises in the production of implantable CGM sensors [147, 354]. Senseonics CGM sensors are approved by the FDA [355]. Senseonics has announced a strategic partnership with Ascensia Diabetes Care and made agreements for up to $80 million to provide liquidity [356].

2.3.1.11 Tandem Diabetes

Tandem Diabetes (Tandem) manufactured the t:slim insulin pump, which is a commercially available and highly innovative device that incorporates Control-IQ technology [351]. The FDA approved this Control-IQ technology [357]. Tandem has officially declared a substantial sum of $60 million in underwritten stock. Tandem has established a collaborative alliance with the JDRF to develop a dual-chamber infusion pump [358]. Tandem and TypeZero Technologies obtained a mutually beneficial licence agreement to commercialise their APS [359]. Abbott and Tandem also announced their partnership to develop advanced AI-based diabetes solutions [360].

2.3.1.12 TypeZero Technologies

TypeZero Technologies produces diabetes assistant systems with the University of Virginia research team [361]. Tandem [359] and Dexcom [362] both collaborate with TypeZero to produce diabetes assistant systems. TypeZero secured a substantial $2.6 million in the year 2017. Dexcom, a prominent entity in the field (see Section 2.1.4), subsequently acquired TypeZero in 2018 to advance the development of AI-based diabetes solutions [363].

2.3.1.13 Ypsomed Group

Ypsomed Group, a Switzerland-based company, operates globally in the distribution of AI-based diabetes products through its mylife Diabetescare brand that serves both clinics and pharmacies [364]. Ypsomed Group’s production of insulin injection pens and infusion systems

has made a significant contribution to the field of self-administered treatment for diabetes [365]. Bigfoot [328], JDRF [366], and Lilly Diabetes [348, 349] have worked on this self-administered treatment for diabetes. Ypsomed Group has emphasised its commitment to the advancement of research and development with a specific objective of allocating CHF100 million towards this endeavour [365].

2.3.2 Diabetes Technology Inventors

Significant advancements in the automated administration of insulin have revolutionised the field of diabetes management and enhanced patient outcomes. Automated devices persistently monitor glucose levels and modify insulin administration through intricate algorithms. The implementation of automated devices for personalised and precise insulin administration has effectively mitigated the risks associated with hyperglycaemia and hypoglycaemia, thereby enhancing glucose management and mitigating the potential long-term repercussions.

Diabetes technology inventors have led an evolution in the field of diabetes treatment, as elucidated by most prominent investors in Section 2.3.2. Table 2.3 presents a list of diabetes technology inventors. The dedication of resources towards technology has served as crucial support for a substantial number of individuals afflicted with chronic conditions, technology supporting glycaemic control and an improved standard of living. The invaluable contributions made by these investors have engendered a profound transformation in the well-being of patients and catalysed the progression of healthcare technology.

Table 2.3

Current Technology Innovations in the Field of Diabetes Management

Reference	Inventor	Progress	Devices or tools	Business partners
[325, 326]	Beta Bionics	iLet Bionic	Insulin Pump and CGM sensor	Lilly Diabetes
[327-329]	Bigfoot Biomedical	Purchased Asante Solutions and forged Ypsomed Group agreement	Insulin pump (Asante Solutions), CGM sensor (FreeStyle Libre), and smart pen (Timesulin)	Lilly Diabetes, Timesulin and Ypsomed Group
[367]	Diabeloop	Using CGM-Accu-Chek	CGM-Accu-Chek	Roche
[333, 334, 362]	Dexcom	Collaborating with the University of Virginia	Insulin Pump (Tandem Diabetes's t:slim X2 and CGM sensor (Omnipod 5)	Insulet Corporation (Insulet)
[335-337]	DreaMed Diabetes	Medtronic investment	Glucositter software and advisor Pro software	Medtronic
[338-340]	EoFlow	Research funding obtained from JDRF	EOPatch patch pump, CGM sensor (POCTech) and controller (TypeZero)	JDRF
[341-343]	Insulet	Dexcom and Insulet have been integrating the Dexcom G6 and G7 CGM into Omnipod Horizon	Insulin pump (Tubeless Omnipod), CGM sensor (Dexcom G6 and G7), and Omnipod Horizon	Dexcom, Omnipod and Automated Glucose Control LLC
[344-349]	Lilly Diabetes	Have been working with Ypsomed Group	Tubed Insulin pump, smart pen, and YpsoPump	Dexcom, Beta Bionics, and

Reference	Inventor	Progress	Devices or tools	Business partners
				Bigfoot Biomedical
[350-352]	Medtronic	Secured investment from Blackstone Life Sciences	Insulin pump (MiniMed 770G and 670G), CGM sensor, and Glucositter software	Blackstone Life Sciences
[147, 354-356, 368]	Eversense Senseonics	Developed Eversense XL Senseonics CGM System	Eversense Senseonics CGM System and Eversense XL Senseonics CGM System	Ascensia Diabetes Care
[351, 357, 359]	Tandem Diabetes	Announced partnership with JDRF and Abbott	t:slim insulin pump	JDRF, TypeZero Technologies, and Abbott
[359, 361-363]	TypeZero Technologies	Dexcom acquired TypeZero Technologies	Diabetes assistant systems	Tandem Diabetes, and Dexcom
[328, 348, 364-366]	Ypsomed Group	Secured Lilly Diabetes collaboration, and partnership with JDRF	AI-based diabetes products	mylife Diabetescare and Lilly Diabetes

2.3.3 Justification for the Use of Literature-Based Research

The choice to undertake this research via a literature-based approach is firmly rooted in feasible and ethical limitations that currently limit direct access to proprietary devices utilised to manage diabetes disease for cybersecurity research purposes. Proprietary devices, including CGM and insulin pumps, are not easily accessible for cybersecurity investigations due to cost, time, and ethical approvals from their manufacturers. Sections 2.3.1 and 2.3.2 provide information on how digital transformation rapidly increased, and the manufacturing companies

of these devices have direct access to their products for research and development purposes. The devices render it economically unfeasible to obtain and examine them directly in an academic research context for cybersecurity purposes.

Beyond the constraints imposed by financial obstacles, ethical and legal considerations significantly restrict access to proprietary systems. Executing cybersecurity assessments on these devices necessitates explicit authorisation from manufacturers, given that ethical protocols and legal constraints dictate adherence to intellectual property rights and considerations of liability. Obtaining such permissions can be challenging, particularly within the constrained timelines characteristic of academic research endeavours, thereby necessitating the exploration of alternative techniques, such as literature-based research and simulation.

This research relies on an extensive literature review to gather information on proprietary devices. Through publicly accessible information via literature, scholarly articles, and comprehensive reports, this study illustrates device manufacturers. This is a great opportunity for future researchers to obtain ethical approval for cybersecurity research to conduct real-world cyber-attacks and propose robust security solutions. This research facilitates a comprehensive understanding of the advancements made by companies, the implementation of diabetes management devices, and the rationale behind the enhancements of models or applications.

2.3.4 Continuous Glucose Monitoring

CGM has become a pivotal device in the lives of individuals afflicted with chronic conditions. The CGM device is a tiny sensor set beneath the skin that quantifies interstitial glucose levels and transmits the data wirelessly to a receiver or smartphone. This device provides real-time feedback regarding the effect of diverse circumstances on glycaemic regulation [369].

CGM enhances the effectiveness of diabetes management by allowing a widespread depiction of glucose fluctuation. A CGM sensor provides continuous feedback about glucose patterns and trends, thereby enabling individuals to make informed and careful choices regarding insulin dosage, dietary choices, and exercise regimens [370]. The data obtained from CGM can be effectively disseminated across various platforms, including smartphones, cloud-based systems, and web-based applications [371].

The CGM sensor serves as a pivotal component within an APS, and different APSs may use the same brand of CGM sensors. Various companies have developed proprietary CGM sensors or used different CGM sensors for their APSs (see Section 2.2.1). Different CGM sensors have been developed to understand glucose patterns; for instance, Medtronic's Minimed CGM sensor was used to understand glucose control patterns. Following this, the Guardian CGM sensor was introduced in 2004 [372]. Subsequently, Guardian RT [373] and Dexcom short-term sensors [374] were launched and tested in clinical trials.

In 2008, Minimed iPro1 was introduced [375]. Minimed iPro2 was introduced in 2011 and obtained approval from the Food and Drug Administration (FDA) [376]. The Guardian Sensor 3 was also introduced; this CGM sensor uses Bluetooth communication and sends readings of glucose levels to the patient’s iOS devices. Different organisations have launched different brands of CGM sensors. The Dexcom 7-Day STS was launched in 2007 with a label indicating that the device lasts for a length of seven days [377]. Guardian RT and Dexcom STS CGM were introduced three days for glucose level monitoring.

In 2012, the G4 Platinum CGM sensor [378] was introduced with an integrated wireless transmitter and receiver within the device. The Dexcom G5 CGM system [70] has enabled patients to transmit their glucose level data to their smartphones. Dexcom G6 CGM System received regulation approval from the FDA [379]. The G6 CGM System was integrated into both the Diabeloop closed-loop system and Tandem's t:slim X2.

In 2008, the FreeStyle Navigator CGM sensor was developed by Abbott and received FDA certification. Following this, the FreeStyle Libre Flash 2 CGM sensor was introduced in 2014. Both the FreeStyle Libre Flash and FreeStyle Libre Pro utilise smartphone devices, and the FreeStyle Libre uses an Android application [73]. Senseonics launched an implanted CGM sensor [147, 354], and secured regulation approval from the FDA [355]. This Senseonics CGM sensor utilises smartphone applications compatible with both Android and Apple devices. In 2017, the Eversense XL CGM system was introduced [368]. The current market includes various CGM brands, and Table 2.4 presents a brief advancement overview of CGM sensors to compare their characteristics.

Table 2.4

Comparison of Currently Available and Previously Continuous Glucose Monitoring Devices.

CGM sensor	**Feature**	**Connectivity**	**User interface**
Minimed	Patterns of glucose control		Windows 95
Guardian RT	Alerts in real time		Software
Dexcom STS	Alerts in real time		
7-Day STS	Alerts in real time	low-powered with radiofrequency	
Minimed iPro	Used for physicians only		
FreeStyle Navigator	Alerts in real time	Radiofrequency transmitter	
Minimed iPro2 CGM	Web-based CareLink and iPro software	Web-based CareLink	iPro software
G4 Platinum	Colour screen, alerts in real- time, and CGM display	Wireless transmitter and receiver	
FreeStyle Libre Flash 2 CGM	Built-in glucose meter and data management software	Near field communication	

CGM sensor	Feature	Connectivity	User interface
G5 CGM	iOS application, Apple smartwatch and real-time CGM display	Bluetooth	Smartphone and iOS systems
Guardian Sensor 3	Data share remotely, SMS messages, and Android application	Bluetooth	Smartphone, smartwatch iOS systems, and remotely web-based
Eversense Senseonics CGM	Real-time CGM display, and Android and iOS applications	Wireless communication and Bluetooth	Android and iOS systems
Eversense XL Senseonics CGM	Real-time CGM display, and Android and iOS applications	Wireless communication and Bluetooth	Android and iOS systems
G6 CGM	Real-time CGM display, iOS application, Apple smartwatch, and Fitbit smartwatches	Bluetooth	Smartphone, iOS systems and Fitbit smartwatches
FreeStyle Libre 14 and pro	Real-time CGM display, software, smartphone, and Android application	Bluetooth	Smartphone and Android application

Table 2.4 presents the utilisation of technology in CGM, and CGM advancement has received much attention due to a CGM sensor's ability to manage diabetes and enhance quality of life. CGM has demonstrated significant advancements in real-time reduction of glycated haemoglobin [380] and a decrease in occurrences of hypoglycaemic episodes [381], making it a highly beneficial advantage from a diabetes management perspective [380].

The CGM reading used for insulin adjustment, where glucose levels within the range of 70–180 Milligrams Per Deciliter (mg/dL) reading may give different meanings for individual. The bolus calculator calculates the glucose level [382].

The formula for calculating the required dosages of insulin is defined as:

$$B = \frac{CHO}{CR} + \frac{GC-GT}{CR} - IOB, \tag{2.1}$$

The Carbohydrate-to-insulin Ratio (CR) represents the amount of carbohydrate that is regulated by insulin, while the Correction Factor (CF) indicates the decrease in glucose level. These two primary parameters are the main indicators for GP to decide on the treatment plan [383]. Carbohydrate (CHO) calculates the amount of carbohydrates. The goal glucose level is given in milligrams per decilitre, and the prior insulin dosage is measured in units (U).

2.3.5 Insulin Pumps

The invention of insulin pumps has revolutionised the landscape of diabetes management and effectively empowered patients to regulate their blood glucose levels with heightened precision and efficacy. Recent technological advancements have led to tiny yet advanced insulin pumps [148] that boast precise and customisable insulin delivery capabilities. Computer algorithms and CGM sensors dynamically adapt insulin therapy based on glucose measurements in real time. The incorporation of an insulin pump with the CGM sensor yields enhanced precision in insulin administration and mitigates the potential risks of hypoglycaemia and hyperglycaemia in hospitals [68]. Further, these insulin pumps have enabled diabetes self-management in home settings [148, 384].

The insulin pump is an advanced device designed to administer insulin subcutaneously while delivering precise and incremental doses of the hormone at regular intervals [385]. This programmable device encompasses a range of adjustable parameters, allowing individuals to finely calibrate the desired dosage settings with their basal rate profiles [351].

In 1983, the initial iteration of the insulin pump was developed by Medtronic [386]. Medtronic's first pump was known as Minimed 502. In 2003, Medtronic subsequently introduced the MiniMed Paradigm 512/712 [387, 388], which was a wireless pump that employed radiofrequency technology. The MiniMed Paradigm REAL-Time 522/722 was introduced to the market in 2006, while the subsequent iteration, the MiniMed Paradigm REAL-Time Revel 523/723, was launched in 2010 [389]. In 2013, the FDA approved the

MiniMed 530G insulin pump, making it the first pump to obtain such recognition [390]. In 2016, the FDA again approved the MiniMed 630G [391], reinforcing its status as a medical technology innovation. Medtronic has since produced hybrid systems such as the MiniMed 670G, 770G, and 780G. New models are yet to be launched.

The OmniPod insulin pump, manufactured by Insulet, is a personal diabetes monitor that was granted FDA clearance in 2005 [341]. The primary characteristics of the personal diabetes monitor include water resistance and wireless connectivity. The FDA has approved the OmniPod DASH insulin pump [392], which was introduced to the market in 2019. The OmniPod DASH system utilises Bluetooth technology for communication, allowing patients to track their personal diabetes monitor data using a smartphone application that is connected to the Dexcom G6 CGM [393].

The t:slim insulin pump, produced by Tandem, was the first manufactured touchscreen insulin pump. This pump obtained FDA approval in 2011. The user interface of the t:slim insulin pump used USB connectivity to establish a connection between the pump and a personal computer [394]. Then, in 2014, Dexcom released the t:slim G4 insulin pump, compatible with their G4 Platinum CGM system [229]. The t:slim X2 pump was introduced in 2016, and it was granted approval by the FDA in 2018 [395].

In 2006, Roche Diagnostics manufactured the Accu-Chek Insight insulin pump. The Accu-Chek Combo insulin pump was then manufactured and received FDA approval in 2012. Using Bluetooth, the insulin pump and glucose monitor were connected [396]. Another 2012 release, the Accu-Chek Spirit Combo insulin pump, uses an infrared interface to link up with a smart glucometer. According to Ulbrich et al., the Accu-Chek Solo insulin patch pump, which was launched in 2018, uses a micropump [397]. The Accu-Chek Solo insulin patch pump can be managed remotely and is available for use in trials. Nevertheless, Roche Diagnostics has not yet received clearance from the FDA.

The Kaleido insulin pump was introduced in 2018 and integrated with the Dexcom G6 CGM System. This pump has not yet obtained approval from the FDA [398]. The Cellnovo insulin pump, which utilises Bluetooth connectivity and is compatible with Android smartphones, was introduced in 2012 for experimental purposes [399]. Following this, the Cellnovo insulin pump was subsequently made available for commercial purchase in 2019, yet it was abruptly recalled with manufacture and commercial activities halted [400]. The Dana Diabecare RS insulin pump was launched in 2018, allowing for remote control using Android and iOS applications [401]. Table 2.5 provides a concise overview of the comparisons and characteristics of several insulin pumps.

Table 2.5

Comparison of Currently Available and Previous Different Insulin Pumps.

Insulin pump	**Feature**	**Communication**	**User interface**
OmniPod	Waterproof with wireless controller	Wireless	Android
MiniMed Paradigm 512/712	Alert	Radiofrequency	
MiniMed Paradigm REAL-Time 522/722	Real time	Infrared and Bluetooth	Personal computer
Accu-Chek Insight insulin pump	Colour screen	Bluetooth	
t:slim insulin pump	Touchscreen and USB connectivity from pump to computer	wired	Personal computer
Accu-Chek Combo insulin pump	Real time	Bluetooth	Personal computer
t:slim G4 insulin pump	Real-time data related to insulin doses	Bluetooth	
t:slim X2 insulin pump	Basal-IQ technology	Bluetooth	Smartphone

Insulin pump	Feature	Communication	User interface
Omnipod DASH	Smartphone applications, Bluetooth wireless communication	Bluetooth and smartphone application	Smartphone
Accu-Chek Solo insulin patch pump	Micropump	Bluetooth and smartphone application	Smartphone
Kaleido Insulin Pumps	Colourful hybrid insulin pump	Bluetooth and Android smartphone	Android smartphone
Dana Diabecare RS insulin pump	Remote control via Android and iOS applications	Bluetooth, Android, and iOS applications	Android and iOS application
Cellnovo insulin pump		Bluetooth and Android smartphone	Android smartphone
Minimed 770G	Auto correction dosing	Bluetooth and Android smartphone	Connect with different devices
Minimed 780G	Auto correction dosing, and Auto dosing	Bluetooth and Android smartphone	Connect with different devices

Similar to CGM sensors, insulin pumps have significantly transformed the management of diabetes by offering a simple and efficient method of administering insulin. These devices provide a multitude of advantages, including greater regulation of blood sugar levels, increased flexibility in administering insulin, and improved overall well-being for those with diabetes. Insulin pumps have considerably decreased the necessity for frequent daily injections. These pumps enable patients to control their blood sugar levels and effectively mitigate likely problems.

2.4 Medical Devices Cybersecurity

According to a recent forecast, the projected valuation of the worldwide medical device market achieved a valuation of approximately USD562.6 billion in the year 2022. With a predicted compound annual growth rate (CAGR) of 6.2% from 2023 to 2031, the medical device market is expected to reach a worth of over USD965.2 billion by 2031 [402].

Implantable medical devices are physically or surgically inserted into the body, and they must remain inserted [403-405]. Health professionals may surgically implant the device based on the patient's medical condition. For example, a patient burdened by severe back pain may find relief through epidural spinal cord stimulation. Similarly, a patient managing cardiac difficulties may consent to the implantation of a pacemaker. Individuals who have Parkinson's disease or epilepsy may also provide consent to implant a deep brain stimulation device [406].

Medical devices can effectively address many management plans for any disease, including cardiac arrhythmia [75], back pain, diabetes, epilepsy, Parkinson's disease, and spinal cord illnesses. Moreover, patients with kidney-related ailments have the potential to offer consent for the utilisation of artificial kidney apparatus [403]. Individuals suffering from diabetes have also been known to use APSs [308].

Medical devices have exemplary features in terms of their various uses, including precise therapeutic targeting and advanced functionality. Therefore, the comparability between two separate medical devices is inherently challenging. For example, a cardiac implant specifically designed to administer therapeutic interventions to a patient suffering from cardiac disease may primarily target the intricate network of the cardiovascular system. A deep brain stimulation implant designed to relieve Parkinson's disease symptoms would mainly focus on regulating the complex central nervous system. Nevertheless, both cardiac implants and deep brain stimulation implants would result in a distinct impact on resources, such as communication protocols and authentication mechanisms.

Due to the direct involvement of patients with implantable devices, the risks of cyber hacking have increased. Numerous instances exist that elucidate the gravity of hacking. A previously mentioned example is how medical professionals deactivated the wireless connection of former Vice President Dick Cheney's pacemaker to prevent it from being hacked [93]. Furthermore, the FDA has identified vulnerabilities in medical devices [235]. Following the identification of numerous vulnerabilities in IoMT devices, a comprehensive investigation has conducted meticulous analyses of these attacks [103, 237-239]. Given the ongoing nature of these investigations, it is essential to prioritise the security of IoMT devices to reduce the risk of hijacking attacks.

2.4.1 Medjacking

Throughout history, instances of mislaid physical documents or the theft of personal computing devices have posed a risk of exposing patients to potential data breaches [407, 408]. Despite the many advantages to medical services, the fact that medical devices may connect to the internet puts patients' lives at risk since hackers can steal sensitive patient data and infect these devices with different cyberattacks. Figure 2.1 depicts the data flow in medical devices and the origins of medical device hijacking. According to [409], data flow defines the type of input and output associated with a given process. The concept of data flow refers to the movement and processing of data across various components or systems [410]. This concept is used across multiple disciplines, such as software engineering and network design, to describe the processes by which data is transmitted and utilised [411].

Figure 2.1

The Origin of Medical Device Hijacking.

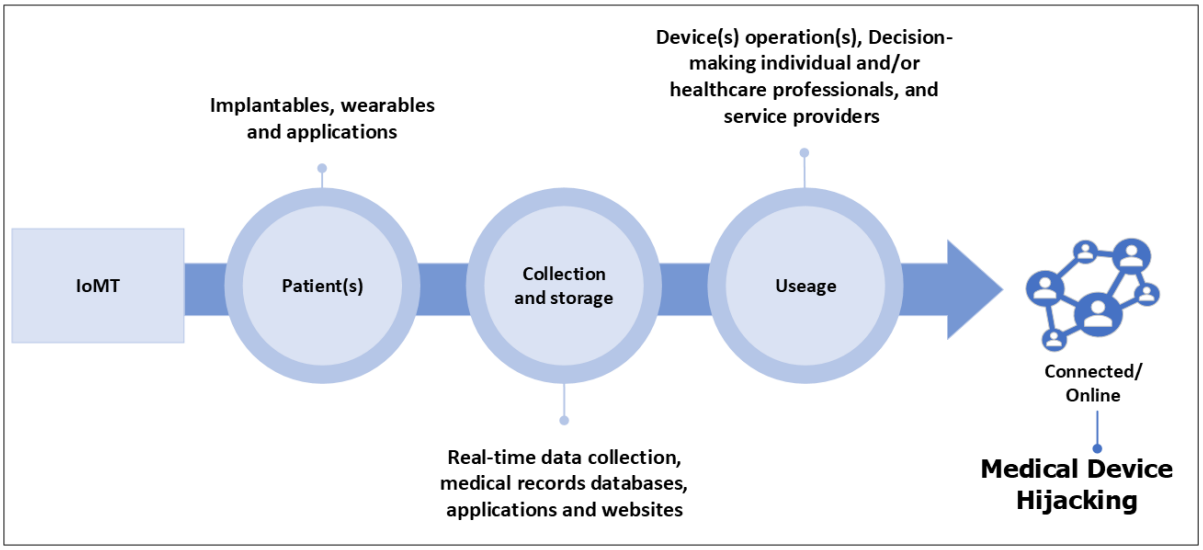


Figure 2.1 presents the IoMT and patient-related connected environment. The IoMT encompasses a range of devices, including wearable sensors, implantable technologies, and remote monitoring systems, which are progressively used to collect and distribute patient data [11, 14, 16, 209, 412-417]. IoMT devices determine variables such as heart rate, blood glucose levels, body temperature, and electrocardiogram readings through integrated sensors [11, 413, 417-419]. The provision and quality of medical services are increasingly dependent upon the advancements in IoT technology. The IoMT technology facilitates unidirectional communication among GPs, patients, and third parties, allowing for the remote and real-time monitoring of various parameters, including disease progression, patient status, and time-series data. Nevertheless, this linkage to the internet makes IoMT devices susceptible to many vulnerabilities, encompassing hijacking attacks. The security vulnerabilities within the data flow may disrupt decision-making processes [179, 209, 414, 420, 421]. For instance, if the data is tampered with during its transmission, it may result in incorrect diagnoses or inadequate treatments. According to [422], the data transmission process within IoMT networks may be susceptible to a range of attacks, encompassing both fake users and fraudulent nodes. Moreover,

adversaries may infiltrate IoMT networks and alter patient data. This may threaten patients' lives, leading to incorrect diagnoses and inappropriate treatment [422]. Another example, according to [78], in a tampering attack scenario, the attacker aims to compromise the integrity of the data contained within the exchanged communications [78]. As stated in [423], adversaries may engage in the passive interception of communications to obtain sensitive information, compromising the patient's privacy and confidentiality. Moreover, adversaries frequently intercept and modify transmitted information through passive attacks, resulting in incorrect diagnoses, altered medication dosages, or other negative outcomes [423].

The medical device's information exchange is essential for the delivery of healthcare services. Treatment or chronic condition management through medical devices is achieved by sharing data with healthcare professionals or service providers. If data are deleted or manipulated during information exchanges, doctors may make wrong choices or perform unsuitable operations. The potential implications of hacking or unauthorised intrusion into medical devices can be highly dangerous [209]. According to [424], hospitals have emerged as highly vulnerable targets for cybercriminals seeking to illicitly acquire patients' confidential information for trade with malevolent entities [424].

2.4.2 Medjacking Threat Vector

The rise of digital transformation has resulted in a heightened focus on implementing resilient cybersecurity measures that safeguard critical data and systems from unauthorised intrusion, exploitation, disclosure, disruption, alteration, or destruction [425]. With the increase and advance of various threats, the domain of cybersecurity has emerged as an essential concern in the realm of public safety [89, 426, 427]. Researchers, manufacturers, regulatory bodies, and healthcare providers in the IoMT field emphasise that robust security measures must be used. This is mainly motivated by the importance of IoMT technologies and their crucial role in providing patients with vital management plans for diseases. The IoMT domain

exposes inherent vulnerabilities and weaknesses, consequently amplifying the potential for hacking incidents. Table 2.6 presents a range of prevalent security risks associated with hijacking, which include but are not restricted to the following subsections.

Table 2.6

Potential Medjacking Security Threats Vectors to Medical Devices and Their Impact.

Reference	Target	Attack	Causes	Impact
[428]	Healthcare website	Phishing	Human factors, reverse engineering	Access personal medical data, release ransomware, and compromise credentials
[83]	Medical devices	Malware	MQTT and BLE protocols weakness	Distrub and or block legitimate operations
[99]	Implantable devices	MITM	BLE weakness	Modify medical doses
[429]	Medical devices	DoS	MQTT and BLE protocol weakness	Disturb time-sensitive functions
[430]	Medical devices	Brute force	MQTT and BLE protocols weakness	Gain unauthorised access

Table 2.6 lists the potential impacts and threat vectors of current medjacking attacks. Various medjacking threat vectors have been illustrated in the literature, and each attack has unique characteristics and consequences. One example is malware, which is described as a program designed to harm or exploit systems. This type of software can cause problems such as data loss, system disruption, and hacking. In another threat vector, phishing risks such as unauthorised access and medical data theft might arise. Table 2.6 outlines each threat vector and its potential consequences, ranging from ransomware that encrypts data and demands payment to MITM attacks that intercept conversations.

BLE and MQTT protocol vulnerabilities are used for MITM, DoS, brute force, ransomware, and phishing attacks [179, 431]. Insecure pairing techniques like Just Works or

weak pass code allow attackers to steal or alter BLE data. Devices may be manipulated into connecting to unauthorised devices, allowing MITM attacks without robust authentication. Excessive connection requests or data packets may impair BLE devices.

Unauthorised users can intercept and read MQTT conversations without Transport Layer Security, adding malicious messages. Lack of authentication can allow unauthorised users to pose as customers or brokers, enabling MITM attacks. Overloading MQTT brokers with connection requests or payloads might disable them [431, 432].

Brute force attacks can compromise MQTT brokers using weak or default credentials. Brokers without account lockout after many failed logins are vulnerable to brute force attacks. Through MQTT topics, malicious payloads may infect users and cause extensive bugs. When misused, MQTT brokers can spread malware or execute destructive orders to all their end users [431].

Due to its low power and simplicity, BLE is vulnerable to MITM, DoS, brute force, malware, and phishing attacks. Easy code pairing and poor encryption are the main reasons. Lightweight MQTT lacks encryption and authentication due to its focus on overhead reduction. MITM, DoS, brute force, malware, and phishing threats threaten MQTT brokers and clients.

2.4.2.1 Phishing

Phishing is a criminal form of social engineering and illicit conduct that calculatingly deceives victims into divulging their personal information or deliberately enabling the dissemination of ransomware [433, 434]. Phishing attacks have become increasingly intricate and often overtly mimic a website the victim intends to use. During phishing attacks, the attacker bypasses security measures to seek comprehensive visibility of the victim’s activities. In a study conducted by Lee in 2023, phishing attacks emerged as the predominant form of cybercriminal activity [428].

Within the broad context of the IoMT and healthcare systems, phishing attacks consist of social engineering approaches aimed at deceiving healthcare professionals and patients. According to [78], social engineering is a technique used to manipulate individuals via strategies such as baiting or pretexting to tempt individuals to share information. This encompasses passwords, names, identification numbers, and confidential information that may facilitate a subsequent cyber-attack.

In [78], the researchers stated that most attacks occurred as a result of social engineering or phishing endeavours, and proposed that that a budget be designated to enhance awareness and facilitate the training of medical personnel, thereby enhancing their technical knowledge to discern any prospective phishing threats [78]. Similarly, according to [435], a phishing attack is characterised as a social engineering technique that exploits email or alternative communication methods to mislead an individual into either opening an attachment or engaging with a link that carries malware. Within the context of the IoMT, a phishing attack can be executed against healthcare professionals or patients through deceptive emails or communications that apparently originate from a legitimate source [435].

A case study was undertaken in a study [436], wherein the researchers stated that phishing attacks define a form of social engineering strategy that exploits purportedly benign messages or emails, which are, in fact, malicious in nature and created for data theft or the transmission of malware. In [436], the researchers designed a phishing attack to replicate an established application that victims frequently use. Participants employed the social engineering toolkit in their experiment to replicate the smart glucose meter application. Furthermore, utilising the Ettercap DNS Spoofing tool, victims were subsequently redirected to the hacker's IP address, facilitating the capture of user credentials [436]. According to [183], a phishing attack involves the impersonates assuming the identity of a legitimate individual or health facility to acquire sensitive healthcare information and device passwords. A phishing

attack aims to collect sensitive information or execute actions that undermine the security of medical devices or healthcare networks.

2.4.2.2 Malware

Malware, a condensed form of malicious software, has the inherent purpose of inflicting harm upon a system or its users. According to Alsubaei et al., IoMT systems are vulnerable to malware attacks (viruses, Trojans, and worms) that infect and impair the systems' operational capabilities. A vast majority of recent online attacks utilise malware [83]. Depending on the attacker's objectives, these malicious programs may include ransomware and disruptive malware.

In [437], the researchers conducted simulations of eight distinct malware variants, encompassing Fuzz20 and Fuzz100, File Manipulation, Information Leakage, Synthetic Malware 1, Synthetic Malware 2, Synthetic Malware 3, and Synthetic Malware 4, targeting smart connected insulin pumps. The researchers classified malware into three distinct categories: (1) Malware that compromises health, (2) Malware that undermines data sensitivity, and (3) Synthetic threats to security. In the first category, Fuzz20 Fuzz100 malware exemplified a form of mimicry malware that disrupted a system's established functionality through the subtle alteration of data, specifically by means of randomisation. Another malware variant within this classification, the data manipulation malware, alters data within the targeted system to interfere with the standard control-flow execution.

In the malware study [437], the first category, malware poses a direct threat to a patient's health. Within the second category, the information leakage malware undermined confidentiality by secretly disseminating private data retained within the system to an unauthorised entity. Within the third category, a series of synthetic malware was deployed to compromise the operational integrity of the targeted system. To assess malware, the researchers proposed a methodology for risk assessment and management. Their approach, termed Finely

Integrated Risk Evaluation (FIRE), is used within a multimodal software model of a smart connected insulin pump.

2.4.2.3 Man in the Middle

In the context of medical devices, an MITM attack occurs when an attacker intercepts or eavesdrops on communication between a user and an application to impersonate one of the parties. This attack aims to obtain sensitive information, such as login credentials, account details, and medical records. This information can then be misused for identity theft, illegal access to medical records, or unauthorised password changes [438, 439]. In an in-depth study, Stergiopoulos et al., demonstrated an MITM attack using a simulation approach. The advisory's MITM attempted to increase the recorded base glucose level, and recorded information was obtained from the patient's sensor. As a result, the attacker increased the patient's bolus level and insulin flow [99].

2.4.2.4 Denial of Service

According to Papaioannou et al., a denial-of-service (DoS) attack is a deliberate attempt to disrupt time-sensitive functions or limit access to authorised resources and facilities. Depending on the service, time-critical intervals might be a few milliseconds to minutes or even hours. Attackers may trigger bandwidth congestion through flooding in medical devices by launching a DoS attack [429].

2.4.2.5 Brute Force

In a brute force attack, the attacker's objective is to identify valid credentials and leverage them to obtain access to a network for multiple purposes, including access to password hashes and tokens [83, 440, 441]. Almaiman and Alqahtani [430], state that an attacker uses a trial-and-guess method to crack a password. The bad actor can utilise software that generates a

vast number of password guesses to gain unauthorised access to IoMT devices and store sensitive data [430].

The attacker tries using a series of credential pairs on a victim's device to guess the valid credentials. If an attacker succeeds, they reuse the same credentials to access a series of other devices. A brute force attacker continues to use the credentials list until all trial lists have been utilised or until the cycle of a defined limit of trials is completed [442-444]. As researchers [445] have estimated, over 300,000 IoT devices are vulnerable to a brute force attack. Similarly, researchers [440] have concluded that nearly every IoT device can be vulnerable to a brute force attack. To launch their brute force attack, the attackers target the terminal network port, Secure Shell (SSH) port, and other ports [440]. The terminal network port is embedded in the internet protocol (IP) or Transmission Control Protocol (TCP) that enables devices to create a connection with a remote system. The SSH is a program used to execute one or more commands on a remote system. In one study, researchers [446] accessed stored data from a device using terminal network and SSH, and leveraged the device with malware. Researchers indicated that once access to a device is obtained, Botnet or DdoS attacks can be performed on a victim's device.

2.4.3 Medjacking Subtypes

Medjacking attacks are assigned to distinct subcategories (i.e., malware, phishing) according to methodology variations in the broad domain of cyber threats. Each medjacking subtype generally encompasses discrete methodologies, tools, or aims. Medjacking subcategories consist of very intricate forms of attacks (see Table 2.7). Cyber adversaries utilise several techniques to infiltrate data, systems, and networks with different intentions (e.g., stealing information and causing damage).

Table 2.7

Medjacking Subtypes Cyberattacks to Medical Devices.

Reference	Attack	Subtype
[83]	Malware	Botnet, worm, Trojan, and spyware
[83]	Phishing	Social engineering
[432]	MITM	ARP spoofing, DNS Spoofing, and IP Spoofing

Table 2.7 lists many forms of attacks, such as botnets, Trojans, spyware, phishing, and MITM attacks. The rising number of cyberattacks towards IoMT devices emphasises the significance of assessing and protecting these devices. According to Alsubaei et al. [83], subtypes of malware include viruses, Trojan horses, worms, and similar entities that can be utilised to initiate malware attacks and infect IoMT systems to hinder their operational functionality [83]. A phishing attack subtype is social engineering, which is a different type of cybercrime that tricks end users into exposing sensitive information [83]. Significant profit or free product or service scams are also associated with phishing attacks.

According to [447], a botnet represents a network of compromised machines or bots, colloquially called zombies, driven by a command and control approach. The botnet is used for various malicious activities, including DDoS attacks, password cracking, keylogging, and cryptocurrency mining. Initially, a botnet within the IoT ecosystem was created utilising the Mirai malware [447]. According to [448], Mirai and Hajime represent modern exemplars of Worm-Type Malware. Worms Mirai and Hajime have not only device-specific vulnerabilities, including default credentials or passwords that revert to their original settings following each reboot but also human vulnerabilities, exemplified by the neglect to alter passwords or the adoption of simplistic password choices. Consequently, Mirai and Hajime remain inadequately

addressed, continuing their capacity to infect and seize control of a substantial collection of IoT devices [448].

In another study in [449], the researchers demonstrated an IoT malware case study encompassing Mirai and BrickerBot, DDoS attacks, and password cracking. The researchers simulated the firmware versions of IoT devices, specifically the NET-GEAR WNAP320, WNDAP350, WNDAP360, and WNAP210 series. The researchers proposed a security detection system for IoT device behaviour, which is predicated on firmware virtualisation and a deep learning model [449]. In [78], the researchers stated that IoMT devices possess the potential to be hacked (as botnets) and subsequently employed to initiate DDoS attacks. Furthermore, botnet attacks are predicated on exploiting vulnerabilities inherent in IoMT devices, thereby transforming them into bots ready to receive directives from the adversary via command and control mechanisms to deliver misleading or false information regarding patients [78].

Another type of malware, known as a Trojan or Trojan horse, constitutes a malicious code designed to cause damage to the system [450]. Malware code can appear as an attachment, utility, or any other entity the victim might engage with or execute. The attacker disseminates the malicious software through social engineering tactics, thereby deceiving victims into downloading and executing Trojan horses [451, 452]. In a study [453], the researchers measured threats utilising honeypots to investigate IoT-based vulnerabilities, ultimately identifying 17 distinct vulnerabilities. The identified vulnerabilities encompass a range of threats, including malware, brute force attacks, privilege escalation, exploit-based alerts, SQL injection, cross-site injection, remote access trojans, Mirai, Gafgyt, and crypto-mining malware, as well as alerts associated with ransomware activities [453].

Reports of spyware have also emerged. According to [454], spyware is a category of malware that monitors user activity without obtaining their consent. Spyware utilises software

vulnerabilities and integrates itself with a normal program [454]. In [455], the researchers demonstrated a case study concerning the security assessment framework for the IoMT. The researchers stated that ransomware is one of the most widespread threats, compromising healthcare information and obstructing access for users or medical personnel to their systems unless a substantial sum (a ransom) is remitted in return for access. Over 78% of medical suppliers have experienced incursions involving ransomware, spyware, or a combination of both [455]. In an investigation [182], the researcher executed a simulation-based enquiry into digital health pathologies, revealing that "A 33-year-old female, 19 weeks pregnant, presents to the Emergency Department expressing complaints of headache and dizziness subsequent to a fall at home. The patient is a victim of domestic abuse; however, the patient is reluctant to interact with available services, attributable to the existence of eavesdropping spyware and GPS-tracking applications on her mobile devices [182].

According to [456], based on spoofing, an MITM attack consists of spoofing wherein the attacker intercepts legitimate communications between distinct hosts, thereby exerting control over the transmitted information, all while the victims remain unaware of the presence of an intermediary. For example, in Domain Name System (DNS) spoofing, the attacker impersonates equipment among various targets. In contrast, in alternative scenarios, such as Address Resolution Protocol (ARP) spoofing, the attacker directly impersonates the target's devices. ARP spoofing involves sending fake ARP packets to link the attacker's medium access control address with the victim's IP address. DNS spoofing redirects victim queries to a hostile server by corrupting the DNS cache. IP hijacking arises when data traffic is redirected to an unauthorised entity's independent system [456].

2.5 Artificial Pancreas System Cyberattacks

The APS uses wireless communication to establish and facilitate connectivity between the insulin pump, CGM sensor, smartwatches, mobile devices, and Android or iOS platforms

to monitor and assess glucose levels effectively (see Tables 2.3 and 2.4). The following subsections discuss APS medjacking, including delineated medjacking parameters and extant suggested solutions for each variant of APS cyberattack alongside their inherent limitations. Current scholarly literature focused on APS medjacking may not be exhaustive, yet it includes a comprehensive range of fundamental investigations. This is identified in the subsections below.

2.5.1 Artificial Pancreas System Tampering Attacks

A tampering attack can be defined as an unauthorised individual's act of deliberate interference with data or a resource to manipulate or eradicate information and compromise the integrity and authenticity of the affected system [457, 458]. A malicious actor utilises basic variables such as cookies, Uniform Resource Locator, query strings, Hypertext Markup Language form fields and headers, and password-cracking techniques to instigate a tampering attack. As previously mentioned, a tampering attack that manipulates data within an APS poses significant risks, such as modification and deletion of medical doses of an insulin pump. To analyse and prevent tampering attacks, implementing robust defence mechanisms aimed at safeguarding sensitive data and communication from unauthorised users is extremely important for APS devices.

2.5.1.1 Tampering: Current Problems

Without the patient's awareness, a tampering attacker might modify the intended parameters of an APS, including the data related to insulin dosages and CGM (low- or high-level) doses of the APS. For example, a tampering attack was conducted utilising a Raspberry Pi 3 and OpenAPS to manipulate the insulin dosages [459]; however, the researchers did not provide a technical description of their experiments. Therefore, an in-depth investigation is necessary to identify current challenges to detect tampering attacks.

Intentional unauthorised access for financial gain or to harm the targeted individual or entity is the usual goal of cybercriminals who launch tampering attacks. Li et al. stated that tampering or impersonation attacks have the potential to acquire system information through the analysis of communication between the insulin pump and CGM sensor [242]. The researchers utilised a commercially available Universal Software Radio Peripheral software radio board to disrupt radio communications in a certain frequency range. Wireless signals with different data, frequency, modulation, and power parameters were also generated. The researchers demonstrated possibilities for attackers to start, pause, and resume insulin bolus dosages by deciphering data packets sent by the insulin pump using its radio protocol, which mimicked the function of the remote control. Moreover, the researchers used reverse engineering to discover the unencrypted connection between the CGM sensor and insulin pump, which may be exploited for tampering or impersonation attacks [242].

According to [460], reverse engineering within the field of computer science describes a typical technique that entails the deconstruction and examination of software, hardware, or other complex systems, to evaluate these entities and extract pertinent design information [460]. In [461], the researchers stated that the process of acquiring knowledge regarding a system by systematically examining its constituent elements, encompassing parameter analysis, output analysis, and the input-output relationship, is referred to as reverse engineering [461].

According to [462], the process of examining an application to determine its functional attributes, internal architecture, and operational mechanisms, encompassing its modules, functions, and algorithms, is referred to as “reverse engineering.”. Moreover, the researchers stated that information technology professionals engage in reverse engineering for a multitude of purposes, including but not limited to: enhancing the functionality of applications in instances where the original developing entity no longer exists, conducting analyses of viruses, worms, and Trojans to extract their signatures and develop protective mechanisms (such as

anti-virus software); as well as decrypting file formats to facilitate improved compatibility and understanding of the underlying code [462].

Li et al. [242] obtained frequency information from the Federal Communications Commission Identification (FCC ID). Generally, the FCC ID offers open-access information on the entity that has been granted certification, including a unique code and validation for a product. Li et al. established a connection with the insulin pump and obtained remote control by utilising a frequency of 915 MHz. This was achieved by connecting a daughter board and antenna with a frequency of 915 MHz to the Universal Software Radio Peripheral board. The researchers successfully sent and received signals inside the 915 MHz frequency spectrum [242]. As a result, researchers completely disrupted the connection between CGM sensors and insulin pumps.

2.5.1.2 Tampering: Existing Solutions

Astillo et al. [463] introduced the concept of specification-based misbehaviour detection. This detection implements the glucose-insulin model of the UVA/Padova Type 1 Diabetes Simulator, the SVM, and the *k*-nearest neighbours' classifiers. The researchers developed a misbehaviour detection agent and applied the unified modelling language to represent the diagram of the detection agent. A Raspberry Pi microcomputer was chosen to emulate the APS component. The process of verification was conducted utilising the UPPAAL tool, and particular attention was given to the scenario of detecting a tampering attack that administered an inaccurate glucose level dose [463]. In this proposed solution, the validation of the identified agent used the Kalman filter estimation technique and the Mahalanobis distance method [463]. However, communication between the CGM sensor and insulin pumps lacked the implementation of any robust encryption algorithm, thereby making the communication vulnerable to potential malicious access and subsequent attacks.

2.5.1.3 Tampering: Remaining Challenges

A cryptographic solution is appropriate to address the tampering attack. However, the primary challenge is in choosing a viable cryptographic method, given an APS employs devices with limited resources. Zheng et al. [464], proposed fingerprint access control and hardware designed to process complex algorithms for this fingerprint authentication. Initially, the patient must provide a fingerprint to achieve authentication. By storing the registered fingerprint in the memory of the insulin pump, the patient may keep their access profile and receive access [464]. Nevertheless, an individual afflicted with a persistent ailment may be in an unconscious state, rendering them incapable of responding to fingerprint authentication.

2.5.2 Artificial Pancreas System Replay Attacks

Replay attacks involve the cybercriminal secretly capturing the transmission of data between two entities and reproducing data, retransmitting data, or acquiring authorisation. during a replay attack, often called a playback attack, adversaries may have the ability to breach authenticated data packets and create a delay in replaying the transmission between authorised users before retransmitting the data packets [103, 458]. Typically, the cyber attacker uses a three-step process to execute a replay attack to mislead the user. First, the attacker secretly monitors (i.e., eavesdrops on) the communication channel. Second, they capture user information such as login passwords and activity logs. Third, the attacker reproduces or replays the data packets to the intended recipient. The authorised user has the belief or assumption that the data packets are being received from another authorised user. However, during a replay attack, data packets have the potential to be modified or even removed to deceive authorised users. The primary defence methods utilised to mitigate replay attacks are session keys (i.e., unique codes valid for a single transaction), timestamps, and one-time passwords.

2.5.2.1 Replay: Current Problems

A replay attack depends on the content of the data and if a cyber attacker replays the content. An attacker can capture APS traffic over a connected network, resulting in the attacker replaying data packets to manipulate or misguide the APS. For example, in [86] Radcliffe found that a replay attack led to non-realistic insulin doses in a CGM device. The researcher demonstrated that the CGM device did not have a defence mechanism or a timestamp during the data packets' broadcast over the network. These results indicate that data packets were possibly being modified by the replay attack. Arduino, a small microprocessor board that allows users to interface with devices and modules to perform any task, was used for controlling the CGM device. Three steps were involved in triggering the replay attack: 1) reconnaissance, where the researcher obtained all information about the insulin pump and CGM device including the module, user manual, frequencies, and modulation; 2) identification, where a vulnerability was identified indicating that the device did not have a defence mechanism or a timestamp during the data packets transmission; and 3) reconfiguration, where theoretical attacks on the insulin pump and CGM sensor were discussed, and the researcher argued that an insulin pump can be reconfigured and transmission from the CGM sensor can be replayed.

In 2016, Johnson & Johnson warned diabetic patients that the Animas OneTouch Ping pump was vulnerable to hacking [202]. Johnson & Johnson is a well-known medical device company; their research team found that unencrypted devices could be forced by cyber attackers to deliver incorrect insulin doses into the patients' bodies. Moreover, the insulin pumps and CGM sensors have proprietary wireless communication methods that might be reverse engineered (see Section 2.4.1.1).

Dexcom G4's CGM security was analysed using software-defined radio based on the HackRF One and GNU Radio. Eavesdropping, jamming, replay, forging, and DoS attacks were analysed and demonstrated. Three steps were used to analyse Dexcom G4's security. The first

step was reconnaissance, where researchers obtained hardware information from publicly available resources, and gathered information. Dexcom G4's transmitter uses a minimum shift keying modulation and transmits on four frequencies: 2.425, 2.45, 2.475, and 2.477 GHz. The packet length 224 bits and 286.4 kHz channel spacing were employed with 49.987 kBit/s data rates. The low-powered radio frequency SimpliciTI network protocol was used. The second step involved software-defined radio based on the HackRF one, and GNU Radio was used to record the data at 2.425 GHz and the found packet that was periodically sent from the transmitter. The third step interpreted findings from step two. The researchers found that the Dexcom G4 protocol did not use cryptographic algorithms; therefore, a cyber attacker could obtain knowledge of Dexcom's G4 protocol to trigger an attack [465].

An APS security analysis was performed using model UPPAAL statistical model checking to demonstrate APS hackability. A scenario of a replay attack was considered to evaluate the security performance [204]. UPPAAL statistical model checking was used to critically analyse the properties and variables of priced timed automata networks. Timed automata is defined as different clock rates under a stochastic semantics analysis. Stochastic semantic analysis is used to understand natural language, including segments of symbols or words. In general, UPPAAL statistical model checking uses the statistical model of group extensions to manage real-time systems and examine undecidable problems by comparing probabilities. Timed automata, a finite automaton, is a mathematical model of computation. The automation clock values increase during the run time at the same speed. Clock values can be examined to enable or disable to evaluate the behaviour of the automata. As a result, the researchers successfully triggered a replay attack; however, the researchers were unable to propose any mitigation solution.

In a study [466], the researcher demonstrated a case study of a replay attack on an insulin pump. The researcher modelled an IoT-based system wherein data was transmitted

between two networks. The system was built as a communication scenario between the RFID sensor tags and the RFID reader, a network between a plant (insulin pump and an auxiliary system) and the command-and-control center to send the real-time control commands and received data for monitoring. The researcher considered a replay attack on the network between the C&C and the plant consisting of RFID sensors, and the insulin pump and the readers [466]. The researchers stated that a replay attack has been discovered within automated insulin delivery systems. In a replay attack, the attacker intercepts and replays legitimate data to manipulate insulin administration. For instance, an attacker could record and subsequently replay a real insulin dosage, thereby delivering a dangerous quantity of insulin to the patient. Consequently, monitoring and safeguarding of these control systems necessitate immediate consideration [466].

2.5.2.2 Replay: Existing Solutions

In [465] Reverberi et al., proposed the use of AES-128 encryption to mitigate eavesdropping, jamming, replay, forging, and DoS security attacks. However, to date, limited investigation has been performed on the proposed AES-128 encryption. Therefore, this encryption must be examined to assess its accuracy for mitigating the rate of attacks. Another study proposed a cancellable fingerprint authentication for insulin pumps by using Delaunay triangulation [464]. However, the patient with the chronic condition may be unable to respond to fingerprint authentication (see Section 2.5.1.3). Nevertheless, fingerprint authentication might be a future direction for researchers. Encryption approaches to strengthen the security of embedded medical devices are challenging due to a lack of computational power, and yet at the developing phase, research communities have argued that embedded medical device data should be encrypted [467, 468].

According to [469], lightweight cryptography has emerged as a viable solution to enhance IoT devices' privacy and confidentiality features. The researcher has compared three

distinct algorithms: AES-128, Speck, and Ascon. AES-128 constitutes an advanced encryption standard characterised by a 128-bit key, Speck represents an add–rotate–xor (ARX) cipher, and Ascon embodies a collection of the lightweight authenticated cipher. The researchers stated that although Speck represents a good option for contexts wherein high throughput and low latency are paramount, and energy availability poses no significant obstacle for IoT devices, AES-128, and Ascon merit consideration due to their energy efficiency and minimal memory consumption. Although the AES is an algorithm the cryptographic community has examined, it encompasses complex mathematical operations involving numerous iterations of substitution, permutation, and mixing [469].

A smart factory and jamming attack have been demonstrated in one study [470]. A smart factory IoT network monitors and controls essential industrial processes with many IoT devices and sensors. Devices monitor temperature, pressure, humidity, and other real-time characteristics to ensure product quality and safety. The network collects and distributes data from an access point. All IoT devices can communicate with the access point. The researchers stated that this design raises security concerns since one IoT device intentionally may function as a jamming device that causes communication disturbance among other devices. Interference can lead to unreliable or missing data at the access point, resulting in suboptimal manufacturing operations choices that could lower product quality. Jamming attacks consume the network channel with malicious signals, making it hard to send or receive normal communications [470].

AES-128, when considered a single solution, may not directly address the issue of jamming; however, it can significantly enhance defence mechanisms through various means. Initially, the concepts of Data Integrity and Confidentiality, Notwithstanding the potential for a jamming attack to interfere with the transmission, AES-128 confirms that any data that may be intercepted remains indecipherable. For instance, via the process of encryption, data receives a transformation that renders it difficult to comprehend to any individual who does not possess

the requisite decryption key [471]. Secondly, Frequency Hopping: AES can safeguard frequency-hopping sequences, which are widely recognised as an effective anti-jamming technique. Through encrypting the hopping pattern, AES ensures that only authorised devices can anticipate and synchronise with the subsequent frequency [472].

Replay attacks encompass the interception of legitimate data packets, followed by their subsequent retransmission to deceive the system. When considered a single solution, it may not directly address the issue of replay; nonetheless, the goal of replay attack defence can be achieved through the following means: Initially: Nonces and Timestamps: Encrypted communications may incorporate distinctive nonces (numerical values utilised singularly) or temporal markers. The receiver confirms these elements to confirm the message's originality and prevent potential replay attacks [473, 474]. Secondly, the implementation of sequence numbers: The encryption of sequence numbers utilising the AES confirms that each packet is assigned a distinct identifier. The receiver checks the sequence to identify and discard any duplicate or out-of-order packets [475].

2.5.2.3 Replay: Remaining Challenges

According to [476], replay attacks involve the fraudulent repetition or a delay of legitimate transmissions. In contrast to various other forms of cyber-attacks, replay attacks do not require prior knowledge of the system's architecture, encompassing specifics regarding controllers and measurement units. As stated in a study [477], replay attacks are comparatively straightforward to execute, given that the attacker requires no antecedent understanding of the system. Similarly, as indicated in [478], individuals with varying hacking knowledge constitute prevalent adversaries. Individuals lacking specialised expertise are typically capable of identifying the accessibility of hosts within control networks and employing standard tools for penetration testing; however, they possess no special knowledge of the communication

protocols related to control systems. This makes it logical to assign varying degrees of threat to external individuals based on their skills [478].

According to the literature [476-478], a cyber attacker can execute a replay assault without needing specialised knowledge of the principal protocol. However, the attacker must be able to intercept data packets by eavesdropping. Subsequently, the attacker interrupts and retransmits the data packets. To counteract a replay attack on an APS, a lightweight mitigation solution is necessary to accommodate the device's limited resources. Further, encrypted session identifiers that are unique for each communication are recommended to effectively prevent replay attacks. OpenAPS does not employ this strategy, therefore making it susceptible to a replay attack. Due to the limited resources of this APS device, the possibility of implementing a unique encrypted session identification is now being explored as a research focus. In one study [466], the researchers demonstrated a replay attack within automated insulin delivery systems. In a replay attack, the attacker intercepts and subsequently retransmits legitimate data to manipulate insulin administration. For example, an attacker could record and subsequently replay insulin dosage, consequently administering a harmful amount of insulin to the patient [466].

2.5.3 Artificial Pancreas System Man-in-the-Middle Attacks

An MITM attack refers to the unlawful interception and alteration of communication between devices and various components by a cyber attacker or unauthorised user [479]. A perpetrator secretly manipulates the data packets delivered between the sender and recipient. However, both parties believe that the sent and received data packets are legitimate. The attacker intercepts data packets and inserts inaccurate or deceptive information.

2.5.3.1 Man in the Middle: Current Problems

MITM attacks can involve malicious code that intercepts or modifies a patient's health information in their device. For example, one study found an attacker could alter the wireless communication of an insulin pump, and the researchers highlighted that the attacker could signal acute overdose attacks because the insulin sensor's communication was unencrypted [480]. The researchers demonstrated that the bad actor could run malicious code in a wireless communication channel to obtain the patient's device information [481].

Smartwatches currently use Android applications to link with diabetes treatment. Connecting diabetic devices requires careful consideration for privacy and security. Chauhan et al. analysed three smartwatch applications, Android Wear, Samsung, and Apple, using MITM proxy to capture and decrypt the communication between smartwatch devices and network traffic. These researchers successfully captured different smartwatch traffic, such as user credentials and patient health data [482]. As a result, smartwatch applications are vulnerable to MITM attacks due to unencrypted communication.

In [483], the researchers stated that the attacker could execute various attack operations, including MITM attacks, impersonation, and replay attacks, to gain unauthorised access to the confidential data communicated by implantable medical devices such as cardiac implants, insulin pumps, deep brain neurostimulators or to manipulate configurations, thereby compromising patient health [483]. In another study [484], the researchers demonstrated smart healthcare systems security analysis using MITM, data injection, and DDOS attacks [484]. In [485], the researcher demonstrated the IoMT security attacks, including d DDoS, MITM, masquerading, and Sybil attacks using the NS2 simulator.

In a study [486], a digital twin reference model linked the human body and medical devices to the IoMT ecosystem was developed to highlight security attacks. The researchers stated that unauthorised access to the digital twin reference model can violate privacy and cause

harm. The researchers stated that an MITM attacker intercepts and perhaps manipulates two-party communications, obtaining or changing sensitive data. The researchers also stated a list of possible security attacks. The list of attacks includes impersonation, sniffing, replay, brute force, tampering, side channel, jamming, DDoS, and clock synchronisation attacks [486].

2.5.3.2 Man in the Middle: Existing Solutions

ML approaches are used for attacking and defending IoT devices. The patient infusion pattern-based access control scheme was proposed to protect against insulin pump overdoses. The ML supervised technique using support vector regression was successfully used to predict abnormal doses [487]. However, according to Qu et al., the proposed solution by Hei et al., in [487] requires an accurate and validated design to deploy properly. Consequently, the suggested method would not work as efficiently as it could in an IoT setting [488].

In another study, a Deep Learning (DL) model and gestures using a Long Short-term Memory (LSTM) network, also described as a Recurrent Neural Network (RNN), was proposed to identify an insulin pump overdose. A threshold was designed to predict valid doses. The threshold value stores insulin doses for three months. If the doses are greater than the stored value, then an altered message using LSTM is generated for the patient to take suitable action [489]. According to Waheed et al. [490], the proposed solution by Ahmad et al. [489] resolved the vanishing gradient problem of RNN and improved the accuracy; however, the model's design and analysis were absent [490]. This proposed solution therefore requires implementation and further analysis.

2.5.3.3 Man in the Middle: Remaining Challenges

The integrity of wireless communication between an insulin pump and CGM sensor can be compromised by several types of MITM attacks. For example, research indicated that acute overdose attacks may have occurred due to the absence of a defensive mechanism [480].

Effective countermeasures against MITM attacks remain limited in proprietary devices. Moreover, other solutions, such as implementing robust cryptographic algorithms, using reliable antivirus and antimalware software, regularly patching software, deploying intrusion detection systems, and enforcing access control measures, are limited. Further device and resource limitations make the task of selecting a suitable defensive mechanism difficult.

2.5.4 Artificial Pancreas System Malware Attacks

In the field of cybersecurity, an attack vector can be defined as a basic pathway that a malicious actor uses to illegally acquire unauthorised entry, which allows the execution of payloads or malware-driven consequences. The potential implications of malware encompass a wide array of adverse consequences, including the disruption of operational processes, the illegal gain of sensitive information, the imposition of unauthorised privileges, and a decrease in system performance. Furthermore, the consequences of malware are dependent upon the attacker and their use of malicious code.

2.5.4.1 Malware: Current Problems

As mentioned in Section 1.3.3, the FDA released their first warning alert on the cybersecurity susceptibilities of the Hospira infusion pumps in 2015. As mentioned in Section 1.3.3, the FDA released their first warning alert on the cybersecurity susceptibilities of the Hospira infusion pumps in 2015, and they released a cautionary notification on the Medtronic MiniMed Insulin Pumps in 2019. As per the FDA's 2019 notice and research conducted by Klonoff and Han, the MiniMed insulin pumps experienced difficulties while incorporating the most recent software or patch upgrades [88, 323]. As a result, the research community were made aware of various malicious software that can potentially attack insulin pumps, causing them to lose their ability to link directly to CGM systems, eventually initiating various malware vulnerabilities.

The usage of IoMT devices for managing chronic illnesses poses several cybersecurity issues, which have attracted considerable interest from the scientific community. Wearable and implantable technologies need strong security features, such as protection against malware infiltrations. Typically, embedded IoMT devices rely on default passwords and pin or key mechanisms for operation, while the transmission and exchange of data lack strong security protocols [89, 93, 94, 491, 492]. According to existing scholarly discussions, APS vulnerabilities provide a susceptibility to damaging malicious attacks [81]. The APS is manufactured to deploy wireless connections throughout many platforms, including cloud infrastructure and mobile devices running on either Android or iOS. Consequently, this integration gives rise to security risks that stem from potential faults in interaction-based processes [87].

2.5.4.2 Malware: Existing Solutions

In [493] Millard, stated that automated security systems can prevent malware attacks. Healthcare providers encounter a significant number of malware attacks daily. One possible cause of these attacks is the inadequate defence mechanisms and shortage of security specialists in the medical field to effectively address the malware threat [493]. The TrapX security organisation states that the expanding volume of malware threats are expanding due to insufficient knowledge and defences (Millard, 2017). Effective defences can incorporate code and a deep understanding of malware. TrapX uses automated techniques and strategic deception to safeguard medical organisations. Automated solutions can imitate and react in the same manner as actual devices that end users might encounter on a healthcare network [493].

2.5.4.3 Synthesis of Artificial Pancreas System Cyberattacks

In recent years, the cybersecurity of insulin pumps and CGM sensors have become more important. The growing number of these devices has resulted in a parallel increase in

digital security and security threats. Section 2.3 explored the problems of medjacking and the use of limited protection measures in the field of cybersecurity. Table 2.8 presents a synthesis of APS cyberattacks to understand medjacking issues. The synthesised summary emphasises the various challenges associated with each type of attack and offers a summary of each one.

Table 2.8

A Synthesis of Artificial Pancreas System Cyberattacks in Current Literature.

Reference	Attack	Target component	Target platform	Defence challenge	Summary
[463]	Tampering	Insulin pump and CGM sensor	System level	Detection agent	Used glucose-insulin model of UVA/Padov Type 1 Diabetes Simulator to design detection agent
[204]	Replay attack	Insulin pump and CGM sensor	System level	Found unencrypted channels	Used a model checker UPPAAL statistical model checking through priced timed automata
[464]	Eavesdropping, and active adversary attack	Insulin Pump	Wireless channel	Found access to the pump was not secure	Used hardware to register fingerprints and stored them in the insulin pump's memory.
[88, 323]	Malware	Medtronic MiniMed Insulin Pumps	Network	Unable to update the new software or patch	Generated a report by the FDA
[494]	Malicious attacks	Insulin pump	Wireless channel	Found lack of security mechanisms	Used pin authentication using visible light signals
[489]	MITM	Insulin pump	Wireless channel		Designed a threshold to predict valid doses.

Reference	Attack	Target component	Target platform	Defence challenge	Summary
[465]	Eavesdropping, jamming replay, forging, and DoS	Dexcom G4 CGM	Communication channel	Cryptographic measures were not found to protect the data	Used software-defined radio setup based on the HackRF one and GNU Radio
[202]	Hijacking and spoofing attack	Animas OneTouch Ping pump	Network		Spoofing attack
[495]	Remote attack via malware	Symbiq infusion pump	Network	Found no software updates	Unauthorised access
[496]	Spoofing	Infusion pump	System level		Reverse engineering
[487]	MITM	Medtronic Paradigm real-time insulin pump and OneTouch meter CGM sensor	Wireless channel	Found no authentication scheme	NA
[480]	MITM	Medtronic insulin pumps MiniMed 511, 512, 522 and Paradigm	Wireless channel	Found no authentication	Used wireless sniffer
[86]	Replay attack	Insulin pump and CGM sensor	Wireless channel	Found no authentication	NA
[242]	Hijacking and impersonation	Insulin pump	Wireless channel	Found no authentication	Reverse engineering

Table 2.8 presents, existing attacks, and defence challenges for mitigating medjacking on APS devices. A wireless channel is a communication medium that utilises electromagnetic waves to transmit data without physical connections such as cables or wires [497, 498]. Wi-Fi, cellular networks, satellite communication, and Bluetooth are examples of wireless channels [499, 500]. A network represents a system of interlinked devices or nodes, encompassing computers, servers, sensors or IoT devices, that participate in communication to facilitate the sharing of resources, data, or services [501-503]. Networks can be classified according to their prospective (e.g., Local Area Network, Wide Area Network), architecture (e.g., peer-to-peer, client-server), or technological structure (e.g., wireless, wired) [504-506]. The difference between a wireless channel and a network is that a wireless channel and a network represent fundamentally different constructs within communication systems, characterised by scopes and functional roles [507-511]. A wireless channel denotes a mechanism through which electromagnetic signals are transmitted without the use of physical connections between devices [497, 498]. Conversely, a network constitutes a broader structure of interlinked devices, including computers, IoT devices, and servers, which communicate to facilitate the sharing of data, resources, or services [501-503]. Fundamentally, a wireless channel constitutes a link for transmitting signals. In contrast, a network represents a structure amalgamating diverse components and protocols to facilitate data exchange and connectivity among devices.

The research community has made significant progress, however, each of the findings can be used to reinvestigate the current issues to propose feasible mitigation solutions for medjacking attacks. The research community and healthcare organisations have exhibited a profound interest in mitigating security challenges in the APS. In the current literature, medjacking has been identified as a serious issue. This literature has stated that cyber-attacks such as tampering, replay, eavesdropping, MITM, DoS, and malicious attacks, were reported in the APS devices.

In [494], the researchers stated, “Many wireless insulin pump systems lack security mechanisms to protect them from malicious attacks”. This study presented their findings; to understand further the meaning of the lack of security mechanisms, this study provided further explanation. The justification for the keyword “lack of security mechanisms”, used in Table 2.7, is also present in the current literature. Various researchers use this keyword; for example, according to [512], IoT's rapid use has generated an extensive risk environment for computer networks, exposing vulnerabilities. These devices lack sufficient security mechanisms without appropriate protection, compromising systems and data [512]. Similarly, according to [513], the rapid growth of the IoT and its lack of security mechanisms render IoT devices a tempting target for malicious actors, including malware [513]. In another study [514], the researchers used conventional strategies for mitigating ransomware, which is generally efficacious within personal computing contexts while inadequate in the context of IoT networks. This shortcoming emerges from the inherently fragmented architecture of IoT infrastructures and a lack of unified security mechanisms [514].

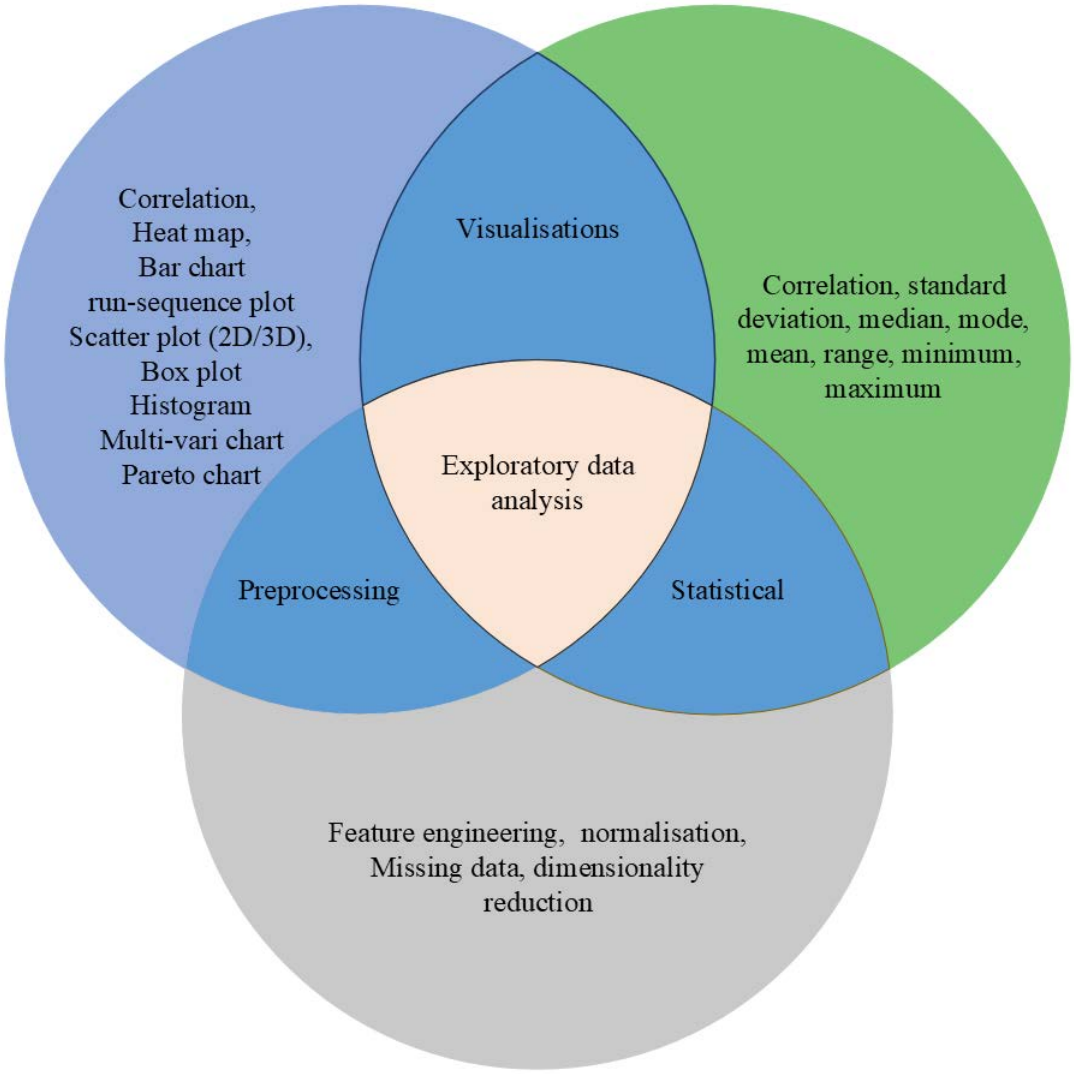
The "lack of security mechanisms in IoT" can refer to IoT devices and systems without protective protections. According to [78], IoMT has security and privacy concerns, such as a lack of security and privacy measures. The researchers stated that IoMT devices are vulnerable to wireless/network attacks since IoMT devices use open wireless connections. An attacker can eavesdrop and intercept both inbound and outbound data and information since most IoMT devices lack security by design or have poor security authentication. The failure to identify and prevent such attacks makes it possible to get unauthorised access. This might lead to increased privileges, malicious injection, or device infection with malware. However, IoMT devices might be hijacked as botnets and used to perpetrate DDoS [78].

In addition to employing social engineering tactics, attackers can exploit the vulnerabilities to mislead patients or healthcare personnel, thereby coercing them into

providing crucial information that could compromise the security of the insulin pump. Multiple consequences are associated with medjacking, as it could negatively impact individual patients, and other important notable consequences can be life-threatening. Hijacked APS devices could deliver incorrect doses or provide incorrect readings related to individual diabetic control. The unauthorised alteration of insulin dosage can give rise to the potential occurrence of hypoglycaemia or hyperglycaemia, thereby posing a significant risk to the safety of the patient.

2.6 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a statistical approach used to interpret raw data. The EDA approach to raw data in the context of cybersecurity provides insight into normal or attack patterns and characteristics [515, 516]. The EDA procedure starts with gathering raw data from devices, preparing the data to manage missing values and outliers, normalising or scaling the data, and converting data types [517]. EDA can identify numerical variables' correlation, mean, median, mode, standard deviation, and quantiles [518]. To comprehend data distribution and variables, EDA outcomes can be represented in histograms, box plots, scatter plots, and charts. These visualisations reveal data trends, patterns, and relationships. Figure 3.1 shows the EDA Venn diagram.



Examining cyber-attacks entails employing diverse methodologies to comprehend the attack and determine patterns to minimise the consequences. Table 2.9 presents techniques used to analyse cyber-attacks.

Table 2.9

Exploratory data analysis techniques for Cybersecurity.

Reference	Technique	Description	Process
[519]	Forensic analysis	Examining systems or devices to gather evidence,	Logs, memory dumps, network traffic, and other data are collected and examined to

Reference	Technique	Description	Process
		discover the cause, and follow the perpetrator.	reconstruct the attack and understand its methodology.
[520]	Packet Inspection	Examining the packets to visualise the efforts of systems or applications.	Deep packet inspection checks packets at different levels for attack activity, such as uncommon patterns and data extraction.
[521]	Log Analysis	Examining system and network logs to identify indications linked to hijacking attacks.	Analyses log entries from different sources to find unusual patterns like failed logins, unauthorised privilege escalation, or suspicious file changes.
[522]	Memory Analysis	Examining an entity's memory for unusual characteristics like inserted code, rootkits, or process behaviour.	Memory dumps and memory architecture are examined to find hijacking patterns.

Log analysis and packet analysis are two separate methodologies used to investigate security-related events. Although both activities require examining data, each has distinct focuses and serves separate purposes. Log analysis entails evaluating logs produced by interconnected systems, applications, and devices to identify signs of compromise indicators related to security events. Logs can originate from several sources, encompassing operating systems, network devices (such as routers and firewalls), servers, applications, and security appliances. Records usually consist of logs documenting system events, user actions, network connections, authentication trials, configuration modifications, and other pertinent data. Log analysis is a process that aids in the identification and investigation of security issues.

Packet analysis entails investigating individual packets to get valuable information about their relationship. To obtain it, researchers collect datasets to conduct the packet analysis. The

dataset consists of unprocessed data that is transmitted between interconnected devices. These packets include headers, payloads, source and destination addresses, and information particular to the protocol being used. Packet analysis is a method used to identify patterns. Although log analysis and packet analysis both can be used to analyse medjacking indicators, both techniques have distinct functions and purposes. Log analysis is the examination of system events and activities that are documented in log files. On the other hand, packet analysis involves analysing the data that is communicated with interconnected devices, which gives detailed information about data patterns.

2.6.1 Exploratory Data Analysis in IoMT

EDA is indispensable in the context of IoMT, as it enables the discovery of significant patterns and the identification of anomalies within datasets [523-529]. The datasets associated with the IoMT typically encompass a variety of data types, including time-series data derived from sensor readings, categorical data reflecting device status, and images obtained from medical imaging devices [530-536]. Visualisation is pivotal in EDA, enabling the discernment of patterns, relationships, and anomalies [16, 209, 537]. Techniques such as histograms for the analysis of distributions [538], scatter plots for the exploration of variable connections [539], time-series plots for the observation of trends [540], and heatmaps for the examination of correlations are of paramount importance [541].

For example, the histogram approach provides a graphical depiction of a single variable's distribution of information or data. In the context of IoMT applications, histograms serve to elucidate the distribution of sensor readings, including but not limited to heart rate and glucose levels, across an array of values [526, 542]. The histogram approach allows for discovering patterns, including normal ranges and skewness within the dataset [543, 544].

The heatmap serves as a guide for examining relationships among variables. In the context of the IoMT, a study [545], used a heatmap representation that employed the Pearson

correlation coefficient matrix to analyse various features, such as fatigue level and heart rate, thereby understanding potential interdependencies among these metrics. Heatmaps provide an effective approach for examining the interrelationships among various variables within a dataset.

Time-series plots provide significant value for IoMT data, enabling metrics visualisation. These plots are essential for monitoring variables such as blood pressure or heart rate over various intervals [545]. Time-series plots enable the identification of trends, periodic patterns, or anomalies that may indicate significant conditions. Each approach functions as a fundamental basis for comprehending the dataset's architecture, identifying anomalies, and preparing data for sophisticated analytical endeavours. Collectively, EDA techniques enable researchers to acquire significant insights and formulate data-driven decisions within the field of IoMT.

2.6.2 Exploratory Data Analysis in IoMT Cybersecurity

Forensic analysis within the context of IoMT cybersecurity mainly involves discovering evidence associated with cyber incidents. The statistical and temporal measurements are a fundamental aspect of EDA within forensic contexts [209, 538, 546]. In [547], the researchers conducted a forensic analysis within the domain of the IoMT. The researchers stated that an attack occurs either in a physical or virtual context, encompassing media that meets the criteria to be classified as evidence at the crime scene. Cyber forensics defines the systematic gathering and analysis of evidence to yield substantive findings. The process of cyber forensics is executed through a series of systematic stages, which encompass a) the identification of evidence, b) the preservation of evidence, c) the acquisition of evidence, d) the analysis of evidence, and e) the reporting of findings [547]. EDA offers a systematic approach to the preprocessing, visualising, and analysing data, thereby determining forensic evidence, including anomalous patterns and unauthorised access [524, 548-550].

Packet inspection represents a fundamental component of cybersecurity within the IoMT environment. Packet inspection includes examining data packets to identify cyber threats [551-554]. Identifying anomalies via packet inspection utilises advanced techniques, including correlation analysis and ML approaches. For instance, in [555], to identify wireless spoofing attacks within the IoMT, the researchers proposed a hybrid approach that integrates spatial correlation analysis, deep learning classification, elliptic curve cryptography encryption, and design science research methodology for attack detection frameworks [555]. Packet inspection offers a systematic method for identifying anomalies and comprehending communication patterns among interconnected medical devices.

The analysis of logs makes an essential element of cybersecurity within the IoMT framework. Logs serve as a comprehensive account of activities, encompassing device communications, user activity, and system activity, which can be investigated to identify anomalies and investigate security incidents. In [556], the researchers proposed Chidroid, a mobile Android application designed to retrieve, collect, and distribute logs from smart healthcare devices to analyse security concerns [556]. Log analysis presents a multitude of practical applications within the realm of cybersecurity for the IoMT.

2.7 Cybersecurity Measures

Cybersecurity measures serve to improve device, data, and network security against medjacking. A broad range of methods exist for the implementation of cybersecurity measures. These methods encompass an extensive variety of techniques, such as access controls, encryption, and intrusion detection. Cybersecurity measures can be proactive [557, 558] and reactive [559] defence strategies.

In the context of vulnerability assessment, proactive strategies identify and mitigate potential security threats and vulnerabilities, thereby preventing a hacker’s exploitation of user. In contrast, reactive strategies are exclusively designed to tackle attacks that have already been

initiated. Reactive defence mechanisms endeavour to determine cybersecurity attacks and subsequently clean up their damaging consequences [559].

The exploration of AI within the field of proactive cybersecurity has emerged as a significant opportunity to mitigate cybersecurity vulnerabilities. As per the existing literature, AI has been used to automate various security-related undertakings, including but not limited to identifying potential threats, promptly responding to cyberattacks, recognising normal and attack patterns, and scanning for vulnerabilities [560].

Automated Cyber Defence (ACD), alternatively referred to as autonomous cybersecurity, serves as an effective safeguard for devices or computer systems, networks, servers, and data. ACD reliably protects end users from a range of serious cyberattacks through the utilisation of automated procedures, cutting-edge technologies, and advanced tools [561]. The main goal of ACD is to reduce the need for human intervention and improve the effectiveness of attack detection, response, and mitigation.

The integration of ACD encompasses real-time threat detection techniques that evaluate system and network activity to identify any anomalous action. Intrusion detection systems and security information and event management solutions are rapidly used in ACD. These systems possess the capability to identify risks by using pre-established patterns and ML techniques [562].

The effectiveness of ACD is based on the utilisation of AI techniques [563]. ACD employs advanced models to analyse data in real time, enabling the models to identify complex patterns and detect anomalies that may fail human perception. In addition to detection patterns, ACD techniques include software patching, user access control, and log analysis. ACD enables continuous surveillance and timely alert mechanisms. The ACD systems effectively incorporate external threat intelligence streams to maintain up-to-date awareness of prevailing

threats and vulnerabilities. The seamless integration of these components significantly augments the efficacy of threat detection and response capabilities.

2.7.1 Justification of How ACD Security Measures Apply to IoMT Devices

Automated security measures based on AI are significant for safeguarding IoMT devices. The advent of AI rapidly transformed the cybersecurity paradigm, enabling the implementation of proactive defence methodologies adept at detecting, predicting, and mitigating cyber threats [111-114, 423]. According to [423], the researchers stated that ML and DL approaches can enhance cybersecurity within the IoMT. The beneficial potential of AI-driven cybersecurity within the IoMT domain has demonstrated considerable significance for safeguarding patient data, thereby facilitating the development of an innovative age that includes tailored and data-driven healthcare [423].

AI-based ACD security measures apply ML techniques to IoMT devices to detect or hunt threats, categorised into five distinct classifications: (i) Supervised Machine Learning, (ii) Unsupervised Machine Learning, (iii) Reasoning techniques, (iv) Graph-based approaches, and (v) Rule-based approaches [114]. These ML approaches evaluate patterns or application logs to discern anomalous activities or trends. According to [564], signature detection determines attacks by comparing the analysed event with established patterns, thereby ensuring high accuracy. Anomaly detection determines anomalous behaviour by learning the system's normal patterns through the analysis of historical data. Any deviation from the normal patterns is identified as anomaly behaviour [564]. Talukder et al., stated that ML algorithms can comprehend information derived from historical datasets, thus facilitating the identification of anomalies or patterns that may signify potential intrusions [565].

To deploy AI-driven approaches to identify anomalies, the researchers typically construct simulated IoMT ecosystems to evaluate AI models and algorithms [288, 484, 566, 567]. The datasets are subsequently used to formulate and demonstrate proof-of-concept

approaches, thereby improving the effectiveness of AI-driven solutions. For instance, the HealthGuard dataset was developed [234] and consisted of eight simulated medical devices, including DoS attacks, and fake data injection attacks. In a further illustration, in [568], the BoT-IoT dataset was developed consisting of a smart fridge, smart garage door, weather station, smart lights, and smart thermostat IoT ecosystem with Botnet attacks examples [568]. AI effectively identifies and mitigates cybersecurity threats by analysing extensive datasets derived from IoMT devices. Automated detection techniques employ ML and DL models, the model development phase encompassing feature engineering, wherein diverse ML and DL models can be trained depending on specific tasks [115].

The diverse models can be employed as ACD security measures; these models encompass, but are not limited to, the Decision Tree (DT) model utilised within the cybersecurity domain [569]. The Random Forests (RF) model applies to ransomware-related cyber threats, facilitating the classification task through categorical-numerical classification strategies [570]. The Support Vector Machine (SVM) model serves to classify malware and detect unauthorised access attempts [571]. The Logistic Regression (LR) model is a valuable tool for executing security operations, including malware and anomaly detection [572]. The Naive Bayes (NB) model demonstrates effectiveness in applications, including email spam filtration and website phishing identification [573, 574]. The Multilayer Perceptron (MLP) model applies to classification tasks pertinent to cybersecurity [575]. The Convolutional Neural Network (CNN) architecture can be used within the cybersecurity domain to classify malware imagery [576]. The implementation of Long-Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models serves as an option for the detection of malware and intrusions [577, 578]. An autoencoder model serves as a mechanism for discerning deviations from established patterns, thereby facilitating the detection of anomalies [579]. Extreme Gradient Boosting (XGBoost), Light Gradient-boosting Machine (LightGBM), and Categorical

Boosting (CatBoost) utilise Gradient-boosting Machines (GBM) are commonly used in the field of phishing detection, malware classification, and intrusion detection [580]. These models can be implemented as ACD security measures for IoMT devices. These models classify tasks pertinent to cybersecurity and utilise them to identify attacks autonomously.

2.7.2 Justification of How Al-Driven Security Helps Prevent an Attacker from Tampering with an IoMT Device

The progression of the IoMT has significantly impacted the healthcare domain. Nonetheless, these advancements open safety risks and weaknesses that compromise patients' lives, potentially leading to medical complications [78, 79]. The IoMT systems and devices are vulnerable to an extensive range of security challenges, including medjacking [80-82], unauthorised access [83], data breaches, and the manipulation of device functionality [84, 85]. The emergence of interconnected medical devices, including insulin pumps [81, 82, 86-89], CGM [81, 90-92], pacemakers [93-98], and infusion pumps [99], has simultaneously targeted for hijacking and various cyber-attacks, as documented in the literature. Furthermore, their increased connectivity renders them susceptible to considerable cybersecurity threats, encompassing tampering attacks. According to [146], any alteration to the data within IoMT systems during collection, transit, or storage phases is classified as a tampering attack [146].

In [581], the researchers stated that data modification or tampering attacks can modify medical data randomly or by specific guidelines. Modifications with metainformation or patient biomarkers pose a significant risk to patients, as modification may lead to inaccurate treatment decisions stemming from incorrect diagnostics associated with this form of MITM attack. For example, a MitM attacker can cause incorrect insulin administration by manipulating the measured values of an insulin pump [581]. Healthcare data possess an increased vulnerability compared to other data categories, as any tampering with such information can result in inaccurate treatment, potentially resulting in life-threatening and

irreversible consequences for patients [582]. Tampering may result in data compromises [582], operational disruptions [429], or potentially compromise patient safety [581].

AI-driven security solutions provide advanced mechanisms to mitigate cybersecurity threats effectively. Using ML models is one of the principal mechanisms by which AI mitigates the risk of tampering. In [421], the researchers stated that anomalies or unusual occurrences within patient data and network traffic can be identified through AI [421]. According to [583], the researchers stated that ML models, such as Random Forests and SVM, can identify anomalous patterns, including unanticipated spikes in heart rate, which may signify potential health concerns or data tampering [583]. In [584], researchers proposed ML-based models, specifically Artificial Neural Networks, Decision Trees, Random Forests, and K-Nearest Neighbours, to identify instances of tampering, DoS attacks, and the injection of fake data within the IoMT [584]. AI models and algorithms examine the behaviours of IoMT devices to discern anomalous patterns that could indicate potential security attacks.

A further significant benefit of AI in the context of IoMT security is its capacity to automate incident response. According to [585], an AI-driven incident response system can autonomously execute measures such as isolating compromised systems, blocking malicious IP addresses, or alerting administrators during a cyberattack [585]. Comparably, according to [586], AI can automate many security-related tasks, including log analysis, threat detection, and incident response [586]. Furthermore, AI systems constantly learn and adapt to threat updates by incorporating current data. As stated in [587], AI can examine and assess the ecosystem within its functions, thereby collecting insights from any newly recognised threats or vulnerabilities and enhancing its algorithms for future evaluations [587].

AI-driven security systems offer resilient, adaptive, and scalable solutions that prevent potential attackers from tampering with IoMT devices [423]. AI effectively examines the distinct challenges associated with the cybersecurity of the IoMT. These measures provide the

reliability, and integrity of IoMT devices, thereby safeguarding patient health and preserving trust in healthcare technologies. As IoMT devices attain greater ubiquity [186, 429, 588], the role of AI will become progressively paramount in safeguarding these systems from ever-evolving threats.

2.7.3 Justification of How ACD Systems Monitor Communication between IoMT Devices to Detect Anomalies that Might Indicate a Medjacking Attempt

ACD approaches are indispensable for safeguarding IoMT ecosystems, as they provide continuous monitoring and identify anomalies or atypical patterns that may signify a hijacking attempt or other cyber threats [209, 581, 588-593]. The examination of logs constitutes a fundamental component of cybersecurity within the IoMT ecosystem [146]. Logs function as an exhaustive record of activities, incorporating device communications, user interactions, and system operations, which can be inspected to detect anomalies. Moreover, within the context of ACD, monitoring IoMT devices through packet inspection constitutes an essential component of cybersecurity in the IoMT ecosystem [553, 554, 594-597]. Packet inspection entails the examination of data packets to discern potential cyber threats [551-554]. Packet inspection provides a systematic method for discerning anomalies and illustrating communication patterns among interlinked medical devices.

In the context of ACD, monitoring IoMT devices through behavioural analysis enables such systems to establish a typical baseline of communication patterns for IoMT devices. Any deviation from these established patterns may signify malicious behaviour. An anomaly can be defined as a deviation from conventional patterns, anomaly detection determines unusual behaviour by learning knowledge of the system's typical patterns by examining historical data. Any deviation from the normal patterns is considered anomalous behaviour [564]. Anomaly detection can be characterised as systematically examining recorded activities to discern significant deviations from occurred instances [598]. In the context of signature detection,

according to [564], signature detection examines attacks by comparing the analysed patterns with established patterns, thereby ensuring high accuracy. These systems employ ML techniques to characterise normal device behaviour. AI-driven ACD security monitoring employs ML techniques to monitor and identify or hunt threats within IoMT devices [114]. ML techniques assess patterns or application logs to identify anomalous behaviours or trends.

To implement AI-driven monitoring techniques for identifying anomalies, researchers commonly develop simulated IoMT ecosystems to assess AI models and algorithms performance [288, 484, 566, 567]. The datasets constitute the foundation for formulating and evaluating proof-of-concept techniques to detect hijacking endeavours. ACD systems provide comprehensive, real-time monitoring of IoMT devices, effectively identifying and minimising hijacking attempts. These systems provide robust security through the utilisation of techniques, including packet inspection, behavioural modelling, and ML techniques [551-554, 594-597].

2.7.1 Artificial Intelligence in Cybersecurity

Intrusion detection, behaviour analysis, access control, biometric authentication, and virus detection are just a few of the many security measures that may benefit from Artificial Intelligence (AI) applications in IoMT cybersecurity [14]. By allowing for the rapid detection of attacks, patterns, and automated responses, AI applications can improve the IoMT's security [564]. AI uses predictive analysis of historical data, which allows for the prediction of possible security flaws or vulnerabilities in IoMT devices. AI in the field of cybersecurity comprises ML and the ML subcategory, DL [599]. In principle, ML is a form of AI capable of adapting autonomously with minimal human intervention. DL is a subcategory of ML that utilises artificial neural networks to emulate the intricate process of knowledge in the human brain. The use of ML and DL models in IoMT devices can efficiently detect potential security issues.

ML models can learn from time series data to create models that can effectively forecast circumstances. ML has demonstrated its effectiveness in addressing problem domains,

including extensive rule-based systems and complex problem-solving scenarios where conventional methodologies are deemed inefficient [599]. Moreover, ML exhibits a remarkable potential for adapting novel information, particularly within a dynamic and progressive environment, such as the IoMT ecosystem. Automated ML capabilities have the potential to enhance security measures within the IoMT ecosystem. In the current literature, ML in intrusion detection yields a high level of effectiveness in detecting attacks [600].

DL applications consist of layers within the system that are interconnected via neural networks, serving as intermediaries for the complex mathematical computations that underlie the processes of learning [601]. DL has shown effectiveness across various domains, including recognition and processing (e.g., processing images, audio, video, text, and patterns), natural language processing, autonomous systems, and robotics [602]. Numerous DL techniques for anomaly intrusion detection have been proposed in the literature [603]. The rapid development of new technologies and the growing interconnection of services and devices have led to various cyberattacks. As the added complexity of attack scenarios has increased, DL techniques have demonstrated their effectiveness in relation to attack detection and classification tasks. Within the DL applications for intrusion detection, deep neural networks develop knowledge from a collection of historical data, including normal and attack samples. Moreover, DL exhibits improved effectiveness in identifying zero-day attacks and complicated attack patterns through its ability to integrate an extensive array of training samples to construct a robust detection model.

2.7.1.1 Machine Learning in Cybersecurity

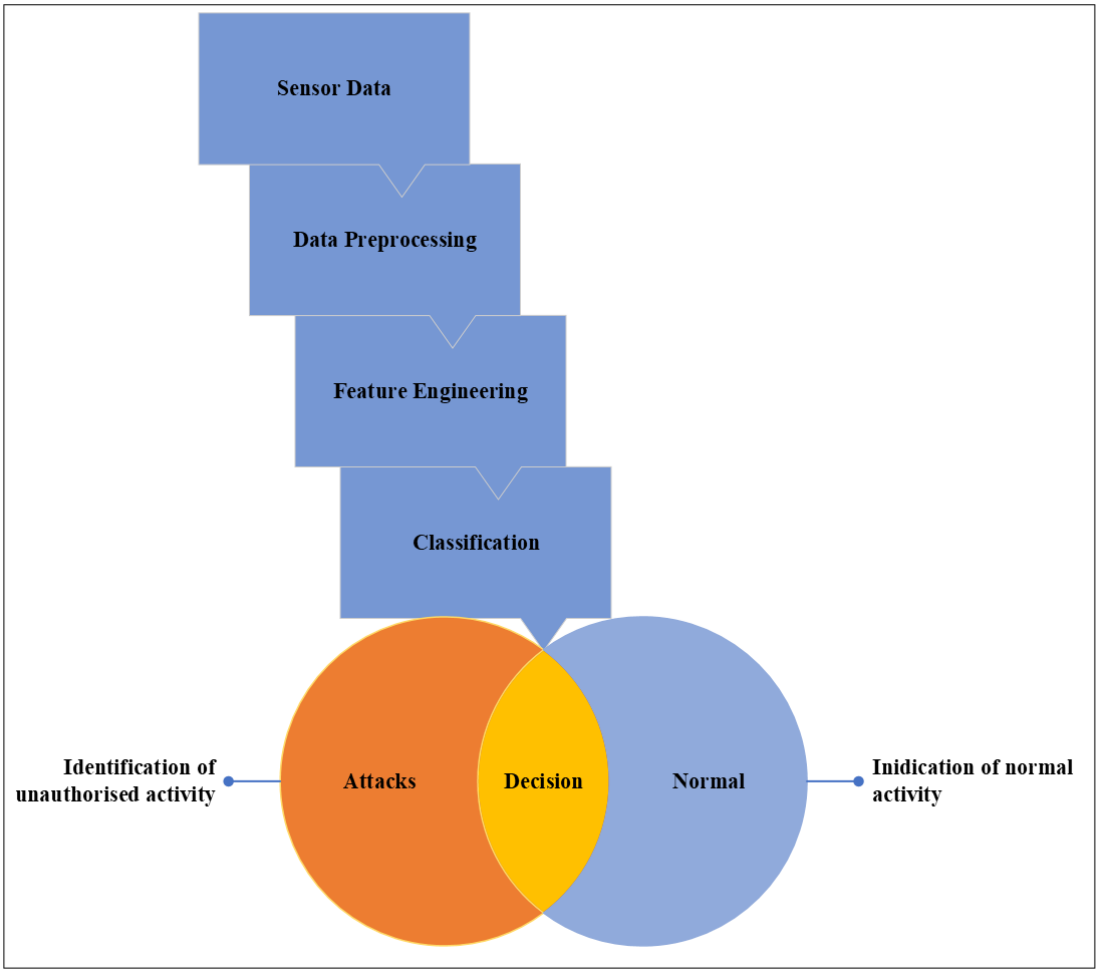
ML is an area of scholarly research that concentrates on the exploration of the theoretical foundations, characteristics, and effectiveness of models and systems designed to facilitate learning [604]. The field of ML is widely recognised for its strong interdisciplinary nature, as it draws upon a multitude of academic disciplines, encompassing but not limited to

information, cognitive science, mathematics, and optimisation theory. ML approaches allow researchers, engineers, and data scientists to examine specific issues without manual processes. Human interaction is unnecessary since robots perform the tasks [605]. ML has compelled cybersecurity researchers to reassess decision-making processes due to its expanded capabilities. ML techniques have been the subject of extensive scholarly investigation over a prolonged period. Significantly, the cybersecurity application accomplishments highlighted in current literature on the cybersecurity have been feasible through the utilisation of ML advancements, and these accomplishments have directly benefited end users' cybersecurity.

ML leverages time series or past data to draw outcomes from forecasts to new information on cybersecurity. ML can stand alone and find data patterns from autonomous devices to make judgements on normal digital activities or cyberattacks. A data collection component collects data to analyse for decision-making [606]. A dataset of autonomous devices can be raw or structural, wherein ML models find a group of tasks on a given dataset. These tasks include identifying unique characteristics, correlations, values, and groups and extracting functions. By categorising datasets for distinct purposes, ML makes it simpler for decision-makers to choose the right strategy. For instance, the *k*-means clustering algorithm can focus on information system discoveries [607] of normal digital activities or cyberattacks. Other than clustering, dimensionality reduction techniques, such as PCA, have been used for intrusion detection [608]. Figure 2.2 shows visualises how an ML model detects attacks.

Figure 2.2

The Standard View of a Machine Learning Attack Detection Process.



The standard ML model consists of preprocessing, feature engineering (e.g., feature selection, dimensionally reduction, or noise reduction), and classification [609]. ML techniques are divided into four forms: supervised, unsupervised [610], semi-supervised [611], and augmented learning techniques [612]. Many variables exist between supervised and unsupervised learning. example, unsupervised learning uses input variables (labelled data) to determine the natural structure of data. Supervised learning requires labelled training data for model building to draw the outcome. Semi-supervised learning combines supervised and unsupervised learning to tackle unsupervised classes. Finally, augmented learning improves the learning process by including interactive and multimedia elements, resulting in higher

levels of engagement. ML benefits of IoMT-enabled cybersecurity include security measures in real time, improved intrusion detection, and improved effectiveness.

2.7.1.2 Deep Learning in Cybersecurity

DL is a subclass of ML that involves multiple layers of non-linear processing to extract and transform discriminative or generative features for pattern analysis [599]. DL methods, also called hierarchical learning techniques, can capture intricate hierarchical representations within their deep architectures [613]. DL techniques have a mathematical framework (e.g., statistics, linear algebra, probability theory, and calculus) that incorporates multiple processing tiers, known as layers, to learn about data representations characterised by varying degrees of abstraction.

The operational structure of DL draws inspiration from the intricate workings of the human brain and its neural processes in signal processing. Deep neural networks are designed to facilitate the process of discrimination (supervised learning), wherein the model learns to make predictions based on labelled data. DL techniques are also divided into four forms: supervised DL [614], unsupervised DL [615], semi-supervised DL [616], and reinforcement DL techniques [617]. Convolutional Neural Networks (CNN) and RNN are examples of supervised DL. Autoencoders, deep belief networks, and restricted Boltzmann machines are used for unsupervised DL. In semi-supervised DL, Generative Adversarial Networks (GAN) and ensemble learning are commonly used. Finally, reinforcement DL consists of Q-learning and game theory.

2.7.1.3 Machine Learning and Deep Learning Models

The automated detection techniques rely on ML and DL models, the model preparation consists of feature engineering, where various ML and DL models can be trained based on specific problems [115]. The list of ML and DL models in cybersecurity as follows:

- The Decision Tree (DT) model is composed of branches that represent possible decisions. Trees consist of three primary nodes: decision, branch, and leaf. A decision node defines an attribute test. Each branch symbolises a potential attribute value. Ultimately, the leaf node signifies the classification of the attacks and malware, the DT model is commonly used in the cybersecurity domain [569].
- Random Forests (RF) model is highly effective in handling large datasets with high dimensionality for classification purposes. The RF model is commonly employed in categorical-numerical classification, rendering it highly valuable for virus and intrusion detection. The application of RF can be used in the ransomware related cyber threats, where categorical – numerical classification can be done in the classification task [570].
- The Support Vector Machine (SVM) model is highly proficient in binary classification tasks, particularly when the data exhibits linear separability or can be transformed into a higher-dimensional space. The SVM is commonly employed in the field of cybersecurity to classify malware and detect unauthorised access attempts [571].
- Collaborative models such as Extreme Gradient Boosting (XGBoost), Light Gradient-boosting Machine (LightGBM), and Categorical Boosting (CatBoost) employ Gradient-boosting Machines (GBM) to iteratively train weak learners and rectify any errors made by previous models. GBM is used for handling diverse data sets and has extensive applications in identifying phishing attempts, categorising malware, and detecting intrusions [580].
- The Logistic Regression (LR) model is a basic yet powerful method for binary classification. The LR model can be employed as a widely used cybersecurity categorisation model due to its interpretability and simplicity. The LR can be used to carry out security activities such as malware detection and anomaly detection [572].
- Naive Bayes (NB) model is a probabilistic model based on Bayes theorem, which assumes independence among features. Despite their apparent simplicity, The NB classifiers have

proven to be highly effective in tasks such as email spam filtering and website phishing detection [573, 574].

- The Multilayer Perceptron (MLP) model is an artificial neural network with input, hidden, and output layers with nodes (neurones). Every neurone in a layer connects to all neighbouring neurones. The MLP model can be applied the cybersecurity related classification tasks [575]. The MLP model learn from signatures and produce predictions by optimising neurone weights and biases using backpropagation and gradient descent algorithms during training.
- The Convolutional Neural Network (CNN) model is used in cybersecurity to classify malware images [576], and recognise Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) [618, 619]. The CNN model excels at capturing spatial relationships in data and extracting crucial visual attributes.
- The Recurrent Neural Network (RNN) model can be employed in the analysis of time-series data. This includes tasks such as data packet analysis and user behaviour modelling, as RNN models excel at analysing sequential data. RNN model variations such as Long-Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models can be employed to detect malware and intrusions because of their ability to capture long-term dependencies [577, 578].
- An autoencoder model is utilised for unsupervised feature learning in an unprocessed dataset and dimensionality reduction. By reconstructing standard data instances and identifying deviations from learned patterns to detect anomalies [579].
- The Generative Adversarial Network (GAN) can produce synthetic data that closely mimics real data distributions. GANs can generate adversarial examples, which can be used to evaluate the robustness of classification models against various attacks [572].

These models illustrate the usage in the cybersecurity domain typically these models classify cybersecurity-related tasks, aiming to detect attacks automatically. The selection of a model includes various factors, such as the features of the data and the complexity of the classification problem, are some examples. Table 2.10 presents the ML and DL models' usage, strengths and limitations of ML and DL models for different tasks in the field of cybersecurity.

Table 2.10

Machine Learning Models for the Cybersecurity.

Model	**Usage**	**Strength**	**Limitation**
DT	Attacks and malware classification	Numerical and categorical data	Low performance with complex trees
RF	Virus and intrusion classification	Optimal for complex trees	Computational costly
SVM	Attacks and malware classicisation	Optimal for high dimensional	Limited only when dealing with big dataset
XGBoost	Attacks and malware classification	Optimal for complex decisions	Hyperparameter tuning requires
LightGBM	Attacks classification	Optimal for big dataset	Hyperparameter tuning requires
CatBoost	Attacks classification	Optimal for categorical data	Computational costly
LR	Malware classification	Optimal for big data	Limited for linear decisions
NB	Phishing classification	Optimal with big dataset	Outperformer due to wrong assumptions may occurs

Model	Usage	Strength	Limitation
LSTM	Attacks classification	Optimal for natural language processing	Computational costly
CNN	Malware image classification	Optimal for malware images	Computational costly
GRU	Malware classification	Computational costly	Limited to capture long term dependencies
MLP	Attacks and malware classification	Optimal for non-linear data	Hyperparameter tuning requires
Autoencoder	Attacks and malware classification	Optimal when dealing with unprocessed dataset	Might limited for big dataset
GAN	Adversarial classification	Optimal for adversarial examples	Training complexity

Choosing an optimal model is challenging, some findings can be drawn, for example, DT model can perform optimally in the attacks' classification task, however, it may be limited for the complex tress. Despite this limitation, DT can still perform optimally in intrusion detection. Similarly, SVMs are successful in high-dimensional attacks classification. To draw the conclusion on the DT and SVM model, the experimental study requires to train same dataset rather than using different dataset for these two models, this can help in comparing the performance of both models. Moreover, XGBoost model can perform accurate in the classification task but require careful configuration in the model preparation phase, therefore, hyperparameters can be used to choose optimised configuration of the XGBoost. similarly, the MLP model works optimally for non-linear datasets but requires hyperparameter tuning.

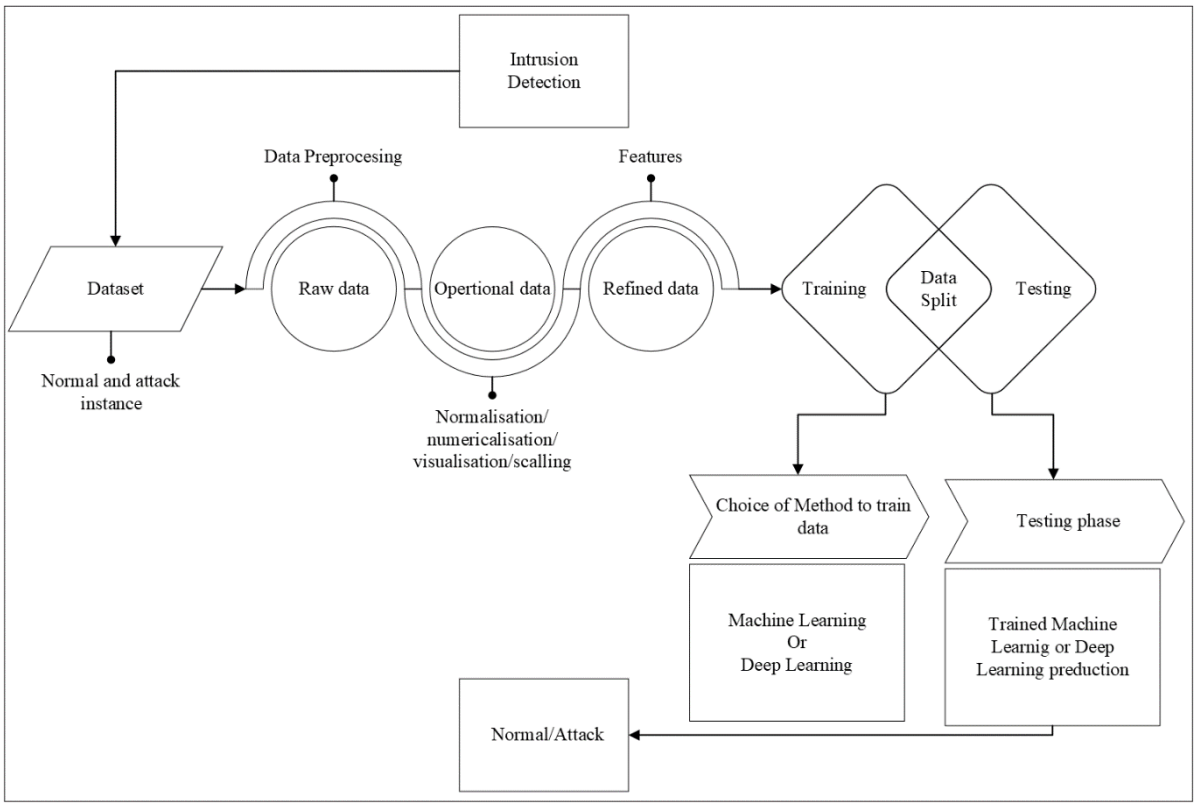
The RF model can be computationally complex, similarly, DL models, like CNNs, excel in picture classification because they can collect spatial characteristics, but they require a lot of computational power. RNN handle sequential input well but struggle with long-term dependence. LSTM networks solve this problem; however, they are complex and computationally intensive. In addition, some other models like NB are efficient, however, their feature independence assumption may be wrong, another model LR is interpretable yet ineffective. GAN model generates synthetic data; however, training instability causes problems. The GRU is computationally costly and may not capture long-term dependencies.

2.7.2 Intrusion Detection Systems

In the cybersecurity domain, intrusion detection systems are applied as either software applications or physical hardware components that determine and pinpoint instances of unauthorised access (i.e., intrusions). Intrusions encompass a sequence of interconnected malicious activities executed by an internal or external intruder to undermine the system's security [600, 620, 621]. The process of an intrusion detection system includes carefully examining data packets, system logs, and other forms of information to identify potential intrusions that pose a threat to the system. The intrusion detection system offers a standard set of functionalities to effectively uphold the integrity and protection of systems. Figure 2.3 illustrates the process of an intrusion detection system.

Figure 2.3

A Machine Learning Intrusion Detection System.



The standard ML intrusion detection system consists of preprocessing data for normalisation, converting words into numbers, or scaling. After preprocessing, data can be split into training and testing phases to detect intrusions. An intrusion detection system begins its operation by collating data derived from observed activities. The system collects data on occurrences and identifies patterns. The detecting component, which employs a wide range of methodologies and related techniques, is the core component of intrusion detection system [620]. Signature or anomaly detection techniques are commonly used by the intrusion detection system for security purposes. These two techniques can be used in conjunction as unified or separate components to primarily enhance detection accuracy [622].

A signature can be defined as a pre-established configuration that corresponds to a recognised instance of unauthorised intrusion. According to Ioulianou et al., the concept of signature detection entails the meticulous examination and comparison of signatures with

observed events to identify potential incidents [623]. Alternatively referred to as misused or knowledge-based detection, signature detection derives its efficacy from the utilisation of knowledge acquired from prior instances of unauthorised access and vulnerabilities. However, this method is not up to the task of spotting new patterns or variants on previous patterns. Furthermore, the task of maintaining current knowledge presents an additional challenge, as it entails an arduous endeavour [624].

An anomaly can be defined as a deviation from conventional patterns of behaviour. Anomaly detection can be defined as the systematic examination of occurred activities to identify significant deviations from these occurred activities [598]. According to Patcha and Park, anomaly detection comprises three steps. The first step is parameterisation, a process of encapsulating the observed behaviour within the device or profile comprising various features that require investigation. These features encompass a series of activities and application functionalities. The second step is training, which involves building a classification model that can distinguish between common and unusual behaviours by modifying the parameterised profiles. The third step is the detection of uncommon instances of activity [625].

2.7.2.1 Medical Device Intrusion Detection Systems

Each medical device's data possesses a high degree of sensitivity that makes the device vulnerable to various medjacking attacks for many reasons (see Sections 2.3 and 2.4). Protecting medical device data from potential threats requires implementing a comprehensive solution that guarantees the utmost security of these data throughout its lifecycle. Such secure solutions encompass data collection, transfer, storage, and processing stages. Safeguarding patients' security against unauthorised intrusion is imperative to prevent medjacking. Various methods may be used to protect medical devices. As explained in Section 2.6.1, ML can offer a potential solution for detecting unauthorised access.

A quadripolar electrode, an implementable pulse generator, and a controller that turns the medical device on or off is known as a deep brain implant. Some implementable pulse generators allow voltage adjustments. Using electrical charges, the DBI regulates Rest Tremor Velocity (RTV) to zero. Changes from zero in RTV cause movement abnormalities and chronic conditions such as Parkinson's disease. An attacker with DBI access may alter the RTV value, threatening the patient's life. To tackle this problem, Rathore et al. [626] proposed advisory attack detection in DBIs using LSTM to forecast RTV values at time *t* to identify deviations. The researchers used a dataset called Physionet and exemplified changing RTV values. However, the researchers concentrated on the RTV characteristic, although a cyber attacker may harm the patient using different features. The model validated attacks with low loss values and training times after several experiments.

The CardiWall dataset was proposed, and a simulation method used, involving pacemakers, implanted cardioverter defibrillators, and 28 malicious scripts targeting defibrillators. The researchers conducted a study on the use of unsupervised ML, namely SVMs, to develop CardiWall. In their study, the dataset consisted of 775 benign samples and the 28 malicious scripts. The analysis outcome was a true positive rate of 0.914, a false positive rate of 0.101, and an area under the curve of 0.947. In their statistical investigation, researchers stated that the radial basis function kernel approach outperforms the polynomial technique [627]. In the CardiWall dataset, the researchers stated that malicious programming was created by modifying clinically valid (benign) programming configurations with the deliberate intent to incorporate harmful configurations that could compromise patient safety. These malicious programs specifically changed Implantable Cardioverter Defibrillators (ICD) configuration parameters. The researchers demonstrated the compromised configurations could lead to life-threatening situations upon implementation. The researchers stated that a total of 28 malicious programmes were developed, each designed to exploit different aspects of device

vulnerabilities at the misconfiguration of ICD's parameters. These configurations were created to describe an array of possible threats, encompassing the misclassification of harmful arrhythmias, delay or absence of detection pertaining to life-threatening conditions, and the incorrect implementation of therapeutic interventions. Each configuration was created for implantable cardioverter-defibrillator programming and arrhythmia management [627].

In the CardiWall dataset, the researchers demonstrated the implications that modifications to the detection zones of ventricular arrhythmias can develop. In a normal configuration, the VT1 (ventricular tachycardia) zone was established to identify arrhythmias at a frequency of 150 beats per minute (bpm), accompanied by a detection counter calibrated to 10 beats. This enabled the identification of any ventricular arrhythmia exceeding 150 beats per minute and persisting for more than 10 beats as VT1. Nonetheless, malicious modifications to these parameters, including increasing the detection rate threshold or the counter value, could delay detection or result in misclassification, thereby permitting perilous arrhythmias to remain undetected. As the arrhythmia rises into the VT2 or VF (ventricular fibrillation) domains, generally identified at rates exceeding 200–220 bpm, subsequent modifications may either obscure detection altogether or result in the incorrect classification of the arrhythmias, culminating in an inability to administer suitable therapeutic interventions [627].

A dataset called HealthGuard was proposed by Newaz et al. [234] that comprised eight simulated medical devices, DoS attacks and fake data injection attacks. During the detection of attacks, different classifiers were proposed, including artificial neural networks, DT, random forest, and *k*-nearest neighbours [234]. Nevertheless, the researchers did not specify the dataset protocol, or other information regarding the implementation of their research.

Ahmad et al. [489] proposed DL detection using LSTM in their research, and an insulin pump was targeted for overdose attacks. A threshold was defined in their system to detect uncommon dosage activities. The patient was prompted to raise or tapped their index finger,

which a gesture sensor captured if the insulin dosage exceeded the threshold. The researchers trained and tested the LSTM and determined a threshold value using the insulin pump system log information from the last three months [489]. However, the researchers stated that they used a combination of the DL and gesture recognition framework to ensure the dosing procedure's accuracy and identify vulnerabilities within the insulin pump system. The threshold value was estimated by analysing a three-month log of the insulin pump system using the LSTM approach. The researchers stated that if the quantity of insulin exceeds the expected amount, the system communicates to the patient to execute the designated gesture. Next, if the outcome of the gesture deviates from the insulin value chosen for injection by the insulin pump, the device emits a warning to alert the patient, thereby enabling the individual to perform the necessary corrective measures. Their study did not provide details about how the model was trained and tested. In another study [628], Rathore et al. proposed a technique to identify false glucose readings using an MLP classifier. The researchers tested their approach on the UCI repository, specifically from the diabetes repository dataset and obtained 93.98% accuracy with their proposed method [628]. The authors illustrate that wireless connections between insulin pumps and glucose metres are especially susceptible to attacks, given their pivotal function in the precise management of glucose infusion. However, the specific dataset is not named in their study. The discussion on overdose attacks in insulin pumps directly relates to vulnerabilities in IoMT devices, an aspect central to this research on APS security.

Thamilarasu et al. [600] proposed a mobile agent-based intrusion detection using an OMNeT++ simulator; three types of mobile agents were designed in their proposed system. (1) sensor agent, functioned as an autonomous mobile programme tasked with identifying a particular class of attack. Each cluster head was tasked with the generation of numerous sensor agents for localised detection within a group of sensors. (2) The Cluster-head agent (CA) was an autonomous mobile programme designed to identify anomalies among cluster-heads within

various interconnected clusters of Wireless Body Area Networks. Finally, (3) the detective agents were responsible when the detection module of a sensor agent failed to characterise the network behaviour as either malicious or normal accurately; in such instances, it triggered an intervention request. The requisition comprised the agent and sensor identification numbers. Upon receipt of such a request, the Cluster Head (CH) generates a signature derived from the request attributes and subsequently stores it in a cache to mitigate the risk of responding to duplicate requests. In their proposed system, SVM, DT, NB, *k*-nearest neighbours, and Random Forest classifiers were used to identify intrusions. The researchers stated a cubic model of polynomial regression was applied to obtain balanced accuracy and avoid overfitting cases; therefore, the DT classification task achieved 99.9% accuracy [600]. However, in polynomial regression, a small number of exceptional data features might have a substantial influence on the nonlinear analysis results [629].

In [630], the researchers proposed a framework to detect DoS and user-to-root attacks in a remote healthcare system. Their proposed framework used three subsystems: master class, agents (sensors), and dataset (activities). The master class used a non-linear SVM anomaly detection strategy, and agents were used for a signature-based intrusion. Datasets were stored to process detection [630]. However, the researchers did not provide details about features and tools.

Schneble and Thamilarasu [631] proposed DoS, data injection, and fake data attack detection for medical devices using ML federated learning based on three components: cluster head, training model, and detection. In their system, the cluster head registered all devices, a training model was used for training the registered devices, and a calculation was made on registered devices to check the average weights and biases of the received models. The researchers computed an false positive rate ratio to evaluate model performance, and they achieved a 99.0 accuracy ratio [631]. The researchers stated that the MIMIC dataset from

PhysioNet comprises six distinct features encompassing 121 records. ML was implemented using the MLP from Sci-kit Learn and executed on Raspberry Pi systems. The researchers stated that attacks were simulated using new patient data. They stated that fifty per cent of the data samples were modified to simulate attacks, while the remaining fifty per cent remained unaltered. The researchers stated that they simulated the system using MATLAB, wherein the Raspberry Pi units were substituted with MATLAB objects representing each device. The MATLAB simulation used a singular-layer feed-forward neural network that functioned sequentially. In the simulation, the period allocated for each training round was determined by dividing the aggregate sequential time by the quantity of participating objects [631].

Alani et al. [632] proposed an ensemble approach for detecting intrusions using the WUSTL EHMS 2020 dataset. In their study, injection and scanning attacks were tested. The researchers claimed to have attained an accuracy rate of 99% [632]. The researchers provided the specifications, including processor: AMD Ryzen 5 3600 (4.2GHz), RAM 128GB, operating system Windows 10 Pro, Python version 3.10.0, Scikit learn version 1.1.3, Matplotlib version 3.5.3, and SHAP version 0.42.0. However, the researchers did not give specific information regarding feature balance and selection. The Scikit-learn library constitutes a vital tool in the field of ML model development and experimentation [633, 634]. Scikit-learn offers a wide range of well-suited ML algorithms for benchmarking and evaluation [635-637]. The ensemble methods provided by Scikit-Learn constitute an effective tool for enhancing model accuracy and generalisation through the strategic combination of multiple models [638-642]. In the context of Scikit-learn, the effectiveness of these techniques is frequently enhanced through the implementation of feature balance and feature selection [643-645]. The EHMS dataset [646], consisting of 32 features with 16 318 samples, was developed on biometric features encompassing electrocardiogram, blood oxygen saturation, temperature sensor, and blood

pressure biometric metrics, with the primary aim of detecting MITM, Spoofing, and Data alteration attacks.

This section details how researchers have studied different case scenarios and proposed solutions for mitigating medjacking using intrusion detection systems. Table 2.10 presents a summary of ML intrusion detection methods applied to several medical devices. These studies sought to emphasise the capacity of ML to identify instances of medjacking in implanted devices. Certain research lacked extensive information about the selection of features and the potential use of dimensionality reduction methods.

Table 2.11

The Current Machine Learning Intrusion Detection Approaches for Medical Devices.

Reference	**Data Sources / Configuration**	**Tools**	**Attacks**	**IDS**	**Feature extraction**	**Dimensionality reduction**	**Machine Learning**	**Limitations**
[627]	Pacemaker		28 malicious scripts	Signature detection	wrapper		SVM	Lack of implementation details
[234]	Medical devices		DoS and fake data injection attacks	Anomaly detection	Wrapper		Artificial neural networks, DT, radio frequency and *k*-nearest neighbours	Lack of implementation details
[489]	Insulin pump		overdose attacks	Signature detection	Information Gain		LSTM	Gesture approach may not be feasible for all patients, and there was a lack of implementation details
[626]	DBI	Deep brain stimulators	Advisory	Signature detection	Information Gain	LSTM based 3D	LSTM	Unsatisfactory model loses value

Reference	Data Sources / Configuration	Tools	Attacks	IDS	Feature extraction	Dimensionality reduction	Machine Learning	Limitations
[628]	CGM sensor		Fake glucose reading	Anomaly detection	Information Gain		MLP	Lack of features and information
[600]	Medical devices	OMNeT++	DoS, data injection	Signature detection	Standard deviation	PCA	SVM, DT, NB, *k*-nearest neighbours, and radio frequency	Polynomial regression applied to increase the accuracy
[630]	Medical devices		DoS and user-to-root attacks	Signature detection			SVM	Lack of implementation and analysis details
[631]	Medical devices		DoS, data injection, and fake data	Signature detection	Wrapper cluster based		Federated learning	Lack of implementation and device details
[632]	Medical devices	Sci-Kit Learn library	Injection and scanning	Anomaly detection	Random oversampling		Ensemble	Lack of feature balance and selection details

2.7.2.2 IoT Intrusion Detection Systems

In [647], researchers proposed a hybrid ML technique termed XGB-RF to identify attacks utilising the N-BaIoT dataset. Random Forest Recursive Feature Elimination (RFRFE) based features were selected, and an XGBoost classifier was trained to identify botnet attacks. The N-BaIoT dataset comprised 115 features; the researcher used the RFRFE feature selection technique, resulting in a refined set of 40 features. The researcher tuned the parameters using a random forest hyperparameter optimisation technique. The effectiveness of the proposed XGB-RF technique was assessed, yielding an accuracy of 99.94% [647]. Nonetheless, the proposed system did not use dimensionality reduction techniques; the researchers reduced the feature set from 115 to 40 through the feature selection RFRFE technique. In [648], the researcher proposed a technique using the N-BaIoT dataset; in their proposed technique, various classifiers, including DT, RF, Bagging Meta Classifier (BMC), Adaptive Boosting (ADB), Gradient Descent Boosting (GDB), and XGBoost, were trained to detect intrusions. Their findings indicate that DT achieved an accuracy of 99.99, BMC attained 99.99, RF reached 99.99, ADB recorded 99.99, GDB 99.98, and XGBoost 99.99 [648]. Nevertheless, the model's findings indicate that there might be an occurrence of overfitting, as each model achieved an accuracy of 99.99.

In [575], the researchers proposed a detection system using the N-BaIoT dataset. The researcher trained the NB, DT, RF, SVM, LR, Single-Layer Perceptron (SLP), CNN, and MLP classifiers to detect intrusions. The researchers used dimensionality reduction techniques, PCA and Linear Discriminant Analysis (LDA). The accuracy of the models derived without the application of dimensionality reduction is as follows: NB 87.0, DT 90.0, RF 93.0, SVM 90.0, LR 90.0, SLP 91.0, CNN 92.0, and MLP 93.0. The results with PCA, NB 67.0, DT 79.0, RF 99.0, SVM 90.0, LR 90.0, SLP 92.0, CNN 93.0, and MLP 94.0. The result with LDA, NB 84.0, DT 87.0, RF 87.0, SVM 72.0, LR 73.0, SLP 85.0, CNN 78.0, and MLP 75.0 [575].

Nevertheless, the feature selection technique was not employed prior to the process of dimensionality reduction, and the hyperparameters were not optimised for each respective model.

In [649], the researchers proposed a technique to detect botnet intrusions, trained NB, RF, Gradient Boosting (GB), and DT using the BoT-IoT dataset. Their investigation calculated the accuracy of RF 98.0, NB 70.0, DT 91.0, and GB 99.0 [649]. However, techniques related to dimensionality reduction, encompassing hyperparameters, were evaluated. In [650], the researchers have developed an intrusion detection system utilising the BoT-IoT dataset to identify botnet attacks. The researchers trained the SVM model to detect botnet attacks, utilising feature selection techniques such as Chi-Square and ExtraTree, and the PCA dimensionality reduction technique. The results indicate that the SVM, without feature selection and dimensionality reduction, achieved an accuracy of 88.37%. In contrast, the SVM employing the Chi-Square method attained an accuracy of 98.66%. Furthermore, the SVM utilising the Extra Trees algorithm recorded an accuracy of 99.29%, while the SVM with Principal Component Analysis (PCA) yielded an accuracy of 62.52%. Notably, the SVM with Factor Analysis (FA) achieved an accuracy of 99.38% [650]. Nevertheless, details of PCA components and FA were not provided; moreover, hyperparameters were not included to optimise the SVM model.

In [651], the researchers proposed a hybrid metaheuristic approach incorporating RNN for intrusion detection, utilising the CICIDS2017 and BoT-IoT datasets. Their proposed system was designated as the Weighted Majority Voting Ensemble Deep Learning (MM-WMVEDL). The Harris Hawk optimisation-based elite fractional derivative mutation (HHO-EFDM) technique was used for the feature selection. In the context of the IoT-23 dataset, models including MM-WMVEDL, CNN, LSTM, autoencoder, and DNN were trained. In relation to the CICIDS2017 dataset, MM-WMVEDL, DeepLog, LSTM-Autoencoder, Bi-LSTM, GRU-

D, HiLSTM, GAN, RNN-AD, and VAE models were trained. Utilising the IoT-23 dataset, MM-WMVEDL achieved an accuracy of 98.21%, while CNN attained 96.84%, LSTM reached 95.62%, autoencoder secured 95.38%, and DNN recorded an accuracy of 93.47%. The outcomes derived from the CICIDS2017 dataset show the following performance metrics: MM-WMVEDL at 99.98, DeepLog at 95.58, LSTM-Autoencoder at 96.75, Bi-LSTM at 97.31, GRU-D at 98.02, HiLSTM at 96.37, GAN at 97.56, RNN-AD at 92.43, and VAE at 91.33 [651]. However, evaluations pertaining to hyperparameter optimisation and dimensionality reduction were not conducted. Furthermore, as indicated in [652], the CICIDS2017 dataset lacks any indications of IoT activity and is considered outdated. In [653], the researchers indicated that the CICIDS2017 dataset is vulnerable to significant class imbalance, adversely affecting detection accuracy and increasing false alarms [653]. In [654], the researchers performed training and evaluation of 1D convolutional neural networks (utilising the ReLU activation function), LSTM, and hybrid CNN+LSTM models to detect intrusions, employing the CICIDS2017 dataset as their primary resource. The examination of the proposed system yields the following results: 1dCNN 95.14, MLP 86.34, LSTM 96.24, and CNN+LSTM 97.16 [654]. However, feature selection, dimensionality reduction, and hyperparameter techniques were not examined.

This section elucidates the techniques used by researchers in examining various forms of attacks and the subsequent proposals for mitigating such threats through the implementation of intrusion detection systems. Table 2.12 outlines an overview of intrusion detection techniques implemented across various IoT devices. Specific studies exhibit specific limitations regarding the selection of features and the prospective application of dimensionality reduction techniques, including hyperparameters.

Table 2.12

Comparison of The Current Machine Learning Intrusion Detection Approaches for IoT Devices

Reference	**Data Sources / Configuration**	**Feature selection**	**Dimensionality reduction**	**Hyperparameter**	**Model (s)**	**Accuracy**	**Limitations**
[647]	N-BaIoT [655]	RFRFE		Random forest	XGBoost	99.94	Dimensionality reduction techniques were not tested
[648]	N-BaIoT [655]				DT, RF, BMC, ADB, GDB, and XGBoost	DT 99.99, BMC 99.99, RF 99.99, ADB 99.99, GDB 99.98, and XGBoost 99.99.	There might be overfitting because each model obtained 99.99 accuracy.
[654]	CICIDS2017 [652]				1d-CNN, LSTM, CNN+LSTM	1dCNN 95.14, MLP 86.34, LSTM 96.24, and CNN+LSTM 97.16.	The feature selection, dimensionality reduction, and hyperparameters techniques were not examined
[575]	N-BaIoT [655]		PCA and LDA		NB, DT, RF, SVM, LR, SLP, CNN, and MLP	Without dimensionality reduction: NB 87.0, DT 90.0, RF 93.0, SVM 90.0, LR 90.0, SLP 91.0,	The feature selection technique was not used before dimensionality reduction, and hyperparameters

Reference	Data Sources / Configuration	Feature selection	Dimensionality reduction	Hyperparameter	Model (s)	Accuracy	Limitations
						CNN 92.0, and MLP 93.0. With PCA, NB 67.0, DT 79.0, RF 99.0, SVM 90.0, LR 90.0, SLP 92.0, CNN 93.0, and MLP 94.0. With LDA, NB 84.0, DT 87.0, RF 87.0, SVM 72.0, LR 73.0, SLP 85.0, CNN 78.0, and MLP 75.0	were not tuned for each model
[649]	BoT-IoT [568]					RF 98.0, NB, 70.0, DT 91.0, and GB 99.0 accuracy	Dimensionality reduction techniques, including hyperparameters, were not tested.
[650]	BoT-IoT [568]	Chi-Square and Extra Tree	PCA, and FA		SVM	SVM without feature selection and without dimensionality obtained 88.37 accuracy, SVM	Details of PCA components and FA is not provided, moreover, hyperparameters also not included

Reference	Data Sources / Configuration	Feature selection	Dimensionality reduction	Hyperparameter	Model (s)	Accuracy	Limitations
						with Chi-Square 98.66 accuracy, SVM with extra tree obtained 99.29 accuracy, SVM with PCA obtained 62.52 accuracy, and SVM with FA obtained 99.38 accuracy.	to optimise the SVM model
[651]	BoT-IoT [568], and CICIDS2017 [652]	HHO-EFDM			For IoT-23 dataset, MM-WMVEDL, CNN, LSTM, autoencoder, and DNN models were trained. For CICIDS2017 dataset, MM-WMVEDL, DeepLog, LSTM-Autoencoder, Bi-LSTM, GRU-D, HiLSTM, GAN, RNN-AD, and VAE, models were trained	With the IoT-23 dataset, MM-WMVEDL 98.21, CNN 96.84, LSTM 95.62, autoencoder 95.38, and DNN 93.47. With the CICIDS2017 dataset, MM-WMVEDL 99.98, DeepLog 95.58, LSTM-Autoencoder 96.75, Bi-LSTM 97.31, GRU-D 98.02, HiLSTM 96.37, GAN 97.56,	Hyperparameter and dimensionality reduction evaluations were not performed

Reference	Data Sources / Configuration	Feature selection	Dimensionality reduction	Hyperparameter	Model (s)	Accuracy	Limitations
						RNN-AD 92.43, and VAE 91.33.	

2.8 Related Datasets Limitations

The emergence of the IoT and the IoMT has significantly revolutionised sectors, including healthcare and manufacturing. These technologies facilitate the instantaneous acquisition of data, information transmission, and processes regulation across diverse settings. Nevertheless, with the expansion of IoT and IoMT ecosystems, safeguarding their security has emerged as a paramount concern, especially through the advancement of IDS. In the field of IDS, the availability of datasets is most important for the effective training and validation of security solutions. However, many datasets presently accessible within this domain are outdated, inadequate, or otherwise inadequate for tackling the evolving challenges associated with IoT and IoMT security.

In a study [656], the researchers proposed an IDS, and highlighted the opportunities and solutions in the field of IoT IDS and concluded that many proposed models in the literature were found to be affected by outdated or imbalanced datasets. In another study [652], the researchers comprehensively reviewed the publicly available datasets. The researchers stated that the datasets employed for analysing network packets in commercial applications are not readily accessible due to privacy concerns. Nonetheless, several publicly accessible datasets, including DARPA, KDD, NSL-KDD, and ADFA-LD, are extensively utilised as benchmarks in the field. The researchers compared the existing datasets and highlighted their limitations. The major limitation was found in the comparison that the following datasets, DARPA 98, KDDCUP 99, CAIDA, NSL-KDD, ISCX 2012, ADFA-WD, ADFA-LD, and CICIDS2017, lack any traces of IoT activity [652].

Furthermore, in [657], the researchers stated that many recent studies have been executed utilising outdated datasets that were not carefully gathered within the context of the IoT, or they employed datasets that are restricted to network and IoT traffic characteristics. The endeavour to identify an accurate and accessible dataset incorporating diverse traffic categories,

including network, IoT, and industrial IoT, alongside an array of complex attacks for multi-classification, coupled with the necessity for a considerable volume of real-world observations, required us to engage in thorough research.

A systematic literature review on unravelling IDS datasets for enhanced cybersecurity understanding was conducted in [658]. The researchers stated that the datasets pertaining to IDS offer a valuable resource for the research community to critically assess and evaluate their methodologies and models related to detection intrusion. Nevertheless, the assessment of the proposed detection methodologies utilising outdated datasets may fail to accurately reflect the efficacy of these methodologies in identifying modern attack vectors, potentially resulting in incorrect findings.

Datasets constitute the core of IDS research, offering the requisite data to train ML models in identifying anomalous behaviour. Within the framework of the IoT and the IoMT, it is imperative that these datasets accurately encapsulate data, device functionality, and potential attack scenarios to ensure their efficacy. Nevertheless, many extant datasets have been impacted by considerable constraints. A wide range of datasets currently in use, including the KDD Cup 1999 and NSL-KDD, were created for conventional network architectures and have since become outdated, as they inadequately represent contemporary IoT devices, protocols, and attack vectors.

The research community highlighted that some available datasets have limitations. They are either outdated or inadequately represent the comprehensive diversity and magnitude of contemporary IoT and/or IoMT ecosystems. Furthermore, datasets about IoMT must confront distinct challenges associated with privacy, ethical considerations, and regulatory adherence, aspects that are currently inadequately represented.

2.8.1 Related Datasets Comparison

Each dataset fulfils distinct research and study objectives, device types, attack scenarios and features influenced by the cybersecurity risks, environments, data types, and devices it encompasses. The heterogeneity inherent in datasets illustrates the wide range of IoT security challenges, encompassing threats such as malware attacks, mobile attacks, and detecting anomalies. This section analysis discusses the specific study objectives, device types, attack scenarios and features related to each dataset, demonstrating their design, distinctive applications, and current limitations (See Table 2.13).

Table 2.13

Comparison of datasets based on objectives, device types, attack scenarios, features, Dataset instances, publication year, domain types, and limitations.

Reference	Year	Dataset name	Study Objective	Device Type	Domain	Attack type	Feature numbers	Dataset instances	Limitations
[659]	2017	Honeypot	Identified WSN attacks		WSN	Botnet, dictionary, bruteforce, recon scripts, launch attack, individual attacks			Outdated: No IoT protocol like MQTT was used.
[660]	2018	Botnet	Dataset was generated containing botnet malware	Profiles: multimedia centre, surveillance camera with additional traffic, surveillance camera	Home	Botnet	CPU and memory usage features		Outdated: No IoT protocol like MQTT was used.
[655]	2018	N-BaIoT	Dataset was generated to develop Botnet operation, including	Simulated nine IoT devices.		Mirai and BASHLITE	115 features	7,068,606	This dataset mainly focused on botnet operations.

Reference	Year	Dataset name	Study Objective	Device Type	Domain	Attack type	Feature numbers	Dataset instances	Limitations
			Mirai and BASHLITE						
[568]	2019	BoT-IoT	Dataset was generated to develop botnet features and detect Botnet	smart fridge, smart garage door, weather station, smart lights, and smart thermostat.	Home	Botnet	32 features	73360900	Botnet attacks were simulated; however, other cybersecurity attacks required different datasets.
[646]	2020	EHMS	Dataset was generated to develop biometric data	Electrocardiogram, Blood Oxygen Saturation, temperature sensor, blood pressure	Health care	MITM, Spoofing and Data alteration attacks	32 features	16 318	The study focused on biometrics Information and attacks related to biometrics information. This dataset mainly focused on biometrics.
[661]	2020	IoT-23	Dataset was generated to develop botnet features and identify botnets	Philips Hue smart lamp, an Amazon Echo, and a Somfy	Home	Malware, including Mirai, Torii, gagfyt, Kenjiro, Okiru, Hakai, IRCBoT, Hajime,	19 features	280,772,878	The IoT-23 was extensively engineered to detect botnet activities, focusing on Mirai and

Reference	Year	Dataset name	Study Objective	Device Type	Domain	Attack type	Feature numbers	Dataset instances	Limitations
				smart door lock.		Muhstik, and hide and seek Botnet detections			comparable IoT malware variants.
[662]	2020	IoT-DDoS	Simulated distributed weather stations to develop weather features	Profiles: Humidity and Temperature data	Industry	Selective forwarding, Blackhole, and flooding attacks	12 features		The nature of distributed weather stations is more suitable for industries.
[663]	2020	DAD	Dataset was generated to develop cold aisle data center features and collect features	Temperature sensors	Industry	Duplication, Interception, and Modification	10 features	101,583	This dataset is more suitable for large data centers with examples of data duplication, interruption, and modification at the data centre level.
[664]	2022	ECU-IoFT	Dataset was generated to develop drone features and identify drone wifi related attacks	Drone	Home and small businesses	Tello API exploit, WPA2-PSK Wi-Fi cracking, and Wi-Fi deauthentication	10 Features	54,492	The study focussed on wi-fi related attacks.

Reference	Year	Dataset name	Study Objective	Device Type	Domain	Attack type	Feature numbers	Dataset instances	Limitations
[665]	2024	WID	Dataset was generated to develop Wi-Fi related features to identify wi-fi related intrusions	IoT device	Industry	Re)Assoc, Disas, Deauth, Rogue AP, SSh, Kr00k, Krack, Botnet, SSDP, SQL Injection, Malware, Website spoofing, and Evil Twin	254 features	6,526,404	The study focused on wi-fi related attacks

Table 2.13 outlines the comparative analysis of datasets, encompassing the year of publication, objective, device category, domain labelling, attack type, features, dataset instances, and limitations. The extant literature and presented datasets unequivocally illustrate that each dataset was designed with a particular research objective, addressing diverse device categories, features, attack techniques, and environments. Current literature indicates that no singular dataset adequately addresses all cybersecurity concerns associated with the IoMT. This constraint provides a valuable opportunity for creating a comprehensive dataset that addresses all cybersecurity issues in every domain.

This research investigates the detection of medjacking within the context of the APS. In current literature, the diversity of datasets, encompassing their scope and scale, depends on the unique features of the issues they address, the domains from which they originate, and the objectives of their investigation [666-668]. For instance, Table 2.5 depicts the comparative examination of datasets, encompassing variations within the datasets, their objectives, device classifications, domain categories, attack type, the number of features, and the count of instances that are distinctive. Literature indicates that variations are fundamental, as datasets are rigorously constructed in accordance with specific objectives, contextual environments, and resources throughout their research phase, utilising available sources and tools [669-671]. Each dataset has been designed to tackle a specific problem, with its structure and scale dependent upon the unique requirements related to that problem or objective. For instance, Table 2.5 illustrates the comparative analysis of various datasets. The datasets in question lack features and samples that are specific to APS, particularly in relation to medjacking attacks. The primary objective of this thesis is to develop and examine a proposed hybrid intrusion detection system for APSs to effectively identify medjacking attacks, with particular emphasis on minimising brute force, DDoS, MITM, and Trojan attacks. The objective of this study is to perform an exploratory data analysis aimed at explaining data patterns, thereby offering

insights into various attack techniques and normal behaviours of the APS context. The APS dataset is essential for comprehending the phenomenon of medjacking within the context of APS.

According to current literature, each dataset was designed to meet specific investigation requirements, and although this specificity is beneficial for focused investigations, it engenders limitations when dealing with broader challenges in IoMT security. For example, the BoT-IoT dataset [568] primarily concentrated on detecting botnet attacks; however, the growing importance of other cybersecurity threats requires distinct datasets. The EHMS dataset [646] concentrated on a biometric dataset encompassing electrocardiogram, blood oxygen saturation, temperature sensor, and blood pressure biometric metrics, with the primary aim of detecting MITM, Spoofing, and Data alteration attacks. Another dataset, ECU-IoFT [664], specified drone-related Wi-Fi attacks intending to detect such intrusions within the drone systems. Similarly, the WID dataset associated with Wi-Fi-related attacks demonstrated various attacks, including Re Assoc, Disas, Deauth, Rogue AP, SSh, Kr00k, Krack, Botnet, SSDP, SQL Injection, Malware, Website spoofing, and Evil Twin attacks within the context of the IoT industry field.

The dataset diversity illustrates how the wide range of research objectives limits the datasets' capacity to generalise across various IoMT contexts. The categories of devices encompassed within these datasets further exemplify this diversity. The wide range of IoMT devices is remarkably diverse, encompassing a variety of applications from complex medical gadgets such as APS to industrial sensors and consumer-oriented devices, including cameras. No existing dataset currently amalgamates data from such a diverse array of devices, thereby leaving substantial gaps in research endeavours aimed at generalising findings across various IoT categories. The emphasis of each dataset on different attacks highlights the limitation of identifying a singular dataset that can comprehensively tackle the entirety of IoMT security

threats. Furthermore, these datasets are generated within diverse environments, such as smart homes, healthcare IoT, industrial IoT, or general-purpose testbeds, thereby constraining their applicability across various domains.

According to dataset limitations, in [658], the researchers explained that the datasets associated with detection systems present an important opportunity for the academic community to rigorously examine and evaluate their methodologies and models concerning the identification of intrusions. Nonetheless, the evaluation of the proposed detection methodologies employing antiquated datasets may poorly represent the effectiveness of these methodologies in determining attacks, thereby potentially leading to incorrect outcomes. A diverse array of datasets presently employed, such as the KDD Cup 1999 and NSL-KDD, were originally developed for traditional network architectures and have subsequently become outdated, as they fail to accurately encapsulate the specifics of modern IoMT devices, protocols, and attacks.

2.9. Research Gap Artificial Pancreas System

In the IoMT scenario, smartwatches and fitness trackers are essential for monitoring heart rate, oxygen saturation, and glucose levels [11, 62, 63, 67]. Insulin pumps and CGMs are essential for managing chronic illnesses like diabetes [68-73]. The various implantable devices, including pacemakers, insulin pumps, and neurostimulators, constantly track and regulate medical conditions such as cardiac arrhythmias, diabetes, and neurological disorders [62, 74-77]. Implantable medical devices are physically or surgically inserted into the body, and must remain inserted [403-405]. Medical devices can assist with cardiac arrhythmia [75], back pain, diabetes, epilepsy, Parkinson's, and spinal cord diseases. Patients with renal diseases can use artificial kidneys [403] patients with diabetes use APSs [308].

However, these advancements result in security risks and vulnerabilities compromising patients' lives, potentially leading to medical complications. The IoMT systems and devices

are vulnerable to a variety of security challenges, including medjacking [80-82], unauthorised access [83], data breaches, and manipulation of device functionality [84, 85]. The emergence of interconnected medical devices, including insulin pumps [81, 82, 86-89], CGM [81, 90-92], pacemakers [93-98], and infusion pumps [99], has simultaneously targeted for hijacking and various cyber-attacks as documented in the literature. In a recent investigation [437], the researchers executed simulations involving eight distinct malware variants, which include Fuzz20 and Fuzz100, File Manipulation, Information Leakage, as well as synthetic malware 1, synthetic malware 2, synthetic malware 3, and synthetic malware 4, targeting smart connected insulin pumps [437].

Many IoMT devices, such as pacemakers and insulin pumps, perform essential life-preserving functions. These devices may be exploited upon compromise, compromising patient safety [1-4]. In addition to safety considerations, a considerable concern relates to the possible depletion of device functionality. For instance, as explained in [183], physical IoMT devices exhibit tampering vulnerability, enabling adversaries to modify operational parameters or disable them entirely or partially. Attackers may exploit specific firmware vulnerabilities, thereby facilitating the deployment of ransomware within the IoMT device and taking control of it [183]. Moreover, according to [184], attacks on implantable and wearable devices may be intentionally designed to deplete the battery life of these devices, consequently presenting a significant risk to patients who rely on them for monitoring or treatment. The phenomenon of battery depletion is complexly linked to diverse manifestations of cyberattacks, especially those instigated by malware breaching a device, which in turn results in excessive consumption of energy [184].

In 2024, Tandem Diabetes initiated a recall for their t:connect mobile application specifically developed for Apple iOS devices. The recall is related to a software malfunction that precipitated the application’s failure, leading to its incessant crashing and subsequent

relaunching in a continuous loop. This behaviour resulted in an accelerated and significant depletion of the insulin pump's battery, thereby preventing the risk of an unforeseen pump shutdown. Interruptions in insulin delivery may precipitate hyperglycemia or diabetic ketoacidosis, both of which constitute serious medical conditions [161].

From a theoretical perspective, the cessation of pump operation will precipitate an interruption in the delivery of insulin, which may culminate in an insufficient administration of this critical hormone. This situation has the potential to precipitate hyperglycemia or, in more severe cases, diabetic ketoacidosis, a condition that presents considerable life-threatening risks owing to elevated blood glucose levels and inadequate insulin supply. In a tangible context, as a consequence of battery depletion, in [162], it has been reported that more than 200 individuals sustained injuries attributable to a technological failure that precipitated an unforeseen cessation of their insulin pumps. The software anomaly necessitated the recall of more than 85,000 iterations of a mobile application, identified as t:connect and developed by Tandem Diabetes Care. The malfunction of this software has elucidated the ramifications of this failure, revealing that more than 200 individuals afflicted with diabetes sustained injuries. Cyber-attacks have an increased impact on the operational and financial aspects of infrastructure, alongside the ethical and social implications. According to [186], the implications of insufficient security in Internet of IoMT healthcare systems encompass the violation of patient confidentiality resulting from eavesdropping, alongside the complicated recognition of critical incidents due to the disruption of standard operations of IoMT devices caused by DoS attacks [186].

The vulnerabilities associated with medjacking within the IoMT are paramount, considering their capacity to affect patient safety profoundly [146]. In traditional healthcare cybersecurity, breaches primarily encompass data theft, during which patient records or financial information are compromised [174]. However, cyberattacks linked to the IoMT may

transcend the confines of simple data expropriation, potentially disrupting or modifying the operational integrity of medical devices, thus presenting significant consequences [103]. A fundamental distinction between hacking within the IoMT and other cybersecurity threats prevalent in the healthcare sector resides in the direct connection to patients. For example, medjacking attacks on insulin pumps [81, 82, 86-89] or pacemakers [93-98] may permit adversaries to alter device functionality, resulting in the delivery of incorrect dosages in relation to a device's operational functionalities.

Ensuring the greatest possible degree of cyber security for APS devices is essential since these devices perform a vital role in the management of diabetes and are susceptible to medjacking. Intrusion detection systems are essential to protecting these devices. In current literature, researchers have primarily concentrated on different intrusion detection system techniques designed specifically for IoMT devices. Nevertheless, there are still significant gaps in the existing research. A major obstacle is the absence of extensive datasets tailored to APS devices. Due to their proprietary nature, these devices pose a challenge for researchers seeking to obtain the dataset and essential information required for cybersecurity analysis.

In current literature, numerous datasets are impacted by a lack of variety in attack types, specifically medjacking attacks. The lack of appropriate datasets poses a challenge to the advancement of efficient IDS. For example, to train ML models for the development of effective IDS, therefore the appropriate dataset is crucial to have access to develop detection techniques. The limited availability of actual data poses a significant challenge for researchers to effectively train and validate these systems.

In the model development, dimensionality reduction techniques, such as PCA, t-SNE, and UMAP, are employed to preprocess data and reduce the feature space, thereby simplifying it for classification models. Multiple classification models, including MLP, LSTM, SVM, DT, and CNN, are utilised to detect intrusions. However, there is a lack of research that specifically

investigates the effectiveness of different dimensionality reduction strategies, and comparative analysis with and without dimensionality reduction techniques. A research gap to examine the impact of with and without dimensionality reduction on the performance of intrusion detection systems can improve the development of detection systems. Furthermore, there is a lack of study in determining the most effective hyperparameters for classification models.

2.10 Summary

The literature review in this chapter demonstrated the multifaceted dimensions encompassing APS, medjacking detection, and defensive methods. A wide range of APS devices are now accessible in the market for managing diabetes. These devices also provide ubiquitous strategies for monitoring health conditions. In general, wearable devices have emerged as a prominent fashion trend in contemporary society. Medical devices are equipped with a variety of sensors that gather data from the user's body to perform assigned tasks. The utilisation of wearable devices in the medical domain has been notably expanding owing to the instantaneous feedback provided to patients and medical professionals.

Existing wearable devices have demonstrated significant vulnerabilities in terms of security measures, particularly in the areas of hijacking devices. To prevent APS devices from being medjacked, it is necessary to implement efficient preventative and defensive mechanisms. ML applications can identify medjacking, which may help protect APS devices and ensure precise doses for patients to manage their chronic conditions.

Various cybersecurity applications exist for securing medical devices, but they primarily rely on basic passwords or keys for access. APS devices are therefore susceptible to medjacking, and the implementation of AI applications potentially provides significant benefits. The literature presented in this chapter has demonstrated the successful use of AI applications in cybersecurity, particularly for the identification of various cyberattacks. Medjacking

detection provides the best chance of safeguarding patients and producing APS devices. the following chapter will examine techniques for implementing cybersecurity analysis in the APS.

Chapter 3. Methodology

This chapter presents the research methodology utilised to investigate the APS medjacking and develop effective proactive mitigation solution. At the outset, the research began with a comprehensive examination of current literature on the APS security vulnerabilities and attack methods to synthesise the medjacking problem, and proactive solutions to examine the limitations of existing solutions. An APS prototype was developed using a simulation approach, and security attacks were conducted utilising a modular design to execute the medjacking security attacks. Subsequently, the dataset, which encompasses normal and attack activities, was gathered to conduct exploratory data analysis. Ultimately, the proactive solution to mitigate intrusion was developed through pilot implementations. The security of APS presents a complex challenge that requires continuous improvement and constant monitoring. This chapter addresses the methodology to address the research questions:

1. What vulnerabilities and attack methods exist within the Bluetooth and Internet of Things communication protocols in the APS?
2. What is the outcome on an APS in a simulated attack environment?

   a. How can an appropriate dataset be developed to effectively study medjacking vulnerabilities in APS?

   b. What exploratory data analysis technique can be used to explore data patterns of medjacking methods, including brute force, DDoS, MITM, and Trojan attacks?
3. What machine learning and deep learning techniques as proactive defences techniques can be used to detect medjacking vulnerabilities in an APS?
4. What dimensionality reduction technique can be performed optimally in the development of a machine learning model for medjacking detection in the APS?

a. What differences can be experienced with and without dimensionality reduction techniques in the development of a machine learning model for medjacking detection in the APS?

3.1 Overview

The research methodology in this study initiates with examining the existing review. This phase examines the APS's security vulnerabilities, methods of attack, and proactive measures to prevent them. This research developed significant research questions and objectives. These questions serve as a guide for the study, directing this research's focus towards addressing the identified gaps and developing proactive solutions to mitigate them.

An APS prototype was developed to investigate proactive mitigation of medjacking threats. A cybersecurity testbed was created to execute the medjacking attacks in a controlled environment. This testbed accurately simulates real-world conditions and various types of attacks. A dataset was gathered for the purpose of analysing and developing a proactive defence technique to mitigate intrusions. This study proposes a hybrid intrusion detection system as a proactive solution to mitigate medjacking in the APS. An evaluation was conducted on a hybrid intrusion detection system. These evaluations gauge the efficiency of the implemented solutions, ensuring that the preventive measure is effective. The evaluations were compiled and analysed, leading to a comprehensive discussion of the research outcomes. This study's discussion addresses the effectiveness of the findings and draws comparisons with existing works in the field of APS medjacking.

3.2 Research Design

Creating an APS prototype, a testbed for cyberattacks, and a proactive mitigation solution involved extensive and systematic research methodology. This study's research methodology comprised two primary research parts, each with its own set of sub-

methodologies. The first part consisted of developing an IoT prototype, establishing a testbed for security attacks, and collecting a dataset. The second part consisted of creating a proactive mitigation solution for the APS medjacking (See Figure 3.1).

This research focused on the crucial step of developing a formal research proposal to solve the research problem including developing an APS prototype and cybersecurity testbed to collect the dataset. The dataset used for engineering features and developing models using machine learning and deep learning techniques, were crucial steps for proactive solution development.

The research proposal included a comprehensive examination of the current art of the APS development, security vulnerabilities, and attack methods. By examining the current literature, this research crafted a concise problem statement emphasising the medjacking concerns present in APS and the imperative need for a robust proactive solution.

Design Science Research methodology was used for a hybrid intrusion detection system as a proactive mitigation solution. The primary focus was creating and evaluating various models, including dimensionality reduction methods, to evaluate performance effectively. The process started with examining the current literature on intrusion detection systems and their limitations. Developing the models required a series of processes, which were then implemented in a controlled setting to demonstrate their functionality and effectiveness in detecting medjacking cases.

Figure 3.1

Research Methodology Outline.

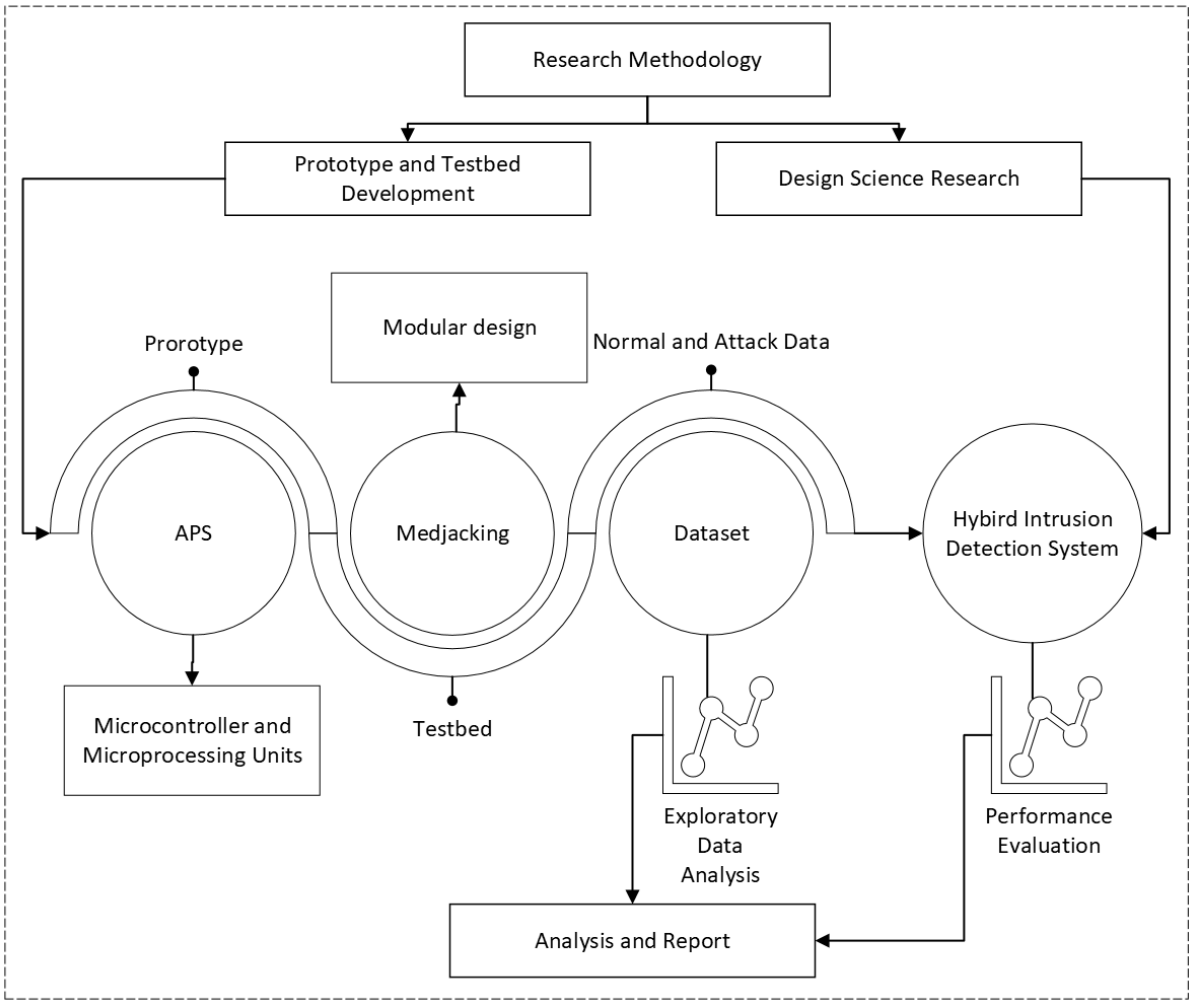


Figure 3.1 outlines the research methodology that systematically investigates and develops medjacking mitigation solutions in this study. The APS prototype and security testbed were essential to collect the dataset. The design science research process was inherently iterative, with design, testing, and refinement to consistently enhance the models. The research design is presented in the following sections.

3.2.1 Prototype and Testbed Development

The APS prototype was simulated using cost-efficient microcontrollers to create a model within the study's scope. The development of an APS employing B. Bergman's minimal model on a Raspberry Pi B and ESP32-WROOM. The DHB actor was assimilated with the

Raspberry Pi. The Raspberry Pi is the primary controller, implementing glucose-insulin dynamics in alignment with the Bergman model. The ESP32-WROOM is a peripheral device for data acquisition and administering glucose and insulin. The Raspberry Pi subsequently calculates the requisite insulin dosage and relays directives to the ESP32 to regulate the insulin administration apparatus. The components of the APS architecture were executed, encompassing communication protocols, gateways, and networks typically found in both public and private digital systems. An IoMT device facilitates communication among multiple devices via diverse gateways within a designated network, and BLE and MQTT communication protocols were employed in this study. The cybersecurity testbed was developed using a modular architecture, wherein distinct modules were utilised to execute specific functions; each module performed a designated task before changing to another process by employing an alternative module. For brute force attacks, this study used the Metasploit framework in conjunction with Kali Linux to conduct brute force attacks. For DDoS attacks, the hping3 tool on the Kali Linux within a virtual machine was used in this study. Ettercap and a virtual machine were used in this study for the MITM attack. Metasploit and virtual machines were used in a controlled environment for trojan attacks.

The development of the APS prototype and testbed serves as a significant milestone in the endeavour to devise a solution to mitigate the phenomenon of medjacking. The prototyping of APS facilitated the evaluation of the proof of concept for mitigating medjacking, thereby assessing its effectiveness and demonstrating potential security vulnerabilities that may emerge. This empirical evidence of the concept served as proof of concept the functionality and potential of the solution, as well as its limitations.

3.2.2 Design Science Research

This study examines medjacking attacks on IoMT devices, specifically APS devices, emphasising the need to use protocols such as MQTT and BLE for accurately modelling real-

world medjacking attack situations. BLE and MQTT have extensively utilised protocols in IoMT because of their unique benefits for interconnected IoMT devices. Subsequently, the formulation of an H-IDS utilised design science research methodology to demonstrate proof of concept. During the Design and Development phase, the foundational system architecture, various models, and algorithms were used in this study as a modular approach. The Development of H-IDS involved the preprocessing of raw data for analytical purposes; in this process, Minmaxscaling served to normalise numerical data, while label encoding facilitated the preparation of categorical data for ML models. Feature selection techniques were used to identify optimal features, including Correlation coefficient, Variance threshold, and Recursive Feature Elimination with Cross-Validation (RFEC) techniques to optimise models and processing.

Dimensional reduction techniques effectively minimise the number of dimensions within a dataset while preserving a maximal amount of pertinent information. Dimensional reduction techniques, including PCA, t-SNE, and UMAP, were utilised and compared for medjacking detection. Two models were designed: the anomaly-based model using an autoencoder and another signature-based model using MLP, SVM, DT, and XGBoost classifiers to identify instances of medjacking in the APS. Hyperparameter tuning technique applied to optimise each model. Dimensional reduction techniques. Essential performance indicators were computed for anomaly detection, including training accuracy, validation accuracy, training loss, and validation loss. The quantitative evaluation entails an assessment of the accuracy of signature detection models, with particular emphasis on metrics such as accuracy, precision, recall, and the F1 score.

3.3 Research Approach

The research design for the APS medjacking encompassed various essential elements that provided comprehensive and practical answers for medjacking mitigation solutions in the

APS. This research implemented a simulated APS prototype, a cybersecurity testbed, to examine four distinct categories of medjacking, including MITM, DDoS, Trojan, and brute force attacks. These four categories of attacks are not merely pertinent to conceivable medjacking situations; they also represent dynamic fields of enquiry within the field of cybersecurity. Subsequently, this study has developed H-IDS using ML and DL models and algorithms or techniques. The models include autoencoder, MLP, SVM, DT and XGBoost. The algorithms or techniques include MinmaxScalar, LabelEncoder, Variance threshold, Correlation coefficient, RFEC, PCA, t-SNE, and UMAP.

This research examined the current literature on state-of-the-art vulnerabilities and attack methods in the APS (See Sections 2.3, 2.4, 2.5 and 2.6). The prototype of the APS was constructed via a simulation technique, using a microcontroller-based framework to develop APS conditions and data streams. This simulation facilitated the modelling of APS device functionalities, thereby establishing a controlled environment for examining diverse medjacking attacks through a modular design architecture, ultimately leading to the development of a hybrid intrusion detection system. Through the simulation of targeted attack scenarios alongside typical usage patterns, the prototype established an exhaustive structure for assessing the effectiveness of the H-IDS in identifying medjacking threats. The study design included a range of quantitative approaches; further detail of the rationale for the quantitative approach is provided in Section 3.3.2. A detailed literature analysis was performed to examine security vulnerabilities and their accompanying solutions systematically. The synthesised literature was used to develop a detection system specifically to detect medjacking instances in the APS. Exploratory data analysis was used to detect relationships within one another's data features. In the context of exploratory data analysis, the Pearson correlation coefficient was utilised to explain the raw data in a more comprehensible manner, with a particular emphasis on discerning correlations between patterns, and these patterns can be prospective indicators

of intrusion detection. Through the computation of Pearson correlation coefficients among the variables, key characteristics exhibiting substantial relationships were identified, which may function as preliminary indicators for detecting anomalies. Collectively, the interpreting raw data approach has established a consistent structure for interpreting unprocessed data. This preliminary analysis of raw data proves useful for advancing intrusion detection, wherein advanced techniques can be used to develop an effective detection system.

The evaluation of the H-IDS involved a thorough analysis utilising descriptive statistics, including measures such as mean score, accuracy, precision, recall, and F1 score. This study aims to provide potential and practical insights into the medjacking of APS devices using quantitative approaches in a well-structured research methodology discussed in the following subsections.

3.3.1 Rationale for Simulation Approach

The simulation proved indispensable in medjacking, as it provided a controlled and secure environment to develop the mitigation solution and assess the effectiveness of the developed mitigation solution, such as detection systems, against reasonable, high-risk attack scenarios, all while limiting the direct involvement of proprietary devices and actual patients. Some key rationales were carefully considered as follows.

- **Issues Pertaining to Patient Safety:** Patient safety assumes the utmost significance in healthcare and medical devices. Medical devices frequently interact directly with patients; thus, any cybersecurity evaluations conducted on these devices within a practical environment entail a risk of malfunction, data breach, or device error, which could potentially cause direct harm to patients. As a result, this research emphasised the implementation of simulated APS to develop the medjacking mitigation solution while safeguarding against any negative outcomes on patient health or safety.

- **Limitations of Proprietary Devices:** Proprietary medical devices pose considerable obstacles for researchers. For instance, proprietary constraints prevent researchers from accessing the internal mechanisms of devices, thereby complicating the conduct of research experiments. This investigation consequently selected a simulated environment that simulates APS functionalities and prospective vulnerabilities.
- **Ethical and Regulatory Obstacles:** Undertaking security research on proprietary devices within medical device environments or involving actual patients would require comprehensive ethical approvals and strict regulatory regulations. Ethics committees and regulatory boards necessitate thorough documentation, meticulous risk assessments, and robust patient protection protocols. Proprietary devices and the processes for obtaining real patient approval frequently entail significant time-based investment and may extend over an indeterminate duration, possibly extending for years. Consequently, proprietary devices and real patient choice fell outside this investigation's limits. Through the adoption of simulation methodologies, the investigation could execute comprehensive cybersecurity assessments without proprietary devices and actual patient involvement.
- **Controlled Environment:** Simulated environments provide a systematically controlled testing space wherein particular cybersecurity attack conditions can be replicated within a limited time frame, thereby facilitating the optimisation of intrusion detection performance. In practical scenarios, the continuous replication of attacks and their methods on a proprietary device is often difficult owing to considerations of device accessibility, associated risk factors, or operational limitations. Using simulation, the research was able to execute comprehensive testing across numerous iterations of MITM, DDoS, Trojan, and brute force attacks.

3.3.2 Rationale for Quantitative Approach

This research has developed an H-IDS that utilises ML, incorporating different techniques in preprocessing, feature selection, dimensionality reduction, and model development, all situated within a quantitative structure. The justification for selecting the quantitative approach is based upon being in line with the study's aims of obtaining quantifiable, data-informed insights regarding the effectiveness and precision of intrusion detection mechanisms. Using quantitative techniques, the investigation could systematically assess and refine various models and algorithms based on exacting metrics, including accuracy, precision, recall, and false-positive rates.

Furthermore, the quantitative approach enabled this study to rigorous comparisons across diverse configurations, encompassing the influence of preprocessing techniques, the effectiveness of dimensionality reduction techniques such as PCA, t-SNE, and UMAP, as well as the performance metrics of several ML models (e.g., autoencoder, MLP, DT, SVM, and XGBoost). This selection allows qualitatively rigorous analyses and improves the models, as the models and algorithms can be evaluated on uniform, quantifiable criteria. By demonstrating quantitative metrics, the research might effectively illustrate the effectiveness of the H-IDS, explaining how techniques can improve detection rates.

3.3.3 Artificial Pancreas System Prototype

The prototype of the APS was built through a simulation methodology, using a microcontroller-centric framework to establish APS conditions and data streams. This simulation approach using microcontrollers enabled the modelling of APS device functionalities, including insulin pump and CGM sensor. This research used cost and time-efficient microcontrollers to develop an APS prototype in this study for various reasons, such as patient safety, limitations of proprietary devices, ethical and regulatory obstacles, and a controlled environment. All these reasons are discussed in section 3.3.1 Rationale for

Simulation Approach. This simulation enabled the modelling of APS device functionalities, thereby creating a controlled environment for the investigation of various medjacking attacks through a modular design architecture, ultimately resulting in the development of a hybrid intrusion detection system. APS prototyping led to the testing of the proof of concept for medjacking mitigation, assessing its effectiveness, and uncovering any security concerns that may arise. This concrete evidence of the concept contributed to demonstrating the functionality and potential of the solution and its limitations. A list of tools, including software and hardware, that were chosen for APS prototype development is presented in Table 3.1.

Table 3.1

List of Tools Used to Developed the Artificial Pancreas System Prototype.

Item	**Purpose**
Raspberry Pi	DBH and MQTT protocol (MQTTRoute)
ESP32-WROOM microcontrollers	Design insulin pump and CGM sensor
Core (TM) i5-10500T CPU 2.30 GHz 2.30 GHz, 16.0 GB RAM	Configure different virtual machines, write codes, monitor the prototype as a dashboard, capture data and perform analysis
Laptops Intel(R) Core (TM) i7-10870H CPU 2.20 GHz 2.21 GHz, 16.0 GB RAM	Configure different virtual machines, write codes, monitor the prototype as a dashboard, capture data and perform analysis
nRF52840 Dongle USB	Bluetooth sniffer
Operating Systems	Windows 10, Ubuntu, and Kali Linux
Arduino IDE	Codes editor
Wireshark	Packet analyser
Portable solid-state drive and USB	Store data
Smartphone	Run different Bluetooth sniffer codes
Jupyter Notebook	Write code
PyCharm	Write python code

This research utilises simulation approaches to develop an APS prototype. The reason for conducting a simulation approach for cybersecurity examination is that the available information for module development of diabetes glycaemic control mostly emphasises functionality rather than cybersecurity performance assessments or defensive mechanism validations. The Raspberry Pi and ESP32-WROOM microcontrollers were selected for developing the APS prototype based on their performance, economic viability, and appropriate applicability to IoT and IoMT ecosystems. According to [672], the researchers stated that the ESP32 is a low-power consumption and cost-effective cost microcontroller. The ESP32 incorporated dual-mode Bluetooth and Wi-Fi capabilities. In another study [673], the researcher used the Raspberry Pi and ESP32-WROOM microcontroller to develop an IoT-based monitoring system, due to its cost-effectiveness and efficiency, alongside its Bluetooth and Wi-Fi functionalities. The research community widely uses these microcontrollers to propose and assess innovative advancements in the field. For instance, a real-time human activity recognition system was introduced in [674], employing Raspberry Pi and ESP32-WROOM microcontrollers, attributed to their cost-effectiveness, reliability, and integrated Bluetooth and Wi-Fi functionalities [674].

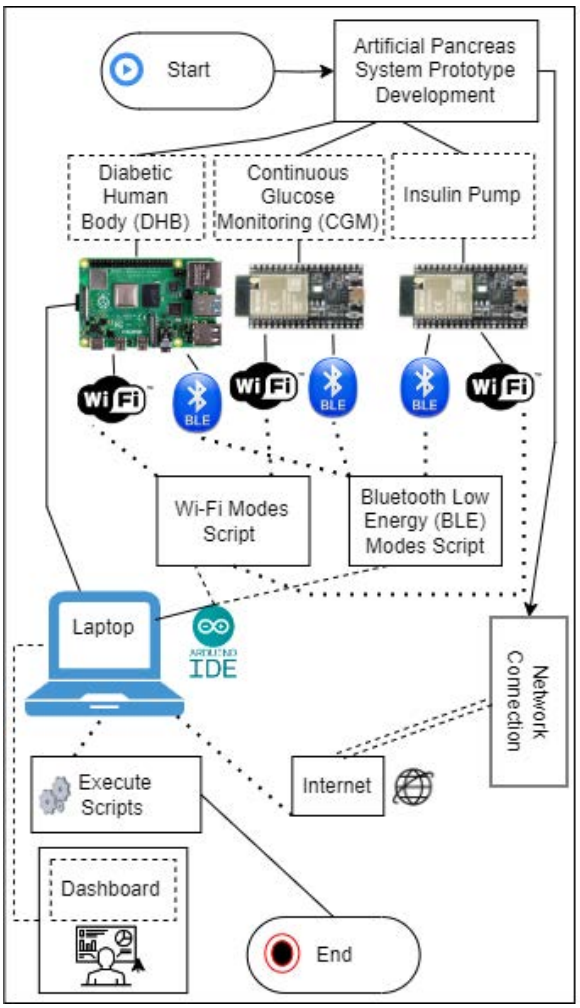
In this research, the simulation method was utilised to develop the APS prototype's simulated insulin pump and CGM sensor using an off-the-shelf ESP32-WROOM microcontroller board for connection with the diabetes glycaemic control modules. A diabetic human body (DHB) actor was designed using the Raspberry Pi B. A CGM sensor and an insulin pump were designed using an ESP32-WROOM microcontroller. Figure 3.2 depicts a flowchart of the APS prototype development.

The development of an APS utilising B. Bergman's minimal model on a Raspberry Pi B and ESP32-WROOM. The DHB actor was integrated with the Raspberry Pi. The Raspberry Pi is the principal controller, executing glucose-insulin dynamics following the Bergman model.

The ESP32-WROOM is a peripheral device for data acquisition and glucose and insulin administration. The Raspberry Pi subsequently computes the required insulin rate and transmits commands to the ESP32 to modulate the insulin delivery devices. In this study, a Bluetooth sniffer served as a passive tool rather than being involved in executing attacks like MITM, DDoS, brute force, and trojan attacks. However, when utilised with additional tools such as Metasploit, hping3, Ettercap, or Nmap, a Bluetooth sniffer can facilitate testing attempts passively but not actively.

Figure 3.2

A Block Diagram of Overall Processes for Developing the Artificial Pancreas System Prototype.



This prototype development enabled the research to circumvent the need for patient participation while conducting various tests, including a cybersecurity exploratory study. The simulation approach to developing an APS prototype enabled the collection of a dataset and development of a medjacking detection approach. Section 2.4 discusses the APS's security vulnerabilities. A crucial security weakness is medjacking [81, 89, 199]. Security research on APS devices, for instance, medjacking detection, is limited due to patient safety, limitations of proprietary devices, ethical and regulatory obstacles, and a controlled environment. All these reasons are discussed in section 3.3.1 Rationale for Simulation Approach. The development of the APS prototype was extremely important in validating the concept, gathering data, investigating medjacking and its solutions, and enhancing the mitigation solution through testing. This study presented a proof of concept, demonstrating the importance of providing a cost and time-efficient solution.

3.3.3.1 APS Prototype Architectural Components

The APS architecture components refer to communication protocols, gateways, and networks to enable IoMT communication. Fundamental communication protocols, including Bluetooth Low Energy (BLE) and MQTT, are imperative for secure and efficient data transmission between wearable glucose monitors and insulin pumps. The prototype employs gateway devices, such as the Raspberry Pi B and ESP32-WROOM, to facilitate data transmission between sensors and control units. Networking devices, including routers and switches, facilitate robust communication within local networks, ensuring continuous and smooth data flow. An IoMT device establishes communication between several other devices through various gateways inside a specific network. To select a dedicated communication protocol, it is crucial to consider many elements, such as network architecture, gateway configuration, and intended application. BLE and MQTT are communication protocols used in IoMT devices. To achieve client and server communication, many additional tools and

software packages, including pre-existing programming libraries, were required to implement communication protocols.

This study used the MQTT broker and MQTT clients for a simulated APS prototype. An open-source framework, Mosquitto MQTT broker and ESP-MQTT was used for IoMT communication. An open-source cloud server, Eclipse Ditto, was configured and deployed for the APS prototype's simulated insulin pump and CGM sensor (See Figure 3.3).

Figure 3.3

A Block Diagram of Overall Processes for Implementing the Cloud Within the Message Queuing Telemetry Transport Protocol.

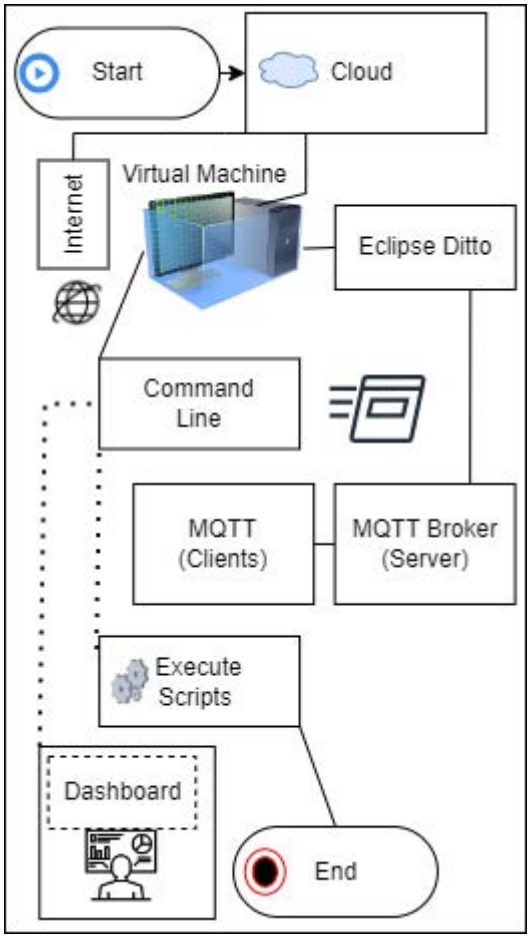


Figure 3.3 depicts the MQTT communication protocol's use in the APS 'prototype's cloud service solution. The APS prototype involved a rigorously developed combination of

architectural elements and protocols to enable real-time, dependable communication among devices. Each component fulfils an essential function and has been selected in accordance with the technical specifications important to the functionality of APS. Initially, fundamental communication protocols, BLE and MQTT, were utilised to enable data transmission among devices. BLE was essential owing to its minimal power consumption, rendering it particularly suitable for wearable devices such as CGMs and insulin pumps. Concurrently, MQTT, a lightweight publish-subscribe messaging protocol, facilitates efficient and reliable data transmission.

The gateway devices, namely the Raspberry Pi B and ESP32-WROOM, were a pivotal function within the APS by facilitating dependable, low-latency data processing and communication. The Raspberry Pi B operates as the principal processing unit, proficient in executing intricate computations important to insulin dosage computations while maintaining connectivity with cloud servers. The ESP32-WROOM is provided with Wi-Fi and Bluetooth Low Energy functionalities, interfaces wearable devices, and a central control unit. It continually transmits data from CGM and insulin devices. Networking devices such as routers and switches were fundamental to establishing a stable local network that interconnects all components of the APS. These devices facilitate smooth interaction among each component, and the network infrastructure facilitates the real-time operation of the APS.

The rationale for the Utilisation of BLE and MQTT in the APS Prototype was that these protocols were strategically selected to fulfil essential requirements related to minimal power consumption, smooth data transmission, and reliable device communication, each of which was essential for the effective development of IoMT eco-system. The role and function of BLE within the APS prototype is that BLE serves as an interface for communication between wearable devices, including CGM and insulin pump. The MQTT protocol is a de facto IoT protocol. Within the APS prototype, MQTT functions as the fundamental framework for

facilitating data interchange between wearable devices and the cloud infrastructure, as exemplified by Eclipse Ditto. This protocol facilitates asynchronous communication, enabling devices to transmit data.

The APS prototype's foundational operations and normal activities have been effectively executed at this stage. Completing this stage assisted the researcher in moving to the subsequent stage: the security testbed. The importance of achieving this stage lies in its essential function in formulating a comprehensive medjacking mitigation strategy for the APS. Specifically, it facilitated the formulation of an H-IDS, enabling it to differentiate between normal and attack patterns; the development of an H-IDS is discussed in Section 3.6.

3.3.4 Cybersecurity Testbed

To demonstrate medjacking attacks, this stage involved developing a cybersecurity testbed comprising a range of virtual machines, operating systems, and programming libraries. The testbed included virtual machines running on both laptop and desktop computers using a diverse selection of operating systems such as Microsoft Windows, Ubuntu, and Kali Linux. These systems are equipped with powerful programming libraries that enable the execution of various security attacks. The list of tools used to design the cybersecurity testbed is presented in Table 3.2. A wide range of tools, software, and programming libraries were employed to create a highly customised infrastructure to ensure the preservation and integrity of all attributes without damaging any APS component. Data were collected during preliminary short-term testing and the execution of various infiltrations on active APS prototype’s subsystems insulin pump and CGM sensor.

Table 3.2

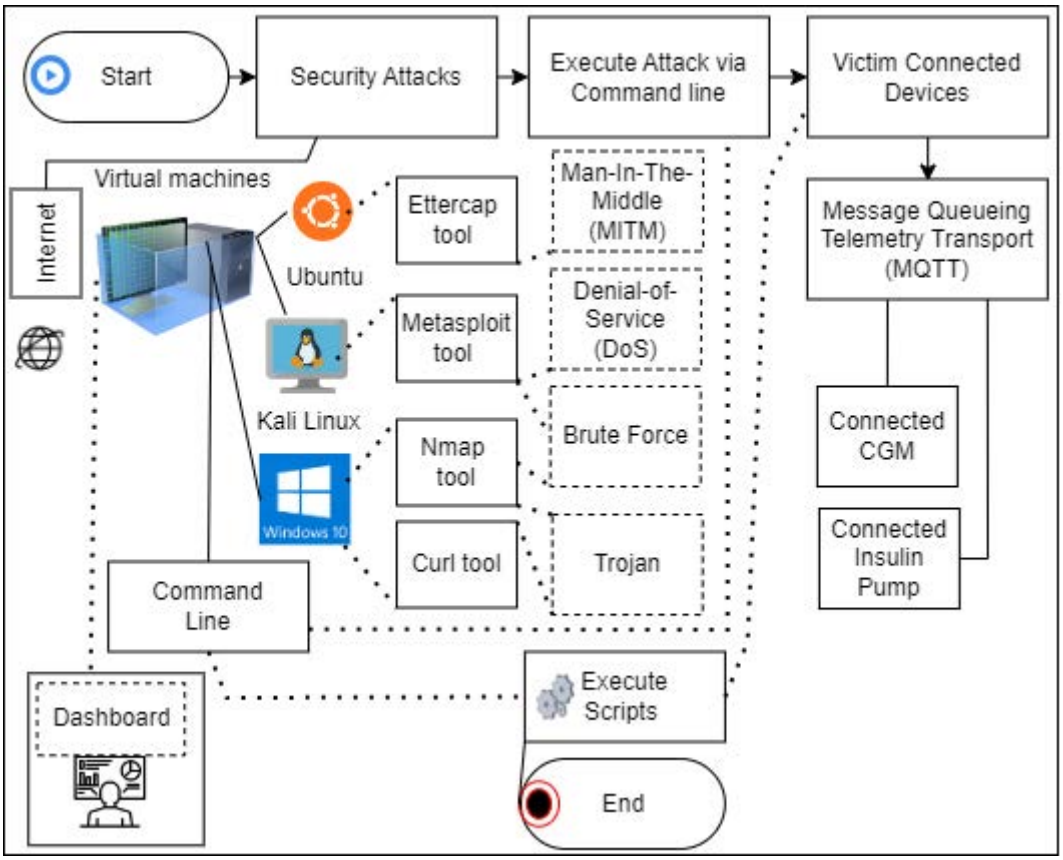
The List of Security Attacks Involved in the Development of a Security Testbed to Infiltrate the Artificial Pancreas System Components.

Attacks	**Tools**	**Infiltration**
MITM	Ettercap	Interrupting communication between APS components and a server. Connecting an attacker's internet protocol (IP) address with the IP address of the APS components
DDoS	hping3	Sending multiple synchronise packets to the server and APS components
Trojan	Metasploit framework and Curl	Sending script backdoor using Metasploit framework and Curl tools
Brute force	Metasploit, Nmap and SSH	Using MQTT-PWN framework

The different security attacks and corresponding programming libraries presented in Table 3.2 are publicly available as open source to exemplify the security attacks. The tools chosen for the cybersecurity testbed, Ettercap, hping3, Metasploit Framework, Nmap, SSH, and Curl, were selected following their operational capabilities, adaptability, and relevance to the study's aims. The research community uses open-source tools due to cost-efficient, ethical, and privacy concerns. In [675], the researchers used Ettercap to conduct MITM attacks. In [676], the researchers used the hping3 tool to execute DDoS attacks. In [677], the researchers used the Metasploit framework and Curl to execute Trojan attacks. In [678], the researchers used Metasploit, Nmap, and SSH tools to execute brute force attacks. A series of controlled infiltration strategies were conducted with the APS prototype over various time durations. The cybersecurity testbed developed in this study is illustrated in Figure 3.4.

Figure 3.4

A Block Diagram of the Cybersecurity Testbed's Development.



The cybersecurity testbed in Figure 3.4 encompasses several security threats such as brute force, DDoS, MITM, and trojan attacks. The cybersecurity testbed enabled the collection of medjacking attack data to perform the comparison of normal activities and attack activities. The study involved obtaining an attack dataset for quantitative analysis and developing a detection strategy for medjacking attacks. The purpose was to assess the reliability of these systems in detecting active attacks and evaluate the overall effectiveness of the built cybersecurity testbed.

3.3.5 Cybersecurity Testbed Limitations

A cybersecurity testbed constitutes an experimental framework designed to replicate real-world environments for the purpose of evaluating and demonstrating security measures,

algorithms, techniques, and frameworks [598, 679-686]. A cybersecurity testbed can be utilised across various domains, such as vehicular networks [687]. The researcher stated that testbeds are essential in the security assessment of in-vehicle computing systems, and that the execution of effective testing techniques is crucial for the successful evaluation of the security of these systems. Knowledge of various testing techniques can prove advantageous in selecting and implementing the most effective technique for achieving optimal outcomes. Aside from the testbed, the researcher stated that the real-world testing configuration could possibly cause circumstances that may compromise the safety of both the vehicle and the testers [687]. Another example is in the domain of industry control; testbeds offer a systematically controlled environment wherein researchers can replicate real-world conditions and examine the responses of components to cyberattacks [679, 680]. Cybersecurity testbeds present numerous benefits, including facilitating a controlled setting wherein researchers can replicate particular attack scenarios [686]. Moreover, testbeds can be customised to incorporate particular hardware, protocols, and attack vectors relevant to the research [687]. Despite these advantages, testbeds encounter obstacles, especially an absence of comprehensive applicability compared to real-world settings.

This study used a simulated dataset; however, a significant limitation lies in the inability to compare it with a real-world dataset utilising the proposed H-IDS. Theoretical datasets are fundamental for comprehending the operational processes of IoMT devices and security attacks and developing ML models aimed at intrusion detection or advancement of any algorithm or technique. Datasets provide high reliability by explaining normal and attack patterns, supported by a variety encompassing an extensive range of scenarios, including rare or unexpected circumstances. Nevertheless, real-world datasets present considerable challenges, including privacy and ethical considerations; these issues are discussed in Section 3.3.1. Testbeds tackle ethical decisions as they do not utilise actual patient and proprietary devices, whereas real-

world datasets necessitate rigours and adherence to privacy and safety regulations. Testbeds facilitated the execution of replicable experiments; however, real-world datasets are dependent upon dynamic occurrences, rendering them inherently non-replicable.

The testbed developed in this research possesses certain limitations. A significant limitation lies in its comparisons with real-world datasets, as the testbed functions within a controlled environment and is limited to account for the investigation of real-world attack scenarios. The investigation of real-world attack causes, and the complications associated with actual patients is limited in this study. The comparison of cybersecurity testbeds and real-world datasets illustrates the inherent trade-offs between control and realism, which are important for advancing IoMT security research. The constraints inherent in the testbed emphasise the necessity for additional investigation and comprehensive testing within more real-world scenarios. Tackling these limitations may enhance the resilience and scalability of IoMT systems, thereby facilitating the advancement of secure and dependable healthcare technologies.

3.4 Data Collection

The existing datasets found in the literature are not suitable for this study due to several reasons. First and foremost, there is a discrepancy between the objectives of this study and those of other existing studies in the literature. The available datasets did not perfectly align with the specific objectives of this study, lacking features, types of devices, or attack scenarios such as medjacking in the APS. Furthermore, these datasets cannot be customized, which restricts their adaptation to align with the current research objectives. Additionally, many datasets have a narrow scope in terms of attack scenarios and may not cover advanced or innovative medjacking techniques intended to be investigated in this study. Consequently, data collection will be performed through APS simulations.

In this study, the data collection unit was created to gather and store data. Several steps were executed to ensure the collection of data. In the event of any errors, troubleshooting was conducted to resolve the issues. Troubleshooting was conducted in response to errors or technological faults that arose during the execution of various tools and programming codes. In this study, the primary objective was to collect a dataset of normal and attack activities to forecast and assess intrusions.

The utilisation of TCPDump, Wireshark, and TShark was vital for collecting the dataset, as each tool offers unique functionalities that, when combined, facilitate comprehensive and effective data acquisition, processing, and analysis. The combined use of TCPDump and Wireshark is vital for collecting and analysing various features, especially within cybersecurity research and intrusion detection systems. Although TCPDump is primarily used to capture packets effectively, Wireshark offers advanced features for comprehensive packet analysis and feature extraction. The complementary application of these tools significantly enhances the dataset collection and feature extraction for ML models. TCPDump constitutes a command-line utility used for capturing packets. In contrast to graphical user interface-based tools, TCPDump functions with remarkable effectiveness, enabling data gathering in resource-limited contexts, such as IoMT devices. TShark tool is the command-line version of Wireshark and is vital for automating extensive analysis. Wireshark and TShark facilitate scriptable packet processing, thereby enabling researchers to collect datasets and extract essential features for intrusion detection.

The packet-based attributes were extracted from the packet capture data and transformed into the (CSV) file format. The dataset encompassed both incoming and outgoing traffic, along with a variety of normal and attack packets. Data packets were collected using several programmes including tcpdump, Wireshark, and Tshark. The tcpdump software was employed via the command line interface, whereby a server (Syslog) was established to store

the traffic recorded by tcpdump. The data characteristics were remotely accessed via the SSH protocol. The utilisation of Tshark tools facilitated the extraction of further characteristics that may not be obtained via the use of Wireshark. Tshark packet analyser and Wireshark was used through a command-line interface.

3.5 Data Analysis Procedures

Cybersecurity became a dynamic field where new cyber threats constantly emerged. EDA technique enabled constant enhancement via insights into data patterns and revealed new trends in attack patterns. EDA provided a fundamental grasp of data characteristics, enabled signature and anomaly detection, and enhanced predictive models. In this study, all packets were captured, processed, and labelled to conduct EDA. The normal and anomalous packets were analysed and compared to understand data patterns. Different packets fluctuated from each other; thereby, to understand these patterns, the Pearson correlation coefficient was applied to precisely visualise and understand the normal and anomalous patterns of the paired packets or signals and dependencies.

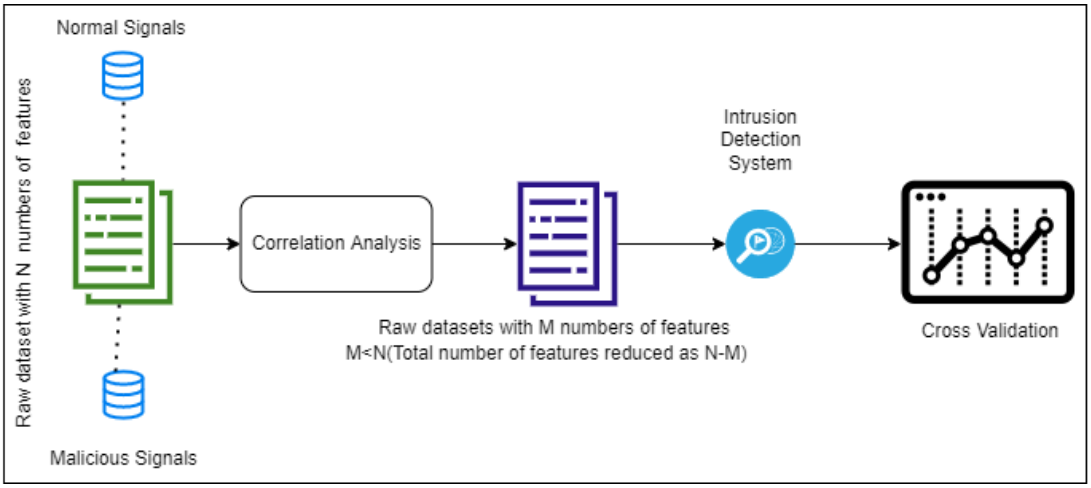
The similarity of correlated signals was measured between two paired signals, and numeric correlation was identified to visualise signals generated by the APS subsystems. The primary objective for identifying the correlation between paired variables and linear numeric signals was to understand how closely paired attributes have a reciprocal relation.

Different subsystem packets performed different tasks within the connected subsystems, and some packets indicated correlation and dependence on another signal. A statistical relationship between some data packets was perceived as positive or negative correlations with paired packets. The correlation of paired packets represented the absolute dependencies of packets. For example, if signal one (i.e., the TCP connection synchronisation) was active, then signal two (i.e., the TCP connection synchronisation acknowledgement) was also active in response. As a result, a correlation analysis was most significant in this study, as it identified

the relationship and dependencies of paired packets. The major direction of this EDA led to identifying linear dependencies and parameters to develop an automated technique to detect anomalies. Figure 3.5 shows the EDA approach used in this study.

Figure 3.5

The Exploratory Data Analysis Approach for the Current Study.



As visualised in Figure 3.5, the EDA approach consisted of correlation analysis and cross validation. Different signal correlations were identified, and heatmaps were created to visualise the dependencies of each linear numeric signal. Some key features (i.e., packets) were identified as having a strong correlation with each other. Trends were also identified, such as different attack correlation dependencies in the same set of signals that changed a lot, and many signals gave different correlation coefficient values in different types of attacks.

In data science, a heatmap is a graphical depiction of data wherein discrete values summarised within a matrix or tabular format are illustrated through a collection of colour gradients. Heatmap is a two-dimensional visualisation mechanism used to represent the scale of data values across dual axes, with colour gradients denoting the intensity or distribution. Heatmaps are commonly used to discern patterns, correlations, and anomalies within datasets, rendering them an invaluable approach for EDA [688-694]. For example, the researchers in [695] stated that heatmap visualisations utilise colour-coded scales to denote data values.

Heatmaps serve as a valuable tool for visualising the distribution of different features related to ransomware samples, thereby assisting analysts in identifying clusters or anomalies within the dataset [695]. In another study [696], the researchers stated that visualisation tools, including heatmaps and graphs, can assist analysts in comprehending the interconnections among various events and their implications for essential resources. Visualisation approaches can also explain which infrastructure assets are prone to increased vulnerabilities or how cybersecurity risks change in accordance with the present organisational context [696].

In a heatmap representation, the numerical values are encoded within colour blocks, wherein variations in intensity signify quantitative differences in expression levels, thereby providing a clear representation [697]. The researchers proposed a heatmap-based DL model for the classification of network attacks and demonstrated that heatmaps enable the understanding of complex data sets by offering a visually engaging and logical approach, thereby rendering them a valuable tool for the visualisation and analysis of data [697].

A correlation study of normal and security attack patterns requires multiple phases. First, the researcher determines system specifications and interactions with other devices and network infrastructure, and then they study specific security attacks. Next, the researcher collects data from normal and security-related events. Subsequently, data must be processed for labelling to maintain the standardisation of both normal and attack data packets. In correlation analysis, statistics are used to investigate the link between normal behaviour and security attack instances. Correlation results must then be interpreted to gain actionable insights and detect potential compromise indications.

The interrelationship among data attributes is significant in the feature selection process for ML models [698-700]. The feature selection process seeks to identify the most appropriate feature for the training of ML models, thereby enhancing performance [701]. Correlation, a statistical metric denoting the interrelationship between two variables, could appear as either

positive (indicating a direct relationship) or negative (which indicates a reverse relationship) [702]. The correlation method allows the formulation of effective choices about the retention or elimination of specific features in ML models [703].

3.6 Proposed Hybrid Intrusion Detection System

An Automated Cyber Defence (ACD) technique using ML and DL models was proposed for the H-IDS. The implementation of ACD strategies in the APS prototype's simulated insulin pump and CGM sensor offered an autonomous and efficient approach to detecting medjacking. ML and DL techniques use both anomaly and signature detection approaches that enhance the effectiveness of medjacking detection. In the field of cybersecurity, the term anomaly refers to the process of deviating from a data point, event, or action that diverges from the established norm to identify unusual patterns, such as new attacks. The anomaly detection method entails the employment of a model that detects new patterns through deviations from established typical patterns.

Models utilised in signature detection are designed to detect established patterns that are linked to various types of attacks. Although signature detection methods are successful in recognising known attacks, they may be limited when detecting unknown attacks that do not correspond to any existing attack signatures. Consequently, the utilisation of anomaly detection methods serves as a valuable complement to this strategy since it enables the identification of previously undetected attacks. The dataset utilised in the proposed H-IDS consisted of a varied collection of labelled instances depicting normal and attack activities. This dataset has been divided into training, validation, and test sets to aid in model training and evaluation. The dataset consisted of both normal and attack packets. Each data packet in the dataset consisted of varied characteristics, including packet sizes, timestamps, and extra metadata offering detailed information about the events.

3.6.1 Hybrid Intrusion Detection System Development Requirements

The development of the H-IDS employed the tools, models, libraries, and algorithms detailed in Table 3.3 below. The H-IDS was founded on open-source libraries and tools and comprised three distinct components: data preprocessing, feature engineering, and the detection engine. Open-source models, libraries, and algorithms helped reduce time and costs. Development was accelerated by focusing on experimentation and innovation rather than developing libraries and tools. Additionally, open-source initiatives frequently excel due to extensive and engaged communities of developers, researchers, and industries that offer novel solutions to various problems. Researchers can use this combined knowledge to solve problems, share ideas, and improve detection system development. The accountability found in open-source resources helps build trust among researchers. Open-source models and algorithms enable time- and cost-efficient rapid development, in-depth analysis, validation, and personalisation, which enhances trust in the models' dependability and safety. Furthermore, detection system development can be customised with specific research goals using open-source models, libraries, and algorithms. Having the ability to personalise models allows for testing different detection classifiers, algorithms, setups, and features to identify optimal outcomes.

Table 3.3

Rationale Requirements of the Hybrid Intrusion Detection System using Machine Learning and Deep Learning Models with Feature Engineering algorithms, Tools, Libraries, and Models.

Component	**Algorithms Tools, libraries, models**	**Description**
Backend application	Jupyter Notebook, Python3, NumPy, pandas, Seaborn, imblearn, Matplotlib, scikit-	The backend handled data processing, logic, and responses with datasets, external services, and the front-end interface

Component	Algorithms Tools, libraries, models	Description
	learn, FCBF, TensorFlow, and Ploty	
Data preprocessing	MinmaxScalar	Transformed numerical features into 0 to 1 to obtain numerical stability
	LabelEncoder	Transformed categorical data into numerical labels
	Drop not a number (NaN)	Dropped NaN
Feature selection	Variance threshold	Removed features with low variance
	RFECV	Identified and selected relevant features using cross-validation to minimise overfitting
	Correlation coefficient	Identified key features using correlation measurement
Dimensionality reduction	PCA	Reduced high-dimensional data while maintaining important information for linear features
	t-SNE	Reduced high-dimensional data into 3D while maintaining important information for non-linear features
	UMAP	Reduced high-dimensional data into 3D while maintaining important information for non-linear features
Detection engine	Signature based	Identified known attack patterns using supervised ML techniques
	Anomaly based	Identified unknown attacks using DL and an autoencoder. The Mean Squared Error (MSE) loss function was used in the autoencoder to measure the differences between the original input and the remodelled output

Open-source libraries and tools were essential to developing H-IDS with Python3 in Jupyter Notebook. The NumPy library is a crucial tool for scientific computing, as it offers

essential features for handling extensive arrays and important mathematical operations needed for data processing in the H-IDS. The pandas library enhances NumPy with robust data structures and operations designed for data processing and analysis. For example, NumPy was imported to manage multi-dimensional arrays and perform mathematical operations for data preprocessing. The pandas library was imported and utilised for many data-related tasks, including data cleansing, exploring, transforming, summarising, handling, structuring, visualising, and integrating with NumPy and other libraries.

Data preprocessing involves shaping raw data into processed data, which includes tasks such as cleaning and improving data. Handling missing values, deleting duplicates, scaling numerical features, encoding categorical data, and balancing the dataset by over-sampling minority classes or under-sampling majority classes was especially important for intrusion detection. Two feature engineering methods were performed. First, key features were selected, and dimensionality reduction algorithms were applied to reduce high dimensions into lower dimensions to capture relevant information. The feature engineering process improved the distinguishability of normal and anomalous patterns. Second, a detection engine was implemented, which involved ML and DL models also known as classifiers for both signature and anomaly detections.

The scikit-learn package was imported and utilised for the implementation of ML and DL models to develop H-IDS, including dimensionality reduction algorithms. Furthermore, TensorFlow was utilised to implement the DL autoencoder. Utilising libraries such as Seaborn and Matplotlib greatly enhances data visualisation. Seaborn, which is based on Matplotlib, provides a user-friendly way to generate attractive statistical visuals that effectively present data patterns related to intrusion detection. Moreover, Matplotlib offers a wide range of tools for creating both static and interactive visualisations. These open-source libraries and tools can be used to develop intrusion detection systems with Python and Jupyter Notebook. By utilising

these components, developers can preprocess data, create ML and DL models, present results visually, and implement scalable solutions to efficiently identify and address security vulnerabilities.

3.6.1.1 Justification of Open-Source Tools

Open-source libraries are extensively utilised in research studies for advancing cybersecurity and various scientific research, attributable to their inherent flexibility, accessibility, and robust community support. The accessibility and cost-effectiveness of open-source libraries represent one of the principal justifications for their utilisation. Open-source libraries are readily accessible, thereby minimising the necessity for costly proprietary software. In the current literature, the research community uses open-source libraries [565, 704-712]. The use of open-source libraries facilitates the development of high-quality research for the scholarly community operating under constrained financial resources, thereby ensuring that economic limitations are not preventing the advancement of innovation.

Open-source libraries' adaptability and integrative potential are suitable for various research domains, including IDS. The libraries NumPy, pandas, and TensorFlow effectively enable multiple phases of the IDS development pipeline, encompassing preprocessing through to model deployment [706-708, 713-717]. These libraries adeptly manage a variety of data formats and ML workflows, therefore allowing the simplification of complex research endeavours. Libraries such as NumPy and pandas demonstrate considerable effectiveness in managing datasets, facilitating the effective dealing of data and extracting pertinent features [705, 712, 718-724]. The processes of visualisation and interpretation have significant importance in the contexts of cybersecurity and other research domains, with open-source libraries such as Matplotlib, Seaborn, and Plotly demonstrating high levels of visualisation in these domains. These libraries provide complex visualisation capabilities, thereby enabling researchers in the analysis of trends, patterns, and outcomes within their datasets [711, 712,

725-727]. Clear representations enhance comprehension, enabling better decision-making throughout the research process.

In the context of ML model development and experimentation, libraries such as scikit-learn and TensorFlow are indispensable resources [633, 634]. Scikit-learn provides a comprehensive selection of ML algorithms suitable for benchmarking and evaluation [635-637], whereas TensorFlow provides the development of DL architectures, including autoencoders, which are often used for anomaly detection [728-731]. These tools enable researchers to complete experimentation with diverse algorithms and architectures, resulting in optimised solutions. Moreover, the Jupyter Notebook is a vital tool for data scientists handling data, code, and visualisations, providing a user-friendly environment for exploration and experimentation [732, 733].

Ultimately, open-source libraries enhance the scalability and deployment of IDS models for practical applications in the real world. Through the utilisation of open-source libraries and tools, researchers can minimise expenditures and save time, thereby directing their research towards innovation. Open-source tools and libraries support consistency, accessibility, and ongoing enhancement, making it particularly suitable for both academic and industrial researchers.

3.6.1.2 Hybrid Intrusion Detection System Development

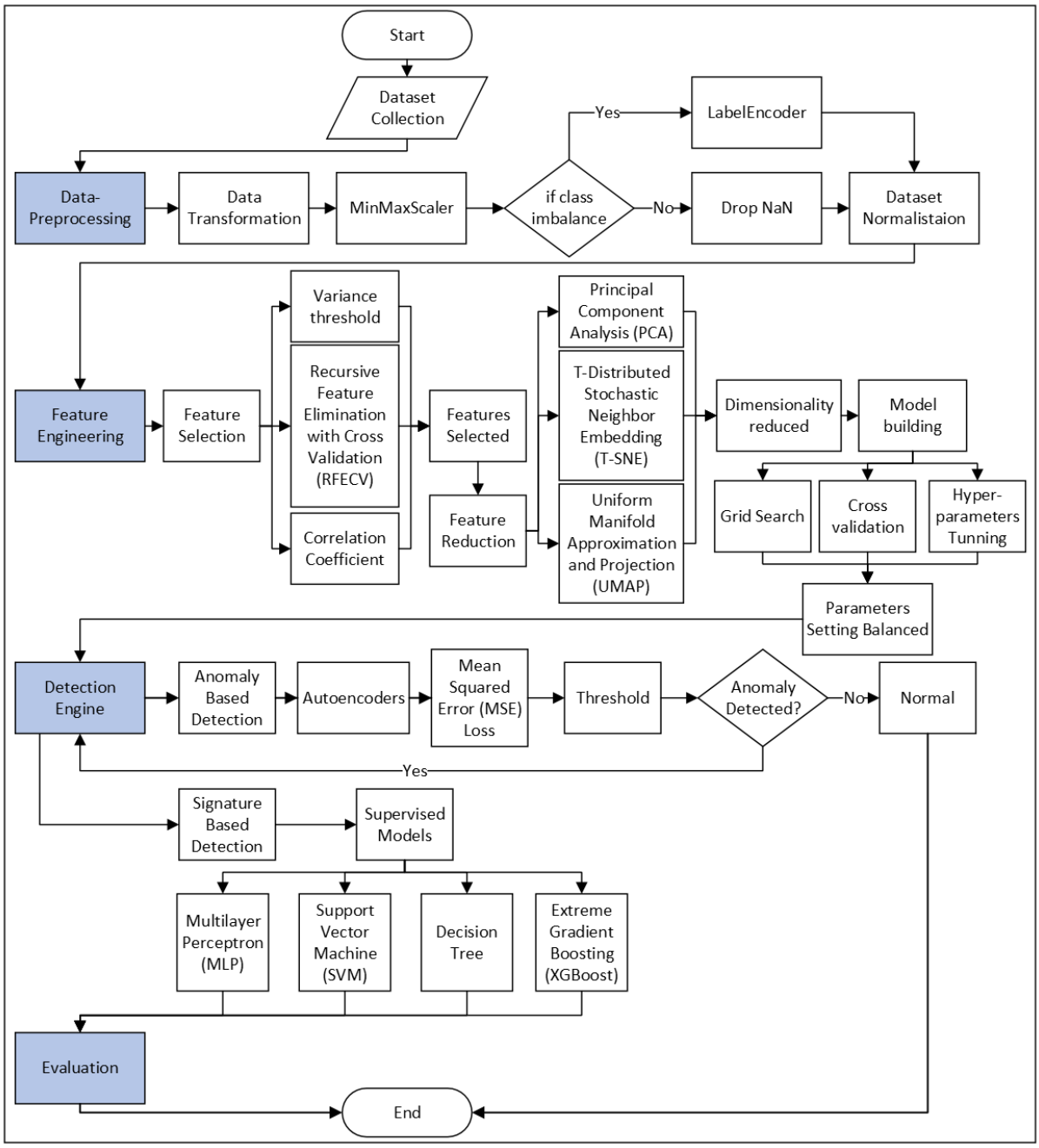
The application of feature engineering techniques was employed to select and extract significant information from the 2022APS dataset. This study gathers normal and attack data packets using an APS prototype and a cybersecurity testbed, as described in Chapter 4. The dataset was collected in 2022 and is therefore referred to as the 2022APS dataset (See Section 4.5). During the next stage, the dataset was used for training ML and DL models and testing sets. The models were trained while fine-tuning hyperparameters to enhance performance. Afterwards, these models were assessed on the testing set to avoid overfitting, and metrics such

as accuracy, precision, recall, and the F1 score were computed to evaluate the models' performance.

Figure 3.8 presents a flowchart of the H-IDS process for mapping out the data preprocessing and feature engineering steps to detect medjacking. In the H-IDS, data preprocessing consisted of different techniques to convert raw data into refined data. Feature engineering was then implemented by extracting relevant features from the dataset. Figure 3.6 captures every step of the H-IDS using ML and DL techniques to detect medjacking. Every element in the flowchart provides an essential part in understanding the organised development of the H-IDS activities through different stages and decision points.

Figure 3.6

A Comprehensive Framework of the Hybrid Intrusion Detection System.



The H-IDS starts at the top of this flowchart by identifying the initial input. As the flow progresses, various processes become apparent and reveal the various components that processes can apply based on predefined conditions. The flowchart thus provides information on the systematic activities and interconnections of each sequential H-IDS activity.

The utilisation of feature engineering algorithms, and ML and DL models, were substantial components of the H-IDS. The ML and DL models employed in this study include the autoencoder, MLP, DT, SVM, and XGBoost. The anomaly detection used an autoencoder, and a threshold for normal and attack data points was set to identify differences. The Mean Squared Error (MSE) loss function was used in the autoencoder to measure the differences between the original input and the remodelled output. An anomaly detection classifies deviations from the normal behaviour of the system using an autoencoder model to learn from historical data and recognise patterns that specify intrusions. In signature detection, the models performed training to detect intrusions by analysing historical data and behavioural patterns. Classification tasks were performed using supervised learning approaches, and the accuracy was improved by employing ensemble methods.

The execution of the proposed H-IDS is presented in Chapter 5, which covers the construction of feature selection, dimensionality reduction, gridsearch hyperparameter, and model development, as well as a detailed discussion of the criteria used for each technique. Chapter 5 aims to provide a solid foundation for model building by utilising relevant characteristics. Chapter 6 presents the presentation of model performance metrics. A comparison of four models, MLP, SVM, DT, and XGBoost with PCA, t-SNE, and UMAP dimensionality reduction techniques, is presented in Chapter 6.

3.7 Validity and Reliability

The effectiveness of an H-IDS was assessed based on its validity and reliability. Validity has measured the effectiveness of the H-IDS to effectively detect intrusions (true positives) and correctly recognise routine activity (true negatives). Reliability was an important factor in an H-IDS's ability to detect intrusions. The capacity of an H-IDS to identify intrusions was significantly influenced by its reliability.

A dataset that included both normal and attack activities was utilised to develop the H-IDS and assessed the validity of the findings was then examined. The dataset was divided into separate subsets for training and testing. The H-IDS models were trained using the training set and evaluated using the testing set. Confusion matrices were created using the testing results to calculate the true positive rate, true negative rate, false positive rate, and false negative rate. Metrics such as precision, recall, and F1-score were subsequently calculated.

Several important performance measures were analysed to assess the accuracy. The metrics include the True Positive (TP), which determined the H-IDS's ability to detect genuine intrusions accurately, and the True Negative (TN), which measured the precision of recognising normal activities. The test assessed the False Positive (FP) and False Negative (FN) to identify the proportion of normal activities incorrectly identified as intrusions and the proportion incorrectly identified as normal activities, respectively. Precision and Recall were measured by calculating the ratio of properly recognised intrusions to the total number of identified attacks, and the ratio of correctly identified intrusions to the total number of real incursions. The F1-Score, a criterion that combines precision and recall, thoroughly evaluated the correctness of the system.

The results exhibited considerable validity, with strong performance metrics that imply accurate identification of intrusions and normal activities. The reliability of findings across several tests and consistent measurements demonstrates a significant degree of dependability without any indication of bias. The subsequent section discusses the performance matrices, which encompass both validity and reliability.

3.7.1 Performance Metrics

The performance metrics' main goal was to determine how well the ML and DL models predicted specific intrusions and discriminated the normal activity. Each model learned from historical data to make implications of new hidden features or interfaces. The main reason for

choosing appropriate performance metrics was to improve the detection models' performance based on tweaking features and tuning the hyperparameters. This study visualised the performance of each implemented detection model.

The classification models are founded on the confusion matrix, also known as an error matrix. The confusion matrix is preferably used in ML and DL models to visualise the classification models' performance. There are typically four confusion matrices, FN, FP, TN, and TP, that are used to measure detection results. A confusion matrix visualises the performance where the output could be two or more classes of the predicted model. To estimate the veracity of each detection model and decide whether to adopt it, different performance metrics for intrusion detection were applied in this study to examine implemented H-IDS. Four common ML performance metrics were used: F1 score, accuracy, precision, and recall.

Accuracy assessment shows the overall efficiency of a classifier within classification models. The accuracy is a percentage of classified events to the total number of events samples. The accuracy of the detection model was measured using Equation 3.1.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Recall or sensitivity assessment, also known as a detection ratio, shows the positive classified events as attacks on all historical data samples. Recall is a percentage of positive detected behaviours in all class observations (TP and FN). The recall of the detection model was measured using Equation 3.2.

$$\text{Recall} = \text{Detection Rate} = \frac{TP}{TP + FN} \tag{3.2}$$

Precision assessment shows the positive predictions generated by classification models. Precision provides the part of true positives over the sum of true positives and false positives.

A high precision ratio specifies that the model has a low false positive rate and a higher level of confidence in its positive predictions. Precision assessment is particularly relevant in applications where false positives are more critical than false negatives, as it ensures that the model is only making positive predictions when it is highly confident that they are accurate. The precision can be measured using Equation 3.3.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.3}$$

The F1 score assessment shows the model's performance better than simple accuracy measurement. Accuracy only calculates the part of positive predictions generated by the classification model; however, the F1 score considers both precision and recall. In other words, while accuracy may give a high value if there are many negatives in the dataset, the F1 score gives a better measure of how well the model performs with respect to positive instances. Therefore, the F1 score is useful when dealing with imbalanced datasets where positive occurrences are much smaller than negative occurrences. The F1 score measurement shows the overall effectiveness of a classification model in terms of correctly identifying positive cases while minimising false positives and false negatives. The F1 score of the detection model was measured using Equation 3.4.

$$\text{F1 score} = \frac{x\ (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{3.4}$$

3.8 Summary

The research methodology in this chapter demonstrated a thorough research design, research approach, data collection, and proposed detection approach to address the research questions and achieve the research aims and objectives. A wide range of tools, along with their description, was provided. The APS prototype and security testbed presented with their tools

and requirements the dataset collection, and exploratory data analysis presented. The development of the Hybrid Intrusion Detection System was discussed, and an assessment of the developed detection system with its performance matrices was presented in this chapter.

Chapter 4. Implementation and Testing of the Simulated APS

Chapter 4 presents the implementation of the APS prototype and its architectural components, cybersecurity testbed, dataset, and Exploratory Data Analysis (EDA). The APS prototype and cybersecurity testbed, both were implemented in a controlled environment to collect a dataset and subsequently, EDA presented. The APS prototype used a simulation approach, which consisted of the CGM sensor, insulin pump, and control algorithm. These architectural components of the APS device were implemented. The security attacks were executed using a command-and-control strategy and simulation approach.

The normal and malicious data packets were captured, stored, and processed to conduct the analysis. The dataset was collected in 2022 and was therefore labelled as 2022APS dataset. The dataset consisted of different data packets, and their values were either numerical or non-numerical with linear and nonlinear patterns, which made it difficult to draw conclusions in some cases. Therefore, the EDA technique was applied to solve these data analysis challenges and forecast when data were in the raw form, regardless of the linear and nonlinear patterns or numerical and non-numerical values. This chapter addresses the first and second research questions:

1. What vulnerabilities and attack methods exist within the Bluetooth and Internet of Things communication protocols in the APS?

2. What is the outcome on an APS in a simulated attack environment?

a. How can an appropriate dataset be developed to effectively study medjacking vulnerabilities in APS?

b. What exploratory data analysis technique can be used to explore data patterns of medjacking methods, including brute force, DDoS, MITM, and Trojan attacks?

4.1 Artificial Pancreas System Implementation

The APS prototype was designed on the philosophies of the object-oriented design methodology. The multithreading setting was enabled by simultaneous implementation over an IoMT platform using BLE communication and MQTT protocols. The APS prototype's design architecture was divided into three segments: the Diabetic Human Body (DHB) simulation including the CGM sensor, insulin pump, and communication component. These segments are demonstrated in the following subsections.

4.1.1 Diabetic Human Body

The DBH component was designed using a modified minimal model to enable the implementation of the APS prototype without a patient. A simulated DHB model was designed using the Raspberry Pi B. The DHB component was simulated to achieve APS functionality and the complementary response between the glucose level and insulin pump was considered the most significant element. In general, the manual mode uses commercial pumps to inject insulin according to the required doses of the insulin. The automatic mode primarily maintains a threshold of the patient's glycaemia, and depending on the measurement, insulin doses are injected, and a computerised automatic system records the patient's history. The DBH component was simulated into two operational principles:

1. The manual mode required a smartphone using BLE protocol to deliver insulin via an insulin pump. Once the BLE connection link was made between the DBH model and user screen, the diabetic human model sent the required blood glucose to the smartphone. When required doses were received on a user's smartphone, the user could manually choose to activate an insulin bolus to inject insulin via the insulin pump.

2. The automatic mode delivered the basal or bolus doses without a user communicating directly with the PID controller. The manual or automatic modes could be changed using a BLE protocol.

The DHB component was simulated to achieve APS functionality and the complementary response between the glucose level and insulin pump was considered the most significant element. The DBH component was designed to inject insulin using a Raspberry Pi B. Bergman's minimal model has been identified as best used among the research community [734, 735] to develop and analyse different methods for diabetic treatment. The modified minimal model in the current study used three differential equations to represent insulin kinetics and glucose levels:

$$\frac{dG}{dt} = -\big(q1 + X(t)\big)G(t) + q1\ G_b \tag{4.1}$$

$$\frac{dX}{dt} = -q_2X(t) + q_3(I(t) - I_b \tag{4.2}$$

$$\frac{dI}{d} = -q_6[G(t) - q_5 + t - q_4(I(t) - I_b \tag{4.3}$$

The parameters of the equations are defined in Table 4.1.

Table 4.1

Equation Parameters and Definition With Unit Parameters

Parameters	Definitions
$G(t)$	Glucose rate (mg/dL)
$X(t)$	Dynamic insulin outcome (min−1)
$I(t)$	Insulin rate and (μU/mL)
Gb	Glucose's basal rate (mg/dL)
Ib	Insulin's basal rate (μU/mL)
$q1$	Glucose clearance value with no insulin (min−1)

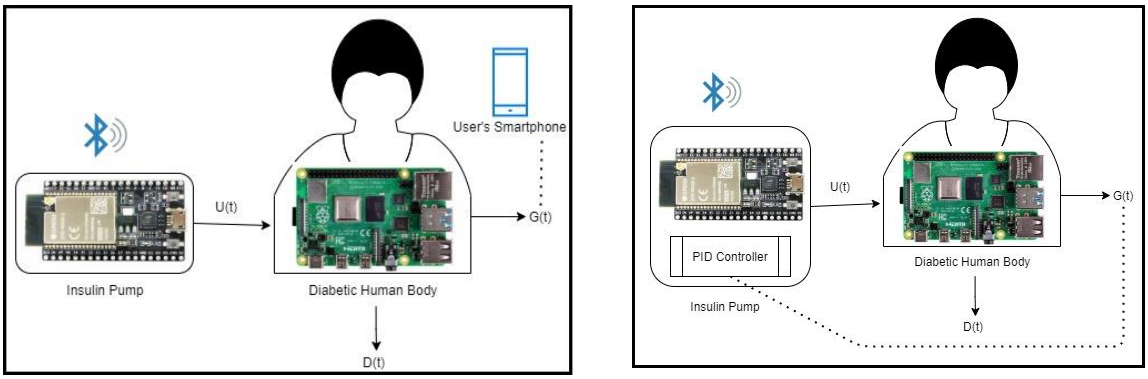
Parameters	Definitions
$q2$	Glucose active clearance value with insulin (min−1)
$q3$	Glucose interest by insulin (min−2 L/mU)
$q4$	Insulin deterioration value (min−1)
$q5$	Glucose target value (mg/dL)
$q6$	β cells value with glucose (mUdLmin−1/Lmg)

Note. µU/mL = microcount per millilitre.

Gb represents the glucose concentration basal plasma. In Equation 4.1, when the $G(t)$ falls below *Gb*, the glucose enters the plasma section. If the *Gb* is exceeded, then glucose will exit the plasma compartment. The enter and exit functions of glucose are comparable to the change between $G(t)$ and *Gb*. As an active insulin effect, $X(t)$ minimises the glucose plasma level and calls the enter function to insert insulin and convert it to energy if needed. When $I(t)$ exceed the insulin concentration basal plasma, the insulin enter function is activated as described in Equation 4.2. When *Ib* falls below *Gb*, the *Ib* exit function is activated. The manual mode continuously calculates glucose levels through the DBH simulation, as shown in Figure 4.1. $U(t)$ performed by a user in a manual mode indicates when the $G(t)$ value is sent via BLE protocol to a user's smartphone; once the $G(t)$ value is received, the user performs the insulin injection process. In automatic mode, the $G(t)$ value is sent to the PID controller to inject the insulin.

Figure 4.1

Manual and Automatic Mode



Note. Panel A (left) depicts the manual mode, the *G*(*t*) value collected, and the *U*(*t*). Panel B (right) depicts the automatic mode and the *G*(*t*) value sent to the PID controller to inject insulin.

Figure 4.1 presents the patient's glycaemia using manual and automatic modes. Both modes play a significant role in diabetic treatment. In general, the manual mode uses commercial pumps to inject insulin according to the required doses. The automatic mode primarily maintains a threshold of the patient's glycaemia, and depending on the measurement, insulin doses are injected. A computerised automatic system records the patient's history.

4.1.2 Insulin Pump

The insulin pump was designed with conventional functions. Insulin was calculated in units; 100 units are equal to 1 mL. In general, the insulin pump delivers insulin to the body in two forms. A bolus term is used for a meal or to correct a high glucose level. A basal term is used for defining basal value to inject insulin between meals and at night. The insulin pump was designed using a cost-efficient, off-the-shelf ESP32-WROOM microcontroller with BLE capability to connect to the glycaemic control and CGM sensor. Insulin pumps successfully provide precise insulin dosages while keeping blood sugar levels within a safe range. From a medical perspective, a non-diabetic individual typically has a blood glucose level of approximately 85 Milligrams Per Deciliter (mg/dL) during the fasting phase [736]. Sugar enters the bloodstream when the individual has a meal. As an outcome, the pancreas delivers the

insulin, and the glucose in the blood is absorbed by the muscles and used to create energy in the body. The pancreas's delivery of insulin controls glycaemia; valid natural insulin keeps glycaemia between 70 or 80 Milligrams Per Deciliter (mg/dL) and 120 mg/dL [737].

Individuals with Type 1 diabetes generally have a pancreas that suffers from a deficiency and loses its capacity to accurately make the required insulin. When a meal is consumed, the blood sugar level does not fall, which may lead to health concerns. When blood glucose levels surpass 180 mg/dL, the body begins to release glucose into the urine; a condition called hyperglycaemia, which is present when blood glucose levels are over 270 mg/dL. According to [738], the modern NovoRapid insulin pump was designed for rapid acting. For example, to normalise blood glucose levels, the insulin units were injected in a cycle phase; if a patient required 12 units of insulin, then the units would be inserted during the cycle phases. Each cycle injects a small amount of insulin, and from its design, this study assumed 0.5 units of insulin injected in five–minute intervals to mimic the real function of the insulin pump. As a result, insulin can be injected 24 times. This assumption was used in the current study to capture event logs.

4.2 Architectural Components

The BLE is characteristically employed in the IoMT network to establish communication between devices, collect attribute values from the peripheral device, and write attribute values to the central device. The communication process between the peripheral and central devices is organised into four phases: advertisement, connection, pairing, and accessing.

The data exchange between the CGM sensor, insulin pump was made using the BLE communication protocol. As this research did not involve an actual patient, a simulation approach was applied to demonstrate normal and attack patterns. The doses of the insulin and CGM sensor were assumed as normal and high, and values are described in Table 4.2. The main objective of this simulation was to implement the communication protocol and collect

inbound and outbound data from the insulin pump and CGM sensor. The doses assumption was made from a study by Mayo Foundation, according to the Mayo Foundation for Medical Education and Research, a 140 mg/dL glucose level is considered normal while a 140–199 mg/dL glucose level shows prediabetes [739].

Table 4.2

Glucose Level and Insulin Pump Values

Glucose level		**Insulin pump value in units**
Value in mmol/L	**Value in mg/dL**	
7.22	130	0
8.3	160	12
9.55	172	12
9.72	175	12
10.61	191	14
12.22	220	30

Table 4.2 presents the glucose and insulin values, which were assumed in the simulation; these values frequently moved between the APS components using the BLE communication protocol. In a real setting, if the CGM value is 8.3 mmol/L then the APS device would inject 12 units of insulin into the diabetic patient. All APS components are designed to treat diabetic patients by providing the required insulin doses, but their dose values vary depending on each patient.

Another IoMT communication, MQTT, is a de facto communication protocol that enables sensors and applications to process information bi-directionally between interconnected devices over the internet. Message queuing involves sending, storing, and retrieving messages between a resource-constrained device or application. This process allows

asynchronous communication, architectural decoupling, and flexibility by not synchronising the source and destination. Message queuing improves distributed system dependability, scalability, and fault tolerance by separating source and destination components, which buffers and retains messages to minimise loss during system or network failures. Telemetry transport has become a key factor in improving message queuing data delivery. The MQTT is renowned for its lightweight feature that handles communication on resource-constrained devices. The MQTT optimises network capacity by efficiently delivering messages through its publish-subscribe architecture. A publisher transmits data to a broker, and the broker reliably sends data to the subscribers. MQTT consists of three major components: the publisher known as the MQTT client, the broker known as the MQTT server, and the subscriber known as the MQTT client. The broker is responsible for handling and distributing messages, and it is subsequently the entity that receives and consumes messages. MQTT messaging implementation was achieved by applying MQTT broker and client applications. The Mosquitto MQTT broker was implemented using Raspberry Pi 4 Model B and ESP32 WROVER B with 8 MB of RAM, which supported both wi-fi and BLE functionalities. An integrated development environment was used to implement the MQTT protocol, and an open-source Espressif IoT Development Framework was used to achieve an integrated development environment, as shown in Figure 4.2.

Figure 4.2

Message Queuing Telemetry Transport Client Libraries Using the Espressif Internet of Things Development Framework

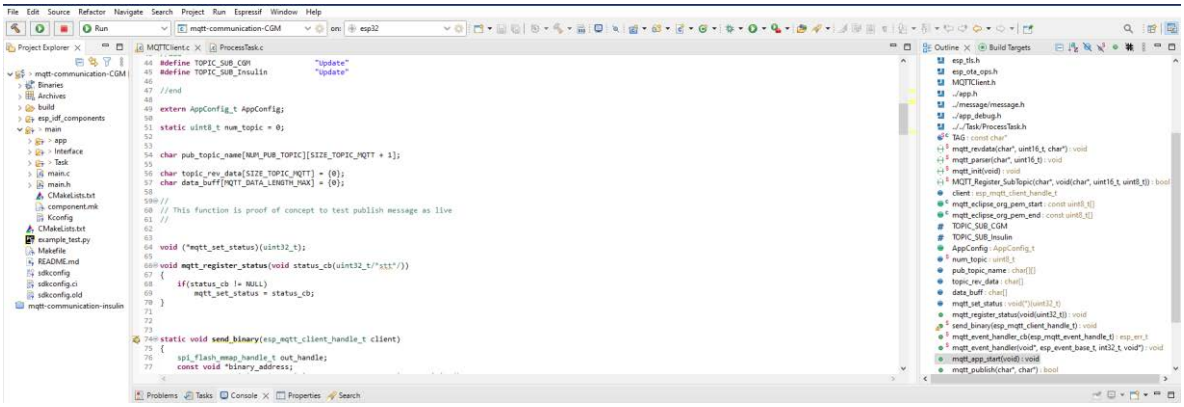


Figure 4.2 presents the Espressif IoT Development Framework, a suite of software development tools and libraries for creating IoT applications on the ESP32 microcontroller platform. IoMT applications that use MQTT brokers can use the networking components' MQTT capabilities. The MQTT broker library was employed to connect to the Espressif IoT Development Framework MQTT brokers. The MQTT client used the publish or subscribe to topics action.

4.3 APS and Cybersecurity Testbed Functionality

The dataset obtained from the APS and the cybersecurity testbed encompassed a series of diverse modular steps. The ESP32 modelled CGM data, transmitting CGM readings to the DBH. The Raspberry Pi, functioning as the APS controller, executed these readings through the application of the Bergman minimal model to calculate insulin dosages, thereby generating the APS communication. These transactions were recorded to establish normal traffic patterns. The communication ports, including Port 80, Port 1883, and Port 8883 traffic, were recorded during normal APS operations without simulated attacks; these patterns were converted into

CSV files and labelled as 0, indicating normal patterns. This normal data was required for the development of H-IDS to train ML and DL models; this study used a modular design approach.

In the process of the attack simulations, a modular design for the brute force attack was employed, and a Kali Linux VM operating the Metasploit Framework was employed to engage the MQTT broker situated on the Raspberry Pi. Metasploit produced a series of repeated authentication attempts directed at the APS, while Wireshark recorded the ensuing login failures; these patterns were converted into CSV files and labelled as 1, indicating attack patterns. The DDoS attack, executed via hping3, generated traffic directed at the IP address of the Raspberry Pi, while Wireshark captured the SYN flood patterns and a corresponding decrease in system response times, which were representative of the DDoS attack. These patterns were converted into CSV files and labelled as 1, indicating attack patterns. In the context of the MITM attack, Ettercap assisted in the interception of APS communications between the ESP32 and Raspberry Pi, while Wireshark captured the packets during transmission; these patterns were converted into CSV files and labelled as 1, indicating attack patterns. Finally, in the context of the trojan attack, the Metasploit and Curl tools were used to implement a payload, while Wireshark captured the connections between the trojan machine and APS, illuminating unusual outbound traffic patterns. These patterns were converted into CSV file and labelled as 1, indicating attack patterns.

This research was conducted with the primary aim of mitigating medjacking attacks and formulating the detection system for the APS. However, various methods might be utilised as discussed in Sections 2.3, 2.4 and 2.6 presented in Chapter Two. This research serves as a proof of concept, aiming to assess the feasibility of ML and DL approaches for efficient intrusion detection, thereby this study established a foundation for enhancing security within the APS.

4.4 Cybersecurity Testbed

Medical devices, and specifically APS devices, have accelerated the digitalisation method of automating accurate medicine doses, measurement, treatment, etc., and establish the link between healthcare, IT professionals (e.g., online or application services) and third parties (e.g., service providers). The digitisation of medical devices has caused cyberattacks, therefore, the cybersecurity research community focused on investigation of security approaches including attack detection and monitoring. However, adapting specific defence methods and their quality of performance is critical due to the cost and privacy of factory-made devices. The lack of evidence-based security solutions, and the limited data available for researchers to develop evidence-based solutions is another factor that has emerged. Therefore, to mitigate cybersecurity attacks, theorical, experimental and mathematical operations (e.g., simulation, emulation, prototype, and testbed operations) have been included in this research to predict or identify potential solutions.

A testbed is a setting platform that uses virtual machines, simulation, and command-and-control strategies. In general, tests are performed in a controlled environment without engaging a real system for multiple operations. The primary objective of developing the testbed in this study was to collect attack data and compare results with the normal data to conduct a quantitative analysis and develop a medjacking detection solution. Developing the testbed was complex and challenging due to mainstream devices being unavailable for multiple time, cost, and ethical reasons. Adequate tools, attack codes, and attack samples also complicated the development. Analysis metrics and feature selection were challenging for visualising and forecasting unique signal changes using the normal and attack datasets. Different tools and infiltration techniques were therefore used, and these activities were not user-friendly because they had distinct processes. Consequently, all attacks in one graphical user interface were nearly impossible.

The cybersecurity testbed in this study featured a modular design, and distinct modules were employed to perform functions, the model performs a specific task and moved to another process utilising a different module. Modular design in literature refers to the integration or separation of numerous models or pieces for development or analysis [740-743]. Alternative terms for modular design include modular architecture [744] and modular programming [745-747]. These modules are designed to be easily swapped out, to keep internal details contained, to encourage autonomy, to make things (e.g., security attacks exemplification) easier to reuse, to assess performance, and to make maintenance easier. This approach enables flexibility, effective parallel development, and collaboration across several modules. The module design for this study's cybersecurity testbed consisted of the security attacks' configuration and execution using different tools, libraries, and attack samples. Different security attacks are described in the following subsections.

4.4.1 Brute Force Attack

Guessing end-user credentials is the foundation of a brute force attack. As discussed in Section 2.3, brute force attacks attempt to find valid credentials and use them to access a network for password hashes and tokens. After accessing the service with a victim's device, the attacker can target other devices interconnected with the same credentials. Most IoT devices choose their passwords, such as the first two letters representing the manufacturer's name, two letters representing the year or series, and the other two letters allocated to the user to set; hence, a password combination can be xx2210, which is easily guessable. Therefore, the attacker abuses the devices with brute force attacks.

In the current literature, the research community employed Metasploit to use exploits, payloads, and other methods. to contribute new findings [568, 682, 748-750]. Cybersecurity experts continually contribute new findings as open sources on Metasploit to understand and evaluate security vulnerabilities, including penetration testing [751-753]. Furthermore, the

MQTT-PWN framework was partially used to understand brute force. This tool is an open-source framework designed to study brute force attacks and other cyber-attacks, specifically for research endeavours. In [754], the researchers stated that MQTT-PWN constitutes a toolkit for penetration testing. Researchers may use this tool to gain the basic knowledge necessary for executing various attacks; the documentation related to this tool can be found in [755].

To exemplify the brute force attack, this study targeted terminal network, the SSH protocol network activities using Metasploit, and the Kali Linux operating system in a controlled environment using a virtual machine. In this part of the study, an interface of Metasploit, msfconsole, was used to exemplify the brute force attack in a controlled setting, as shown in Figure 4.3.

Figure 4.3

The Interface of the Metasploit Using Kali Linux Operating System

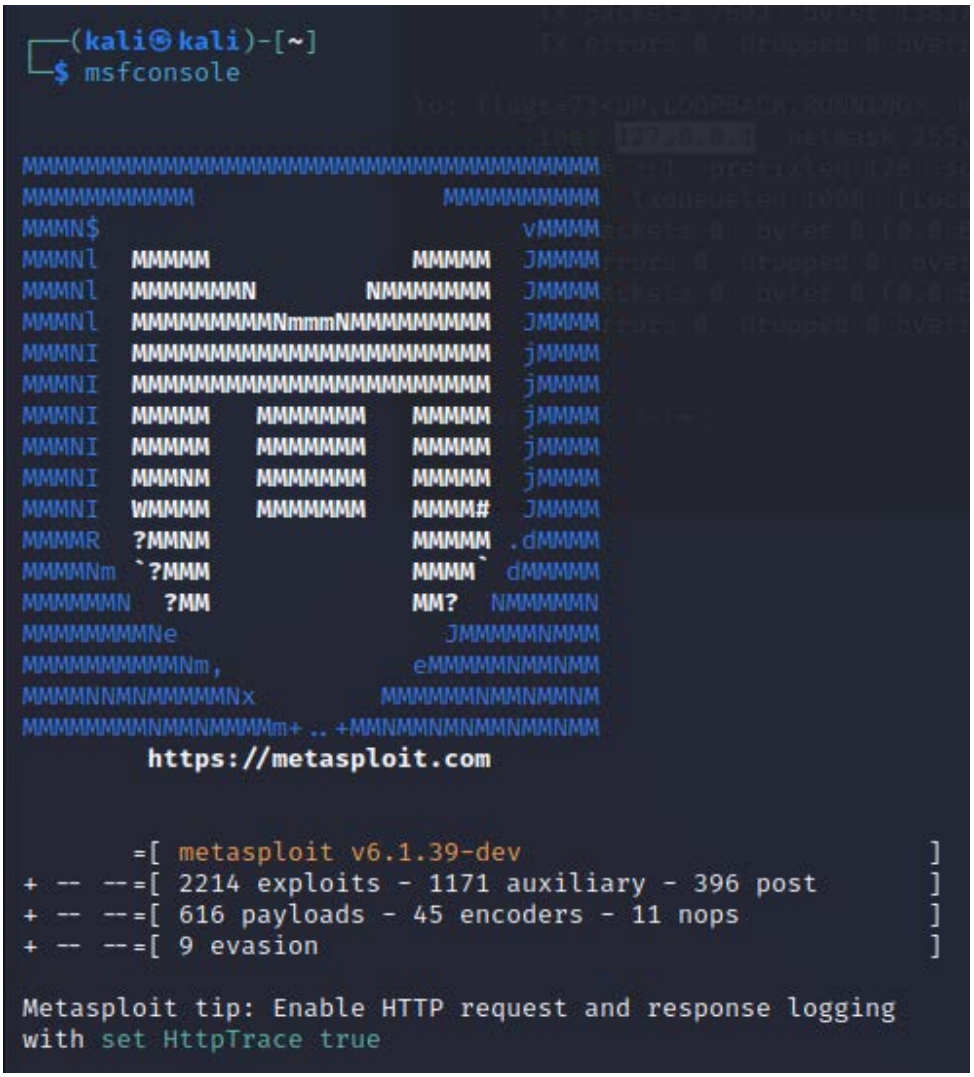


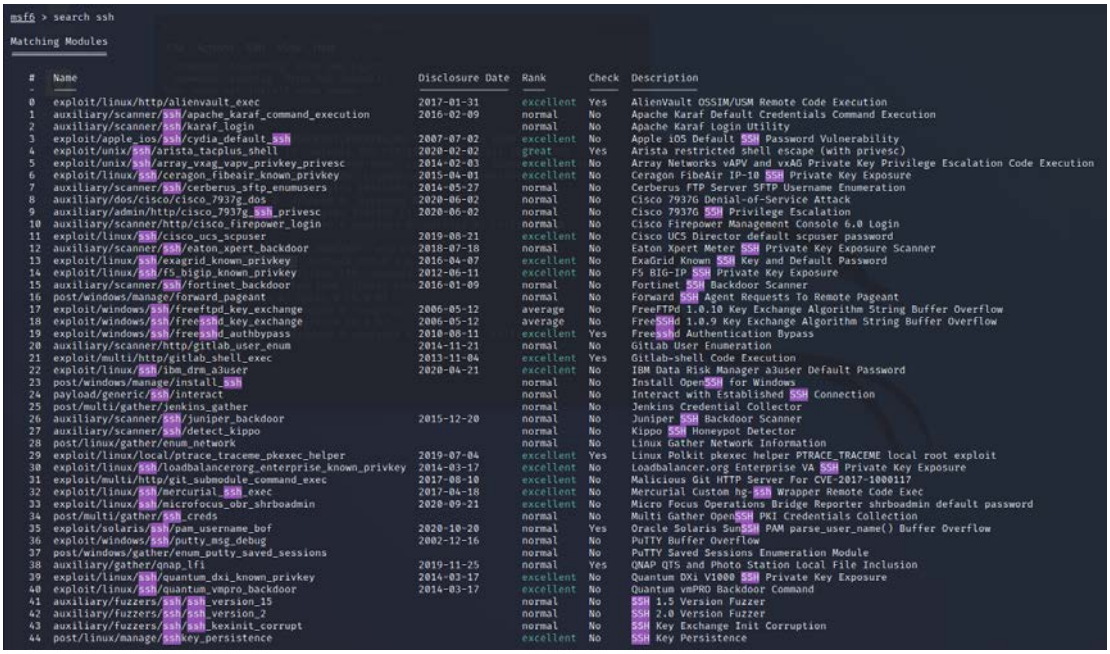
Figure 4.3 shows Metasploit's interface, which has 2214 exploits, 1171 auxiliaries, 396 posts, 616 payloads, 45 encoders, 11 no operations, and 9 evasions. These interface features use Metasploitable to execute various security attacks and vulnerabilities. The Metasploit's exploit is a series of scripts or commands that use a bug or vulnerability to victimise the devices. Auxiliaries are used in pen testing, including fuzzing, scanning, and sniffing. Posts commands can be executed after exploits. After a device has been compromised, posts command can be utilised to carry out more acts of exploitation. Payloads create roots (i.e., entry points) to enter a device by using various types of malware, viruses, or bugs. Once a payload fits into the device,

the encoders can use further commands to exploit the device by taking advantage of the device's vulnerabilities. No operations deactivate the device a single clock cycle. Evasions use commands to exclude a system's defence mechanisms, such as antivirus applications.

The Metasploit framework was used for executing the brute force attack, while an Nmap service was utilised to identify the open ports. Terminal network and SSH ports were the main features of the targets. Figure 4.4 depicts the lists of the SSH exploits stored in the Metasploit database. As the APS prototype's subsystems operated on a Windows operating system, the SSH exploits were specifically linked to the Windows operating system.

Figure 4.4

The List of all Secure Shell Protocol



```
msf6 > search ssh

Matching Modules

   #  Name                                                      Disclosure Date  Rank       Check  Description
   -  ----                                                      ---------------  ----       -----  -----------
   0  exploit/linux/http/alienvault_exec                        2017-01-31       excellent  Yes    AlienVault OSSIM/USM Remote Code Execution
   1  auxiliary/scanner/ssh/apache_karaf_command_execution      2016-02-09       normal     No     Apache Karaf Default Credentials Command Execution
   2  auxiliary/scanner/ssh/karaf_login                                          normal     No     Apache Karaf Login Utility
   3  exploit/apple_ios/ssh/cydia_default_ssh                   2007-07-02       excellent  No     Apple iOS Default SSH Password Vulnerability
   4  exploit/unix/ssh/arista_tacplus_shell                     2020-02-02       great      Yes    Arista restricted shell escape (with privesc)
   5  exploit/unix/ssh/array_vxag_vapv_privkey_privesc          2014-02-03       excellent  No     Array Networks vAPV and vxAG Private Key Privilege Escalation Code Execution
   6  exploit/linux/ssh/ceragon_fibeair_known_privkey           2015-04-01       excellent  No     Ceragon FibeAir IP-10 SSH Private Key Exposure
   7  auxiliary/scanner/ssh/cerberus_sftp_enumusers             2014-05-27       normal     No     Cerberus FTP Server SFTP Username Enumeration
   8  auxiliary/dos/cisco/cisco_7937g_dos                       2020-06-02       normal     No     Cisco 7937G Denial-of-Service Attack
   9  auxiliary/admin/http/cisco_7937g_ssh_privesc              2020-06-02       normal     No     Cisco 7937G SSH Privilege Escalation
   10 auxiliary/scanner/http/cisco_firepower_login                               normal     No     Cisco Firepower Management Console 6.0 Login
   11 exploit/linux/ssh/cisco_ucs_scpuser                       2019-08-21       excellent  No     Cisco UCS Director default scpuser password
   12 auxiliary/scanner/ssh/eaton_xpert_backdoor                2018-07-18       normal     No     Eaton Xpert Meter SSH Private Key Exposure Scanner
   13 exploit/linux/ssh/exagrid_known_privkey                   2016-04-07       excellent  No     ExaGrid Known SSH Key and Default Password
   14 exploit/linux/ssh/f5_bigip_known_privkey                  2012-06-11       excellent  No     F5 BIG-IP SSH Private Key Exposure
   15 auxiliary/scanner/ssh/fortinet_backdoor                   2016-01-09       normal     No     Fortinet SSH Backdoor Scanner
   16 post/windows/manage/forward_pageant                                        normal     No     Forward SSH Agent Requests To Remote Pageant
   17 exploit/windows/ssh/freeftpd_key_exchange                 2006-05-12       average    No     FreeFTPd 1.0.10 Key Exchange Algorithm String Buffer Overflow
   18 exploit/windows/ssh/freesshd_key_exchange                 2006-05-12       average    No     FreeSSHd 1.0.9 Key Exchange Algorithm String Buffer Overflow
   19 exploit/windows/ssh/freesshd_authbypass                   2010-08-11       excellent  Yes    FreeSSHd Authentication Bypass
   20 auxiliary/scanner/http/gitlab_user_enum                   2014-11-21       normal     No     GitLab User Enumeration
   21 exploit/multi/http/gitlab_shell_exec                      2013-11-04       excellent  Yes    Gitlab-shell Code Execution
   22 exploit/linux/ssh/ibm_drm_a3user                          2020-04-21       excellent  No     IBM Data Risk Manager a3user Default Password
   23 post/windows/manage/install_ssh                                            normal     No     Install OpenSSH for Windows
   24 payload/generic/ssh/interact                                               normal     No     Interact with Established SSH Connection
   25 post/multi/gather/jenkins_gather                                           normal     No     Jenkins Credential Collector
   26 auxiliary/scanner/ssh/juniper_backdoor                    2015-12-20       normal     No     Juniper SSH Backdoor Scanner
   27 auxiliary/scanner/ssh/detect_kippo                                         normal     No     Kippo SSH Honeypot Detector
   28 post/linux/gather/enum_network                                             normal     No     Linux Gather Network Information
   29 exploit/linux/local/ptrace_traceme_pkexec_helper          2019-07-04       excellent  Yes    Linux Polkit pkexec helper PTRACE_TRACEME local root exploit
   30 exploit/linux/ssh/loadbalancerorg_enterprise_known_privkey 2014-03-17      excellent  No     Loadbalancer.org Enterprise VA SSH Private Key Exposure
   31 exploit/multi/http/git_submodule_command_exec             2017-08-10       excellent  No     Malicious Git HTTP Server For CVE-2017-1000117
   32 exploit/linux/ssh/mercurial_ssh_exec                      2017-04-18       excellent  No     Mercurial Custom hg-ssh Wrapper Remote Code Exec
   33 exploit/linux/ssh/microfocus_obr_shrboadmin               2020-09-21       excellent  No     Micro Focus Operations Bridge Reporter shrboadmin default password
   34 post/multi/gather/ssh_creds                                                normal     No     Multi Gather OpenSSH PKI Credentials Collection
   35 exploit/solaris/ssh/pam_username_bof                      2020-10-20       normal     Yes    Oracle Solaris SunSSH PAM parse_user_name() Buffer Overflow
   36 exploit/windows/ssh/putty_msg_debug                       2002-12-16       normal     No     PuTTY Buffer Overflow
   37 post/windows/gather/enum_putty_saved_sessions                              normal     No     PuTTY Saved Sessions Enumeration Module
   38 auxiliary/gather/qnap_lfi                                 2019-11-25       normal     Yes    QNAP QTS and Photo Station Local File Inclusion
   39 exploit/linux/ssh/quantum_dxi_known_privkey               2014-03-17       excellent  No     Quantum DXi V1000 SSH Private Key Exposure
   40 exploit/linux/ssh/quantum_vmpro_backdoor                  2014-03-17       excellent  No     Quantum vmPRO Backdoor Command
   41 auxiliary/fuzzers/ssh/ssh_version_15                                       normal     No     SSH 1.5 Version Fuzzer
   42 auxiliary/fuzzers/ssh/ssh_version_2                                        normal     No     SSH 2.0 Version Fuzzer
   43 auxiliary/fuzzers/ssh/ssh_kexinit_corrupt                                  normal     No     SSH Key Exchange Init Corruption
   44 post/linux/manage/sshkey_persistence                                       excellent  No     SSH Key Persistence
```

Figure 4.4 depicts Metasploit's list of all SSH exploits with their disclosure dates, which indicate the dates, ranks, checks, and descriptions. Dates indicate the initial dates of disclosure, ranks indicate normal or excellent feasibility, and descriptions indicate further action. These descriptions identified a series of vulnerabilities, including exploits. Multiple attacks could be executed using a series of scripts or commands, and the user guide was provided in the

Metasploit for mastering this tool. The brute force attack was launched in a controlled environment to demonstrate its ability to infiltrate the APS prototype.

4.4.2 Distributed Denial-of-Service Attack

A DDoS attack can be executed using hping3, which is a robust and open-source command-line tool that exemplifies DDoS attacks. Researchers have demonstrated DDoS attacks using hping3 [756]. In another study, intrusion detection was proposed, and the system used hping3 to execute DoS attacks [757]. Hping3 could generate and transmit packets to a targeted system, which is one of its primary functions. The user may customise the number of hping3 packets to TCP, IP, or UDP ports and flags. The hping3 can also scan ports to reveal which services are running on a target and which ports are open for the purpose of security assessment.

In this study, a DDoS attack was implemented using hping3 and Kali Linux with a virtual machine as shown in Figures 4.5 and 4.6. Three steps were involved. First, the command-line script flooded the APS's components with synchronise (SYN) requests. Second, the SYN–acknowledgement (SYN–ACK) was sent from a central machine response. Third, ACK messages were delivered to establish a connection.

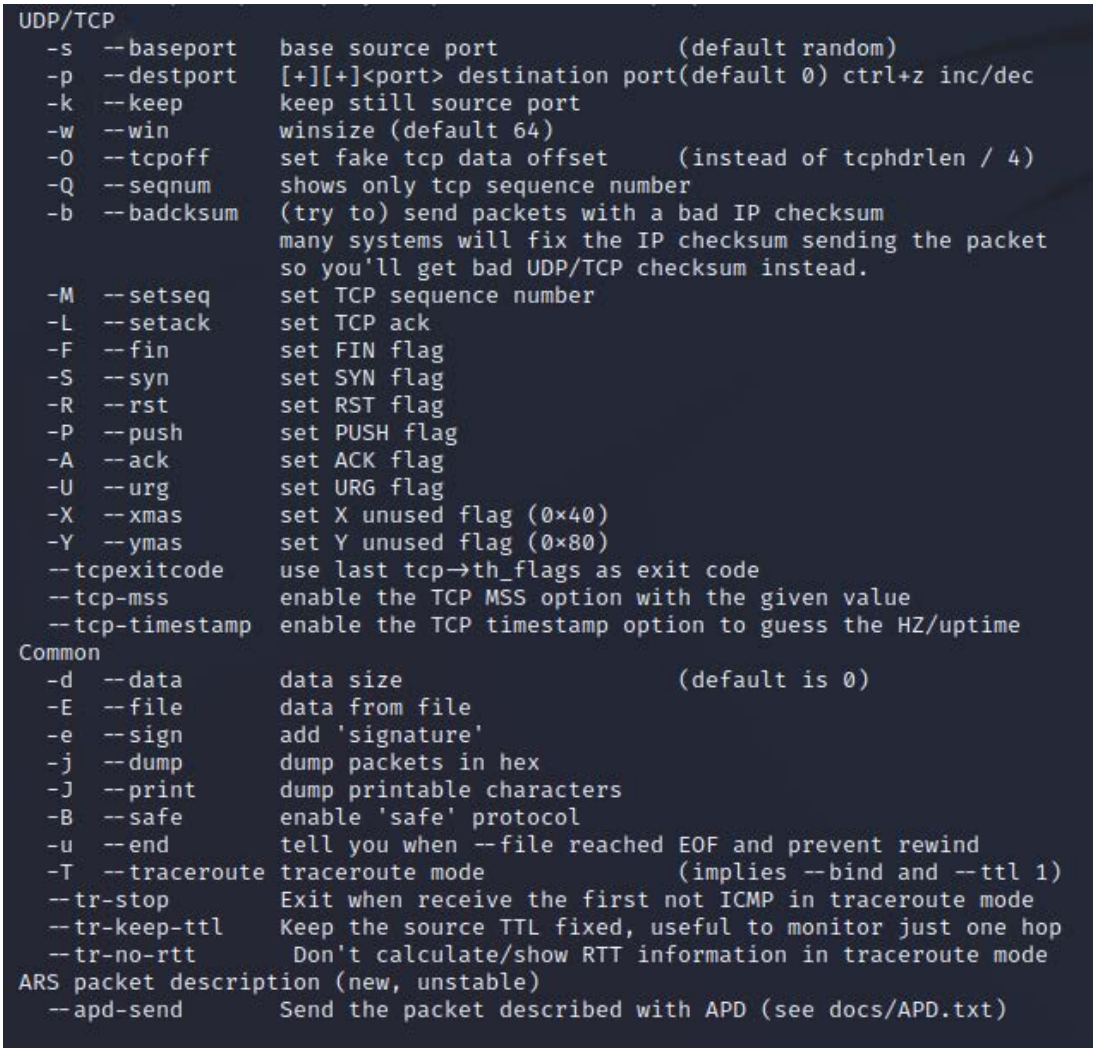
Figure 4.5

The Hping3 Commands for the Distributed Denial-of-Service Attack Using Kali Linux With the Virtual Machine

Note. The commands in this figure are continued in Figure 4.6.

Figure 4.6

The Remaining Hping3 Commands for the Distributed Denial-of-Service Attack Using Kali Linux on the Virtual Machine



Note. The initial commands are presented in Figure 4.6.

Figures 4.5, and 4.6 show how the different commands of the hping3 tool can be used for generating DDoS attacks. Hping3 was mainly configured for the creation of packets using a command line execution. The commands consist of S, P, flood, and rand-source. S was used to set the SYN flag, P was used for setting the port, flood was used for frequently sending requests, and rand-source sent packets to the APS's subsystems. The DDoS attacks used hping3, which may randomise signal origin, to bypass security mechanisms that could damage a single device. DDoS could delay response and prevent server interface use.

4.4.3 Man-in-the-Middle Attack

An MITM attack was implemented using an open-source tool named Ettercap to sniff and spoof DNS. The DNS provides the directory for communicating with the desired destination. If a user intends to call someone, then the user must follow the steps to make a call, such as looking up the phone directory and pressing the call button. To understand how the DNS works, if the APS prototype's subsystems intend to connect to the internet, the subsystems automatically redirect to the programmed location already stored in a kernel, or manually provide the location of their desired destination. The kernel is a facilitator between the hardware and software that provides interaction using the programmed script.

In the current literature, the Ettercap is described as a multipurpose tool that is supported by different operating systems for security auditing and protocol analysis [758]. Ettercap is commonly used to exemplify MITM attacks; it enables users to capture the communication between devices to acquire and transmit any type of data, including passwords and other sensitive information. Features supported by Ettercap include live connection sniffing, content filtering, active and passive protocol examination, and capabilities to launch an ARP poisoning attack, which redirects traffic through the attacker's machine. Cybersecurity professionals and academic researchers use Ettercap for various reconnaissance and attack features [427, 758, 759].

To execute an MITM attack, this study targeted DNS and ARP protocols using Ettercap in a controlled environment with a virtual machine. In the current study, the assumed attacker's objective was to intercept communication and continuously enable eavesdropping. The primary aim of using the Ettercap was to enter promiscuous mode and manipulate the target device through DNS spoofing and ARP poisoning. Promiscuous mode refers to the state in which all packets, regardless of whether they have a designated destination for the medium access control, are forwarded to the kernel for processing.

The simulated attacker stored the APS prototype's subsystem credentials through an eavesdrop DNS port and forged responses to the device. The attacker applied DNS cache poisoning with fabricated data and eliminated the cache from the DNS server. The MITM attacks used Ettercap with all operations performed using the kernel command line, and some commands were required for executing the attack. The Ettercap unique identifier, Ettercap group identification, and different parameters are shown in Figure 4.7.

Figure 4.7

The Ettercap Configuration Parameters

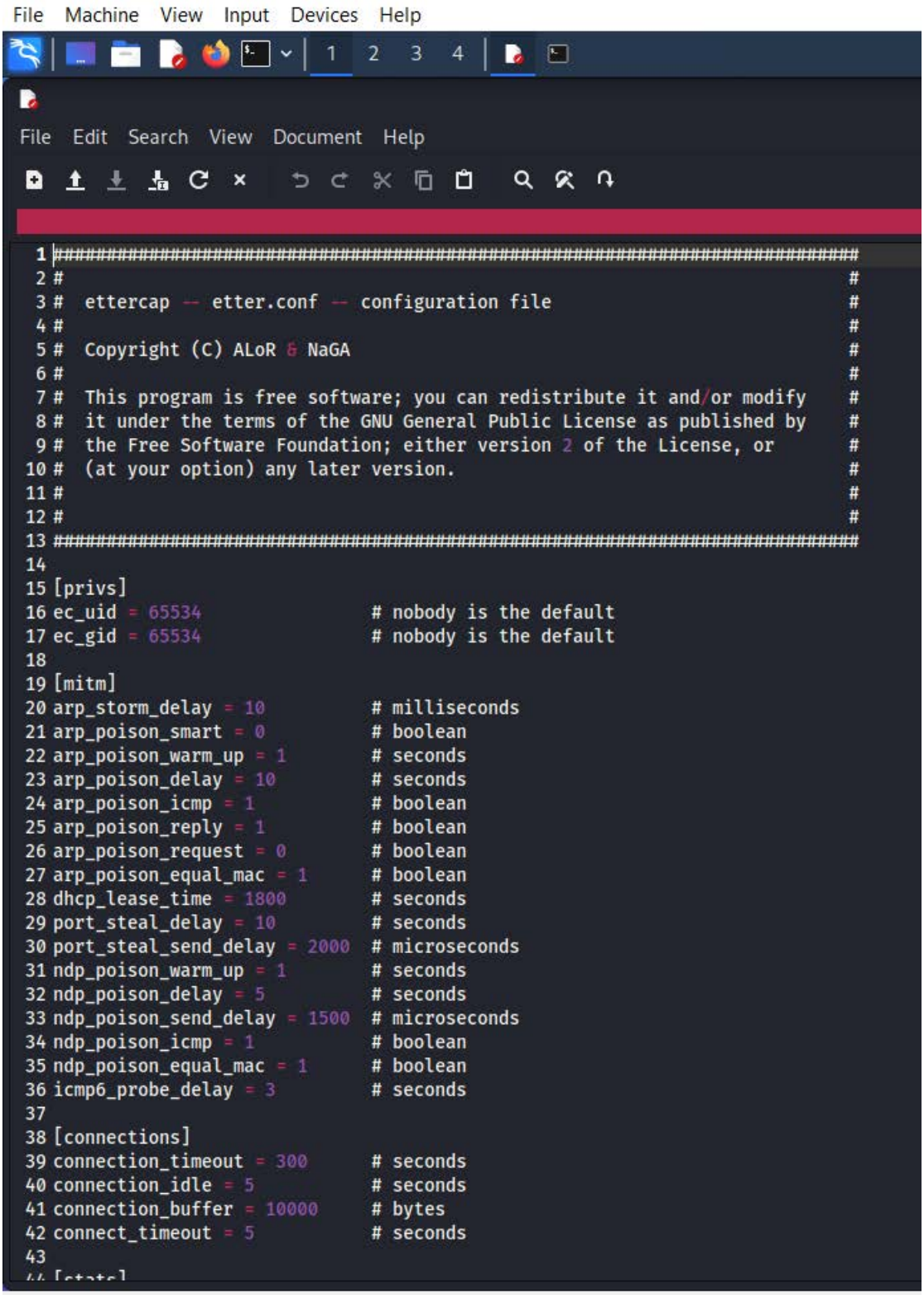


Figure 4.7 shows different parameters such as arp_storm_delay with milliseconds, arp_poison_smart with boolen, arp_poison_warm_up with seconds, and arp_poison_delay with seconds. These parameters were programmed to launch the MITM attack. The Linux operating system was used in this attack, based on operating system requirements, and the SSL segmentation was required to enable redirection as shown in Figure 4.8.

Figure 4.8

The Linux Operating System for the Secure Sockets Layer Segmentation

```
163
164 ######################################
165 #       redir_command_on/off
166 ######################################
167 # you must provide a valid script for your operating system in order to have
168 # the SSL dissection available
169 # note that the cleanup script is executed without enough privileges (because
170 # they are dropped on startup). so you have to either: provide a setuid program
171 # or set the ec_uid to 0, in order to be sure the cleanup script will be
172 # executed properly
173 # NOTE: the script must fit into one line with a maximum of 255 characters
174
175 #----------------
176 #     Linux
177 #----------------
178
179    redir_command_on = "iptables -t nat -A PREROUTING -i %iface -p tcp -d %destination --dport %port -j REDIRECT --to-port %rport"
180    redir_command_off = "iptables -t nat -D PREROUTING -i %iface -p tcp -d %destination --dport %port -j REDIRECT --to-port %rport"
181
182 # pendant for IPv6 - Note that you need iptables v1.4.16 or newer to use IPv6 redirect
183    redir6_command_on = "ip6tables -t nat -A PREROUTING -i %iface -p tcp -d %destination --dport %port -j REDIRECT --to-port %rport"
184    redir6_command_off = "ip6tables -t nat -D PREROUTING -i %iface -p tcp -d %destination --dport %port -j REDIRECT --to-port %rport"
185
```

Figure 4.8 presents some parameters to redirect the SSL dissection, such as the iface parameter destination port, to define the rule for the interface. The source port communication was assigned with the redirect command, and the response port was aligned with Ettercap, which eavesdropped continuously. The DNS parameters were assigned in the kernel line to redirect legitimate nodes to a redirected destination; multiple operations could then be performed based on the attacker's intention, as shown in Figure 4.9.

Figure 4.9

The Kernel Line to Redirect Legitimate Nodes to a Redirected Destination

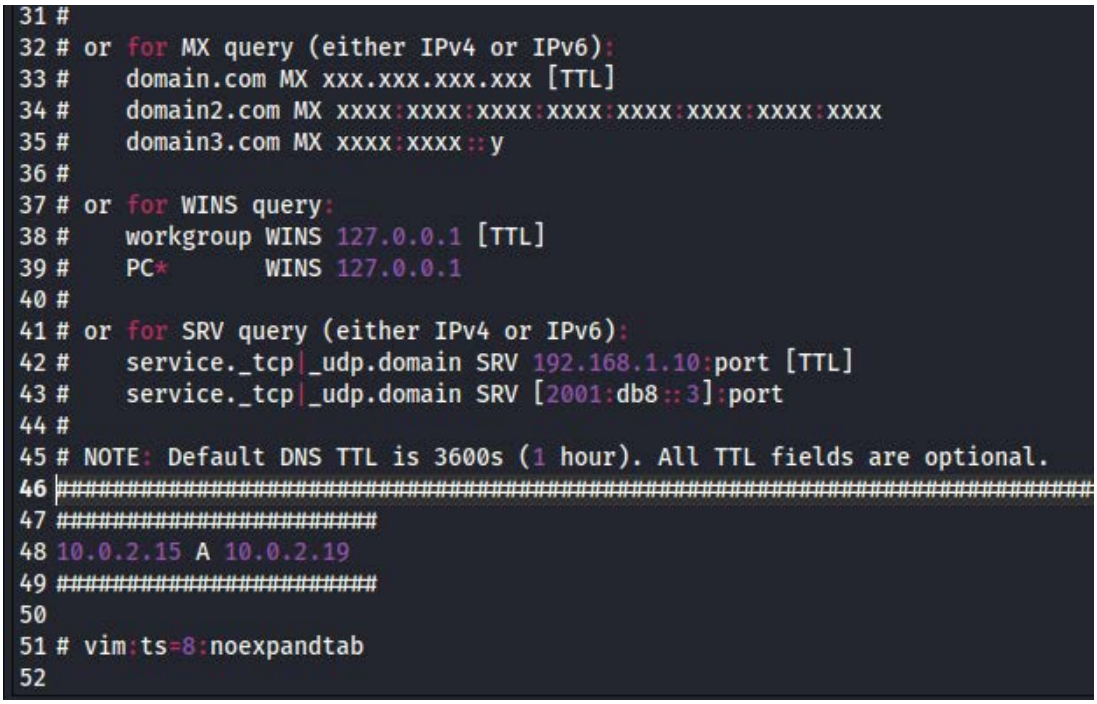


Figure 4.9 shows the source port communication that was redirected to another destination at the bottom line in the command. The purpose of this command was only to exemplify the MITM attack in a controlled environment while using the command-and-control environment to demonstrate how the MITM attack could infiltrate the APS's subsystems.

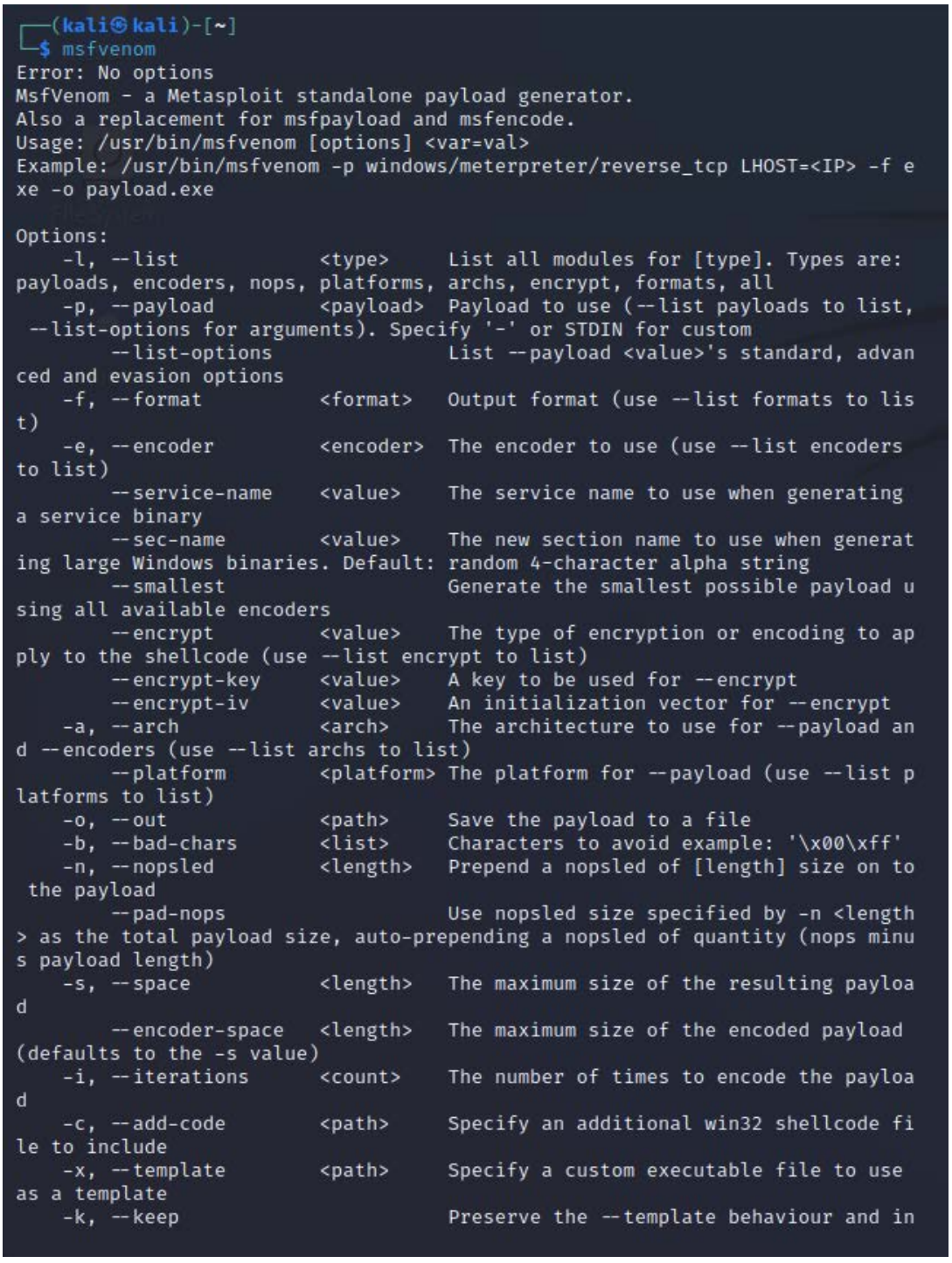
4.4.4 Trojan Attack

A Trojan attack, also called a Trojan horse, is a malware code written to harm the system [450]. Malware code can be an attachment, utility, or any other item that the victim may access or load. The attacker transmits the malware code using social engineering to mislead victims into downloading and executing Trojans [451, 452]. If the victim becomes entrapped in the execution of the Trojan, the malware will anonymously install threatening features in the background and, therefore, establish a backdoor [760]. After a backdoor is established, the attacker remotely accesses the hacked system. An intruder can steal credentials or manipulate system data via the backdoor. In this study, Metasploit, msfvenom, and curl tools were used

for the backdoor script, and both tools were executed in the command line. The Metasploit was executed in a terminal command msfconsole. The APS's subsystems were chosen for a Trojan attack, and the msfvenom tool was used in the command line to generate the malicious payload (see Figure 4.10).

Figure 4.10

The msfvenom Used to Execute Backdoor Unauthorised Access for a Trojan Attack



```
┌──(kali㉿kali)-[~]
└─$ msfvenom
Error: No options
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f e
xe -o payload.exe

Options:
    -l, --list            <type>     List all modules for [type]. Types are:
payloads, encoders, nops, platforms, archs, encrypt, formats, all
    -p, --payload         <payload>  Payload to use (--list payloads to list,
 --list-options for arguments). Specify '-' or STDIN for custom
        --list-options               List --payload <value>'s standard, advan
ced and evasion options
    -f, --format          <format>   Output format (use --list formats to lis
t)
    -e, --encoder         <encoder>  The encoder to use (use --list encoders
to list)
        --service-name    <value>    The service name to use when generating
a service binary
        --sec-name        <value>    The new section name to use when generat
ing large Windows binaries. Default: random 4-character alpha string
        --smallest                   Generate the smallest possible payload u
sing all available encoders
        --encrypt         <value>    The type of encryption or encoding to ap
ply to the shellcode (use --list encrypt to list)
        --encrypt-key     <value>    A key to be used for --encrypt
        --encrypt-iv      <value>    An initialization vector for --encrypt
    -a, --arch            <arch>     The architecture to use for --payload an
d --encoders (use --list archs to list)
        --platform        <platform> The platform for --payload (use --list p
latforms to list)
    -o, --out             <path>     Save the payload to a file
    -b, --bad-chars       <list>     Characters to avoid example: '\x00\xff'
    -n, --nopsled         <length>   Prepend a nopsled of [length] size on to
 the payload
        --pad-nops                   Use nopsled size specified by -n <length
> as the total payload size, auto-prepending a nopsled of quantity (nops minu
s payload length)
    -s, --space           <length>   The maximum size of the resulting payloa
d
        --encoder-space   <length>   The maximum size of the encoded payload
(defaults to the -s value)
    -i, --iterations      <count>    The number of times to encode the payloa
d
    -c, --add-code        <path>     Specify an additional win32 shellcode fi
le to include
    -x, --template        <path>     Specify a custom executable file to use
as a template
    -k, --keep                       Preserve the --template behaviour and in
```

Figure 4.10 depicts the msfvenom, which includes a payload generator and encoder. Msfvenom worked smoothly with other Metasploit components. These tools allow payloads into exploit modules. Msfvenom generated a payload by a command line that used Kali Linux

to obtain unauthorised access to the APS subsystems. Subsequently, the payload was accessed through a direct URL. After the payload was hosted, a curl command was created to download the payload into the APS subsystems. Curl command retrieved the payload from the hosted location and saved it locally on the APS subsystems. Once the payload was executed on these subsystems, it established a connection back to the attacker system. At this stage of backdoor establishment, the simulated intruder could further exploit the hacked machine or incorporate it into an additional cybersecurity compromise.

4.5 Dataset Collection Unit

Units of information, known as data packets, are sent between devices and the network infrastructure. Data packets consist of sensor readings, control signals, status updates, or other pertinent data produced by devices during their interactions with other devices. Information contained in the packets can be systematically gathered, recorded, and processed for multiple purposes, such as dataset visualisation, forecasting, and drawing conclusions through cybersecurity analysis on targeted variables in an established system. Data collecting for cybersecurity analysis includes collecting data from both normal functioning and security events.

This research utilised a modular design within the cybersecurity testbed that enabled systematic and regulated data collection for the simulation and records of a variety of medjacking scenarios. Within a modular design, each module was assigned to a different type of medjacking, including brute force, DDoS, MITM, and trojan attacks in a controlled environment. By ensuring that each attack simulation functions independently (for instance, a DDoS simulation is kept different from a Trojan attack simulation), this configuration minimises the potential for cross-contamination among datasets. Consequently, the data captured from each attack is clean and useful.

Every packet captured within the modular testbed included fundamental attributes necessary for specific intrusion detection, including time, scr.host.ip, dst.host.ip, dst.proto.arp_ipv4, arp.opcode, scr.proto.arp_ipv4, checksum_icmp, seq.le_icmp, timestamp.transmit_icmp, contentle_http, request.method_http, request.full.uri.http, request.ver_http, response_ http, ack_tcp, checksum_tcp, connection.fin_tcp, connection.rst_tcp, connection.syn_tcp, connection.syn.ack_tcp, dst.port_tcp, flags_tcp, flags.ack_tcp, payload_tcp, seq_tcp, src.port_tcp, port_udp, stream_udp, time.delta_udp, qry.name_dns, qry.name.len_dns, qry.qu_dns, qry.type_dns, retransmission_dns, retransmit_request_dns, con.ack.flags_mqtt, con.flag.cleansess_mqtt, con.flags_mqtt, hdr.flags_mqtt, len_mqtt, msg_mqtt, msg.type_mqtt, proto.len_mqtt, proto.name_mqtt, topic_mqtt, topic.len_mqtt, ver_mqtt, attack_label, attack_type. All these packets are further explained in section 4.4.2 Packet Description. This modular configuration facilitated the precise extraction of relevant data from each simulated attack. Each module within the testbed produced its respective logs and outputs, which were subsequently transformed into CSV format. The structured data organised in CSV files was optimised for efficient integration into ML models.

Data were gathered through normal operations such as sensor readings, network interaction and communication protocols. For normal operations, the APS prototype subsystems were implemented, and the subsystems interacted through the different prefixed communication protocols of the network. The communication protocols included unique sets of rules to create connections and transfer and receive packet-based information on formatting specified for data transfer. The next step was security attack data collection. Data were collected during attack scenarios, and the security attacks described in Section 4.3 were exemplified as proof of concept. The exemplified attacks included in this study were brute force, DDoS, MITM, and Trojan.

The packets obtained sensor activities, such as sent and received packets, delays, acknowledgement, and attacks. These sensor activities were used as evidence to compare behaviour and activities in the network. All packets were stored in a binary format, also known as packet capture, that roped nanosecond timestamps. All packets were captured using packet analysis software. Packet sniffers and Wireshark software packages were used to capture every item of packet traversing over the network. The packets were captured in the raw form and stored in an external device for processing into a readable form that visualised various packets for passive analysis.

Packet-level passive analysis is a significant traceback method that visualises sensor data to perform deep packet inspection of inbound and outbound data packets over the network. In general, this technique was applied to identify traces of malicious packets, and further traces were compared to identify major differences in activities. Passive monitoring was ideal due to less complex restrictions. In this study, packet-level passive analysis was founded around inbound and outbound network packets. These packets provided convenient information about network events, deep packet inspections, and network statistics that were carefully analysed to differentiate normal and attack packets.

4.5.1 Packet Processing

In this study, the data of normal and attack events were captured to synthesise data packets and conduct the packet-level analysis. Each packet was carefully stored in the external drive for packet processing. Following the data collection process, the packet processing phase consisted of multiple steps to manage raw data for analysis. The dataset contained different traces that were useful to identify normal and security attack operations or events. Attributes were extracted from the packet capture data and converted into CSV file format. The dataset contained both incoming and outgoing traffic covering a wide range of packet categories, including both normal and attack packets. Collected data packets used various programmes

such as tcpdump, Wireshark, and Tshark. The tcpdump software was employed via the command line interface, whereby the Syslog server was established to store the traffic recorded by tcpdump. The packet attributes were remotely accessed via the SSH protocol. Following the packet processing, Tshark tools were used to allow for extracting additional characteristics that may not be accessible through Wireshark. Tshark packet analyser was used with Wireshark through a command-line interface.

4.5.2 Packet Description

Packets of normal and malicious activities were collected using packet sniffers and Wireshark, and these packets were extracted using Tshark open-source software tools. The packets contained several types of data, such as source and destination information, timestamp logs, and other relevant details. Different packets performed different tasks within connected devices. Various packets were extracted to analyse the major differences between normal and attack patterns. The packet types and descriptions are presented in Table 4.3.

Table 4.3

Packets Representing the Type of Attribute and Their Description.

No.	Name	Description
1	time	Start time
2	scr.host.ip	IP address source host
3	dst.host.ip	IP address destination host
4	dst.proto.arp_ipv4	Portion of IPv4 Target IP address
5	arp.opcode	Portion of IPv4 opcode response
6	scr.proto.arp_ipv4	Sender IP address
7	checksum_icmp	Error detection
8	seq.le_icmp	Identifier
9	timestamp.transmit_icmp	Start timestamp

No.	Name	Description
10	contentle_http	Length
11	request.method_http	Request
12	request.full.uri.http	Full request
13	request.ver_http	Version
14	response_ http	Response
15	ack_tcp	Acknowledgement segments
16	checksum_tcp	Error detection
17	connection.fin_tcp	Terminate connection
18	connection.rst_tcp	Reset segment connection reset
19	connection.syn_tcp	Start connection
20	connection.syn.ack_tcp	Start connection and acknowledgement segment
21	dst.port_tcp	TCP destination port
22	flags_tcp	Indicate information
23	flags.ack_tcp	Indicate information and acknowledgment
24	payload_tcp	Payload
25	seq_tcp	Segment
26	src.port_tcp	TCP source port
27	port_udp	Source or destination port
28	stream_udp	Stream value
29	time.delta_udp	Last packet time
30	qry.name_dns	Query request
31	qry.name.len_dns	Query request length
32	qry.qu_dns	Query request question
33	qry.type_dns	Query request type
34	retransmission_dns	Resend

No.	Name	Description
35	retransmit_request_dns	Resend request
35	con.ack.flags_mqtt	Acknowledge segments
36	con.flag.cleansess_mqtt	Specify persistent and non-persistent connection
37	con.flags_mqtt	Specify link with payload
38	hdr.flags_mqtt	Variable header
39	len_mqtt	Packet length
40	msg_mqtt	Message
41	msg.type_mqtt	Message type
42	proto.len_mqtt	Length field
43	proto.name_mqtt	Protocol name
44	topic_mqtt	Specific topic
45	topic.len_mqtt	Specific topic length
46	ver_mqtt	MQTT version
47	attack_label	Normal packet flow allocates 0, and attack packet flow allocates 1
48	attack_type	Attack type

The packets presented in Table 4.3 were exchanged inbound and outbound traffic. All packets were captured and stored in an external drive. To visualise packets and analyse pair association, a correlation algorithm was applied to analyse the differences between packets. This analysis identified which pair positively or negatively correlated, and the analysis provided insight into the signals between connected devices.

The packet-based information comprised meta-information about the transmitted packets from communication protocols in the network. The flow-based information comprised meta-information about the activities that occurred in the network connections. In general,

packet-based feature data are network information (packet headers, sequence number, flags, and checksum), transport information (TCP, UDP, Internet Control Message Protocol [ICMP], and IP), and traffic information (inbound and outbound metadata). Flow-based feature data are volume information (payload size, flow size) and inbound and outbound information (byte size per message). Both packet-based and flow-based data can be used to predict the performance or take appropriate action for forecasting.

Attributes or characteristics of data packets indicate either normal or attack behaviour. Cybersecurity professionals use these attributes to detect attacks. For example, source and destination IP addresses contribute a significant role in investigations, especially when unusual patterns such as a high volume of data packets to or from certain IPs indicate possible attacks (e.g., DDoS attacks). Another example of attack production can be based on data packet sizes: very large or small packet sizes can indicate injection attacks or malware dissemination. Moreover, investigating flags, including SYN, ACK, reset segment connection, and FIN flags can also indicate MITM attacks.

4.5.3 Packet Labelling

Packet labelling refers to the process of assigning class labels or categories to data packets. Packet labelling is particularly important in supervised ML, where the models are trained on a labelled dataset to find patterns and correlations between features and labels. In supervised ML, the model learns to map input features (packet data) to the right labels during training. Typically, a dataset consists of a training set, which is used to train the model, and a testing set, which is used to evaluate the model's performance on new data. In this study, the packets in the dataset were labelled to identify the correlation between normal and attack behaviours. Two classes were categorised where 0 represented normal and 1 indicated attack pattern. The binary classification was applied to separate the legitimate sensors' data and attack machines' data. Table 4.4 presents the labelling of legitimate and attack data representation.

Table 4.4

Binary Classification of Legitimate and Attack Behaviours.

Type	Class	Description
Attack_label	0	Legitimate
Attack_label	1	Attack

The classifications presented in Table 4.4 were used to analyse the attack and normal pattern using correlation identification between different data packets.

4.5.4 Performance Evaluation

In this study, the implementation of the prototype and testbed were accomplished. Quantitative performance evaluation was made using Pearson's correlation coefficient to determine the correlation between one signal and another random signal. As previously discussed, the signal (packets) carries source and destination, and acknowledgement information (see Table 4.3). Some signals were identified as correlated with one signal and another random signal. The Pearson correlation coefficient method was used in this study to identify relationships between paired variables, denoted by *r*, where *r* calculates a linear correlation of paired variables (See Equation 4.4 below). The outcome of *r* could be between +1 and −1, which shows the strength and the absolute dependencies or relationships among paired variables.

$$r = \text{covariance}(X, Y) / (stdv(X) * stdv(Y)) \quad (4.4)$$

Where r represents the Pearson correlation coefficient, X and Y are the two variables, and stdv represents the standard deviations. The r can be determined by dividing the covariance of the two variables by the product of their standard deviations. Measuring the standard deviation indicates the level of variation for each variable, whereas covariance shows how the two

variables vary in relation to each other. The primary objective of measuring the correlation was to identify and understand how random variables have a reciprocal relation. The degree of correlation can be strong (from 0 to + or − 1) or weak (from 0 to + or − 1); if no correlation of paired variables is determined, different correlated frequencies that are either strong or weak can be visualised. Table 4.5 describes the possible ratio ranges of the random correlated variables (associate signals).

Table 4.5

The Correlations Coefficient Values with Explanation.

Correlation coefficient value	**Explanation**
From ± 0.90 to ± 1.0	Very robust correlation or (± 90% − ± 100%)
From ± 0.70 to ± 0.89	Robust correlation or (± 70% − ± 89%)
From ± 0.40 to ± 0.69	Moderate correlation or (± 40% − ± 69%)
From ± 0.10 to ± 0.39	Low correlation or (± 10% − ± 39%)
From 00 to ± 0.9	Insignificant correlation or (00% − ± 0.9%)

The possibility of insignificant correlation occurring within paired linear numeric signals can be observed in Table 4.5. The correlation values indicate positive and negative correlation, where two pair signals are either positively or negatively correlated with another signal. Different signals performed different tasks within connected devices. Some signals had numeric values, and others had non-numeric values. The correlation method identified linear numeric associations and dependencies between one linear numeric signal and another random linear numeric signal. The performance evaluation using the Pearson correlation coefficient in this study is presented in Sections 4.5.1.1 and 4.5.1.2.

Within the context of the IoMT, the examination of correlations among device features yields essential insights pertinent to the identification of anomalous behaviour [761, 762].

IoMT systems generally demonstrate consistent relationships when operating under normal conditions. For example, insulin pumps, glucose sensors, and network traffic often show identifiable correlations, establishing a foundation of expected behaviour. Nevertheless, in the context of anomalous occurrences, including cyberattacks or system malfunctions, these correlations may change, synthetically increase, or display unpredictable patterns, thereby functioning as a critical indicator of prospective threats.

Certain categories of attacks exert distinct impacts on these correlations. For example, In the context of a brute force attack, specific features demonstrate determined correlations, whereas others indicate inverse correlations. Data packets include contentle_http, response_http, ack_tcp, and connection.rst_tcp appear distinctive patterns that diverge from normal patterns. Detecting brute force attacks generally requires an examination of traffic for indicators such as anomalous increases in data flow; any irregular pattern may denote the occurrence of a cyber-attack. In the context of Trojan attacks, data packets such as ack_tcp, connection.syn_tcp, and connection.syn.ack_tcp serve as a sign of the underlying attack pattern associated with these intrusions. Unusual connections between internal systems and unknown or recognised malicious domains or IP addresses may signify the presence of backdoor intrusions. In a DDoS attack, malicious traffic floods the system, thereby capturing IoMT devices or servers. A MITM attack involves the interception and potential modification of communications between IoMT devices and their corresponding servers. This results in disrupted correlations between transmitted and received packets.

4.5.4.1 Rationale for Using Pearson’s Correlation

Pearson's correlation is often chosen for many reasons in the context of cybersecurity risks related to the IoMT within EDA. EDA provides insight into the interrelationships among numerical variables, including their correlation, mean, median, mode, standard deviation, and quantiles [518]. Primarily, the correlation technique is adept at quantifying the strength and

direction of linear associations between two continuous variables [763-767]. In the IoMT ecosystem, numerous metrics, including sensor readings and network traffic, may demonstrate linear correlations under normal conditions [768-770]. Employing Pearson's correlation facilitates the detection of deviations from expected linear patterns, thereby providing an invaluable approach for identifying anomalies.

Another merit of Pearson's correlation is its straightforwardness and clarity of interpretation. Correspondingly, the correlation coefficient spans from -1 to +1, with values approaching +1 or -1 signifying strong positive or negative linear associations, while values proximate to 0 imply the absence of correlation [763, 771-773]. This straightforward interpretation allows expedited variation detection within the context of the IoMT, where prompt and precise insights are essential for cybersecurity analysts. Pearson's correlation is particularly adept at analysing continuous variables common within IoMT ecosystems. Typical instances encompass heart rate, glucose levels, packet dimensions, and transmission intervals. By focusing on these continuous data streams, Pearson's correlation identifies anomalies in forecast relationships during EDA.

When compared with alternative correlation methodologies, Pearson's correlation frequently demonstrates greater significance in IoMT contexts. Spearman's rank correlation effectively identifies monotonic relationships; however, it exhibits diminished sensitivity to nuanced linear variations [774-777]. However, linear relationships hold significant importance in anomaly detection within the IoMT. Therefore, Spearman's rank correlation chances are less to perform well due to the monotonic relationships technique. Another technique, Kendall's tau is more appropriately utilised for ordinal data and rankings [778-780], thereby making it less applicable to the continuous and high-frequency data produced by IoMT devices. Ultimately, Pearson's correlation demonstrates both computational efficiency and significant assistance from major data analysis libraries, including NumPy and pandas [781]. Pearson's

correlation is particularly well-suited for examining continuous, linear relationships common to IoMT systems, thereby making it an appropriate selection for EDA in IoMT.

4.5.4.2 Normal Data Correlation

The Pearson correlation coefficient has been used in many cybersecurity studies. Consequently, the present study perceived performance evaluation using the Pearson correlation coefficient as an appropriate analytical technique. Correlation analysis was used to identify data patterns and prepare data for additional analysis processes. The normal APS data packets were visualised and examined using the Pearson correlation coefficient. The results presented in this section were consistent with addressing the research sub-question: What exploratory data analysis technique can be used to explore data patterns of medjacking methods, including brute force, DDoS, MITM, and Trojan attacks?

The correlation coefficient of normal signals was quantified to ascertain a numerical linear association between two paired signals. Comprehensive details of each signal are provided in Table 4.3. The quantitative examination of normal signals showed a consistent pattern of change in the other signals when one signal changes. Two signals correlated when a change in one signal was linked by a predicted variation in the other signal. To understand the signals' correlation and their relationship with other signals, the Pearson correlation coefficient algorithm was applied to calculate the association or similarities of paired signals, as shown in the heatmap below (Figure 4.11).

Figure 4.11

The Estimated Normal Data Correlations Between One Signal and Others Using Pearson's Correlation Coefficient.

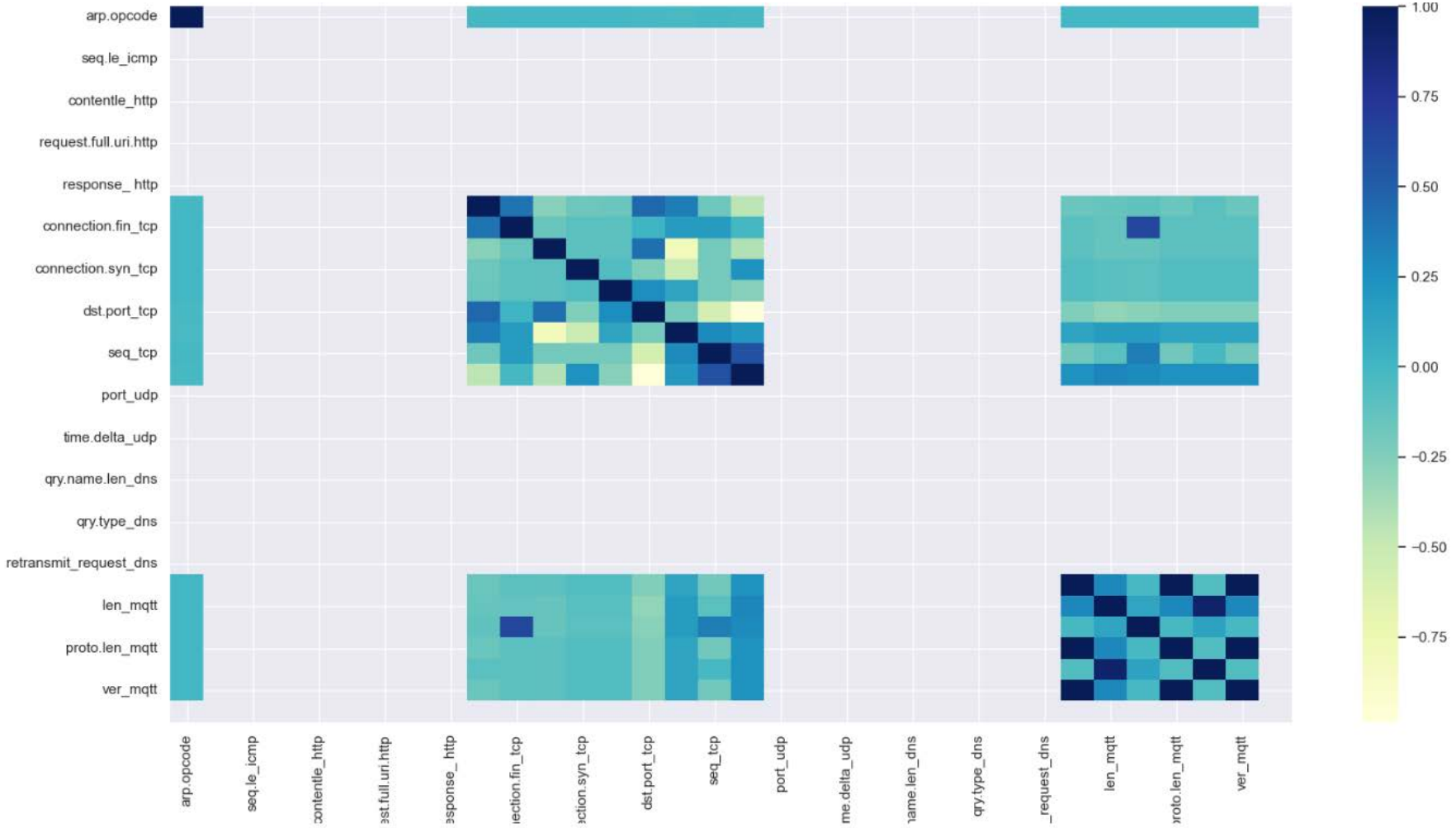


Figure 4.11 illustrates the numerical linear correlation between one signal and another. The various colours represent a proximity range of 0.75–1.0, indicating either a positive or negative correlation frequency. The relationships between each signal were analysed to determine the extent of use for predicting and modelling data. This analysis was based on the frequencies of numeric linear correlations, which enabled the labelling and description of the associations and helped interpret the linked signals' ranges. The varying coefficients indicated a relationship, while a non-numeric value in a signal represented a null value and did not capture correlation frequency. The presence of 0 correlation in certain portions indicated a lack of numeric association between certain signals, as determined by correlation coefficient analysis. This study presents an estimated numerical linear correlation between various signals. Further narrows to understanding the correlation from 1:15 numeric signals and 15 to the remaining signals illustrated in the heatmaps is visualised in Figures 4.12 and 4.13, respectively.

Figure 4.12

The Estimated Normal Data Correlations Between One Signal and 15 Signals Using Pearson's Correlation Coefficient.

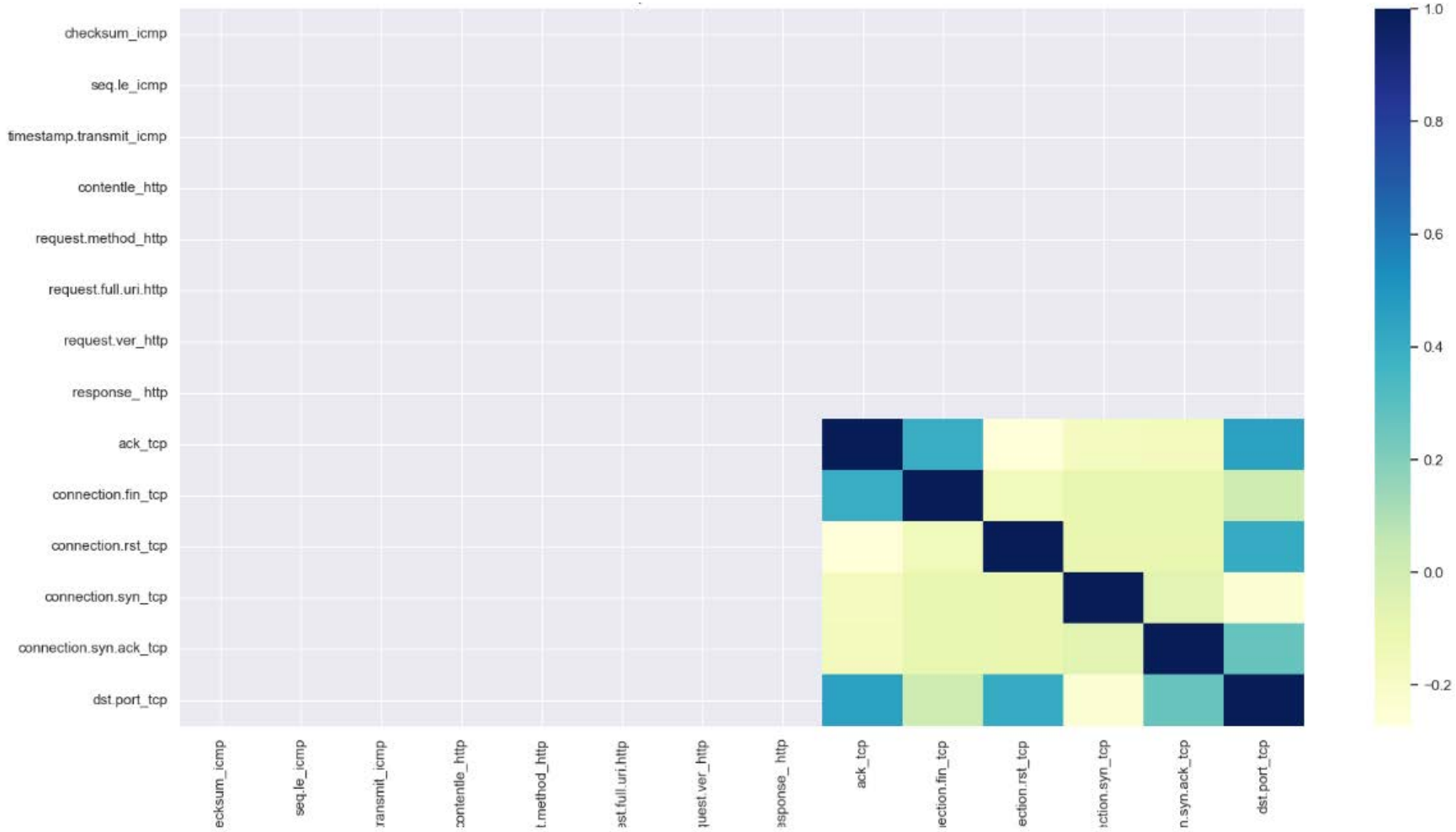
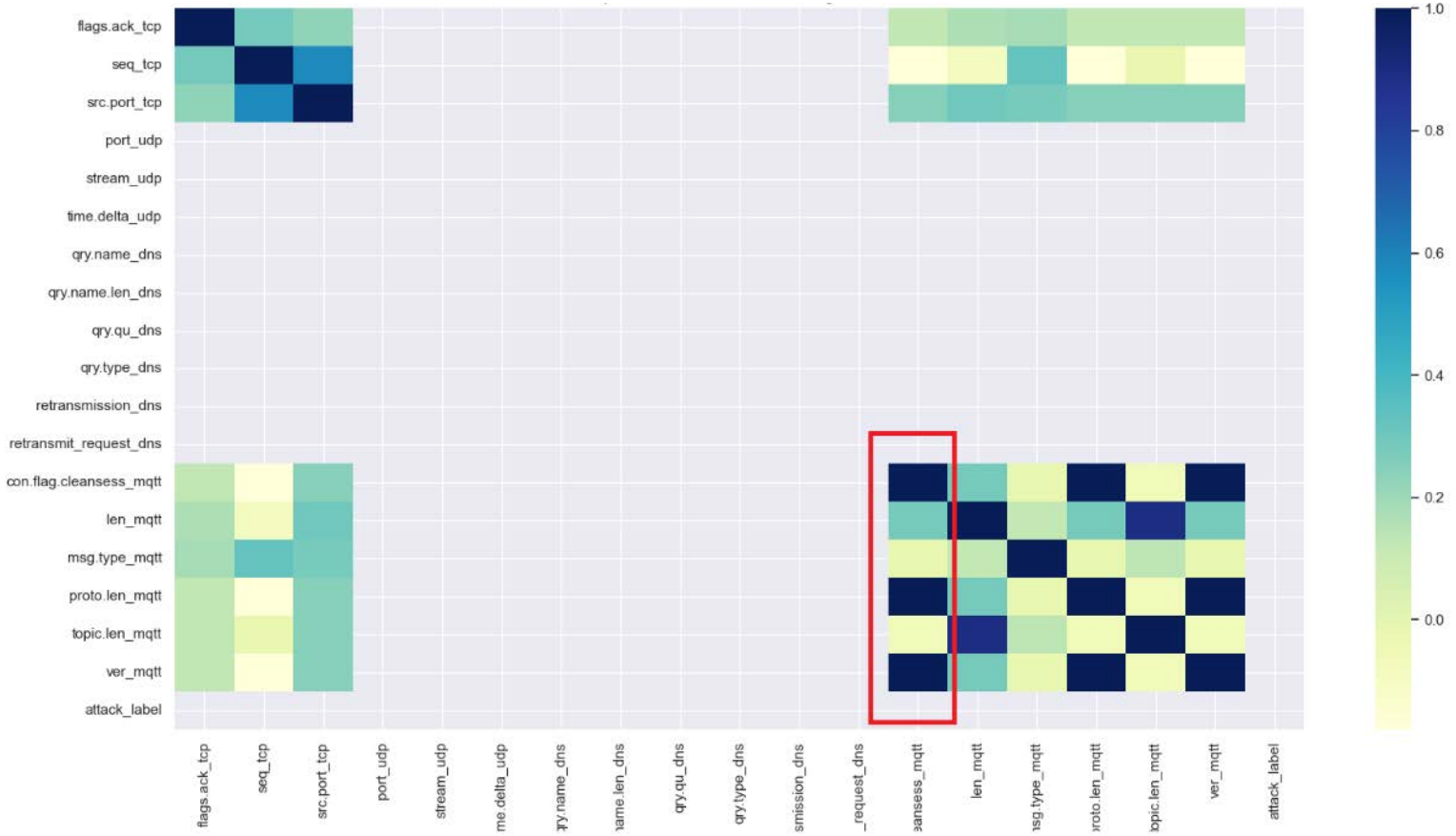


Figure 4.13

The Estimated Normal Data Correlations Between 15 Signals and Remaining Signals Using Pearson's Correlation Coefficient.



Figures 4.12 and 4.13 depict the numerical linear correlation of different signals to predict statistical relationships between these signals. Some signals are positively correlated, and some are negatively correlated; for instance, the frequencies outlined by the red rectangle in Figure 4.13 represent a strong positive correlation between one signal and other signals, indicating that these frequencies were particularly useful for understanding the direct relationship between signals.

The normal data were examined and visualised using the Pearson correlation coefficient to determine the link between normal APS device activities. The assessment found a significant relationship among the typical APS operations. The strong positive correlation between normal APS device functioning has various effects. For example, it implies that several actions were interconnected and driven by similar factors. Analysing these points of intersection can aid in predicting and enhancing the performance of APSs in the context of cybersecurity. This performance evaluation demonstrates the significance of utilising the Pearson correlation coefficient to designate the correlations of normal activity in APS devices.

4.5.4.3 Anomalous Data Correlation

A correlation analysis was conducted on the anomalous data to understand and compare anomalous patterns, distinguish anomalous behaviours, and visualise the anomalous raw data for the performance evaluation. Research that identifies cybersecurity vulnerabilities can inform evidence-based security solutions. To examine the APS device's vulnerability to medjacking, the anomalous data packets of the attacks were evaluated using the Pearson correlation coefficient. The correlation of anomalous signals was used to identify the relationships between paired signals. The descriptions of all packets are presented in Table 4.3. Various packets executed distinct actions. For instance, DDoS attacks frequently generated flooding through source and destination IP addresses, particularly in cases where there were unexpected patterns, such as a large amount of data packets to or from certain IPs. Data packet

sizes are another indicator of attack generation that indicates Trojan distribution. Different flags, including SYN, ACK, reset segment connection, and FIN flags indicate normal or attack occurrence. Therefore, it was most suitable to determine the correlation of each signal. A statistical study was conducted to visualise several attack signals for DDoS, MITM, brute force, and Trojan attacks. In the evaluation analysis of the anomalous data, some attack signals were identified as having positive and negative correlations with each other or similarities between paired signals. The correlation between various attack patterns is explained as follows.

4.5.4.4 Brute Force

This study exemplified a brute force attack to obtain credentials and access to the APS's subsystems in a virtual machine setting, where no machines were harmed. Brute force attack data were examined using the Pearson correlation coefficient to visualise the attack pattern as compared to normal patterns and with other attacks patterns (See Figure 4.14).

Figure 4.14

The Estimated Brute-Force Attack Correlations Between One Signal and Others Using Pearson's Correlation Coefficient.

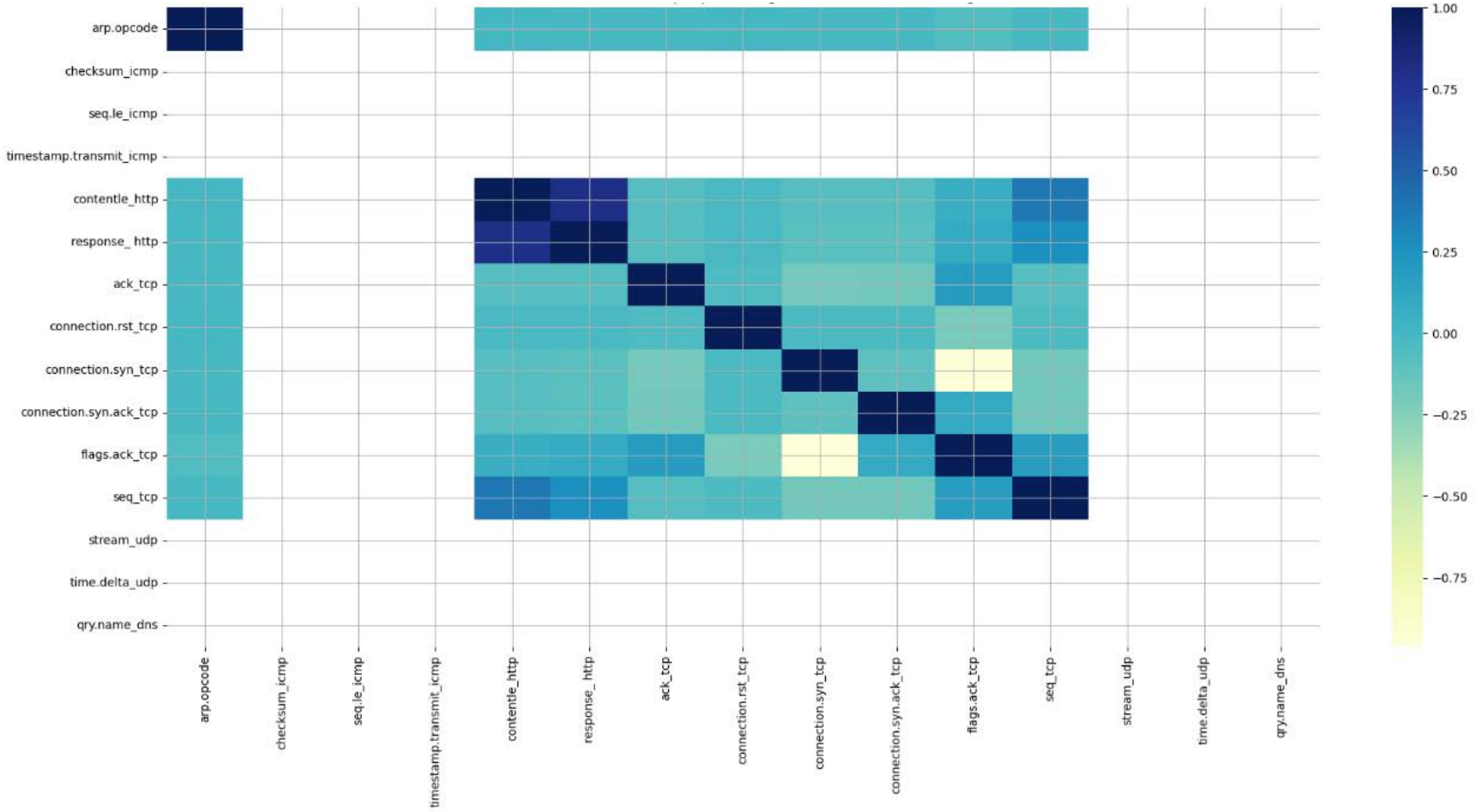


Figure 4.14 illustrates the numerical linear correlation of various packets associated with a brute force attack. Certain signals exhibit high correlations while others show negative correlations. This figure visualises the unique network traffic patterns that were developed in the presence of brute force attacks, such as contentle_http, response_ http, ack_tcp, and connection.rst_tcp data packets that contributed unique patterns as compared to normal activities. Identifying brute force attacks typically requires watching network traffic for signs such as unexpected increases in traffic to authentication endpoints, uncommon user-agent strings in HTTP requests, and repetitive use of common passwords. The brute force attack's data pattern was unique compared to other attacks in this study as visualised in the heatmap above.

4.5.4.5 Distributed Denial of Service

In this study, DDoS attacks depleted the resources of the APS's subsystems and rendered a service inoperable. This research exemplified a DDoS attack using a virtual machine. Three steps were involved in the TCP handshake process. First, a programmed script using a command line initiated a synchronise flood request by continuously sending SYN requests to a target machine. Second, an SYN Acknowledgement (SYN-ACK) was generated from the central machine to respond with an acknowledgment. Third, ACK messages were sent to complete the connection establishment. This study examined DDoS attack trends and features and visualised raw data into a heatmap, as shown in Figure 4.15. The Pearson correlation coefficient was used to estimate the DDoS correlation between one signal and other data signals.

Figure 4.15

The Estimated Distributed Denial-of-Service Attack Correlations Between One Signal and Others Using Pearson's Correlation Coefficient.

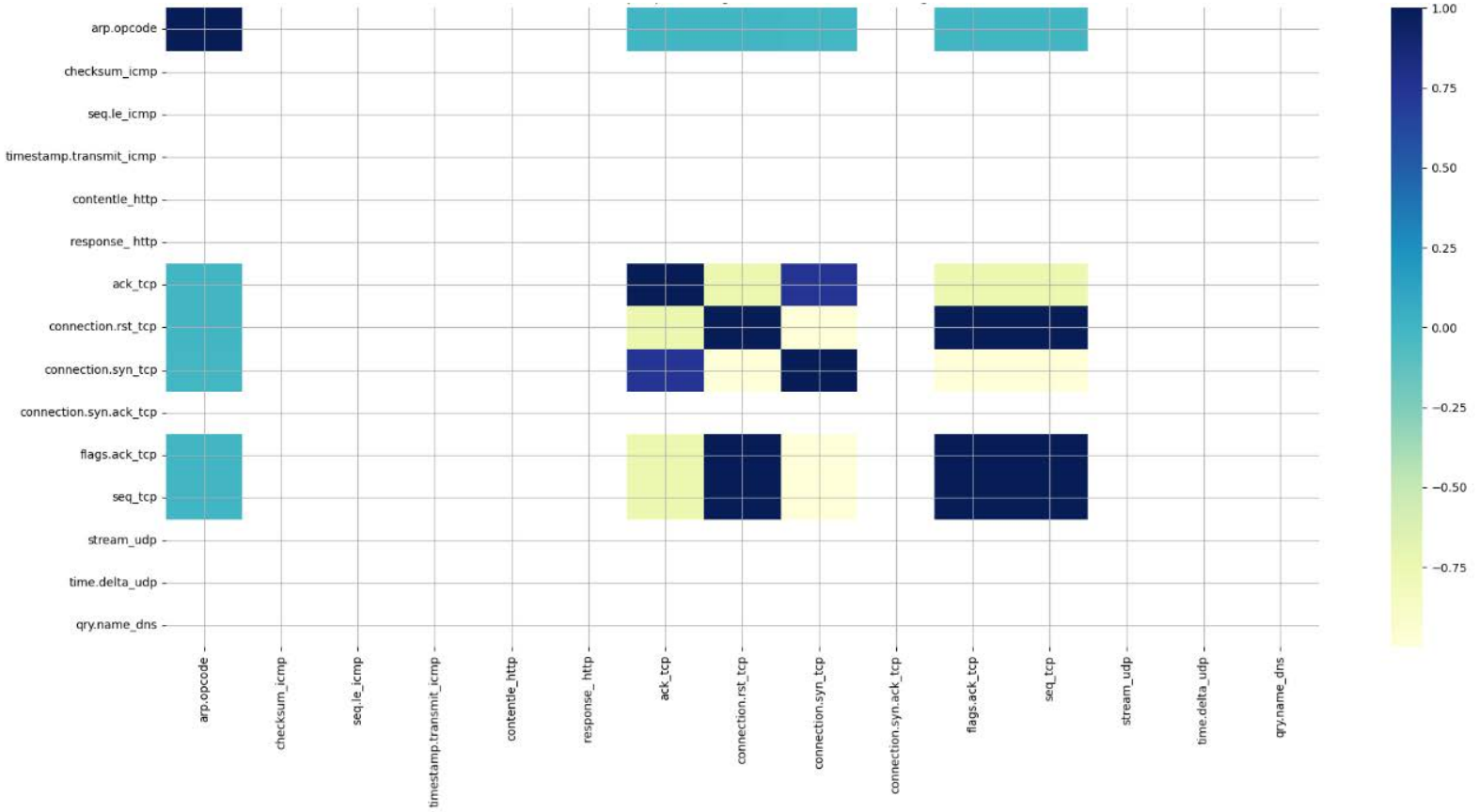


Figure 4.15 depicts the numerical linear correlation of different packets associated with DDoS attacks, demonstrating the statistical relationships among them. Certain indicators show strong positive relationships while others show negative relationships. During the DDoS attack, different data packets were transmitted to flood network resources to represent the typical nature of this attack. DDoS attacks may include various packet types, but certain characteristics of the packets can indicate their connection to such attacks. For example, overloading a target server with TCP and SYN packets indicated the pattern of DDoS attacks, which disrupted the server's capacity to handle incoming connection requests. The connection.rst_tcp, ack_tcp, connection.syn_tcp and flags_tcp packets are typical data packets linked to DDoS attacks. The data pattern of DDoS attacks appeared distinct from other attacks, as shown in the heatmap above.

4.5.4.6 Man in the Middle.

This study focused on targeting DNS and ARP protocols to activate MITM attacks. The simulated attacker used the DNS port to execute an attack to control the APS subsystems' IP address. The attacker attempted to capture credentials when a DNS request was sent to the DNS server; during this phase, the attacker forged responses to the subsystems. This study analysed trends and characteristics of MITM attacks and presented the raw data in a heatmap, as depicted in Figure 4.16. The Pearson correlation coefficient estimated the MITM attack's data correlation between one signal and other data signals.

Figure 4.16

The Estimated Man-in-the-Middle Attack Correlations Between One Signal and Others Using Pearson's Correlation Coefficient.

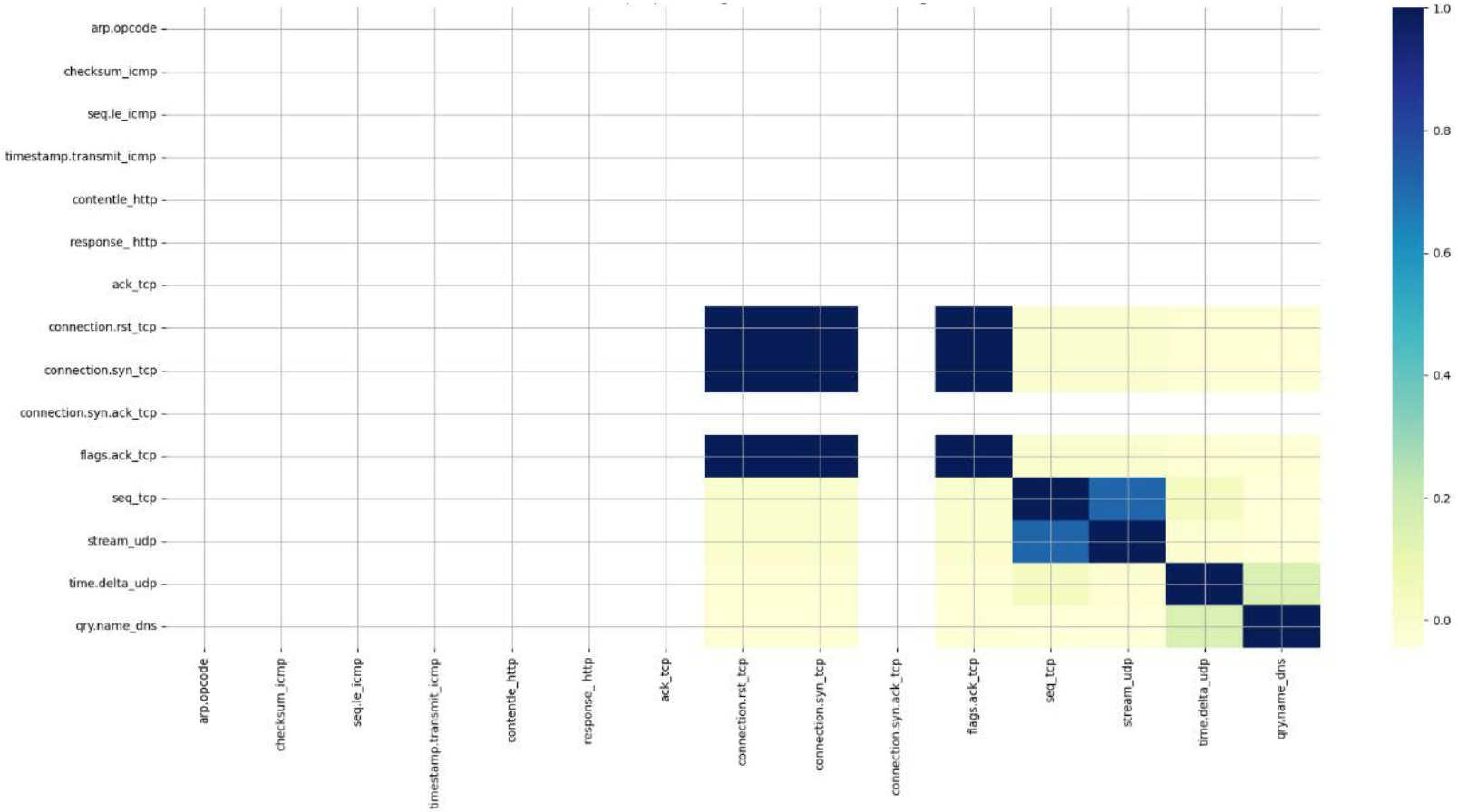


Figure 4.16 shows the numerical linear correlation of various packets linked to MITM attacks. Some indicators show strong positive connections while others indicate negative connections. The MITM attack was examined, and patterns were visualised, such as the DNS spoofing packets associated with the attack patterns. If this was not a simulated attack, the attackers would have misled the victim into connecting with their servers by intercepting DNS requests and providing false DNS records. The DNS packets showed the MITM attack presence; the data packets represented different patterns in the MITM attack as compared with different attack and normal patterns.

4.5.4.7 Trojan Attack.

This study analysed Trojan attack patterns and presented the raw data in a heatmap to visualise the correlation as depicted in Figure 4.17. The Pearson correlation coefficient was used to estimate the Trojan attack's data correlation between one signal and other data signals.

Figure 4.17

The Estimated Trojan Attack Correlations Between One Signal and Others Using Pearson's Correlation Coefficient.

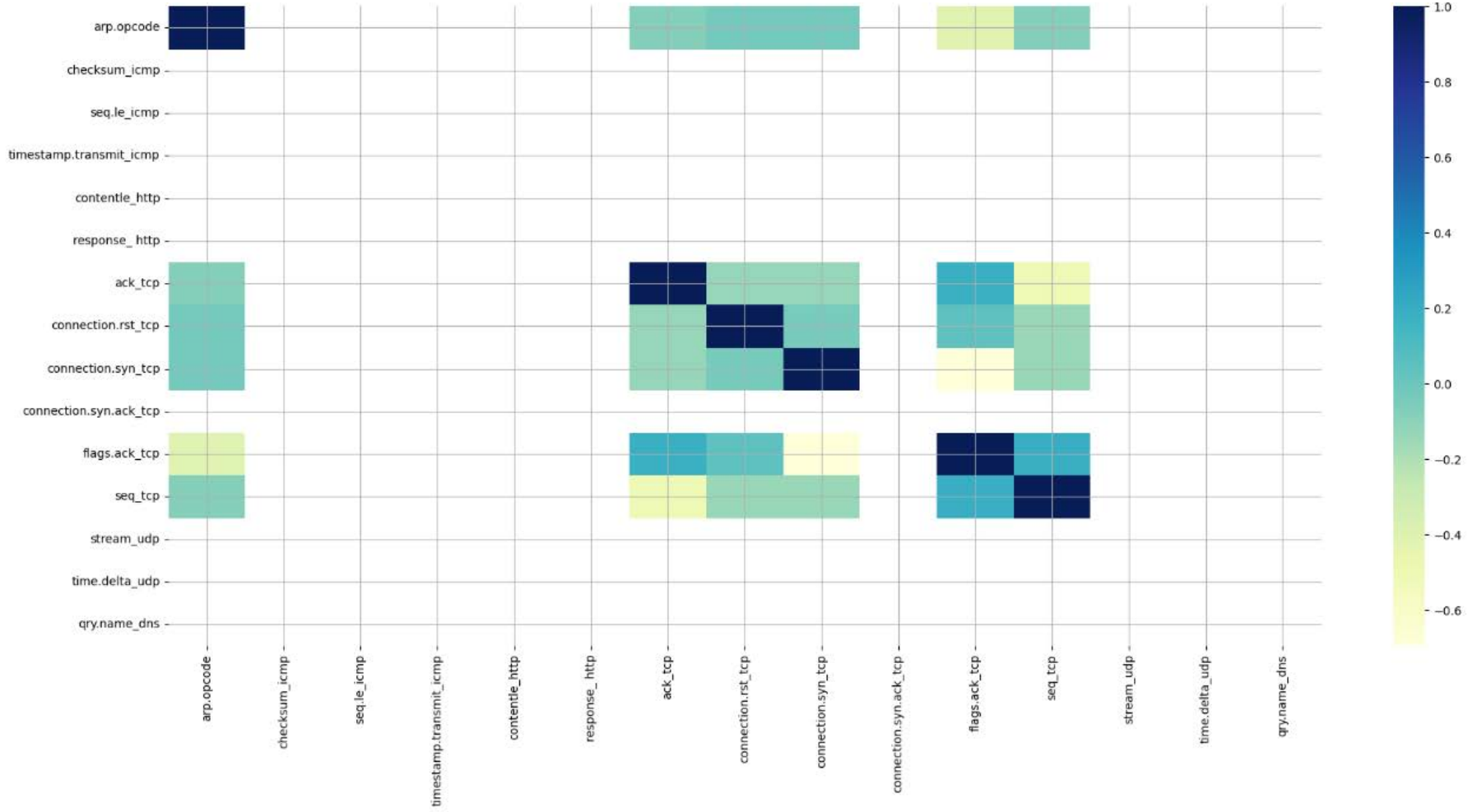


Figure 4.17 presents the numerical linear correlation of packets associated with Trojan attacks. Some signs indicate strong positive correlations while others indicate negative correlations. The backdoor was created as a connection with remote command-and-control servers, where frequent communication involved periodic requests for guidance, updates, or transferring data to the server. The heatmap shows that the communication pattern differs from other attacks in this study. The ack_tcp, connection.syn_tcp, and connection.syn.ack_tcp data packets indicate the Trojan attack pattern. Unusual connections from internal systems to suspicious or known malicious domains or IP addresses can indicate backdoor attacks.

Figures 4.14, 4.15, 4.16, and 4.17 present the correlations of the brute force, DDoS, MITM, and Trojan attacks. The correlations between these attack signals have exhibited distinct patterns through the Pearson correlation analysis. The correlated patterns between brute force and Trojan attacks have exhibited a substantial difference. Nevertheless, a discernible variation in intensity exists when comparing the DDoS and MITM attack patterns, which indicates a possible connection between these attacks. Attackers could try to gain unauthorised access using MITM tactics in the background while launching DDoS attacks as a diversion strategy.

Brute force data packets have shown correlation with different packets, and the contentle_http, response_http, ack_tcp, and connection.rst_tcp data packets primarily exhibited unique patterns that indicated a brute force attack. The correlation of DDoS attacks has shown that data packets such as connection.rst_tcp, ack_tcp, connection.syn_tcp and flags_tcp indicated the DDoS attack. In the MITM attack, the correlation of the DNS packets has shown different patterns that indicated the MITM attack. The data packets present during the MITM attack were distinct compared to other types of attacks and normal patterns. When examining the Trojan attacks the ack_tcp, connection.syn_tcp, and connection.syn.ack_tcp data packets

indicated the Trojan attack pattern. All patterns were presented in the heatmap figures to understand the attack packet correlations.

This study conducted a comprehensive medjacking correlation analysis and identified different attack correlation patterns. Understanding the correlation patterns of medjacking attacks enables more complete evaluations by identifying hidden dependencies between attack patterns and mitigating possible risks. The examination of correlation patterns may help the future identification of attack patterns that indicate the occurrence of coordinated or sequential attacks carried out by the same threat actor. Pearson's correlation analysis was used to investigate the different features of medjacking. Identifying these correlation patterns can enhance security measures by providing information on patterns and trends for developing defence techniques, tools for detecting threats, and plans for mitigating attacks based on tailored individual patterns.

4.6 Summary

The implementation of the APS subsystems was accomplished using a simulation approach. The simulation has provided a virtual environment where this research exemplified the functionality of the APS device without the need for proprietary devices and real patients. The simulations provided a cost-effective and efficient way to perform IoMT functionalities and collect datasets for further analyses. A cybersecurity testbed for the APS subsystems was developed to evaluate the medjacking vulnerabilities of the device. The testbed mimicked the medjacking conditions. Key components of the testbed included the APS subsystems, an IoMT communication environment, and attack tools (virtual machines, operating systems, and scripted frameworks). In this study, data gathering focused on normal and attack-related actions in the simulated environment. The simulated environment monitored and recorded APS device activities, medjacking activities, and communications, including sensor readings, device interactions, and network traffic patterns under normal and attack situations.

Chapter 5. The Development of the Hybrid Intrusion Detection System

Chapter 5 reports the development of a Hybrid Intrusion Detection System (H-IDS) by combining signature and anomaly detection methods to detect medjacking vulnerabilities in the APS. The H-IDS was developed using both Machine Learning (ML) and DL (Deep Learning) models to detect medjacking. ML and DL models have significantly enhanced the identification of known and unknown attacks, particularly when incorporated with conventional signature-based and anomaly-based detection designs. Unknown attacks represent a distinct category of attacks that exploit newly identified vulnerabilities. For known attacks involve to those that leverage vulnerabilities that have previously been recognised or signatures created. ML and DL models present dynamic, efficient, and adaptive approaches for addressing these threats by utilising a range of models, including Autoencoders for anomaly detection and supervised classifiers such as Multi-Layer Perceptron (MLP), Decision Trees, Support Vector Machines (SVM), and XGBoost for signature-based detection, as proof of concept.

To comprehend the operational mechanisms of the detection of unknown attacks, the autoencoder model was developed for anomaly detection in this study, which possessed the capability to directly process a dataset in its unrefined state, characterised by noise and lack of structure, without necessitating extensive preprocessing. This approach enabled the identification of unknown attacks that diverge from established or previously known signatures or known attacks. Conversely, the dataset requires a preprocessing step to enable the generation of signatures for known attacks. This preprocessing approach encompasses cleaning, normalising, and feature engineering (i.e., feature selection and dimensionality reduction) to develop signatures, which subsequently serve as the foundation for training models aimed at signature-based detection. A hybrid approach integrated both signature-based and anomaly-

based detection mechanisms. ML and DL models are trained and tested as proof of concept to detect known and unknown attacks.

To enhance effectiveness, the ML and DL models and algorithms were used with signature-based feature engineering, grid search, and classification. In anomaly detection, autoencoder is a DL method, a form of artificial neural network utilised to detect anomalies. Models utilised in signature detection are designed to detect established patterns that are linked to various types of attacks. The 2022APS dataset utilised in the proposed H-IDS consisted of a varied collection of labelled instances depicting normal and attack activities. This dataset has been divided into training, validation, and test sets to aid in model training and evaluation. The 2022APS dataset consisted of both normal and attack packets. Each data packet in the dataset consisted of varied characteristics, including packet sizes, timestamps, and extra metadata offering detailed information about the events. This chapter will specifically answer the fourth research question:

4. What dimensionality reduction technique can be performed optimally in the development of a machine learning model for medjacking detection in the APS?

a) What differences can be experienced with and without dimensionality reduction techniques in the development of a machine learning model for medjacking detection in the APS?

5.1 Hybrid Intrusion Detection System

The H-IDS was created using the open-source tools, models, libraries, and techniques discussed in Chapter 3, Table 3.3. The H-IDS consisted of three main components: data preprocessing, feature engineering, and the detection engine. Chapter 3 also presented the flowchart (see Figure 3.6) and development requirements for the H-IDS. The H-IDS received the initial input for data preprocessing, provided in Section 5.1.1, and feature engineering,

presented in Section 5.1.2. The detection engine presented in Section 5.2.3. An autoencoder model was trained for the anomaly detection. The Mean Squared Error (MSE) loss function was used to measure the differences between the original input and the remodelled output. In signature detection, the models were developed to classify normal and attack signatures. The dataset was discussed in Chapter 4. The 109,360 normal and attack samples offered abundant data for training and evaluation. There were 18,565 insulin device data packets and 18,227 CGM data packets; these data were labelled as normal data packets. The brute force (18,313), DDoS (18141), MITM (18,012), and Trojan (18,102) data were labelled as attack data packets.

The data collections of normal and attack activities were performed in Chapter Four, encompassing a range of data, including source and destination information, timestamp logs, TCP, UDP, ICMP, and IP; SYN, ACK, reset segment connection, and FIN flags as well as traffic information, and further relevant details. The data packets were exchanged in both inbound and outbound traffic. The packet-oriented data encompassed meta-information pertaining to the transmitted packets derived from the communication protocols within the network. Diverse packets executed different tasks within interconnected devices. The various packet types and their corresponding descriptions can be found in Table 4.3, Chapter Four. All data packets are explicitly discussed in Chapter Four: 4.4 Dataset Collection Unit, 4.4.1 Packet Processing, 4.4.2 Packet Description, 4.4.3 Packet Labelling. The details of H-IDS components are present in the following sections.

5.1.1 Data PreProcessing

Data preprocessing was a fundamental step in developing ML and DL models. The primary rationale behind the preprocessing approach was to convert raw data into a multi-dimensional format. Preprocessing techniques yielded an optimised dataset vector that was utilised as input for ML and DL models during the training phase to achieve optimal outcomes. As a result, this process enabled the discovery of knowledge from the 2022APS dataset. In

general, an unprocessed dataset may consist of noisy, unreliable, and unbalanced data characteristics that potentially impact training and analysis. Furthermore, unprocessed datasets may lead to overfitting or underfitting problems.

MiniMaxScallar and LabelEncoder were utilised to transform the data for the feature engineering step. The utilisation of the MinMaxScaler aimed to prepare data features within a (e.g., “within the 0–1”) range. The input data used in the MinMaxScaler process consisted of numerical characteristics that displayed variations in scales, units, or ranges. All features transformed with the range to minimise the possibility of biases that may arise from varying feature scales. All features were scaled, and the scaling process was based on linear transformation. This transformation consisted of subtracting the minimum value recorded in each feature from the data points, which was followed by dividing the result by the range. The range was determined by subtracting the minimum value of the corresponding attribute from the maximum value. LabelEncoder was used to transform categorical labels into numerical values, which was an essential prerequisite for analytical and modelling processes. The input data utilised by the LabelEncoder consisted of categorical features wherein each feature represented unique categories. The Labelencoder procedure allocated a unique integer value for each category. The sequence of encounters determined the assignment of numbers, and this process was crucial in converting the categorical data into a numerical representation. This numerical data was subsequently utilised as input for the ML and DL models.

The MinMaxScaler approach obtained the equitable transformation of all aspects. This approach reduced the possibility of any factor exerting an unbalanced impact on the modelling process. There were multiple reasons for using MinMaxScalar, including but not limited to adjusting the data to a specific (e.g., “within the 0–1”) range, which was essential and significant to scaling input features. Scaling the data to a consistent range helped prevent features with larger magnitudes from impairing those with smaller magnitudes in model

training; this technique ensured that each feature had an equal impact on the learning process. Second, MinMaxScaler maintained the original distribution when scaling data, which helped maintain the data distributional features and uphold key statistical properties. Moreover, the MinMaxScaler is primarily utilised in data preprocessing tasks due to its stability that guarantees uniform and dependable outcomes across various datasets and applications, particularly in situations where the data lacks significant outliers or non-normal distributions. In this study, data preprocessing was implemented to convert the 2022APS dataset into a coherent and compatible format. This operation aimed to transform the 2022APS dataset for the next step of H-IDS feature engineering. The preprocessed 2022APS dataset achieved a refined, optimised, and balanced structure, as MinMaxScallar was applied to transform the dataset in a scalable and consistent range. The 2022APS dataset was then prepared for further data feature engineering processes. Table 5.1 presents the different procedures conducted in the data preprocessing phase.

Table 5.1

Data Preprocessing Operations for 2022APS Dataset.

Operation	**Major**	**Description**
Data cleaning	Cogency	All data features were converted into numeric and Boolean formats.
	Segregation	All data were balanced for normal and attack types.
	Completeness	Unknown or missing data values were fixed. In general, the null value and not a number were set to 0, and non-numeric values were removed.
	Uniformity	Source and destination IP addresses were fixed into datum categories.

Operation	Major	Description
Data conversion	Non-numeric	The data containing communication protocol types, and normal and attack types in the nominal form with strings format, were applied using LabelEncoder to balance data.
	Numeric	Each feature's numeric value differed based on the activity logs.
Categorical encoding	LabelEncoder	Categorical columns were converted into numerical values.
Data transformation	MinMaxScaler	Feature values were categorised into a given range. Scaler was used when the upper and lower limits were identical.

Enhancing the 2022APS dataset using MinMaxScaler and LabelEncoder was important to ensure the strength and dependability of ML tasks. Both LabelEncoder and MinMaxScaler methods transformed the dataset for the ML tasks. The MinMaxScaler serves a pivotal role in the domain of feature scaling, thereby ensuring uniformity across the diverse features present within the dataset. In datasets characterised by features exhibiting distinct ranges (for instance, one feature may extend from 0 to 1000, whereas another may span from 0 to 10), in this case, the chances of becoming biased towards features with larger values may be experienced. Consequently, the MinMaxScaler was applied to transform features into a standardised range, conventionally between 0 and 1, thus ensuring that all features contribute equally. Conversely, LabelEncoder transformed categorical information into numerical values by assigning a distinct integer to each category, a requisite for most ML model development. Many ML models are engineered to handle numerical inputs; consequently, categorical data is transformed into a numerical format. The Label Encoder accomplishes this by allocating integer values to each category, thereby enabling the model to interpret and process the information accurately. Furthermore, LabelEncoder removed uncertainty in representing categorical data and guarantees uniformity and transparency across a dataset.

5.1.1.1 Justification of MinMaxScaler

The MinMaxscalar is a critical procedure for normalising the features. In various studies related to intrusion detection [782-789], the researchers utilised a MinMaxScaler technique to normalise data inputs (i.e., scaling values within the interval of 0 to 1 and -1). Specifically, MinMaxScaler enhances model effectiveness in developing intrusion detection systems utilising ML models. In a study related to intrusion detection [790], the researchers stated that a data normalisation technique is required to ensure that data with distinct ranges are standardised to a uniform scale. The normalisation technique confirms that the model does not incorporate inconsistent ranges into its calculations, which could potentially lead to ineffectiveness within the overall learning model. The MinMaxscalar is a normalisation technique that systematically converts each data with varying ranges to a uniform scale of 0 to 1, and -1. The researchers stated that the data was normalised [0, 1, and -1] using MinMaxScaler [790].

In this study, the MinMaxscalar was important for converting each data feature with varying ranges to a uniform scale of 0 to 1 to develop an MLP model to detect medjacking. According to the literature, normalised data facilitates accurate convergence during gradient-based optimisation processes in ML models such as MLP. As stated in [791], the transformation of scaled features facilitates a more efficient convergence of gradient descent, a factor of significance for neural networks used in Intrusion Detection System applications.

In this study, to develop an SVM model to detect medjacking, the MinMaxscalar was important for converting each data feature with varying ranges to a uniform scale of 0 to 1. According to the literature, the SVM model performs optimally when applied to scaled data [792]. The MinMaxScaler ensures that each feature contributes uniformly, a critical factor for accurate margin computation and optimal kernel efficacy in SVM models.

In this study, the development of the XGBoost, and DT models required MinMaxscalar to convert each data feature with varying ranges to a uniform scale of 0 to 1. According to the literature, in the context of XGBoost and DT, the application of MinMaxScaler is essential for scaling data features. In [793], the researchers used MinMaxScaler for data normalisation, thereby enhancing the effectiveness of tree split computations and, consequently, improving the overall efficiency of the model.

According to the literature [782-793], developing an intrusion detection model requires normalised data for accurate identification. The MinMaxScaler is a fundamental technique that normalises data, and it is known as a process significantly enhanced through feature scaling implementation. The MinMaxScaler technique serves to normalise data, thereby facilitating the capacity of models to identify complex anomalies in relation to a baseline of expected behaviour. The notion of normalisation, encompassing Min-Max scaling, finds its roots in fundamental statistical procedures. Scaling serves to standardise variables within the context of multivariate analysis, defined as a collection of statistical techniques used to examine data encompassing multiple variables concurrently. The MinMaxScaler constitutes a fundamental preprocessing procedure in developing ML and DL models to detect intrusions.

5.1.1.2 Justification of LabelEncoder

The LabelEncoding technique is used to transform categorical features into their numerical representation. In intrusion detection applications, categorical variables such as attack types, protocol, and service types, including TCP, UDP, and HTTP, encompass protocol type, service, flag, and attack. Machine learning models typically require numerical inputs, and the LabelEncoder effectively converts categorical variables into numerical labels, thereby maintaining a concise feature set. The technique systematically transforms each categorical value within the dataset into a numerical representation [794]. The LabelEncoder ensures that categorical variables are uniformly represented in a numerical format, thereby maintaining

their intrinsic significance without leading to the dimensionality of the feature set, a crucial aspect for models to interpret the input data accurately.

The LabelEncoder technique is a fundamental preprocessing step in developing ML models. Numerous studies have developed intrusion detection using LabelEncoder as an essential preprocessing step [788, 794-801]. The LabelEncoder is particularly advantageous for handling categorical data, a common dataset characteristic that frequently encompasses protocol types, attack classifications, and categorical indicators pertaining to network services or flags. Within this context, the LabelEncoder converts categorical features into integer representations, thereby facilitating the effective and efficient processing of categorical data using MLP, SVM, XGBoost, and DT models. The transformation of categorical variables into integer representations facilitates ML models' processing and interpretation of critical features.

5.1.1.3 Justification of Why MinMaxScaler and Label Encoder Steps Are Critical in the Overall Detection Process

In the field of data science, a MinMaxScaler pertains to its function of normalising numerical data, whereas a Label Encoder transforms categorical data into a numerical format. Both represent essential preprocessing stages in the field of data science, facilitating the effective operation of ML and DL models across a variety of data types. The MinMaxScaler and Label Encoder represent essential preprocessing techniques in the development of a model aimed at intrusion detection. Identifying intrusions within the APS systems necessitates the implementation of robust preprocessing techniques to adequately prepare data for ML and DL models, thereby ensuring both accuracy and effectiveness in intrusion detection.

In this study, the MinMaxScaler serves the purpose of normalising numerical features within the 2022APS dataset to a designated range [0, 1, and -1]. The APS intrusion detection datasets encompass numerical attributes, including time, scr.host.ip, dst.host.ip, dst.proto.arp_ipv4, arp.opcode, scr.proto.arp_ipv4, checksum_icmp, seq.le_icmp,

timestamp.transmit_icmp, contentle_http, request.method_http, request.full.uri.http, request.ver_http, response_ http, ack_tcp, checksum_tcp, connection.fin_tcp, connection.rst_tcp, connection.syn_tcp, connection.syn.ack_tcp, dst.port_tcp, flags_tcp, flags.ack_tcp, payload_tcp, seq_tcp, src.port_tcp, port_udp, stream_udp, time.delta_udp, qry.name_dns, qry.name.len_dns, qry.qu_dns, qry.type_dns, retransmission_dns, retransmit_request_dns, con.ack.flags_mqtt, con.flag.cleansess_mqtt, con.flags_mqtt, hdr.flags_mqtt, len_mqtt, msg_mqtt, msg.type_mqtt, proto.len_mqtt, proto.name_mqtt, topic_mqtt, topic.len_mqtt, ver_mqtt, attack_label, attack_type. The MinMaxScaler technique was most important because in case of without any of scaling, features that have larger values may dominate those with smaller values, resulting in biassed outcomes. The MinMaxScaler in this study facilitated the equitable contribution of all features to the ML and DL models by standardising their respective values.

In this study, the Label Encoder constitutes a crucial preprocessing technique, employed to transform categorical data into numerical representations. The 2022APS datasets encompassed categorical attributes, including protocol types (e.g., TCP, UDP) and attack classifications (Normal, Attack). ML and DL models necessitate the utilisation of numerical inputs, thereby delivering encoding an indispensable process. Label encoding ensures the uniform representation of categorical features, thereby facilitating the model's capacity to process and derive insights from them with effectiveness. In the context of supervised learning models, the encoding of target labels, wherein "Normal" is denoted as 0 and "Attack" as 1, serves an essential role in the association of input features with their corresponding classifications. This transformation ensures uniformity in data representation and enhances the ease of interpretation of the outcomes.

The simultaneous application of MinMaxScaler and Label Encoder signifies the appropriate preprocessing of both numerical and categorical features, thereby enhancing the

effectiveness of intrusion detection mechanisms. The MinMaxScaler mitigates the potential bias of models towards features characterised by larger ranges, whereas the Label Encoder facilitates the incorporation of categorical data. The MinMaxScaler and Label Encoder serve as essential preprocessing techniques in the field of intrusion detection. Through the normalisation of numerical features and the encoding of categorical data, these techniques improve the accurate processing of data by models and enhance the effectiveness of intrusion detection.

5.1.2 Feature Engineering

Feature engineering refers to the structured process of choosing, generating, or adjusting input data points (also known as features) to enhance the predicting capacities and classifications of the models by optimising features using the available data. Different techniques can be used in feature engineering, including feature selection, reduction, transformation, and visualisation. In the data preprocessing phase, the 2022APS dataset enhanced its structural representation, prepared data transformation, and enabled visualisation at an optimal level.

In the feature engineering phase, feature selection and dimensionality reduction algorithms were used to prepare the dataset for model building. First, feature selection techniques were utilised to determine a subset of the most relevant features from the 2022APS dataset. Second, dimensionality reduction and different techniques were utilised to prepare data into a low dimensionality to enhance the computational efficiency of the ML models and minimise the potential for overfitting problems. Both feature selection and dimensionality reduction significantly contributed to enhancing the performance of the ML models employed in the study. The use of feature engineering significantly impacted the models' forecast accuracy, generalisation capabilities, and classification performance. Both techniques are described in Sections 5.1.2.1 and 5.1.3.

5.1.2.1 Feature Selection

In this study, feature selection involved the identification and selection of a subset of relevant variables from the 2022APS dataset. According to [701], the feature selection process is considered an essential phase in developing ML models for numerous reasons, as it significantly impacts the model's performance. The process involves identifying and selecting the most pertinent features from the dataset, simultaneously eliminating irrelevant or redundant ones. The researchers in a study [701], concluded that "Feature selection is effective in preprocessing data and reducing data dimensionality". Moreover, the researcher also mentioned "feature extraction/selection still plays an essential role such as improving learning performance, preventing overfitting, and reducing computational costs". According to [703], feature selection denotes the process of deriving a subset from an original feature set based on specific selection criteria, which identifies the relevant features within the dataset.

Feature selection was significant in this research for several reasons. Initially, this method decreased the dimensionality of the input space, thereby minimising the risk of overfitting, lengthened training periods, and inadequate generalisation. Additionally, the utilisation of feature selection resulted in enhanced model performance. This was achieved by prioritising the most informative features that have a significant impact on the prediction of the target variable. In this study, correlation coefficient, the variance threshold, and Recursive Feature Elimination with Cross Validation (RFECV) techniques were utilised to identify and select key features. These techniques involved identifying how strong, weak, irrelevant, and redundant features were in the dataset.

In the context of the 2022APS dataset, these techniques are applied to identify the relevance of features; the correlation coefficient technique was used to identify the features' strengths. Further, to identify relevant features, the variance threshold technique and the statistical variability of each feature are identified in Table 5.2. Another technique, RFECV,

applied to identify relevant features, is present in Table 5.3. This study utilised features identified from the RFECV technique because it integrates the significant benefits of recursive feature elimination with the approach's robustness of cross-validation.

The correlation coefficient quantifies the degree of linear relationship between pairs of variables. When two features demonstrate a substantial correlation, they signify corresponding information, potentially resulting in redundant information. Within the domain of feature selection, the correlation coefficient functions as an important technique for identifying and potentially discarding one of the features that demonstrate significant correlation, thus minimising multicollinearity and enhancing the model's complexity optimisation. In the variance threshold, the approach discards features distinguished by negligible variance, which are simply of minimal relevance to the predictive effectiveness of the model. Features illustrating a near-zero variance among samples often possess negligible informative value, as they provide inadequate differentiation between distinct classes or outputs. The variance threshold represents a simplistic yet effective technique for eliminating low-variance features, thereby optimising the dataset's structure. Recursive Feature Elimination with Cross-Validation (RFECV) represents an advanced technique within the domain of feature selection related to machine learning models. RFECV technique establishes feature ranking when combined with model validation to determine the most important subset of features. Starting with all the features, it systematically eliminates the least consequential feature(s) based on the effectiveness of a specified model utilising the tree-based model. Cross-validation is used at each phase to ensure that the selection process demonstrates a substantial level of generalisation, thereby providing RFECV with an effective technique for determining the most pertinent features for the predictive endeavour.

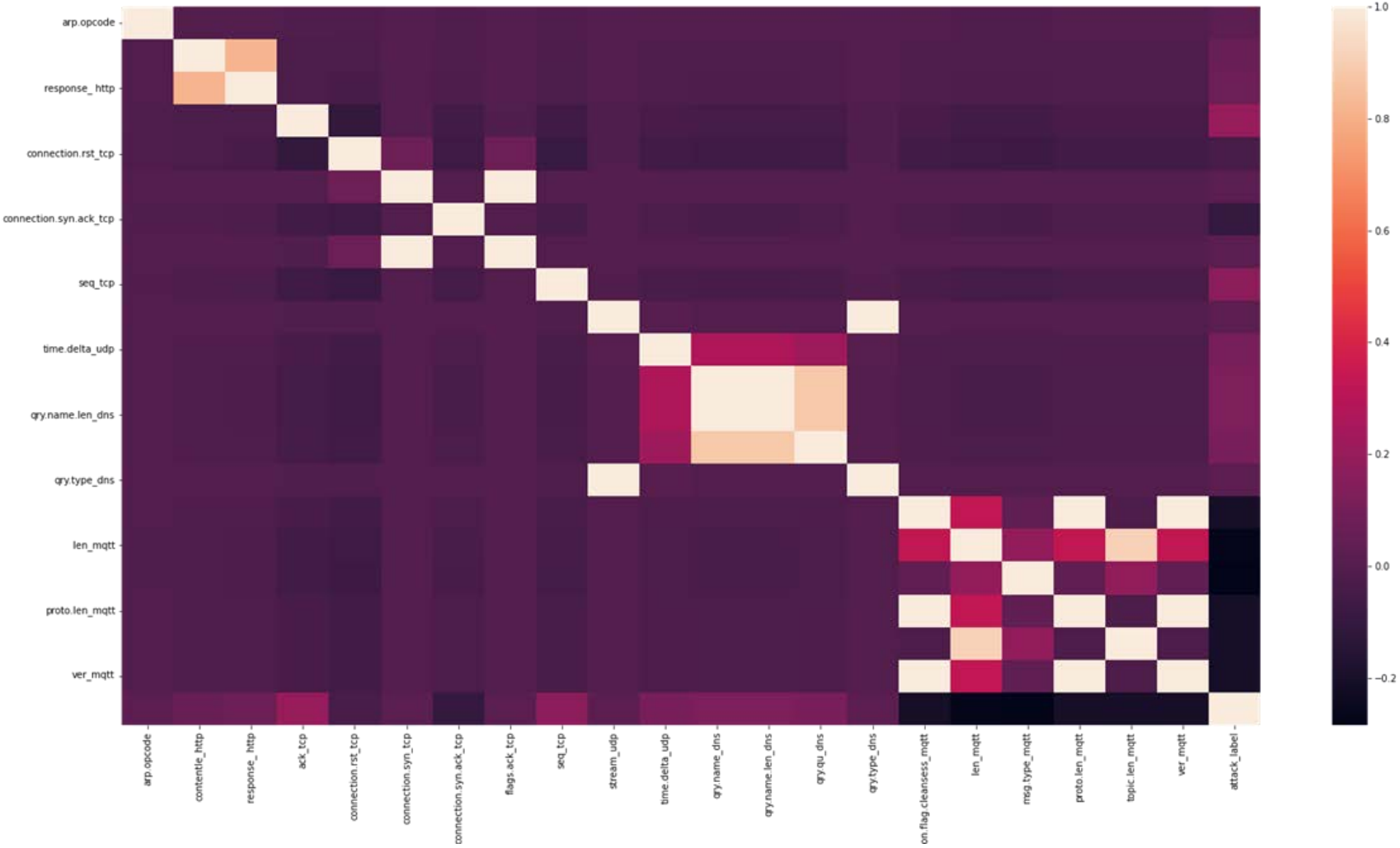
5.1.2.1.1 Correlation Coefficient

This study examined the effectiveness of correlation coefficient feature selection. This technique was used to analyse the relationship between each feature and the target variable by measuring their strength and direction. The primary rationale for identifying a linear relationship was to meet the requirements of the detection system, which relied on computing numerical linear statistics to detect intrusions. Numerical variables possess a crucial role in detection systems for various reasons. Primarily, linear features provided a quantitative understanding that enabled the ML and DL models to quantify the intrusions.

The quantitative data were used to learn the operation of the system and promptly identify intrusions. The correlation coefficient measured the strength of the linear connection between two variables. This linear measure offered valuable insights into the robustness and perspective of the relationship within different data features. The Pearson correlation coefficient of 1 indicated a positive linear relationship, whereas −1 indicated a negative linear relationship, and 0 implied no linear relationship between the variables. Figure 5.1 presents a heatmap that determines the association or similarities of paired signals.

Figure 5.1

Correlations Computed Between One Feature and Others in the Feature Section Process.



The different colours in Figure 5.1 symbolise a range of proximity from +1 to −1. An analysis was conducted to examine the relationships between each feature and assess their utilisation in data prediction and modelling. This analysis was conducted using numeric linear feature correlations that enabled the understanding of noteworthy features in the feature section task and, subsequently, classification tasks. The process required the computation of the correlation coefficient between each feature and the target variable. Features that show a high absolute correlation coefficient (close to 1 or −1) were strongly correlated with the target and were usually kept for further analysis. On the other hand, features with low absolute correlation coefficients (close to 0) were seen as weakly correlated and could be ignored or more carefully examined. By establishing a predetermined threshold, such as an absolute correlation coefficient surpassing 0.5, features that exhibited strong correlations with the target variable were included in the model.

5.1.2.1.2 Variance Threshold

In this study, the variation threshold was implemented to identify and eliminate features that have low or no variation. The variance threshold technique started by estimating the variance for each feature in the dataset. The variance threshold served as an index for removing features that failed to meet the required level of variance. The constant, 0-variance features were removed, as they showed insufficient variation. The variance was calculated as the average of the squared deviations between each data point and the mean of the dataset. The objective of this computation was to assess how much the values of each attribute deviated from their corresponding means. Features exhibiting low variance were identified as those characterised by data values that demonstrated a high degree of consistency and low variation from the mean. The features with low or no variance were removed for the predicting modelling task. Table 5.2 presents all features with their variance; a total of 48 features were provided in the feature selection phase and the variance threshold technique selected 20 features.

Table 5.2

Feature Selection Outcome Based on Variance Threshold Technique.

No.	Feature	Variance
1	arp.opcode	0.005439
2	contentle_http	15307.09
3	response_ http	0.011022
4	ack_tcp	1.36E+17
5	connection.rst_tcp	0.108296
6	connection.syn_tcp	1667.048
7	connection.syn.ack_tcp	0.034254
8	flags.ack_tcp	27982.53
9	seq_tcp	8.31E+10
10	stream_udp	0.31096
11	time.delta_udp	1299.988
12	qry.name_dns	0.029729
13	qry.name.len_dns	0.029729
14	qry.qu_dns	10862.4
15	qry.type_dns	0.174915
16	con.flag.cleansess_mqtt	0.02023
17	len_mqtt	23.66471
18	msg.type_mqtt	4.161976
19	proto.len_mqtt	0.323678
20	topic.len_mqtt	2.235919

Table 5.2 provides a statistical overview of the variability represented by each feature. The labels of the individual features are listed in the column and each row refers to a distinct

feature. Features with low or no variances were removed, as they contained uninformative or noisy data for the ML task. High variance features were selected, as they contained significant information for the modelling task. This technique enhanced the effectiveness of the ML task by eliminating features that were assessed as unlikely to significantly impact the modelling process. The removal of low-variance features reduced the risk of overfitting, a phenomenon in which models learn to include noise or irrelevant patterns from the dataset.

5.1.2.1.3 Recursive Feature Elimination with Cross Validation

The Recursive Feature Elimination with Cross Validation (RFECV) technique was applied in the feature selection phase to identify relevant features for the modelling task. While using the REFCV technique, an ML model was applied to identify the effectiveness of each feature in the 2022APS dataset. The DT model was applied for cross validation to examine the importance of individual features in the dataset. The cross-validation process in the RFE technique split the dataset into five folds (the *k*-fold). The 2022APS dataset contained 48 features with 109,360 instances of normal and different attack samples; these instances offered abundant data for training and testing. This technique selected the numeric features to identify the importance of each feature. The training process used $k - 1$ of the folds for model training with the remaining fold being allocated for testing purposes. This process iterated five times, which enabled a thorough evaluation of the model's accuracy across various subsets of data. Table 5.3 presents the computed accuracy of each fold.

Table 5.3

Accuracy of 1:5-Folds Using Recursive Feature Elimination with Cross Validation to Show the Features' Importance.

No.	Features name	Split0 accuracy	Split1 accuracy	Split2 accuracy	Split3 accuracy	Split4 accuracy
1	arp.opcode	93.032	93.109	87.756	92.899	92.712

No.	Features name	Split0 accuracy	Split1 accuracy	Split2 accuracy	Split3 accuracy	Split4 accuracy
2	checksum_icmp	94.545	94.614	94.595	94.399	94.239
3	seq.le_icmp	94.586	94.641	94.691	94.408	94.312
4	timestamp.transmit_icmp	98.852	99.094	99.044	98.824	98.710
5	contentle_http	82.822	98.518	95.331	98.408	98.710
6	response_ http	83.005	98.692	95.487	98.683	98.893
7	ack_tcp	83.005	98.692	95.487	98.683	98.893
8	connection.rst_tcp	83.005	98.692	95.487	98.683	98.893
9	connection.syn_tcp	83.005	98.692	95.487	98.683	98.893
10	connection.syn.ack_tcp	83.005	98.692	95.487	98.683	98.893
11	flags.ack_tcp	83.005	98.692	95.487	98.683	98.893
12	seq_tcp	83.005	98.692	95.487	98.683	98.893
13	stream_udp	83.005	98.692	95.487	98.683	98.893
14	time.delta_udp	83.005	98.692	95.487	98.683	98.893
15	qry.name_dns	83.005	98.692	95.487	98.683	98.893
16	qry.name.len_dns	83.005	98.692	95.487	98.683	98.893
17	qry.qu_dns	83.005	98.692	95.487	98.683	98.893
18	qry.type_dns	83.005	98.692	95.487	98.683	98.893
19	retransmission_dns	83.005	98.692	95.487	98.683	98.893
20	retransmit_request_dns	83.005	98.692	95.487	98.683	98.893
21	con.flag.cleansess_mqtt	83.005	98.692	95.487	98.683	98.893
22	len_mqtt	83.005	98.692	95.487	98.683	98.893
23	msg.type_mqtt	83.005	98.692	95.487	98.683	98.893
24	proto.len_mqtt	83.005	98.692	95.487	98.683	98.893
25	topic.len_mqtt	83.005	98.692	95.487	98.683	98.893

No.	Features name	Split0 accuracy	Split1 accuracy	Split2 accuracy	Split3 accuracy	Split4 accuracy
26	ver_mqtt	83.005	98.692	95.487	98.683	98.893
27	attack_label	83.005	98.692	95.487	98.683	98.893

Table 5.3 shows the RFECV results of 1:5 folds, showing the highest and lowest accuracy attained during the cross-validation procedure. The highest level of accuracy attained across all folds indicates the significance of the features. The RFECV technique iteratively performed this evaluation of the model's efficacy with mean test accuracy and standard deviation of accuracy over the folds. The mean test accuracy was calculated across all folds during the cross-validation procedure. The model involved training and evaluation five times using *k*-fold cross validation. The mean test accuracy obtained by averaging the accuracy across all folds, provided a comprehensive evaluation of the model's performance using the selected feature subset. The standard deviation measured the degree of variation in the test scores indicating the model's performance metrics. Using *k*-fold cross validation, the standard deviation was calculated by analysing the five test scores obtained from each fold. Table 5.4 presents the computed mean test accuracy and standard deviation.

Table 5.4

Mean Test Accuracy and Standard Deviation Shows the Features' Importance.

No.	Features name	Mean test accuracy	Standard deviation
1	arp.opcode	91.901	0.020
2	checksum_icmp	94.478	0.001
3	seq.le_icmp	94.528	0.001
4	timestamp.transmit_icmp	98.905	0.001
5	contentle_http	94.758	0.060

No.	Features name	Mean test accuracy	Standard deviation
6	response_ http	94.952	0.061
7	ack_tcp	94.952	0.061
8	connection.rst_tcp	94.952	0.061
9	connection.syn_tcp	94.952	0.061
10	connection.syn.ack_tcp	94.952	0.061
11	flags.ack_tcp	94.952	0.061
12	seq_tcp	94.952	0.061
13	stream_udp	94.952	0.061
14	time.delta_udp	94.952	0.061
15	qry.name_dns	94.952	0.061
16	qry.name.len_dns	94.952	0.061
17	qry.qu_dns	94.952	0.061
18	qry.type_dns	94.952	0.061
19	retransmission_dns	94.952	0.061
20	retransmit_request_dns	94.952	0.061
21	con.flag.cleansess_mqtt	94.952	0.061
22	len_mqtt	94.952	0.061
23	msg.type_mqtt	94.952	0.061
24	proto.len_mqtt	94.952	0.061
25	topic.len_mqtt	94.952	0.061
26	ver_mqtt	94.952	0.061
27	attack_label	94.952	0.061

A high mean accuracy, along with a low standard deviation, indicates consistently high performance across all folds that demonstrate the reliability of the selected feature subset. The

test accuracy and standard deviation offered valuable insights into the overall performance and consistency of the model using the chosen feature subset in RFECV with *k*-fold cross validation. The outcome of RFECV has shown the importance of the features. Subsequently, these features represent the important features to use for classification tasks in the detection models of the H-IDS. The mean test accuracy outcome of the RFECV technique for feature selection is shown in Figure 5.2.

Figure 5.2

Mean Accuracy of Features Using the Recursive Feature Elimination with Cross Validation Feature Selection Technique.

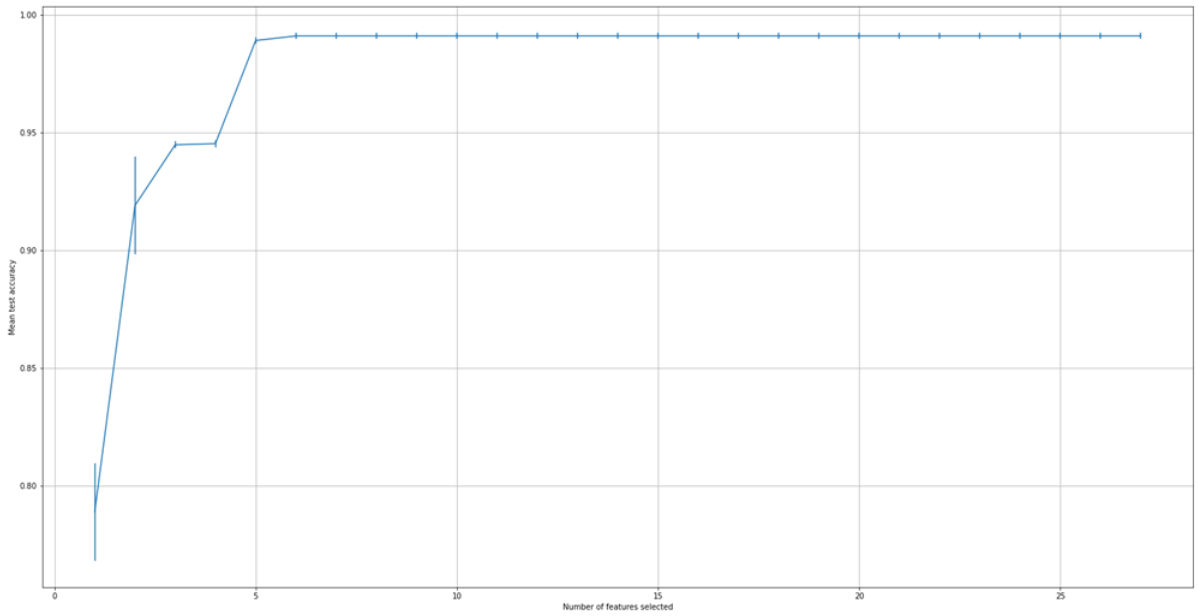


Figure 5.2 presents the validation accuracy of each feature. The cross-validation results show the importance of features with an accuracy maintained at 90%, an upper accuracy of 98.905, and a lower accuracy of 91.901. These features were selected using RFECV feature selection. By selecting the most relevant features during the training process, the model's predictive performance improved while mitigating the risk of overfitting.

5.1.3 Dimensionality Reduction

In this study, dimensionally reduction techniques, including PCA, t-SNE, and UMAP were utilised and compared for the intrusion detections in the APS devices. The dimensionality reduction process was essential for enhancing intrusion detection within APS devices, as this technique contributed to the simplification of high-dimensional data and enhanced the effectiveness of detection processes. Dimensionality reduction techniques lessen computational challenges by reducing the number of features while retaining crucial information. These techniques are discussed in sections 5.1.3.1, 5.1.3.2, and 5.1.3.3.

5.1.3.1 Principal Component Analysis

PCA was used for linear representation to show the relationship between two or more feature variables in the 2022APS dataset. In this technique, a new set of orthogonal axes (uncorrelated) divided the multivariate dataset into a series of orthogonal components. These components were selected to maximise the amount of variation; the process is described in the following subsections.

5.1.3.1.1 Standardisation of the 2022APS dataset

The standardisation stage transformed the original data into a uniform scale and range with a mean (average) of 0 and a standard deviation of 1 to maintain the same scale. As previously mentioned, the 2022APS dataset has 109,360 samples consisting of 48 features. The feature selection techniques were used to identify the most relevant features, and the RFECV technique identified 27 features that were used for dimensionality reduction in the PCA algorithm. The features of the 2022APS dataset were standardised by subtracting the mean of each feature and dividing by the standard deviation of that feature for *i* in the range of 1 and *n*.

$$S_{io} = R_{io} - \mu_i / \sigma i \quad (5.1)$$

In Equation 5.1, *S* is a standardised value, *i* is the feature, and *o* is the observation in the dataset. Further, μ_i is the mean, *R* is the raw value of *i-th* feature and *o-th* observation, and σi is the standard deviation of the *i-th* feature across all observations.

5.1.3.1.2 Covariance Matrix

In this study, the covariance matrix identified principal components as a part of PCA for dimensionality reduction. The mathematical expression used to compute the covariance between two features, *X* and *Y*, is:

$$C(X, Y) = (1 / (n - 1)) \sum\_[802] \wedge n * (x_i - \bar{X}) * (y_i - \bar{Y}) \quad (5.2)$$

Where *n* is the number of observations, x_i and y_i are discrete features; $\bar{X}$ and $\bar{Y}$ represent the means of *X* and *Y*, respectively. The covariance matrix *C* computed correlation among the various features available in the dataset. The components (eigenvectors) are utilised to determine the patterns that include the maximum variance in the data.

5.1.3.1.3 Eigenvectors and Eigenvalues

Eigenvectors found directions of maximum variance (principal components) within a dataset, whereas eigenvalues measured the corresponding magnitudes (length) of variance along these directions. The eigenvectors and eigenvalues calculation were formulated with *C* as the covariance matrix, λ as the eigenvalue, v_n as the correlate with eigenvectors, and *I* as the identity matrix.

$$(C - \lambda I)\, v_n = 0 \quad (5.3)$$

5.1.3.1.4 Explained Variance

The calculation of explained variance for a principal component can be expressed as the ratio of the eigenvalue associated with that component to the total sum of all eigenvalues.

Equation 5.4 measures the explained variation of the *i-th* principal component can be computed by dividing the *i-th* eigenvalue by the sum of all eigenvalues:

$$EV(i) = \text{Eigenvalue}(i) / \sum\_[802]\text{^n} * \text{Eigenvalue}(t) \quad (5.4)$$

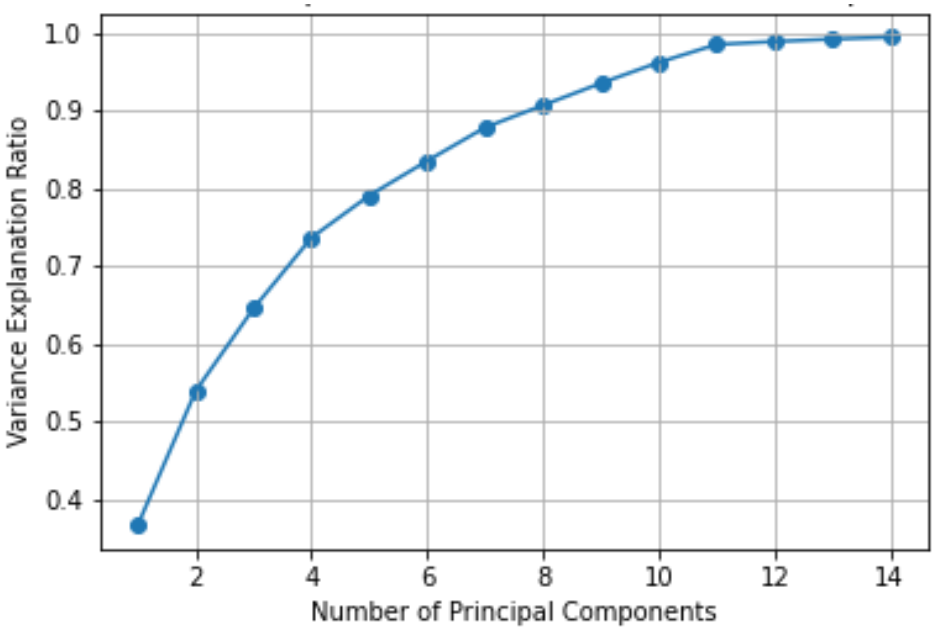
In this equation, *EV*(*i*) represents the explained variance associated with the individual *i-th* principal component. Eigenvalue(*i*) represents the individual *i-th* eigenvalue related to the *i-th* principal component. Eigenvalue(*t*) represents the eigenvalue related to the total *t-th* principal component.

5.1.3.1.5 New Features

PCA computed the covariance matrix, which was then followed by the process of eigenvalue decomposition. The process of decomposition facilitated the system in discerning the principal components. These components were determined as linear combinations of the initial attributes. The explained variance was then computed to observe the optimal principal component. The outcome of the explained variance is shown in Figure 5.3.

Figure 5.3

Principal Component Analysis Results: Explained Variance with Different Numbers of Principal Components.



In Figure 5.3, the y-axis denotes the ratio, and the x-axis represents the count of principal components. As the number of principal components increases, the percentage of explained variance progressively increases. The 95% retention of information was achieved with 10 components. The 99.47 retention of information was achieved with 14 components.

5.1.3.2 t-Distributed Stochastic Neighbor Embedding

Four steps were involved in the t-SNE. First, pairwise similarities were computed in the high-dimensional space using a Gaussian distribution. Second, similarities within the low-dimensional space were identified. Third, a similarity metric for the low-dimensional map was determined, and similarities were computed among the points inside the low-dimensional space. Fourth, the similarity between the high- and low-dimensional maps was optimised through the utilisation of the gradient descent technique to minimise the Kullback-Leibler divergence. The process of the t-SNE is described in the following subsections.

5.1.3.2.1 High-Dimensional Pairwise Similarities

This step calculated conditional probabilities that could accurately represent the probability of one data point selecting another as its neighbour. Equation 5.5 was used to compute high dimensional conditional probabilities.

$$P(i, j) = \exp(-\|xi - xj\|^2 / (2\sigma^2)) / \sum_{k \neq I} \exp(-\|xi - xk\|^2 / (2\sigma^2)) \quad (5.5)$$

The variable $p(i, j)$ represents the conditional probability in higher dimensional space that data point i would select data point j as its closest point. The variables xi and xj represent the data points of the high dimensional. The parameter σ is used to calculate the width of the Gaussian distribution to compute similarity.

5.1.3.2.2 Low-Dimensional Pairwise Similarities

The second step of t-SNE involved the formation of pairwise similarities within the low-dimensional space using a Gaussian kernel algorithm. The low-dimensional space was utilised to facilitate the embedding of the data to determine conditional probabilities. These conditional probabilities described the similarities between a mapped data point and its selection of another data point as its neighbour based on their relationships within this newly formed space. The mathematical equation to compute the low dimensional conditional probabilities is:

$$q(i, j) = \exp(-\|yi - yj\|^2) / \sum_{k \neq i} \exp(-\|yi - yk\|^2) \quad (5.6)$$

The variable $q(i, j)$ denotes the conditional probability that, inside the lower-dimensional space, data point i shows similarity to data point j. The variables yi and yj represent the squared Euclidean distance between the low-dimensional coordinates.

5.1.3.2.3 High- and Low-Dimensional Differences

The third step of t-SNE minimised the Kullback-Leibler divergence (relative entropy) to compute probability distribution and further understand how one point differs from another. The minimisation of relative entropy measured the dissimilarity of probability distributions, specifically the high dimensional conditional probabilities.

$$C = \sum\_i \sum\_j p(i, j) * \log(p(i, j) / q(i, j)) \quad (5.7)$$

The variable C computed dissimilarity or disparity between two probability distributions.

5.1.3.2.4 Visualise Low-Dimensional Embedding

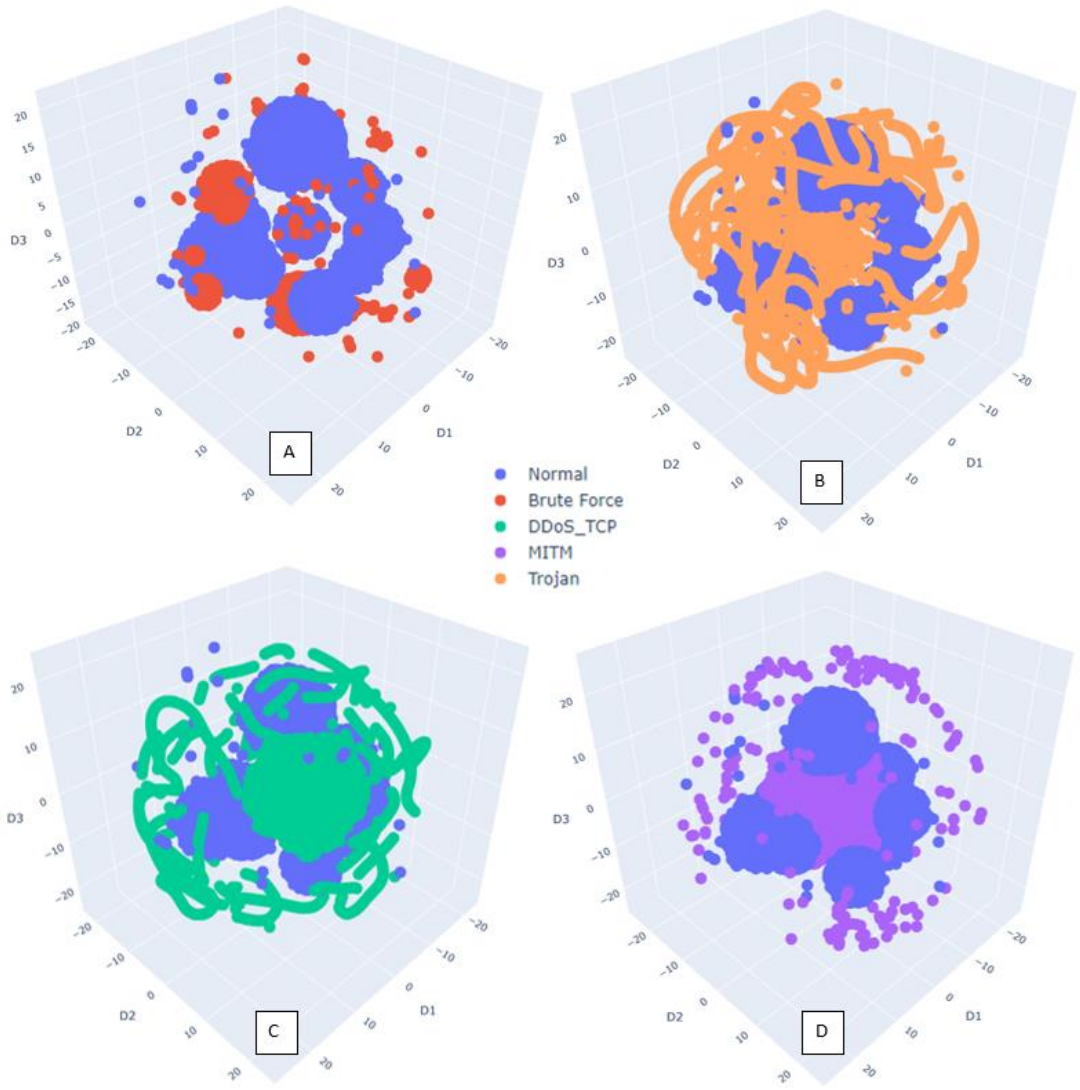
In the context of this low-dimensional embedding, each data point was defined by a specific collection of coordinates. These coordinates were computed through the optimisation process conducted in Step 3 (Section 5.1.3.2.5).

5.1.3.2.5 t-Distributed Stochastic Neighbour Embedding Outcomes

The number of nearest neighbours chosen throughout the iterative updating process represented the degree of perplexity. In general, a bigger sample set involves a higher degree of perplexity. Figures 5.4 and 5.5 show the dimensionality reduction in 3D representation, where different numbers of perplexities are applied. The t-SNE technique enabled a lower-dimensional space and provided numerous advantages for classification tasks. Primarily, t-SNE greatly improved visualisation by reducing high-dimensional data to a lower-dimensional space while maintaining the similarities between data points. In addition, t-SNE effectively eliminated any extraneous noise and irrelevant fluctuations in the data by placing a high value on maintaining the similarities between data points. The feasibility of generating scatterplots or other visualisations in a low-dimensional environment was achieved by lowering the dimensionality of the data.

Figure 5.4

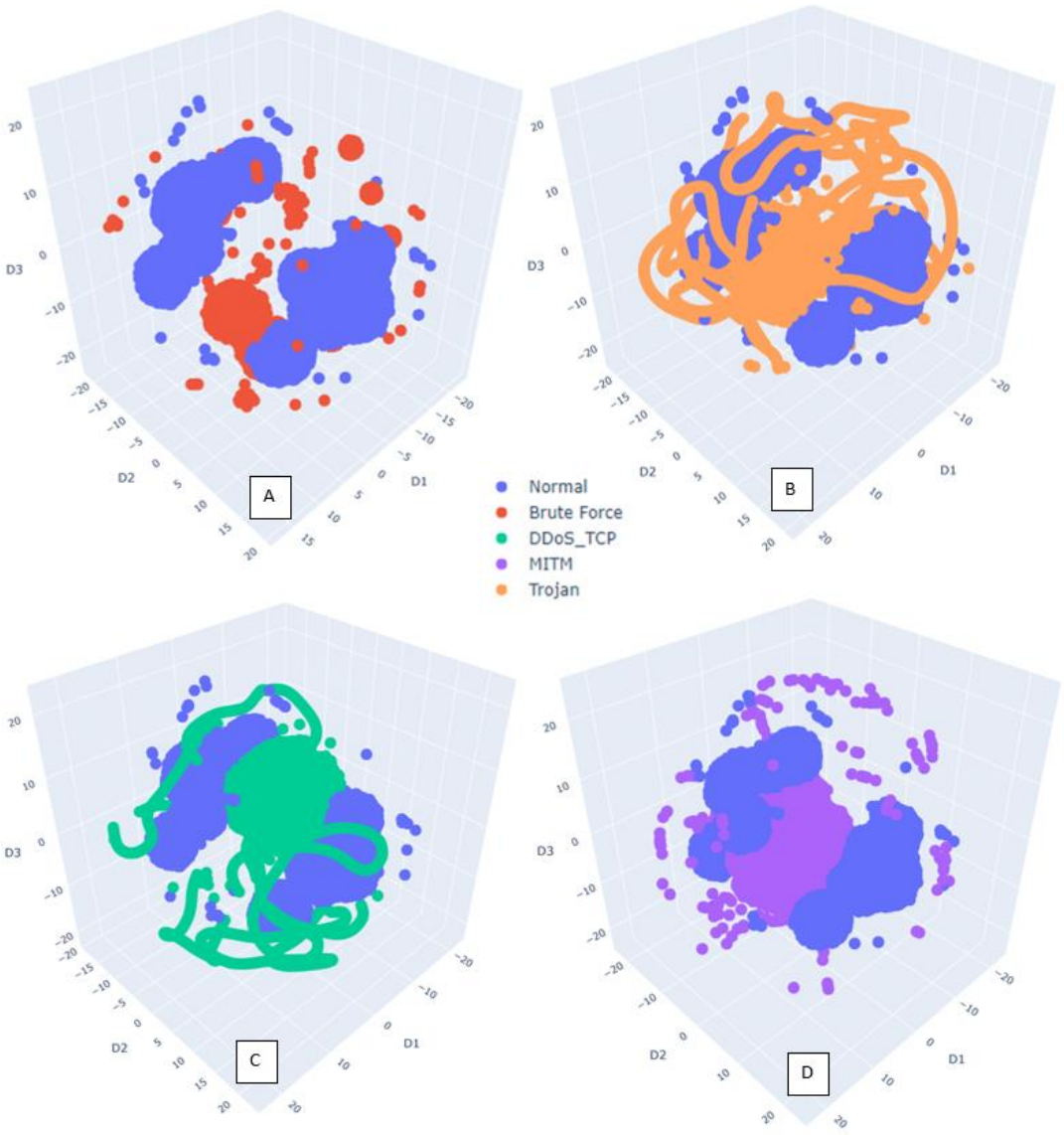
3D Visualisation of t-Distributed Stochastic Neighbor Embedding Dimensionality Reduction with Perplexity Set to 20.



Note. A = a dataset of normal and brute force attacks comprising 3D clusters. B = a dataset of normal and Trojan attacks with a 3D cluster representation, C = a dataset of normal and DDoS attacks with the 3D cluster, and D = a dataset of normal and MITM attacks.

Figure 5.5

3D Visualisation of t-Distributed Stochastic Neighbor Embedding Dimensionality Reduction with Perplexity Set to 40.



Note. A = a dataset of normal and brute force attacks comprising 3D clusters. B = a dataset of normal and Trojan attacks with a 3D cluster representation, C = a dataset of normal and DDoS attacks with the 3D cluster, and D = a dataset of normal and MITM attacks.

5.1.3.3 Uniform Manifold Approximation and Projection

The 2022APS datasets were visualised with a high dimensional by projecting them onto a lower-dimensional space. The UMAP process is described below.

5.1.3.3.1 Uniform Distribution

In this initial stage, neighbourhood identification involved the calculation of neighbour interactions among data points within a domain of high dimensionality. This calculation was performed to determine the nearest neighbours of each data point using the Euclidean distance measurement. The nearest neighbour of a data point was established based on its closest neighbours. Subsequently, the fuzzy topological representation set was constructed using the similarities between data points in this local neighbourhood that represented the local structure of the data.

5.1.3.3.2 Fuzzy Topological Representation

The next step involved creating a fuzzy simplicial set acting as a mathematical representation of the local relationships among data points. The mathematical representation quantified the probability of a connection between two data points in the graph founded around the similarity or proximity.

5.1.3.3.3 Low-Dimensional Optimisation

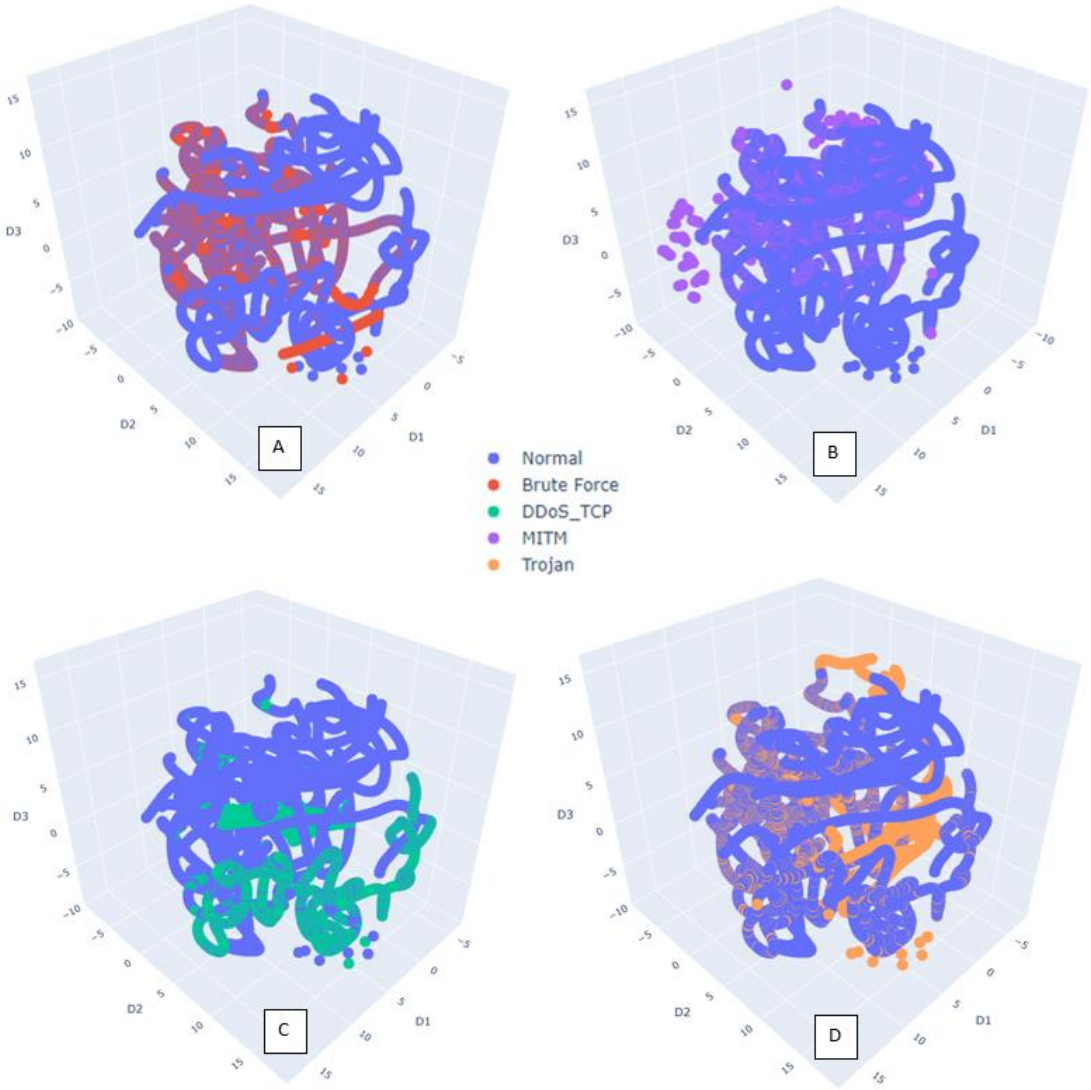
The low-dimensional optimisation consisted of two key tasks. The first task was to maintain the basic fuzzy topological structure that belonged in the high-dimensional space. The lower-dimensional visualisation of a fuzzy simplicial set maintained the proximity of highly interconnected points. The second task was to discover a reduced-dimensional representation that maintained the embedding within the fuzzy simplicial set. The optimisation task was aimed to minimise a cost function that quantified the degree of correlation between the fuzzy simplicial set in the high-dimensional space and the set’s equivalent in the lower-dimensional space.

5.1.3.3.4 Dimensionality Reduction

The UMAP proceeded to embed the given data into a lower-dimensional space represented in 3D. The UMAP captured intricate relationships and manifold structures that exist within high-dimensional data. Figure 5.6 shows the dimensionality reduction in 3D representation.

Figure 5.6

The 3D Visualisation of Uniform Manifold Approximation and Projection Dimensionality Reduction.



Note. A = a dataset of normal and brute force attacks comprising 3D clusters. B = a dataset of normal and Trojan attacks with a 3D cluster representation, C = a dataset of normal and DDoS attacks with the 3D cluster, and D = a dataset of normal and MITM attacks.

The t-SNE and UMAP visualisations presented in this study constitute three-dimensional representations of the data, wherein features are clustered according to their interrelations and proximity to one another within the decreased dimensional structure. In the t-SNE visualisations, the emphasis lies on maintaining the validity of local relationships among

data points. Consequently, points that exhibit proximity in the original high-dimensional space continue to remain combined in the three-dimensional representation.

Conversely, the UMAP visualisations effectively maintain the validity of both local and global structures. UMAP not only adeptly preserves the coherence of local clusters but also enhances the representation of the overarching relationships among clusters with greater devotion in comparison to t-SNE. Through the examination of these visual representations, data scientists or analysts can discern patterns, including compact clusters, distinct groupings, and any intersections, which can be analysed within the framework of the ML models. The insights are thoroughly examined in Chapter 6.

5.1.4 Hyperparameter Tuning

The hyperparameter tuning technique was applied to determine the most optimal combination of hyperparameters for the H-IDS. Hyperparameters determined the behaviour and effectiveness of the model while exerting a profound influence on its capacity to obtain knowledge and generate accurate predictions of parameters. The hyperparameter tuning technique was chosen to mitigate the occurrence of underfitting or overfitting phenomena. Several common instances of hyperparameter tunning involve the optimisation algorithm's learning rate, the neural network's number of layers and neurones, the strength of regularisation, the batch size, and the depth of DTs. Each of these parameters carried a pivotal role in shaping the model's performance. The process of hyperparameter tuning consisted of pivotal steps. The first step in this process was to establish a search criterion with a comprehensive range or set of values for each hyperparameter. This process defined the parameters to investigate potential hyperparameter configurations. Subsequently, the second step in this process was to select a search technique that encompassed an array of approaches ranging from the systematic grid search to randomised search. The third and fourth steps of this process were to evaluate the matric and cross validation, respectively.

5.1.4.1 Justification of Hyperparameter Tuning

Hyperparameter tuning significantly enhances ML and DL model performance by optimising the balance between model accuracy, generalizability, and computational efficiency [803-811]. Through the systematic tuning of hyperparameters, the model's decision boundaries can be thoroughly refined to identify even the most complex patterns that indicate the presence of intrusions [812-815]. The tuning process enhances precision and decreases the occurrence of false positives [816], a factor of crucial significance in detection systems, wherein the failure to detect intrusions or the generation of excessive false alerts may pose considerable risks.

In the context of the MLP model, the tuning process primarily emphasises the adjustment of the learning rate and batch size to enhance both the speed of convergence and the stability of the model [817]. Managing the learning rate to an ideal parameter facilitates consistent weight adjustments without overshooting [691, 818]. These tuning adjustments yielded a model capable of more effectively discerning nuanced intrusion patterns with a reduced number of epochs, representing a considerable benefit for identifying complex patterns. The significance of hyperparameters within the MLP model is fundamental, not solely in identifying medjacking patterns but also across various domains, including medical diagnosis [809, 811, 819].

The optimisation of hyperparameters for the DT model generally encompassed the maximum depth and the minimum number of samples requisite for node splits [820, 821]. Tuning the DT not only enhances interpretability but also aids in the identification of attack structures, which has demonstrated effectiveness in discerning the different types of attacks.

The tuning of SVM demonstrates the significance of the regularisation parameter (C) and the kernel function [822, 823]. Through the adjustment of parameter C, which is subject to evaluation within the range of 0.1 to 100, the model can achieve a balance between margin width and classification accuracy, resulting in a resilient framework capable of identifying

anomalies [824], in a comparable manner, SVM can be optimised for threats such as medjacking. The adjustment of hyperparameters significantly improves the sensitivity of the SVM to different attack patterns, thereby diminishing noise and minimising false positives [825]. According to literature illustrates the significance of hyperparameters within the SVM model, particularly concerning the effectiveness of attack detection.

Ultimately, in the context of XGBoost, hyperparameters, including maximum depth, learning rate, and the number of estimators, can be adjusted to allow balanced tree development for discerning complicated attack patterns [826, 827]. Moreover, the adjustment of subsample and feature sampling ratios creates a form of regularisation, thereby facilitating robust generalisation across various attack types [828, 829]. These adjustments can fine-tune the XGBoost model to achieve satisfactory precision in medjacking patterns.

The adjustment of hyperparameters across each ML and DL model resulted in developing a more identifying and resilient detection system capable of identifying complex and medjacking-oriented intrusions with satisfied precision. These tuning enable the models to generalise effectively while preserving a high sensitivity to various complex and advanced attack patterns. Every tuning decision is pivotal in enhancing detection rates, diminishing false positives, and optimising computational efficiency, all essential for establishing a reliable detection system.

5.1.4.2 Grid Search

This study applied grid search, also known as Grid search cross validation (GridsearchCV), that identified a set of hyperparameters for the prediction models. During the grid search process, the model conducted training and testing for every combination of hyperparameters and subsequently measured the accuracy. The ensuing procedure of optimal parameters involved the specification of potential values or alternatives for each hyperparameter. The ranges were carefully constructed to accommodate significant

improvements with the potential to improve the performance of the model. For example, the study examined learning rates at specific constant and adaptive values. The hyperparameter grid is presented in Table 5.5, where different instances were set to identify optimal hyperparameters for the classification tasks in the H-IDS.

Table 5.5

Model and Hyperparameter Grid Used to Identify Optimal Hyperparameters.

Model	**Dimensionality reduction**	**Hyperparameter Grid**	
		Name	**Description**
MLP	PCA, t-SNE, and UMAP	Hidden layers	[(10,), (20,), (59,), (50,50,50), (50,100,50), (100,)]
		Activation	[‘tanh,’‘relu,’‘sigmoid’]
		Alpha	[0.0001, 0.05]
		Learning_rate	[‘constant’,‘adaptive’]
		Cross validation	*k*-fold
DT	PCA, t-SNE, and UMAP	Max_depth	[10,20,30,40,50]
		Min_samples_split	[2,3,4]
		Max_leaf_nodes	[3,4,5,6]
		Criterion	[‘gini,’‘entropy’, ‘log_loss’]
		Splitter	[‘best,’‘random’]
		Cross validation	*k*-fold
SVM	PCA, t-SNE, and UMAP	Degree	[3,4,5]
		Max_iter	[100,150,200]
		kernel	[‘poly,’‘rbf’]
		Gamma	[‘scale,’‘auto’]
		Cross validation	*k*-fold

Model	Dimensionality reduction	Hyperparameter Grid	
		Name	**Description**
XGBoost	PCA, t-SNE, and UMAP	Learning_rate	[0.005,0.2,0.8]
		Num_feature	[8,9,10]
		Max_depth	[50,100,150]
		Booster	[‘gbtree,’‘gblinear,’‘dart’]
		Sampling_method	[‘uniform,’‘subsample,’‘gradient_based’]
		tree_method	[‘auto,’‘exact,’‘approx,’‘hist,’‘gpu_hist’]
		Cross validation	*k*-fold

Table 5.5 presents the hyperparameter grid that functioned as a foundation that comprised all conceivable combinations of the hyperparameters and their corresponding values. The grid served as a navigational aid for the grid search algorithm and facilitated systematic exploration of the parameter space. During the grid search procedure, the algorithm systematically examined all possible combinations inside the grid, conducted model training and testing using the specific hyperparameters, and afterwards summarised the performance outcomes. The optimal configuration for the ML models was determined by selecting the combination of hyperparameters that resulted in optimum model performance.

Cross validation was conducted to identify optimal hyperparameters. Cross validation divided the dataset across various splits to evaluate the results of the model on each section to obtain average results of each iteration. In this study, a *k*-fold cross validation was conducted to assess the generalisation performance and identify optimal parameters. Cross validation involved several iterations wherein each iteration consisted of two phases: training and testing. In each iteration, one of the five folds was designated as the validation set, also known as the test set. The remaining four folds were allocated for model training. The model completed five

cycles of training and testing, computed the accuracy performance, and computed the standard deviation (see Table 5.6).

Table 5.6

The Performance Measure of k-Fold Cross Validation.

Model	Dimensionality reduction	Mean test accuracy	Standard deviation test score
MLP	PCA	89.239	0.006
	t-SNE	90.908	0.010
	UMAP	90.908	0.010
DT	PCA	86.143	0.010
	t-SNE	86.143	0.010
	UMAP	86.143	0.010
SVM	PCA	82.586	0.013
	t-SNE	73.081	0.022
	UMAP	87.631	0.004
XGBoost	PCA	99.158	0.001
	t-SNE	99.014	0.001
	UMAP	93.645	0.001

Table 5.6 presents the *k*-fold cross-validation scores that were computed as the average performance metrics collected over the five rounds. The performance metric provided an in-depth assessment of the model's prospective performance on data that are not known. Cross validation has been considered a more reliable approach for evaluation compared to a single train–test split. The result was validated based on cross-validation outcomes to capture the differences in the data split and reduce the probability of getting a biased outcome.

The optimal hyperparameters were collected for the classification task using a *k*-fold cross-validation grid search. The optimal hyperparameters were determined by comparing the performance of the model across different combinations of hyperparameters and selecting the combination that provided the best overall performance. Table 5.7 presents the optimal hyperparameters.

Table 5.7

The Optimal Hyperparameters for Classification Task.

Model	**Dimensionality reduction**	**Optimal hyperparameters**	
		Name	**Description**
MLP	PCA	Hidden layers	(50, 100, 50)
		Alpha	0.0001
		Activation	tanh
		Learning rate	Adaptive
	t-SNE	Hidden layers	(50, 100, 50)
		Alpha	0.05
		Activation	Relu
		Learning rate	Adaptive
	UMAP	Hidden layers	(50, 100, 50)
		Alpha	0.0001
		Activation	Relu
		Learning rate	Adaptive

Model	Dimensionality reduction	Optimal hyperparameters	
DT	PCA	Max_depth	10
		Min_samples_split	6
		Max_leaf_nodes	2
		Criterion	Gini
		Splitter	Best
	t-SNE	Max_depth	10
		Min_samples_split	5
		Max_leaf_nodes	2
		Criterion	Gini
		Splitter	best
	UMAP	Max_depth	10
		Min_samples_split	6
		Max_leaf_nodes	2
		Criterion	Gini
		Splitter	best
SVM	PCA	Degree	3
		Max_iter	150
		kernel	Poly
		Gamma	Auto
	t-SNE	Degree	3
		Max_iter	100
		kernel	Poly
		Gamma	Scale

Model	Dimensionality reduction	Optimal hyperparameters	
	UMAP	Degree	4
		Max_iter	100
		kernel	Poly
		Gamma	Auto
XGBoost	PCA	Learning_rate	0.2
		Num_feature	10
		Max_depth	50
		Booster	Gbtree
		Sampling_method	Uniform
		Tree_method	Hits
	t-SNE	Learning_rate	0.2
		Num_feature	3
		Max_depth	50
		Booster	Gbtree
		Sampling_method	uniform
		Tree_method	approx
	UMAP	Learning_rate	0.005
		Num_feature	3
		Max_depth	50
		Booster	Gbtree
		Sampling_method	Uniform
		Tree_method	approx

Table 5.7 presents the optimal hyperparameters for the classification task. The process of identifying the optimal hyperparameters involved exhaustively exploring all possible

combinations of hyperparameters within the predefined grid, and each combination was evaluated using cross validation. The optimal parameters led to the highest cross-validated performance score. These hyperparameters were then utilised to train the final model on the complete training dataset, and the performance of the model was assessed on an individual test dataset to ensure its ability to predict effectively.

5.1.4.3 Hyperparameters Limitations

This study acknowledges the limitation of not examining alternative hyperparameter optimisation techniques, including randomised search or Bayesian optimisation. The Grid search technique was chosen for this study due to hyperparameters configuration was unknow to the dataset. In this study, Grid search was used to identify a systematic and comprehensive examination of the hyperparameter space, thereby developing confidence that the most optimal combination of parameters for the specified dataset and models was determined. Although grid search was selected for its systematic and comprehensive examination of the hyperparameter space, alternative techniques may possibly yield similar outcomes. The study did not include the examination of randomised search or Bayesian optimisation techniques. Future studies may focus on the comparative analysis of all hyperparameters within the framework of enhancing the effectiveness of ML models. The rationale for choosing grid search was to provide a systematic and exhaustive examination of the hyperparameter space. According to [813], the random search technique uses a specified range of values and randomly selects parameter configurations to determine the optimal settings. Grid search is primarily advantageous when the optimal parameter configurations remain unidentified, whereas random search is more appropriate for enhancing performance in accordance with user-specified metrics [813].

Grid Search systematically investigates combinations of hyperparameters to determine the optimal configuration settings for ML models [830]. In [831], researchers compared various hyperparameter techniques, specifically greedy search and swarm intelligence, and determined

that optimisation strategies, including random search and grid search, demonstrate significant potential [831]. In the research presented in [832], the researchers conducted a comparative analysis of Grid Search and Random Search techniques for hyperparameter tuning within the framework of the Extreme Gradient Boosting algorithm aimed at predicting chronic kidney failure. The researchers determined accuracy and F-measure values based on hyperparameters. The results have shown that when a dataset was without normalisation and oversampling, the grid search yielded an accuracy of 98.75%, whereas the random search achieved an accuracy of 96.78%. Dataset with min-max normalisation, a grid search that achieved an accuracy of 98.75%, while a random search yielded an accuracy of 96.78%. Dataset with Z-score normalisation, a grid search accuracy of 98.75%, while the random search achieved an accuracy of 96.78% [832].

The randomised search technique, for example, investigates a random subset of the hyperparameter space, thereby enhancing computational efficiency in extensive search domains while concurrently maintaining a substantial probability of identifying near-optimal configurations [833]. Comparably, Bayesian optimisation utilises probabilistic models to systematically enhance the search process through iterative refinement grounded in previous evaluations [834, 835]. The choice of hyperparameters may depend on either improving the model accuracy based on the user's preferences or identifying the best optimal parameters for realistic configuration when parameter configuration is unknown [813].

5.2 The Detection Engine

The H-IDS integrated both anomaly and signature detection techniques, which utilised supervised and unsupervised learning techniques. This system is referred to as hybrid due to the use of anomaly and signature methods during its development as a proof of concept. In cybersecurity, anomaly uses unsupervised techniques while signature uses supervised techniques. The H-IDS development investigated different supervised and unsupervised

techniques separately to provide a systematic investigation of each technique based on different factors such as the hyperparameter tunning and different dimensionality reduction techniques. The H-IDS development provided a mechanism for learning from results. This study investigated each model separately to learn from the results and improve where required in future research.

Anomaly-based detection tested an unsupervised ML technique, where the DL method, namely the autoencoder, was trained and tested to detect any abnormalities or deviations from the established typical behaviours acquired during the training phase of the autoencoder. Signature-based detection tested supervised ML techniques in which different supervised ML models were tested separately to identify the best model for the hybrid approach. The signature detection approach utilised ML and DL models that were trained using the 2022APS dataset.

The effectiveness of pattern classification and discrimination between normal and attack occurrences was assessed by calculating the performance of both detection strategies for the APS prototype. The hybrid method proved effective in detecting the vulnerabilities that are commonly linked with medjacking. The H-IDS technique enabled the system to rapidly adapt to constantly evolving threat environments while also preserving its capacity to identify persistent threats through the application of existing signatures.

5.2.1 Anomaly Detection

In the formulation of a proof of concept for the detection of anomalies in the APS. This study used an autoencoder, a type of neural network unsupervised learning. Employing Mean Squared Error (MSE) as the principal evaluative metric, this proof of concept was examined to determine the autoencoder's effectiveness in identifying attack behaviours through the reconstruction of normal data patterns. The MSE is the principal metric for detecting anomalies, quantifying the difference between the original input and the reconstructed output. Critical performance metrics such as training accuracy, validation accuracy, training loss, and

validation loss provided the model's effectiveness throughout the training phase. In this proof of concept, five distinct steps were undertaken: 1. data collection and processing, 2. development of the autoencoder model, 3. training of the model, 4. anomaly detection utilising mean squared error, and 5. evaluation.

The process of anomaly identification utilised an autoencoder with the TensorFlow framework. The initial step was the acquisition of the 2022APS dataset, which comprised normal and attack patterns. The collected data was subsequently processed through preprocessing; in this process, MinMaxScallar was applied to transform the 2022APS dataset.

In step two, the design of the autoencoder model, the model development comprised an input layer, an encoder, a bottleneck layer, a decoder, and an output layer. The encoder effectively condenses the input data into a reduced-dimensional space, capturing key behaviour patterns. The bottleneck layer functions as a condensed representation of the input data. The decoder subsequently reconstructs the input data from this condensed representation while the output layer attempts to align with the original data. The autoencoder was trained to minimise the MSE, a metric that quantifies the average squared deviation between the original input and its reconstruction.

In step three, the model training, upon establishing the autoencoder architecture, the subsequent step involved training of the model utilising the training dataset comprised of normative data. The dataset was subsequently split into two subsets: a training set and a test set. The training set consisted of normal patterns, whereas the test set had a combination of both normal and anomalous patterns. The autoencoder architecture was designed to align with the distinct features presented by the 2022APS dataset. The autoencoder model was developed using a loss function, such as MSE, and an optimiser, such as ‘adam.’

The subsequent phase involved training the autoencoder using the 2022APS dataset training set. During the training process, the neural network aimed to minimise the MSE loss

function, which served as a measure of the difference between the input data and the reconstructed output derived by the autoencoder. To detect anomalies, the loss function for an autoencoder was applied to set the threshold for the anomaly detection approach. The MSE loss function was used in the autoencoder to measure the differences between the original input and the remodelled output. The MSE loss function is presented in Equation 5.8:

$$MSE = 1/n * sum((x\_i - x'\_i)\text{^}2) \quad (5.8)$$

Where n is the number of samples in the dataset, x_i is the original input, and x'_i is the remodelled output. The goal of training the autoencoder was to minimise the MSE loss function by balancing the weights and biases of the DL.

Step 4: Anomalous detection utilising MSE, the trained autoencoder employed on the test set, encompassing both normal and anomalous data instances. The reconstruction error was quantified as the MSE computed for each instance. Following the completion of the training process, the model completed evaluation using the test set. The autoencoder completed training and was employed to reconstruct the data originating from the test set. Subsequently, the reconstruction error was calculated for each data point utilising the mean MSE metric. Step 5: Evaluation of the developed model assessed in accordance with the subsequent performance metrics, Training accuracy, Training validation, training loss, and validation loss are discussed in Chapter 6.

5.2.1.1 Justification of Autoencoder

The autoencoder model provides significant adaptability and can be utilised across various domains, such as cybersecurity [836], healthcare [837], manufacturing [838], and finance [839]. In the context of cybersecurity, autoencoders serve as an essential application for anomaly detection, especially in recognising unusual patterns that could indicate anomalous behaviour. For example, in a study [730], the researcher developed an anomaly detection model

utilising an autoencoder to identify malicious attacks in cyber-physical systems. The researchers employed MSE to detect anomalous behaviour. Similarly, in another study [840], the researchers focused on an anomaly detection model that utilised an autoencoder for identifying probing and DDOS-Smurf attacks. Another study in [841] also conducted a study on autoencoders and proposed anomaly detection in simulated nuclear data. Current literature shows that autoencoders are considered a competitive method, particularly in the field of anomaly detection. Conventional anomaly detection techniques such as K-Means may be obstacles when dealing with high dimensionality of features or complex relationships among features [842].

Autoencoders demonstrate the feasibility of reconstruction-based anomaly detection throughout the training process to reconstruct patterns. Deviations from established patterns lead to increased reconstruction errors in the output phase. This anomaly detection approach operates unsupervised, indicating that it does not necessitate labelled data for its functionality. The capability to identify anomalies through reconstruction errors positions autoencoders as a feasible choice, particularly in contexts where labelled anomaly data is limited or insufficient. Another reason autoencoders demonstrate strong performance is their scalability with neural networks. When constructed with deep neural network architectures, autoencoders can manage extensive datasets comprising variables, effectively capturing a diverse range of normal behaviours.

5.2.1.2 Rationale for Autoencoder

The autoencoder model defines a paradigm of unsupervised learning [843]. Autoencoders, functioning as unsupervised deep learning models, acquire efficient representations of data through the compression of input data into a latent space, subsequently reconstructing it [844, 845]. The learning capacity of this autoencoder is paramount for detecting anomalies within internet traffic, including MITM attacks, DDoS attacks, brute force

attempts, and trojan intrusions, all of which frequently manifest in complex patterns that diverge from conventional data patterns. Autoencoders provide substantial benefits in identifying cybersecurity threats, attributable to their intrinsic architecture and operational efficacy in tasks pertaining to anomaly detection. For instance, one investigation [730] formulated an anomaly detection model using an autoencoder, while another research [840] similarly designed an anomaly detection model utilising an autoencoder.

In hijacking instances, attackers may illicitly acquire access to sessions, IP addresses, or alternative communication channels [846-848], frequently exploiting user credentials or system privileges to evade detection. Hijacking attacks are generally delineated by divergences from established patterns of behaviour, encompassing irregular login instances, unforeseen session lengths, and unusual resource access [849]. The autoencoder model demonstrates important effectiveness attributable to its fundamental architecture: it is systematically trained to reconstruct standard patterns precisely. In the context of intrusion detection, autoencoders exhibit a capacity to reconstruct legitimate activities with considerable precision, thereby yielding a minimal reconstruction error. In the framework of using autoencoders to detect cyber-attacks, the MSE metric assumes a crucial function as the principal measure of reconstruction error, facilitating the identification of anomalous patterns that may indicate an attack [850-852]. Throughout the training phase, the autoencoder is refined to minimise the mean squared error between the input and the reconstructed output. The autoencoder achieves minimal MSE values for normative data patterns, signifying that the reconstructed data strongly resembles the original dataset.

An additional benefit of autoencoders lies in their capacity for scaling across diverse data types [853, 854], providing the autoencoder model particularly adept at identifying many hijacking attack variations. Hijacking attacks can manifest across various network protocols and user interactions [855], and the autoencoder demonstrates effectiveness in managing

multiple data inputs, including IP packet attributes, session characteristics, and user authentication information.

The autoencoder's ability in unsupervised learning represents a significant benefit in the field of intrusion detection, particularly in scenarios where attack data is limited or unlabelled [856-858]. This is particularly relevant considering the frequently limited availability of labelled data for certain cyber-attacks, such as hijacking incidents. Unsupervised training enables the autoencoder to effectively illustrate standard behaviour without dependence on explicit labels, a critical capability given that hijacking attacks might manifest as entirely novel patterns that were not encountered during the training phase. Consequently, an autoencoder can identify novel or variant hijacking attempts, even when such variations have not been explicitly annotated during the model's development phase.

5.2.1.3 Autoencoder Justification for APS

The autoencoder model presents considerable benefits in identifying medjacking attacks, especially within the context of APS devices. The autoencoder model involves the compression and reconstruction of data throughout the training process [844, 845], as an autoencoder is an unsupervised neural network [813]. Thus, insights are acquired into the basic structure and normal behavioural patterns inherent in APS device data. In the context of insulin pumps and CGM devices, the observed data patterns may signify systematic trends in the management of diabetes or the interactions with associated devices.

One of the fundamental techniques used with the autoencoder model for the anomaly detection objective involves assessing reconstruction errors, which are frequently quantified through the metric of MSE. The MSE metric is pivotal as the primary indicator of reconstruction error, enabling the detection of anomalous patterns that may signify an attack [850-852]. Through the processes of data compression and reconstruction training, the autoencoder formulates a model that aims to minimise reconstruction errors. Any variations

from this acquired behaviour, such as unauthorised access or commands generated by medjacking, result in an increased reconstruction error, signifying an anomaly.

Moreover, the autoencoder model effectively recognises general anomalies and discerning complex deviations from established data patterns, using MSE as a pivotal metric [579, 838, 840, 841, 850, 851, 859]. The autoencoder model demonstrates exceptional feasibility in discerning deviations from the established patterns within the compressed data representation, thereby facilitating the detection of subtle anomalies. For instance, an unforeseen increase in command frequency or unusual connection attempts to the pump or CGM may precipitate a visible reconstruction error, thus facilitating the precise identification of attacks.

Another merit of the autoencoder model lies in its capacity for adaptation to the dynamic environments of the IoMT. Devices within the IoMT frequently undergo firmware updates [105, 107], communicate with other devices, and facilitate the integration of new devices, sensors, applications, and solutions within the ecosystem [254]. Furthermore, IoMT devices might encounter new data types or experience data diversity [860, 861]. The autoencoder model can be retrained or fine-tuned [862] on updated data, thereby facilitating its adaptation to evolving usage patterns or emerging modalities of medjacking attacks. Utilising MSE as the metric for anomaly detection, the autoencoder's inherent adaptability facilitates its capacity to identify emerging threats, thereby sustaining a proactive security paradigm as threat models progress in line with innovations in IoMT technologies.

5.2.1.4 Results and Performance of the Autoencoder Justification

The performance metrics of the autoencoder model, encompassing training accuracy, validation accuracy, training loss, and validation loss, indicate the effectiveness of the autoencoder in anomaly detection. Numerous studies have investigated the identification of anomalies through the application of autoencoders across various contexts, including medical

diagnosis, image classification, fault detection, and security breaches [851, 863-869]. Similarly, these performance metrics can also be employed to assess the model's effectiveness in relation to medjacking attacks on IoMT devices, such as insulin pumps and CGMs. Each metric illustrates the model's generalisation capabilities and robustness to deviations that may signify a security breach. Utilising these performance matrices, researchers can evaluate the effectiveness with which the autoencoder recognises complex anomalies that represent medjacking attacks, wherein illicit commands or access endeavours compromise the expected functionality of the device.

This study evaluated autoencoder performance, encompassing both training and validation accuracy as well as loss metrics. The training accuracy assessed the autoencoder's ability to reconstruct the training data accurately, while the validation accuracy examined its ability to distinguish between normal and attack data. The training loss quantifies the difference between the original training dataset and its corresponding reconstructions during the training procedure. Comparably, the validation loss evaluated the difference within the validation set, encompassing both normal and attack instances. A decreased validation loss denotes the autoencoder's ability to differentiate between normal and attack inputs, simultaneously decreasing the loss associated with normal data during the training phase. The findings of the autoencoder model employed for detecting medjacking attacks will be discussed in section 6.1, titled Anomaly Detection Performance.

5.2.2 Signature Detection

In this study, signature detection used supervised ML and DL techniques to detect intrusions. The supervised ML and DL techniques involved feature engineering to select, transfer and create input data variables. The 2022APS dataset was utilised for signature detection during the ML and DL model's development. To enhance detection effectiveness, feature engineering was required to convert unstructured data into a structured form, and to

determine the most useful features. Finally, structured form data were transformed into more effective patterns.

Feature selection, dimensionality reduction, and hyperparameter techniques were used to transform the dataset for the next phase. This study focused on the most relevant features, which enabled the model to perform optimally to detect intrusions. The sub-phase of feature engineering was dimensionality reduction, which reduced the dimensionality while maintaining important information. A grid search technique was used to identify hyperparameters. Ultimately, classification models were essential for signature detection.

The ML and DL models such as DT, SVM, MLP and XGBoost can be used to detect the known signatures as discussed in Table 2.7 (Chapter Two), every ML and DL model in the cybersecurity field has its own set of trade-offs. Selecting the right model requires considering interpretability, processing resources, and task complexity. In this study, four classification models, MLP, XGBoost, DT, and SVM, were trained, tested, and compared to evaluate the effectiveness of the H-IDS. The models were trained on a training set to assign class labels to new instances. During the testing, performance matrices evaluated the effectiveness of the model and proved the model's suitability. The testing phase played an essential part in validating the accuracy and precision of the model's forecasting.

5.2.2.1 Justification of Signature Detection

According to [622], signature-based techniques are instrumental in safeguarding against various attacks, and these techniques are initially assessed and preserved on the device within the IoT, with each signature associated with a particular type of attack. Signature-based techniques are commonly used and require a distinct signature for each attack [622]. The implementation of signature-based detection can be considered as a low-complexity technique for resource-constrained IoT devices. In [870], the researchers proposed a signature-based technique for the detection of malware within IoT environments. The researchers stated that

signature-based detection enhanced the system by identifying and storing common signatures associated with a malware group. The signature-based detection technique shows numerous advantages compared to current solutions, encompassing elevated accuracy, no false alarms, and reduced memory resource requirements [870]. According to [871], the most common technique for malware detection is signature-based detection, which is esteemed for its minimal false alarm rate, rendering it crucial and extensively employed in anti-malware tools [871].

The incorporation of signature detection within an APS system, which is principally designed to identify medjacking attacks, including zero-day vulnerabilities like medjacking, is fundamentally rooted in its function as a foundational layer that strengthens and increases the comprehensive detection architecture. In establishing foundational defence mechanisms, signature detection proficiently discerns and minimises recognised threats. By tackling recognised attack patterns, the system is equipped to eliminate common and repetitive assaults (such as brute force and DDoS), which enables ML and DL models to focus solely on the more specific and unusual anomalies that may signify zero-day threats. As indicated in [872], signature-based components consistently demonstrate a significant degree of accuracy in identifying known threats, thereby rendering them an indispensable asset within any intrusion detection system [872]. In [873], the researchers investigated the foundational performance of a signature-based intrusion detection system, aiming to evaluate the advantages and disadvantages of the ML-based technique. The researchers replicated a research study, using a deep neural network model for intrusion detection. The researcher suggested a comparative analysis of ML models to assess the broader relevance of ML techniques [873]. This study designed and compared four ML models with three dimensionality reduction techniques for the medjacking in the APS, as a proof of concept for first baseline protection.

In the context of rapid attack detection, signature detection using ML approaches illustrates excels in determining established attack patterns, a capability that proves

especially advantageous in scenarios where known vulnerabilities may be present. The rapid attacks detection of frequently known threats can significantly minimise the risks associated with acknowledged threats. In the context of low false positive rates, signature based with ML techniques frequently observed exhibit greater accuracy. As highlighted in [874], signature-based detection has long been the preferred approach among researchers and security professionals due to its numerous advantages. Signature-based detection demonstrates effectiveness in identifying established worms, viruses, and instances of attacks while maintaining an exceptionally low rate of false positives [874]. Although signature detection has conventionally concentrated on established patterns, its integration within a zero-day detection system is pivotal for establishing fundamental security, enhancing resource efficiency, and strengthening a comprehensive, multi-layered defence architecture. These factors, in collaboration, improve the system's capacity to identify both known and unknown medjacking threats within the APS.

In this research, the H-IDS using ML and DL techniques is developed and tested as a proof of concept, wherein the anomaly detection component was trained to identify deviations from normal behaviour. The anomaly detection methodology did not involve feature engineering to the 2022APS dataset. The anomaly detection component facilitates the model's autonomous evaluation of unknown attacks, as feature engineering has not been implemented, thereby allowing it to recognise anomalies that do not conform to established patterns, thus enabling the detection of zero-day vulnerabilities. In contrast, the signature-based component functions primarily as an initial filter for recognised medjacking, wherein feature engineering is employed to identify pertinent features. The study outlines a proof of concept and a preliminary validation of the proposed techniques. The proof of concept elucidates the essential functionality and establishes a basis for subsequent advancement.

5.2.3 Model Development Pipeline

The pipeline performed a crucial role in the model development by outlining the steps involved in feature classification tasks, such as training and testing sets, and hyperparameter tuning. The process involved extracting features and subsequently dividing the feature data into a training set (75%) and a testing set (25%). The training set data were utilised to train the model, whereas the test data were used to assess the performance of the trained model. After dividing the data into two parts, the mode carried out a basic normalisation process. Table 5.2 illustrates the data samples and details for training and testing splits.

Table 5.8

The Data Samples and Split Details

Data sample	**Total samples**	**Training**	**Testing**
Insulin	18565	13923	4641
CGM	18227	13670	4556
Brute force	18313	13734	4578
DDoS	18141	13605	4535
MITM	18012	13509	4503
Trojan	18102	13576	4525

The purpose of normalisation was to ensure that all features utilise a standardised scale. This split training and testing set strategy was applied for anomaly and signature detection. One of the challenges encountered while collaborating with different ML classifiers was the optimisation of hyperparameters. Hence, it was imperative to determine the hyperparameter that would result in the optimal classification performance for both normal and attack classification. To avoid the issue of classification overfitting, this study employed grid search cross-validation to fine-tune the hyperparameters for each classification model. During the grid

search, the dataset was divided into training and testing sets with 75% allocated for training and 25% for testing. The hyperparameters were cross validated using a *k*-fold approach, and the mean accuracy was calculated for the optimal hyperparameter.

This study used cross-validation in sections 5.1.2.1.3 and 5.1.4, to find the optimal features, and to determine the most appropriate hyperparameters, respectively. However, after identifying the most effective features and hyperparameters, cross-validation was not conducted on the classification models. The choice was made because the grid search method determines the optimal combination of hyperparameters that produces the highest average performance across the cross-validation folds.

5.2.4 Summary

The development of H-IDS was accomplished in this chapter. The feature selection process was applied to choose a subset of pertinent features. Utilising feature selection techniques significantly enhanced the performance of the H-IDS by reducing data dimensionality and eliminating redundant or irrelevant features. After the selection of optimal features, dimensionality reduction techniques were applied to reduce the number of features while still maintaining essential information. The process of model preparation involved fine-tuning the hyperparameters to mitigate the risk of overfitting or underfitting scenarios. A grid search cross-validation technique was utilised to thoroughly explore a designated subset of hyperparameter combinations to identify the most optimal configuration for an ML model. After identifying the optimal features, the ML classifiers were applied to predict the intrusion cases. These classifiers acquired and used knowledge from the input data to make predictions about the class labels of unseen instances. To identify anomalies, an autoencoder was utilised to train on unlabelled datasets, enabling them to acquire a compressed understanding of typical behaviour. The deviation from acquired knowledge revealed the existence of anomalies.

ML and DL models, such as autoencoder, MLP, DT, SVM, and XGBoost, can autonomously identify intrusions in APS devices. These models efficiently acquire knowledge of patterns to detect anomalies. Within the context of anomaly detection, the autoencoder model can compress and accurately reconstruct normal operational data from APS devices as demonstrated in this study. Signature detection involved training classification models such as MLP, DT, SVM, and XGBoost to automatically identify attack patterns using the 2022APS dataset containing both normal and attack behaviour. This study primarily aimed to demonstrate the feasibility of automatically detecting intrusions using ML and DL models in simulated APS devices. ML and DL models possess the capability to make intelligent decisions based on recognisable patterns, enabling them to differentiate between normal and attack patterns. Ultimately, the specific production preferences are determined by the manufacturer or software development company, who may decide which logs, user interface, and graphical interface to incorporate. The fundamental idea was to differentiate between normal and attack patterns using ML and DL models without any human supervision. There is an opportunity for improvement in the detection models, depending on the preferences of APS manufacturers or software development for the final production.

Chapter 6. Analysis and Comparison of Results

Chapter 6 reports the analysis of results and comparison of the performed experiments in the Hybrid Intrusion Detection System (H-IDS). This analysis is an essential component in extracting valuable insights, validating each test, and engaging in quantitative data-driven choices across the context of scientific pursuits. This assessment allows for an estimation of strengths and weaknesses in various H-IDS configurations, such as different dimensionality reduction techniques, that improve current methods while developing optimal detection strategies. The effectiveness and efficiency of H-IDS were evaluated using quantitative metrics that provided objective measures: accuracy, precision, recall, and the F1 score. Data visualisation helped to present the findings in a visually captivating and easily understandable way, making it easier to spot patterns, trends, and anomalies in the data. This chapter will specifically answer the third, fourth and fourth (a) research questions:

3. What machine learning and deep learning techniques as proactive defences techniques can be used to detect medjacking vulnerabilities in an APS?
4. What dimensionality reduction technique can be performed optimally in the development of a machine learning model for medjacking detection in the APS?
    a. What differences can be experienced with and without dimensionality reduction techniques in the development of a machine learning model for medjacking detection in the APS?

6.1 Anomaly Detection Performance

An autoencoder performance evaluation, consisting of training and validation accuracy and loss, was computed in this study. Training accuracy evaluated the autoencoder's effectiveness in accurately recreating the training data, whereas validation accuracy evaluated its effectiveness in determining between normal and attack data. Training loss measured the

difference between the original training data and their reconstructions throughout the training process. Similarly, validation loss assessed the difference for the validation set, which comprised both normal and attack cases. A low validation loss signifies the autoencoder's accurate discrimination between normal and attack input, while simultaneously minimising the loss for normal data throughout the training process.

Four medjacking attacks, brute force, DDoS, MITM, and Trojan attacks, were tested using an autoencoder technique to detect the anomalies. The unlabelled dataset was utilised in this process. As discussed in Chapter 5, the 2022APS dataset consisted of normal and attack instances, and the dataset was subsequently split into two subsets: a training set and a test set. Four experiments were performed, and each test computed the performance of the autoencoder. In experiment 1, normal and brute force attacks were examined; the autoencoder consisted of an encoder and a decoder. The autoencoder model used a Mean Squared Error (MSE), and an adam optimiser to detect the anomalies, the MSE loss function was used in the autoencoder to measure the differences between the original input and the remodelled output. Experiment 2 examined the DDoS attacks, Experiment 3 examined MITM attacks, and Experiment 4 examined Trojan attacks.

6.1.1 Anomaly Detection Brute Force Attacks

In the brute force attack, the autoencoder focused on learning knowledge about the distinguishing features of valid login attempts, and the autoencoder subsequently detected anomalies by measuring the disparity between the initial data and reconstructed data forms. The autoencoder effectively identified brute force attacks when the login attempts made by an attacker deviate from the learned pattern of normal logins. The training and validation accuracy are shown in Figure 6.1.

Figure 6.1

Anomaly Detection Brute Force Attacks Training and Validation Accuracy.

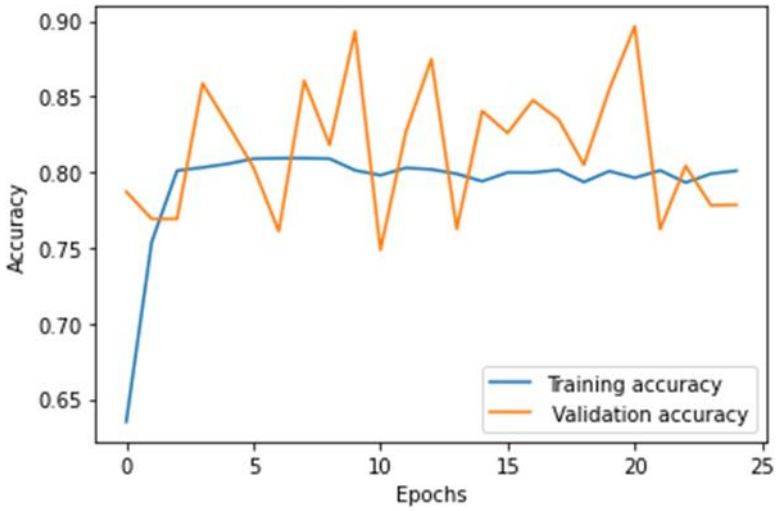


Figure 6.1 illustrates training accuracy in approximately the 80%–82% range and validation accuracy in the 76%–88% range over 25 epochs. The training accuracy serves as an indicator of the model's aptitude in accurately reproducing typical instances from the training dataset. The validation accuracy assessment computed the ratio of correctly reconstructed instances, encompassing both normal and anomalous data. The high validation accuracy indicated the model's adeptness in reconstructing both normal and anomalous instances and effectively discerning anomalies.

In the context of anomaly detection using an autoencoder, the primary metrics employed for evaluation included training and validation loss. The training loss metric quantified the difference between the original and reconstructed training data. The assessment of validation loss, computed on normal and brute-force attack data, served as an indicator of the model's ability to generalise novel and previously unseen data. The training loss and validation loss are shown in Figure 6.2.

Figure 6.2

Anomaly Detection Brute Force Attacks Training and Validation Loss.

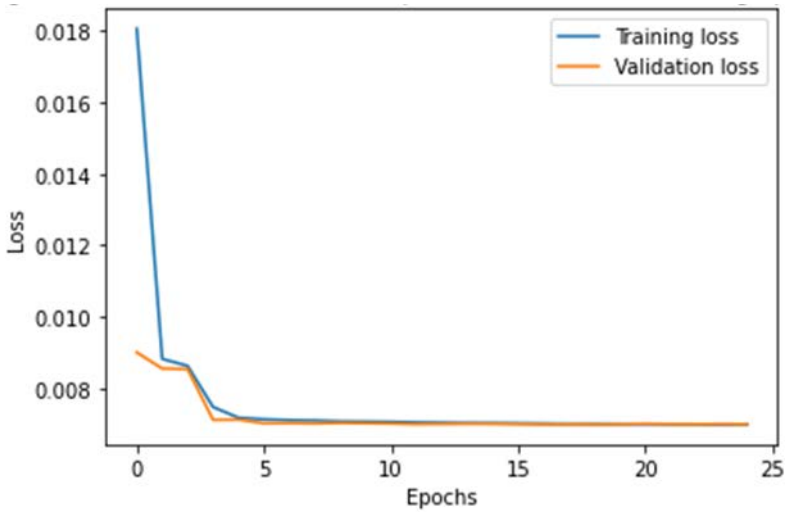


Figure 6.2 illustrates the training loss ratio in approximately the 0.080–0.004 range and the validation ratio in the 0.018–0.004 range over 25 epochs. The outcome minimised both losses and ensured that no overfitting occurred in the autoencoder model. The low losses also indicated the autoencoder's capacity to effectively capture typical patterns and discern anomalies within the 2022APS dataset.

6.1.2 Anomaly Detection Distributed Denial-of-Service Attacks

The process of anomaly detection brute force attacks computed and computed the performance; similarly, the normal patterns and DDoS attack patterns, the learned patterns during training, and the testing of the 2022APS dataset were evaluated in this experiment and computed the performance. The process of monitoring the reconstruction error during the training phase successfully detected deviations. Training and validation accuracy are shown in Figure 6.3.

Figure 6.3

Anomaly Detection DDoS Attacks Training and Validation Accuracy.

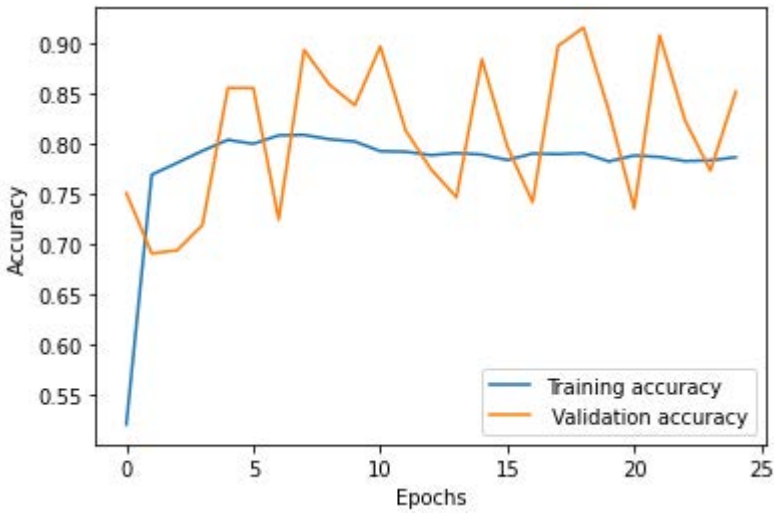


Figure 6.3 depicts a training accuracy of 78%–82% and a validation accuracy of 70%–88%. The model's training accuracy has shown its ability to reproduce typical cases from the training dataset. The validation accuracy evaluation measured the ratio of normal and anomalous cases that were properly reconstructed. The model's high validation accuracy showed its ability to recreate normal and anomalous occurrences and identify abnormalities. The autoencoder model's performance for anomaly detection, using the 2022APS dataset, computed the training loss and validation loss as shown in Figure 6.4.

Figure 6.4

Anomaly Detection DDoS Attacks Training and Validation Loss.

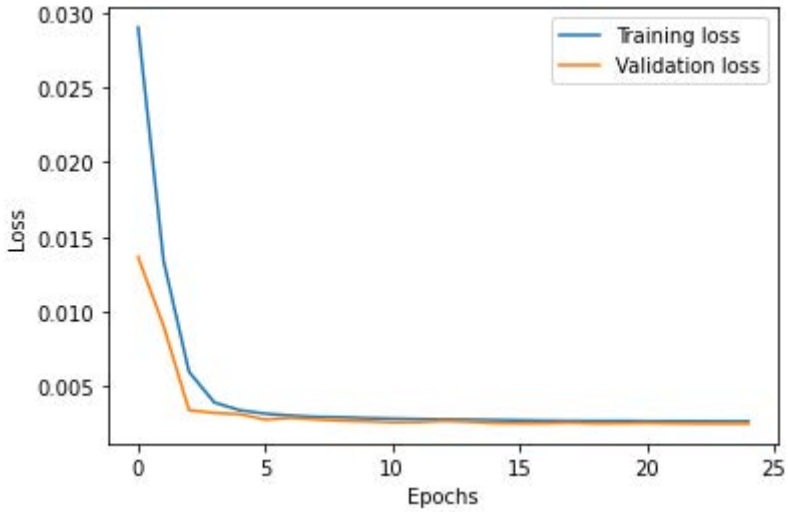


Figure 6.4 shows a training loss ratio of 0.030–0.003 and a validation loss ratio of 0.014–0.003 over 25 epochs. The autoencoder's minimal losses showed its ability to capture common patterns and identify dataset anomalies.

6.1.3 Anomaly Detection Man-In-the-Middle Attack

The model’s encoder and decoder were trained. Reconstruction error was monitored during training, and a threshold analysis detected anomalies. Instances beyond the threshold were identified as an attack. The training accuracy and validation accuracy are shown in Figure 6.5.

Figure 6.5

Anomaly Detection MITM Attacks Training and Validation Accuracy.

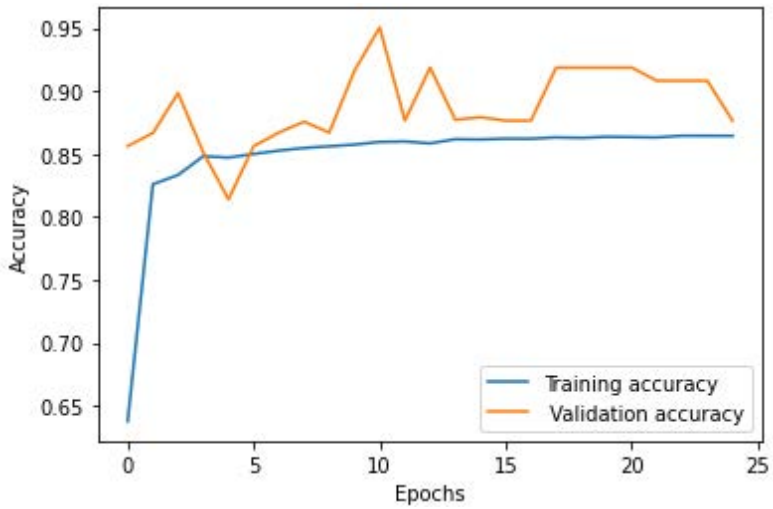


Figure 6.5 shows a training accuracy of 75%–82% and a validation accuracy of 85%–94%. The training accuracy indicated the model could recreate typical situations from the training dataset. Figure 6.6 shows the autoencoder model's anomaly detection performance, computed training, and validation losses.

Figure 6.6

Anomaly Detection MITM attacks Training and Validation Loss.

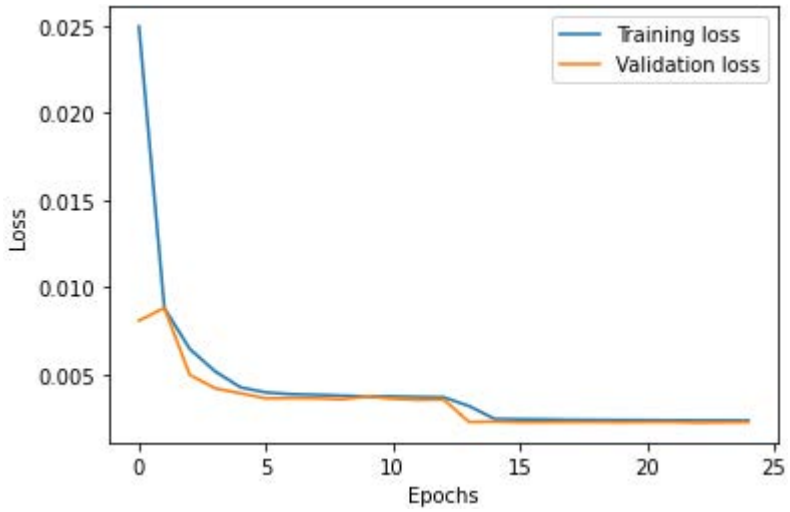


Figure 6.6 illustrates the training loss ratio of 0.080–0.003 and the validation loss ratio of 0.025–0.003 over 25 epochs. The autoencoder model's minimal losses exemplified its effectiveness in capturing patterns and discerning anomalies within the dataset.

6.1.4 Anomaly Detection Trojan Attacks

In the last experiment, the autoencoder model was evaluated for its effectiveness in detecting both normal behaviour and Trojan attacks. The foundation of a predetermined threshold facilitated the detection of possible Trojan attacks during real-time analysis. Instances that exceed the established threshold could indicate the existence of a Trojan attack. The training accuracy and validation accuracy are shown in Figure 6.7.

Figure 6.7

Anomaly Detection Trojan Attacks Training and Validation Accuracy.

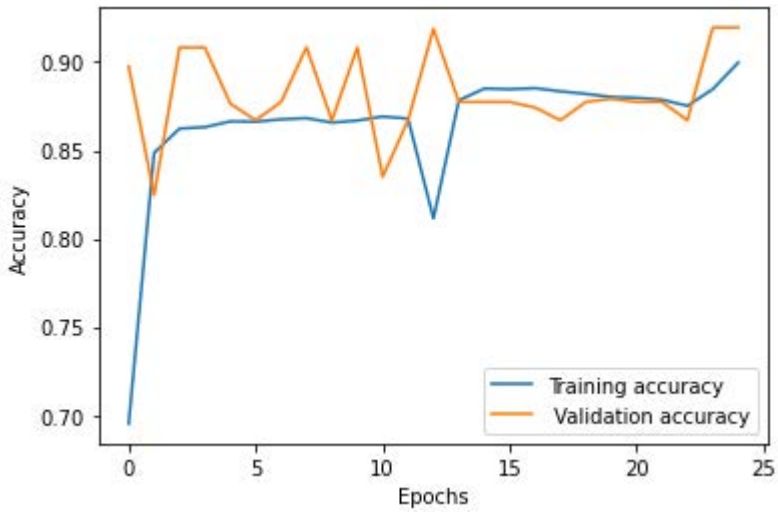
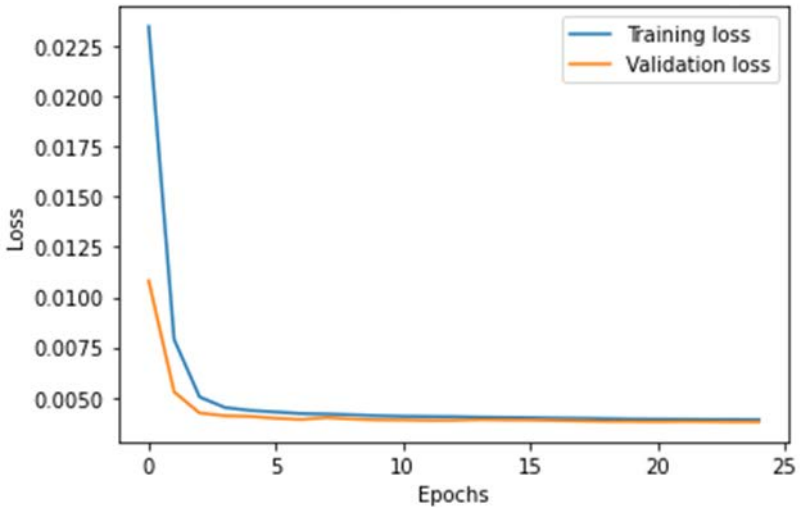


Figure 6.7 shows a training accuracy of 75%–91% and a validation accuracy of 83%–92%. The training accuracy indicates that the model could recreate typical situations from the training dataset. The model's high validation accuracy indicated it could replicate normal and abnormal events and identify anomalies. Figure 6.8 shows the autoencoder model's anomaly detection performance using the normal and DDoS dataset to compute training and validation losses.

Figure 6.8

Anomaly Detection Trojan Attacks Training and Validation Loss.



The autoencoder model's performance metrics were evaluated for different medjacking attacks. With the brute-force attack patterns versus normal patterns, the model predicted anomalies with accurate pattern reconstruction and low training and validation losses. Similarly, the DDoS, MITM, and Trojan attacks generated low losses. The autoencoder exhibited consistent effectiveness in terms of accuracy and loss across a range of attacks, thereby indicating its versatility in detecting anomalies and maintaining resilient performance in normal instances.

6.2 Signature Detection Performance

Signatures are unique characteristics associated with normal and medjacking patterns; the ML and DL techniques were implemented and tested for automated detection of intrusions. The 2022APS dataset was used for the development of the ML and DL models. The dataset was transformed using MinMaxScalar and LabelEncoder; feature selection was made, dimensionality reduction was performed, and grid search cross-validation techniques were applied to identify optimal hyperparameters. These processes were used to prepare classifiers for prediction intrusions. Automated responses to detected medjacking patterns were made possible through classification. The classifiers accurately predicted the intrusion patterns, and

their performance was thoroughly evaluated using performance metrics. As discussed in Chapter 5, the MLP, XGBoost, DT, and SVM classification models were tested and compared for the H-IDS. The accuracy, precision, recall, and F1 scores for each classifier were computed and compared with different classifiers to determine which classifier performed optimally for the H-IDS. The results from testing the four classification models are discussed in Sections 6.2.1, 6.2.2, 6.2.3, and 6.2.4.

6.2.1 Multi-Layer Perceptron

Preparing the MLP classifier involved feature engineering, including data transformation, feature selection, and dimensionality reduction of the 2022APS dataset. The dataset was then divided into separate sets for training, validation, and testing. The training set was utilised to train the model, while the validation set was utilised to fine-tune hyperparameters and evaluate model performance during training. Ultimately, the test set was utilised to evaluate the performance of the trained model. The MLP classifier performance was computed based on the non-dimensionality reduction and with PCA, t-SNE, and UMAP dimensionality reduction techniques to compute the differences and performance of each technique. The MLP model was compared with 10 PCA components and 14 PCA components to examine the impact of dimensionality reduction on the outcomes. Furthermore, the MLP model was evaluated in the absence of dimensionality reduction techniques. These five experiments are outlined below.

6.2.1.1 Multi-Layer Perceptron with 10 PCA Components

In the MLP model development, PCA (10 components) was used for dimensionality reduction, grid search was used for hyperparameter optimisation to validate optimal features, and TensorFlow for DL capabilities was used in the preparation of MLP classifiers to detect intrusions. The 2022APS dataset for this study was prepared by following a series of steps.

First, the data were read from a CSV file using pandas. Next, the data were split into features (*x*) and the target variable (*y*). Finally, the dataset was further divided into training and testing sets using the split function from the scikit-learn framework. This study incorporated various preprocessing steps and the MLP model into the construction of the model pipeline. The MLP model was built using the TensorFlow framework with DL capabilities. The pipeline included a standard scaler for scaling the features and PCA (10 components) for reducing dimensionality. This study outlined the parameters that need to be adjusted during the grid search, as discussed in Chapter 5. The outcome of the MLP classifier's confusion matrix using PCA (10 components) and optimised hyperparameters is shown in Figure 6.9.

Figure 6.9

Confusion Matrix Results of Multi-Layer Perceptron with 10 PCA Components.

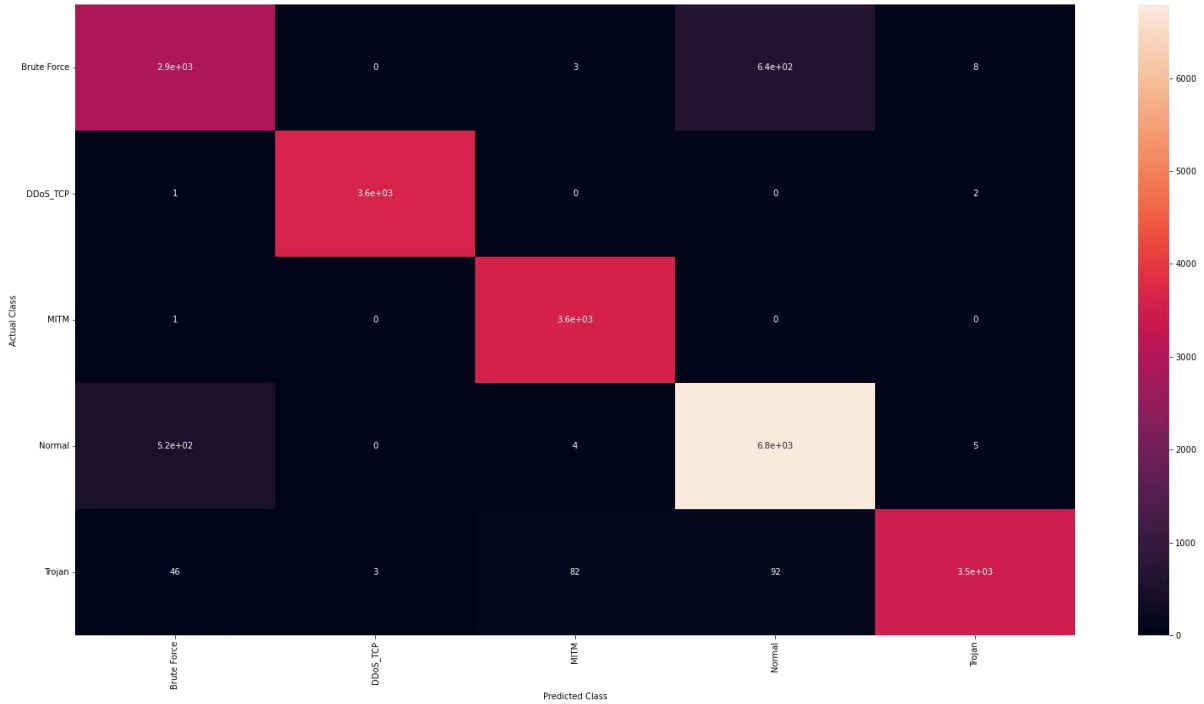


Figure 6.9 depicts the confusion matrix of the MLP model using PCA (10 components), typically represented as a five-by-five matrix, where the actual class labels exist in a row, and the predicted class labels exist in the columns. The four confusion matrices: TP, FP, TN, and FN, were obtained to examine the model's performance. The overall performance of the MLP

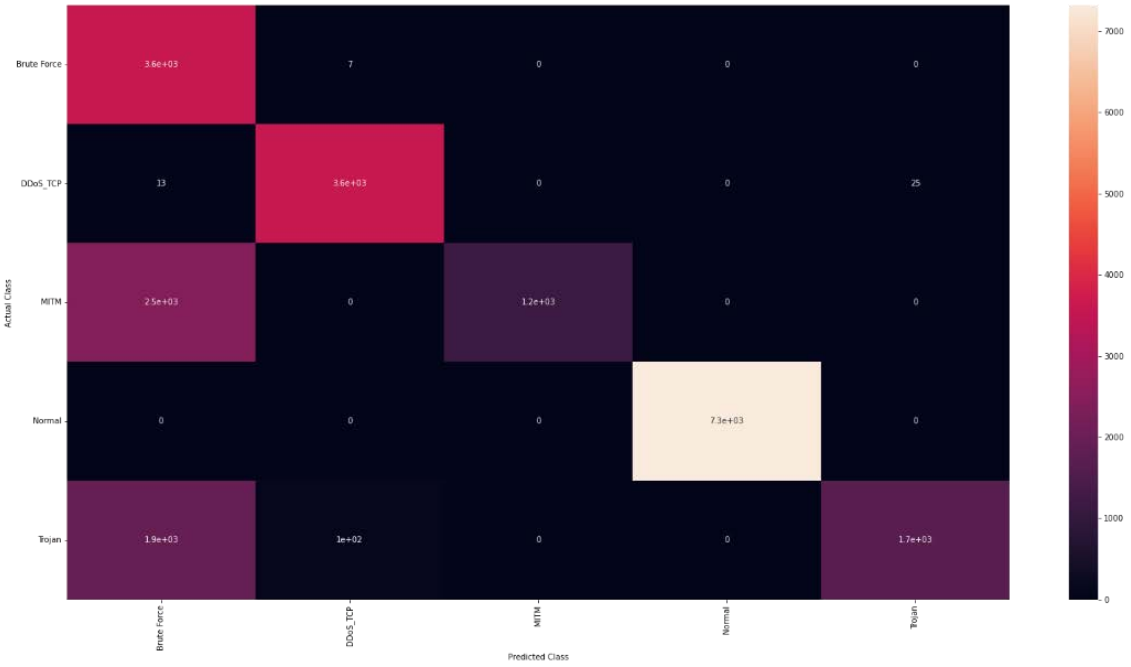
classifier with PCA computed accuracy was 79.142, precision was 89.868, recall was 79.142 and the F1 score was 78.262. These performance matrices were computed using the four confusion matrices.

6.2.1.2 Multi-Layer Perceptron with 14 PCA Components

Section 6.2.1.1 evaluated the MLP model's performance using 10 PCA components. This section compared the effectiveness of the MLP model using 14 PCA components to determine how PCA dimensionality reduction impacted the findings. Figure 6.10 depicts the MLP model's confusion matrix with 14 PCA components.

Figure 6.10

Confusion Matrix Results of Multi-Layer Perceptron with 14 PCA components.



The confusion matrix of the MLP model using 14 PCA components is shown in Figure 6.10, displayed as a five-by-five matrix, with the predicted class labels in the columns and the actual class labels in the row. The four confusion matrices TP, FP, TN, and FN were gathered to assess the effectiveness of the model. With 14 PCA components, the performance of

the MLP model calculated accuracy of 79.261, precision of 90.182, recall of 79.261, and F1 score of 78.428.

6.2.1.3 Multi-Layer Perceptron with t-SNE

The outcome of MLP experiment with t-SNE provides a comprehensive understanding of the data's structure and sheds light on the MLP model's classification behaviour when applied to the transformed data. The data from the t-SNE transformation and MLP classification performance were analysed through visualisations of the decision matrices and computing performance metrics. This analysis indicated the effectiveness of t-SNE for reducing dimensionality and the MLP model's capability to detect intrusions. The performance of the MLP model was subsequently tested using the testing dataset. A multitude of assessment measures encompassing accuracy, precision, recall, and the F1 score, were computed to validate results. Through the integration of t-SNE with hyperparameter optimisation, the MLP model was developed. The model effectively captured pertinent data patterns and was fine-tuned for the normal and attack pattern classification tasks. The outcome of the MLP model's confusion matrices using t-SNE and optimised hyperparameters is shown in Figure 6.11.

Figure 6.11

Confusion Matrix Results of Multi-Layer Perceptron with t-SNE.

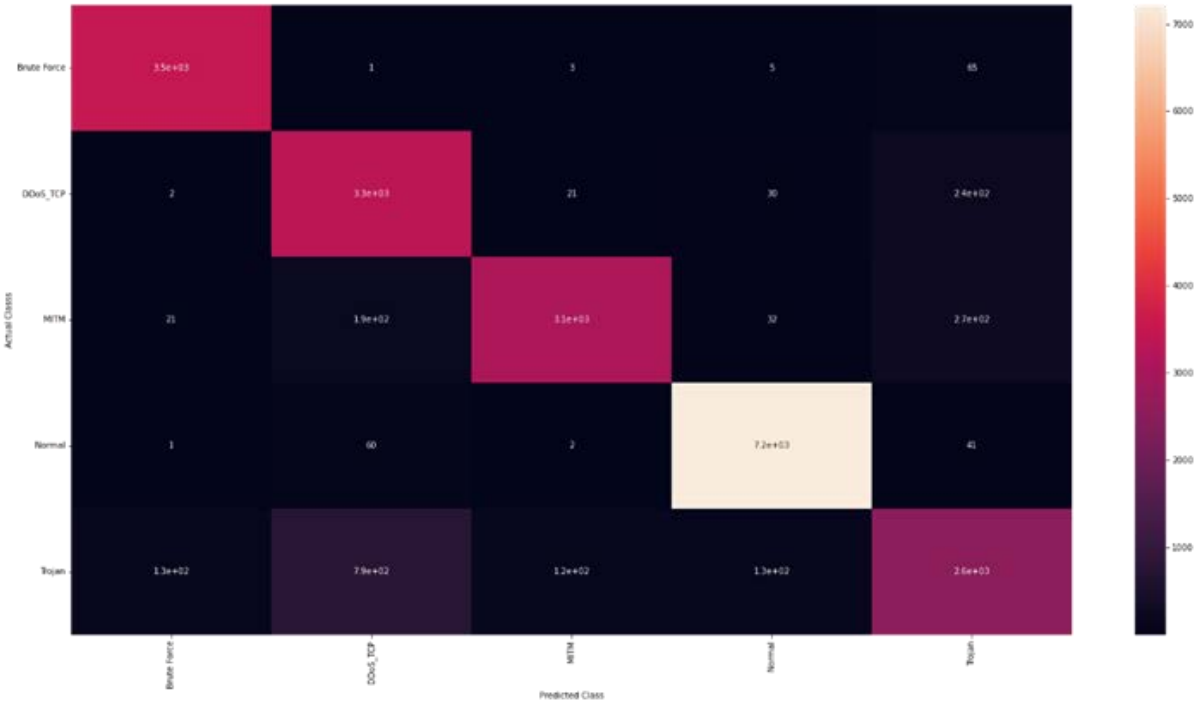


Figure 6.11 depicts the MLP model that used confusion matrices and t-SNE to obtain an accuracy of 90.138, a precision of 90.413, a recall of 90.138, and an F1 score of 90.028.

6.2.1.4 Multi-Layer Perceptron with UMAP

In the MLP experiment with UMAP, the 2022APS dataset was transformed, followed by a phase of processing through feature engineering. After completing the model training, MLP with UMAP assessed the performance of the MLP model on the testing set by calculating various performance metrics. Through the integration of UMAP dimensionality reduction with grid search hyperparameter tuning, the MLP model was developed for the normal and attack pattern classification task, resulting in improved classification results and overall model efficacy. The outcome of the MLP model's confusion matrix is shown in Figure 6.12.

Figure 6.12

Confusion Matrix Results of Multi-Layer Perceptron with UMAP.

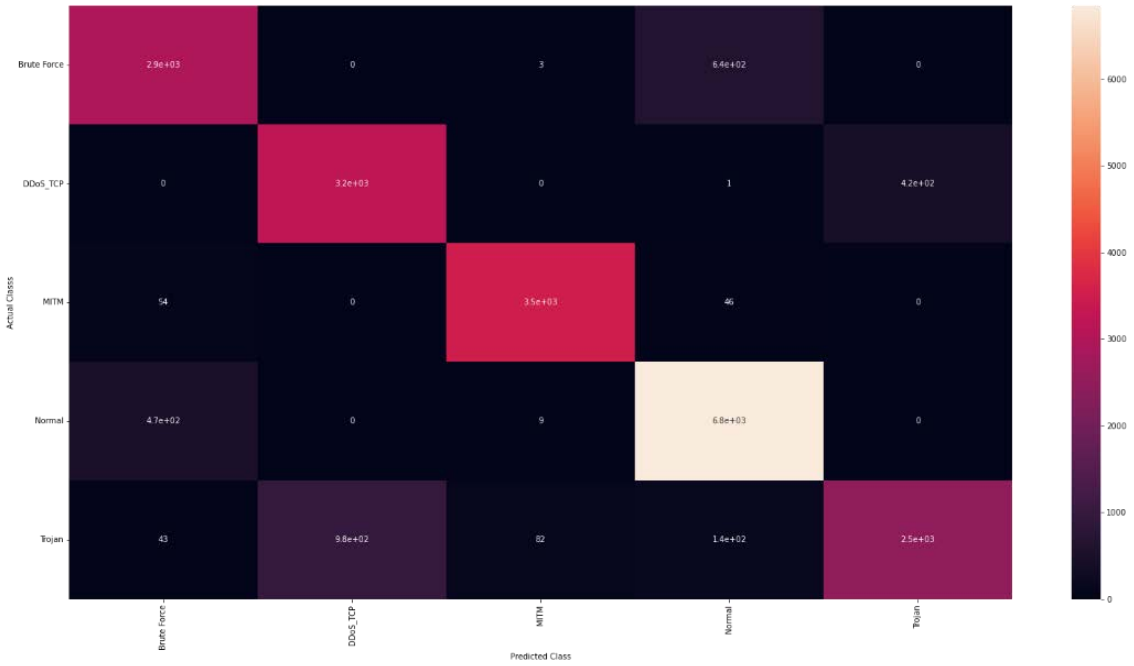


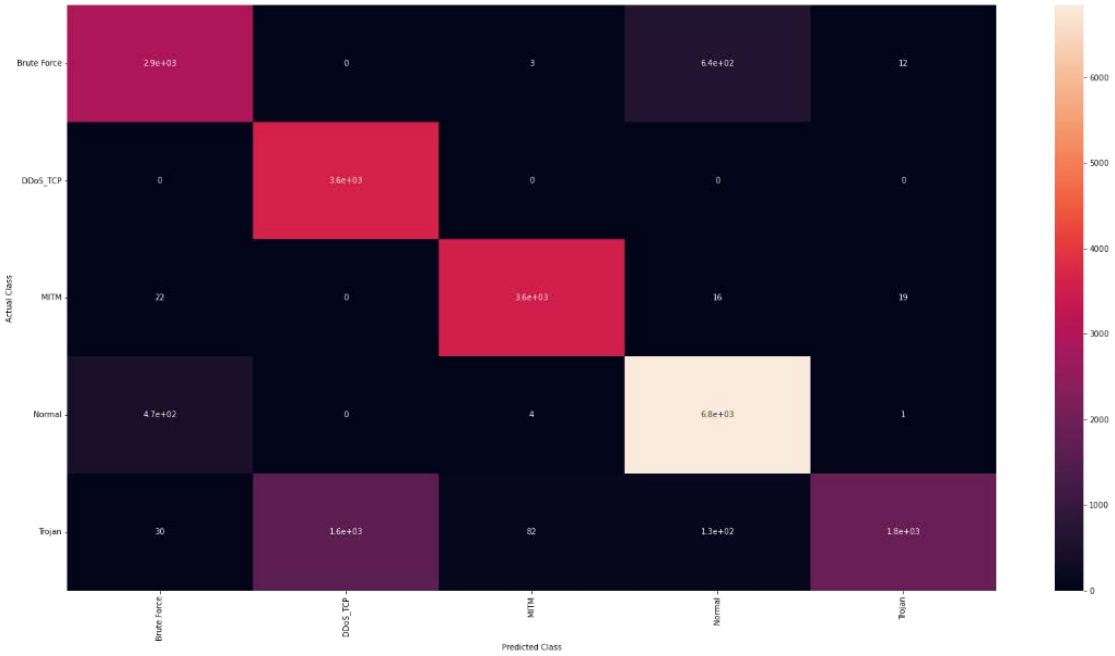
Figure 6.12 depicts the MLP model that used confusion matrices and t-SNE to obtain an accuracy of 86.809, a precision of 86.972, a recall of 86.809, and an F1 score of 86.600.

6.2.1.5 Multi-Layer Perceptron without Dimensionality Reduction

The MLP model, both with and without the application of dimensionality reduction techniques, was trained and tested, yielding results that indicated only negligible differences in the performance of PCA with 10 and 14 components. The outcome of the MLP model without dimensionality reduction confusion matrix is shown in Figure 6.13.

Figure 6.13

Confusion Matrix Results of Multi-Layer Perceptron without Dimensionality Reduction.



The performance of the MLP model was evaluated and compared for medjacking detection in the APS. The model's effectiveness was evaluated based on different dimensionality reduction techniques, the MLP evaluation scores are presented in Table 6.1.

Table 6.1

Performance of the Multi-layer Perceptron with PCA (10 and 14 components), t-SNE, and UMAP Dimensionality Reductions Techniques, and without Dimensionality Reduction for Medjacking Detection.

Model	Dimensionality Reduction	Accuracy	Precision	Recall	F1 Score
MLP	10 PCA Component	79.142	89.868	79.142	78.262
	14 with Components	79.261	90.182	79.261	78.428
	t-SNE	90.138	90.413	90.138	90.028
	UMAP	86.809	86.972	86.809	86.600

Model	Dimensionality Reduction	Accuracy	Precision	Recall	F1 Score
	None	85.881	88.965	85.881	84.835

6.2.2 Decision Tree

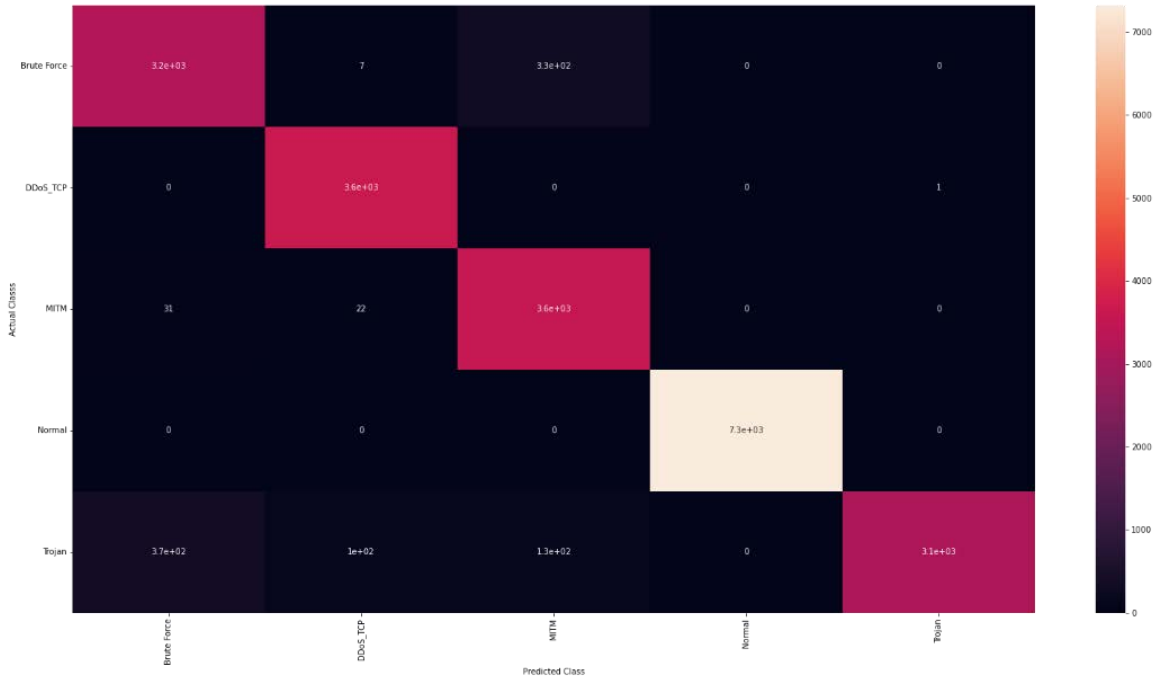
In this study, a DT model, using with and without dimensionality reduction techniques, and grid search technique, was examined for medjacking detection. The DT model started with the 2022APS dataset at the root node. As specified by the root node determination rule, the model chose the 2022APS dataset features to split into subgroups and generate tree branches. Each branch was a tree route that represented a feature value. The DT model was constructed by recursively splitting the dataset to create a hierarchy of decision rules used to discriminate the normal or anomalous patterns. Five experiments were performed to investigate the DT model for the medjacking detection, these experiments are presented below.

6.2.2.1 Decision Tree with 10 PCA Components

In DT with PCA (10 components), the model was developed to detect the medjacking. The DT model utilised PCA for dimensionality reduction, and grid search for hyperparameter tuning. The performance of the trained DL model on the testing set was evaluated for its accuracy, precision, recall, and F1 scores. The DT model effectively captured pertinent data patterns in the classification task. The DT model’s confusion matrix using PCA (10 components) and optimised hyperparameters is shown in Figure 6.14.

Figure 6.14

Confusion Matrix Results of Decision Tree with 10 PCA Components.



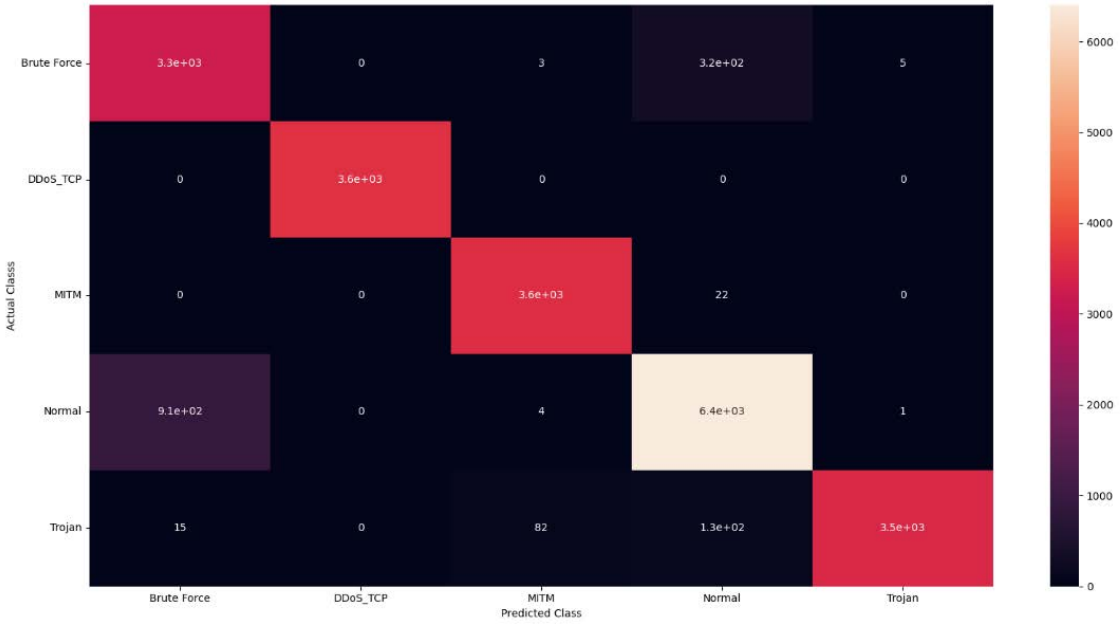
The DT model used confusion matrices that were computed with an accuracy of 95.459, a precision of 95.722, a recall of 95.459 and an F1 score of 95.421.

6.2.2.2 Decision Tree with 14 PCA Components

Section 6.2.2.1 analysed the results obtained with 10 PCA components. This section tested the DT model using 14 PCA components to examine how different numbers of PCA component dimensionality reduction affected the outcome, providing a thorough evaluation of the model's effectiveness. The confusion matrix of the DT model, which has 14 PCA components, is shown in Figure 6.15.

Figure 6.15

Confusion Matrix Results of Decision Tree with 14 PCA Components.



The DT model with 14 PCA components used confusion matrices computed with an accuracy of 96.964, a precision of 97.207, a recall of 96.964, and an F1 score of 96.910.

6.2.2.3 Decision Tree with t-SNE

In DT with t-SNE, optimised hyperparameters were used in the development of the DT model for the medjacking classification task. The evaluation of the classification performance on the testing data involved analysing accuracy, precision, recall, and the F1 score to assess the effectiveness of the DT mode. Figure 6.16 illustrates the result of the confusion matrix for the DT model.

Figure 6.16

Confusion Matrix Results of Decision Tree with t-SNE.

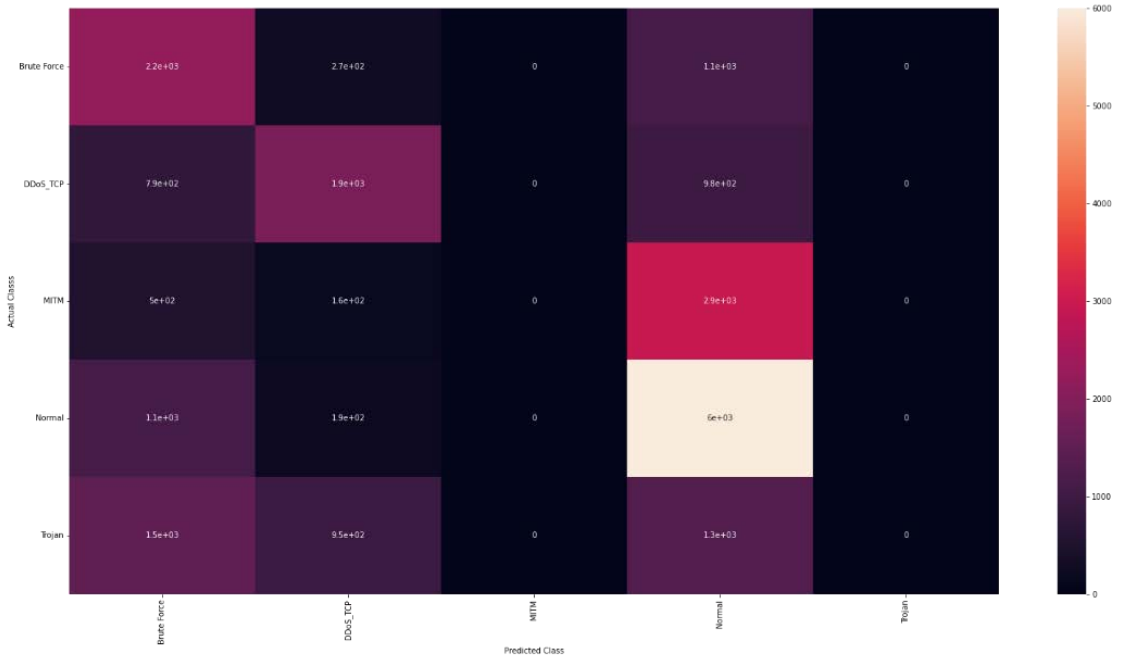


Figure 6.16 depicts the DT model that used confusion matrices to compute an accuracy of 47.970, a precision of 46.733, a recall of 47.970, and an F1 score of 46.792. The effectiveness of t-SNE dimensionality reduction in comparison to PCA demonstrated inconsistency. The t-SNE algorithm for the dimensionality reduction in the 2022APS dataset was not performed optimally as compared to PCA. The DT model's performance in predicting the medjacking could be limited when using t-SNE; however, PCA performed optimally.

6.2.2.4 Decision Tree with UMAP

The DT model's performance was also evaluated using UMAP and optimised hyperparameters. In this test, the performance of the trained DT model on the testing set was evaluated to assess its effectiveness in the classification task. The outcome of the DT model's confusion matrix using UMAP and optimised hyperparameters is shown in Figure 6.17.

Figure 6.17

Confusion Matrix Results of Decision Tree with UMAP.

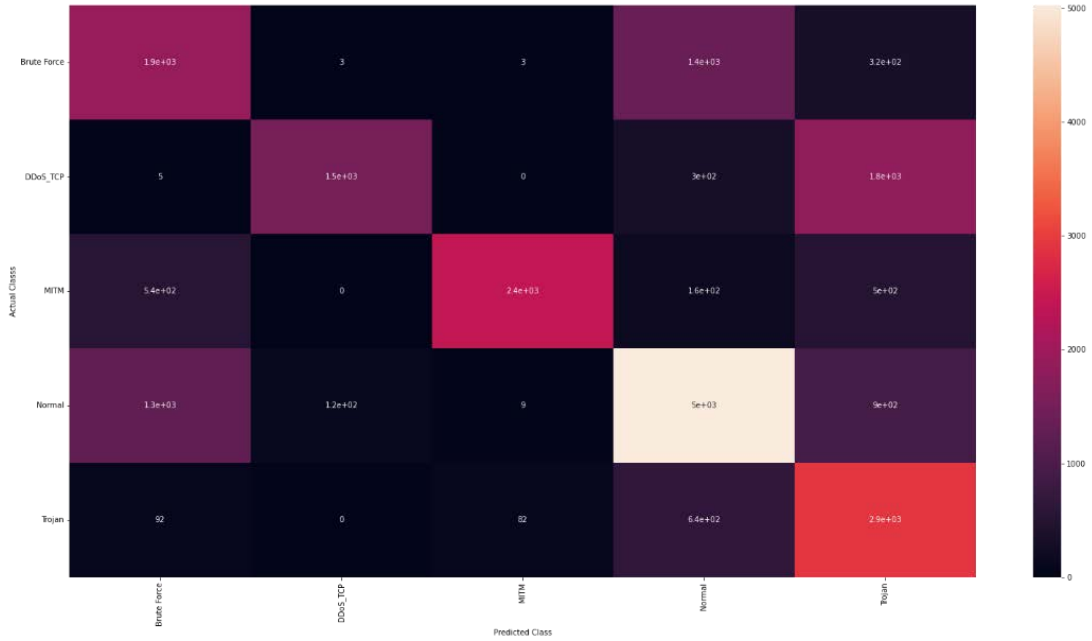


Figure 6.17 depicts the DT model that used confusion matrices to compute an accuracy of 62.961, a precision of 69.664, a recall of 62.961, and an F1 score of 63.519.

6.2.2.5 Decision Tree without Dimensionality Reduction

The DT model was trained and tested with and without dimensionality reduction, resulting in minimal differences in performance observed when using the DT model. The outcome of the DT model without dimensionality reduction confusion matrix is shown in Figure 6.18.

Figure 6.18

Confusion Matrix Results of Decision Tree without Dimensionality Reduction.

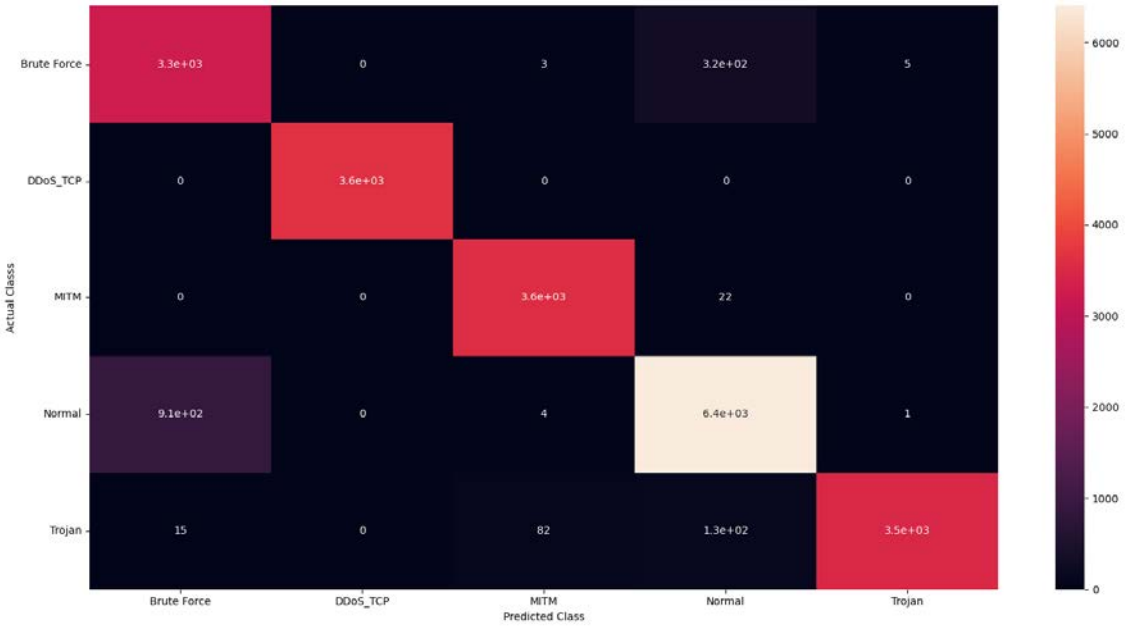


Figure 6.18 depicts the DT model without a dimensionality reduction approach that used confusion matrices to compute an accuracy of 98.509, a precision of 98.549, a recall of 98.509, and an F1 score of 98.488. The DT model's development with PCA (10 and 14 components) demonstrated optimal performance in comparison to UMAP and t-SNE. PCA performed better due to a range of factors. Primarily, the DT model inherently relied on linear binary splits along feature axes that were highly suitable for capturing linear patterns in the data. PCA, being a linear technique for reducing dimensionality, effectively maintained the linear associations among variables, thereby facilitating a consistent integration with the decision-making mechanism of the DT model during the classification task. The DT model evaluation scores are presented in Table 6.2.

Table 6.2

Performance of the Decision Tree with PCA (10 and 14 components), t-SNE, and UMAP Dimensionality Reductions Techniques, and without Dimensionality Reduction for Medjacking Detection.

Model	Dimensionality Reduction	Accuracy	Precision	Recall	F1 Score
DT	10 PCA Components	95.459	95.722	95.459	95.421
	14 PCA Components	96.964	97.207	96.964	96.910
	t-SNE	47.970	46.733	47.970	46.792
	UMAP	62.961	69.664	62.961	63.519
	None	98.509	98.549	98.509	98.488

DT with PCA (10 and 14 components), t-SNE, and UMAP were important for determining the DT model's effectiveness when using different dimensionality reduction techniques. A hierarchical framework of decisions was established based on the relative significance of various features. The PCA technique with 10 and 14 components in the DT model development simplified this procedure by offering distinct and comprehensible principal components that facilitated the DT model's execution of logical and transparent divisions. With PCA 10 and 14 components, the differences observed are negligible. Results were also captured without dimensionality reduction, and the DT model was observed to perform better without dimensionality reduction. However, UMAP and t-SNE, despite their adeptness in capturing intricate non-linear patterns, frequently produced less comprehensible outcomes. The advantages of PCA over UMAP and t-SNE, when used in conjunction with DT model development for normal and attack pattern classification, could be attributed to several factors. First, PCA exhibits a high level of compatibility with the linear characteristics inherent in the

DT model. Second, PCA offers optimal performance in terms of interpretability, allowing for a clear understanding of the underlying patterns and relationships within the data. Third, PCA demonstrated effective capabilities in reducing the dimensionality of the data.

6.2.3 Support Vector Machine

The SVM model was tested in the classification task of discriminating normal and attack patterns. The 2022APS dataset was trained according to the SVM model's philosophies. During the development of the SVM model, three experiments were performed with different dimensionality reduction. Optimal parameters were also identified using the GridSearchCV technique, and finally, outcomes were evaluated to assess the effectiveness of the SVM model. These experiments are presented below.

6.2.3.1 Support Vector Machine with 10 PCA Components

The model development of the SVM comprised PCA for dimensionality reduction and grid search for hyperparameter tuning. The 2022APS dataset was split into a training set to train the SVM for the classification task, and the testing set was used to evaluate the effectiveness of the SVM model's performance in detecting medjacking. The SVM model's confusion matrix using PCA and optimised hyperparameters is shown in Figure 6.19.

Figure 6.19

Confusion Matrix Results of Support Vectors Machine with 10 PCA Components.

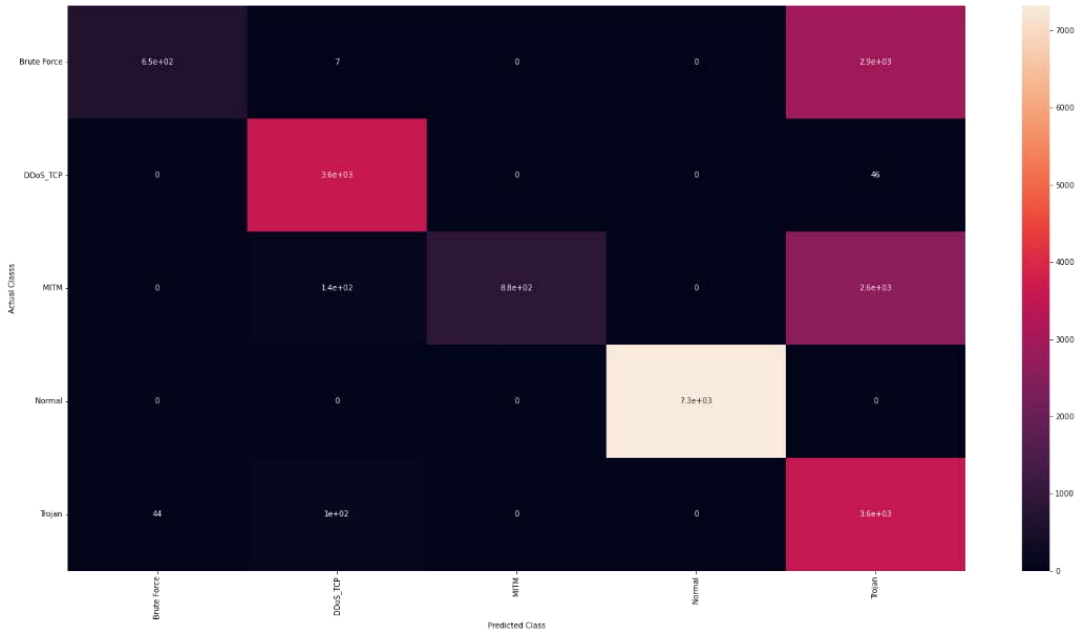


Figure 6.19 depicts the SVM model that used confusion matrices to compute an accuracy of 73.194, a precision of 87.505, a recall of 73.194, and an F1 score of 70.309. Three other experiments were performed using different dimensionality reduction techniques to compare the optimal performance of the SVM model for detecting the medjacking.

6.2.3.2 Support Vector Machine with 14 PCA Components

Section 6.2.3.1 examined the SVM model's performance using 10 PCA components. This section compared the effectiveness of the SVM model using 14 PCA components to determine how the PCA technique impacted the findings. Figure 6.20 depicts the SVM model's confusion matrix with 14 PCA components.

Figure 6.20

Confusion Matrix Results of Support Vectors Machine with 14 PCA Components.

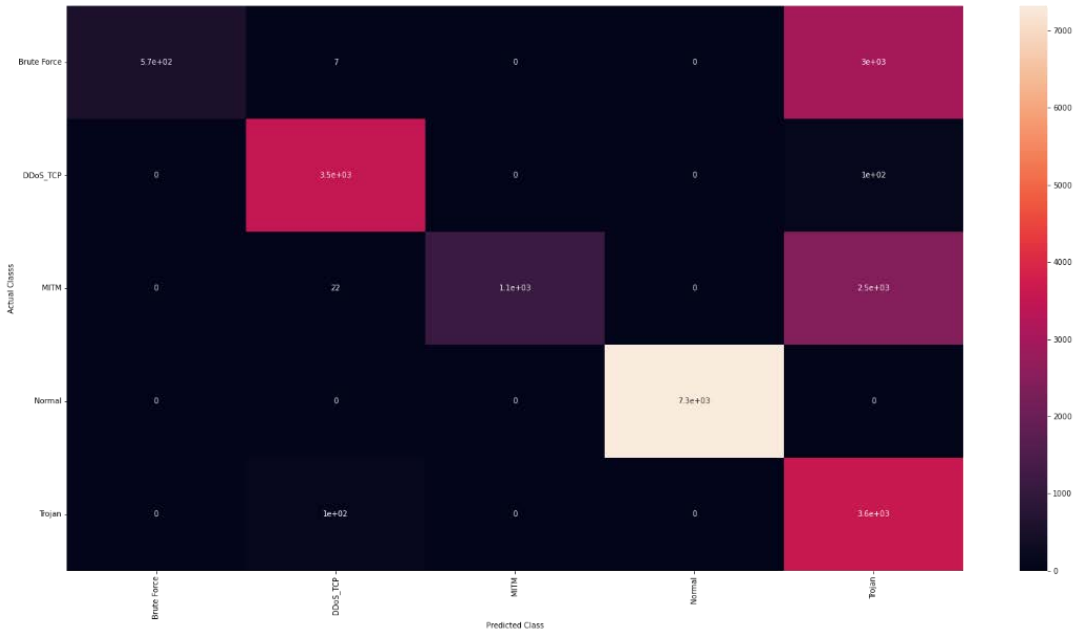


Figure 6.20 depicts the SVM model that used confusion matrices to compute an accuracy of 73.925, a precision of 89.076, a recall of 73.925, and an F1 score of 71.444.

6.2.3.3 Support Vector Machine with t-SNE

In this experiment, the SVM model was developed using t-SNE. In the model development pipeline, the 2022APS dataset was split into a training and testing set for the SMV model training phase, and the testing set was used to evaluate the SMV performance. The performance of the SVM model's confusion matrix using t-SNE and optimised hyperparameters is shown in Figure 6.21.

Figure 6.21

Confusion Matrix Results of Support Vectors Machine with t-SNE.

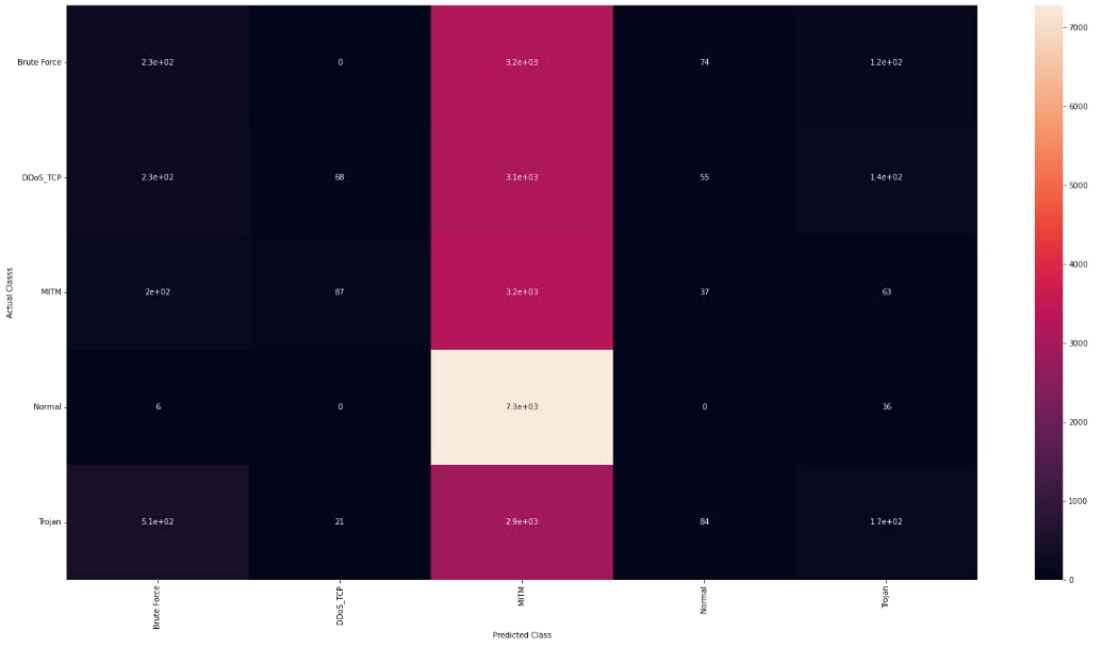


Figure 6.21 depicts the SVM model that used confusion matrices to compute an accuracy of 16.893, a precision of 17.812, a recall of 16.893, and an F1 score of 15.103. The confusion matrices presented in Figure 6.16 show the worst performance when using t-SNE in the SVM model, and other performance matrices have shown unsatisfactory outcomes. The use of PCA in the SVM model was satisfactory compared to t-SNE. PCA was used with the SVM model's linear nature since SVM classifiers succeed at linearly separable data. The PCA technique focused on linear feature relationships, which enabled the SVM model to determine the best class separation hyperplane. Unlike the SVM model, t-SNE's non-linear variations could make data non-linearly separable and less useful. PCA presented interpretable main components that enabled the SVM model to establish distinct decision limits. However, the model failed to classify data since t-SNE's unclear, non-linear embeddings did not match the idea of clear decision boundaries. Another issue was data density. The t-SNE technique created different information quantities in the converted space, which might cause unequal class

distribution. PCA had a better-balanced data density, making the SVM model more robust for the classification task.

6.2.3.4 Support Vector Machine with UMAP

In this experiment, UMAP was applied, and the 2022APS dataset was split into a training and testing set for both training and evaluating the efficiency of the SVM model's ability to detect the medjacking. The performance of the SVM model's confusion matrix using UMAP and optimised hyperparameters is shown in Figure 6.22.

Figure 6.22

Confusion Matrix Results of Support Vectors Machine with UMAP.

Figure 6.22 depicts the SVM model that used confusion matrices to compute an accuracy of 36.452, a precision of 36.271, a recall of 36.452 and an F1 score of 35.178.

6.2.3.5 Support Vector Machine without Dimensionality Reduction

The SVM model was trained and tested both with and without dimensionality reduction techniques, showing that SVM performs better with PCA and performs optimally without dimensionality reduction. Additionally, the results show negligible differences in performance

between the PCA with 10 and 14 components. Figure 6.23 illustrates the confusion matrix for the SVM model without dimensionality reduction.

Figure 6.23

Confusion Matrix Results of Support Vectors Machine without Dimensionality Reduction.

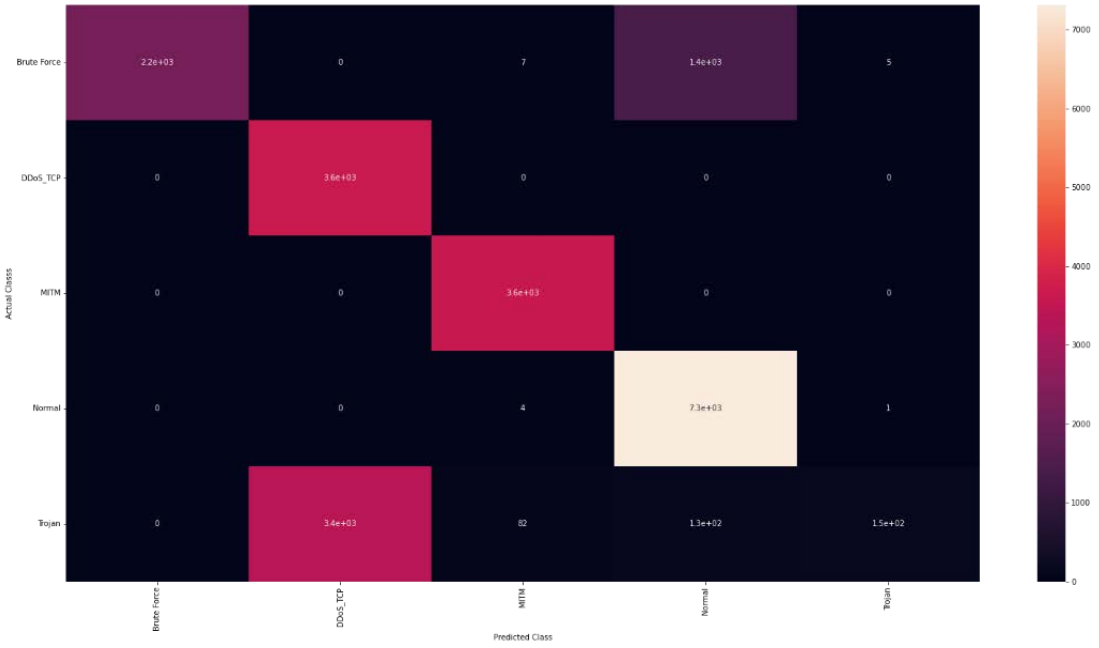


Figure 6.23 depicts the SVM model that used confusion matrices to compute an accuracy of 82.923, a precision of 87.384, a recall of 82.923 and an F1 score of 81.589. The comparison of performance matrices shows the SVM model using PCA (10 and 14 components) performed better than UMAP and t-SNE in classifying normal and attack patterns, and the results show the SVM model performed optimally without dimensionality reduction. The SVM evaluation scores are presented in Table 6.3.

Table 6.3

Performance of the Support Vector Machine with PCA (10 and 14 components), t-SNE, and UMAP Dimensionality Reductions Techniques, and without Dimensionality Reduction for Medjacking Detection.

Model	Dimensionality Reduction	Accuracy	Precision	Recall	F1 Score
SVM	10 PCA Components	73.194	87.505	73.194	70.309
	14 PCA Components	73.925	89.076	73.925	71.444
	t-SNE	16.893	17.812	16.893	15.103
	UMAP	36.452	36.271	36.452	35.178
	None	82.923	87.384	82.923	81.589

The SVM model developed linearly separable data by design, and with PCA (10 and 14 components), the model performed better than other dimensionality reduction techniques. The SVM model was also trained and tested without dimensionality reduction; the results show the SVM model performed optimally without dimensionality reduction. UMAP and t-SNE created non-linearities in the data that impacted the SVM model's performance. PCA generated comprehensible principal components, making the SVM model's decision bounds easier to define. UMAP and t-SNE developed complex non-linear embeddings with uncertain decision boundaries that may impact the SVM model's performance.

6.2.4 Extreme Gradient Boosting

This study examined the effectiveness of the XGBoost model in detecting medjacking. The XGBoost hyperparameters were determined using GridSearchCV. Following this, the dataset was divided into separate training and testing sets. Three experiments were performed to compare the performance with different dimensionality reduction techniques and classifiers.

Finally, the performance evaluation was conducted and compared using the following three experiments.

6.2.4.1 Extreme Gradient Boosting with 10 PCA Components

In this experiment, the XGBoost model was developed to classify normal and attack patterns. PCA was chosen for dimensionality reduction in this experiment. The dataset was split to train the model, and performance evaluation was conducted to evaluate how accurately the model predicted the patterns that were trained. For this purpose, the testing set was used in the model. The performance of the XGBoost model's confusion matrix using PCA and optimised hyperparameters is shown in Figure 6.24.

Figure 6.24

Confusion Matrix Results of Extreme Gradient Boosting with 10 PCA Components.

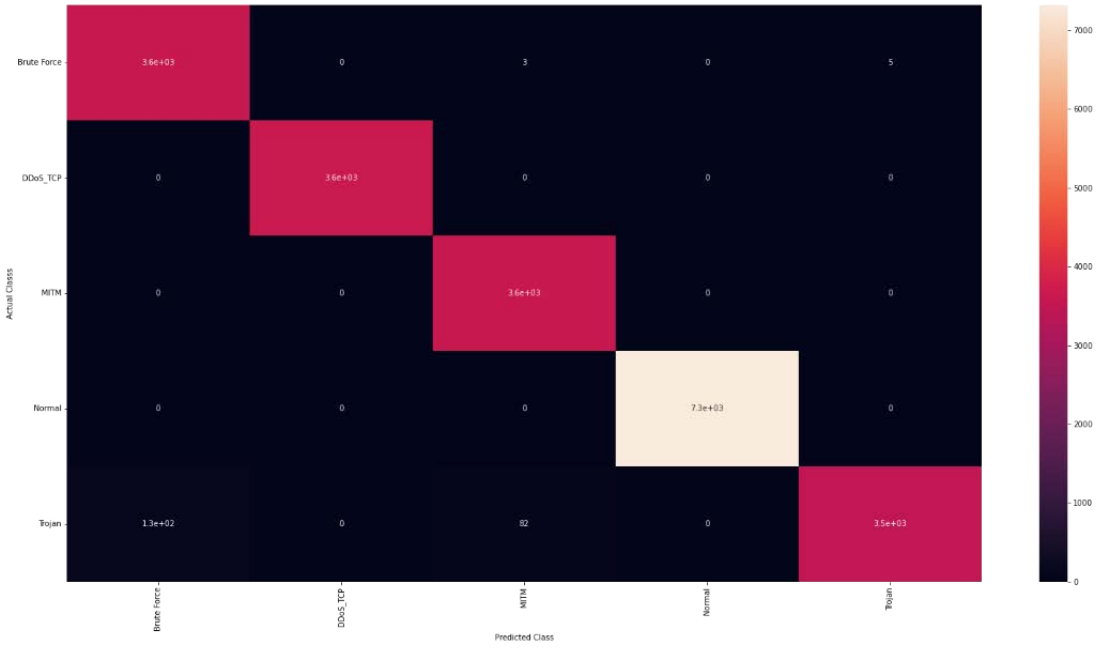


Figure 6.24 depicts the XGBoost model that used confusion matrices to calculate an accuracy of 98.975, a precision of 99.004, a recall of 98.975, and an F1 score of 98.968.

6.2.4.2 Extreme Gradient Boosting with 14 PCA Components

Section 6.2.4.1 computed the XGBoost model's performance using 10 PCA components. This section compared the XGBoost model's performance using 14 PCA components to determine how dimensionality reduction impacted the findings. Figure 6.25 depicts the XGBoost model's confusion matrix with 14 PCA components.

Figure 6.25

Confusion Matrix Results of Extreme Gradient Boosting with 14 PCA Components.

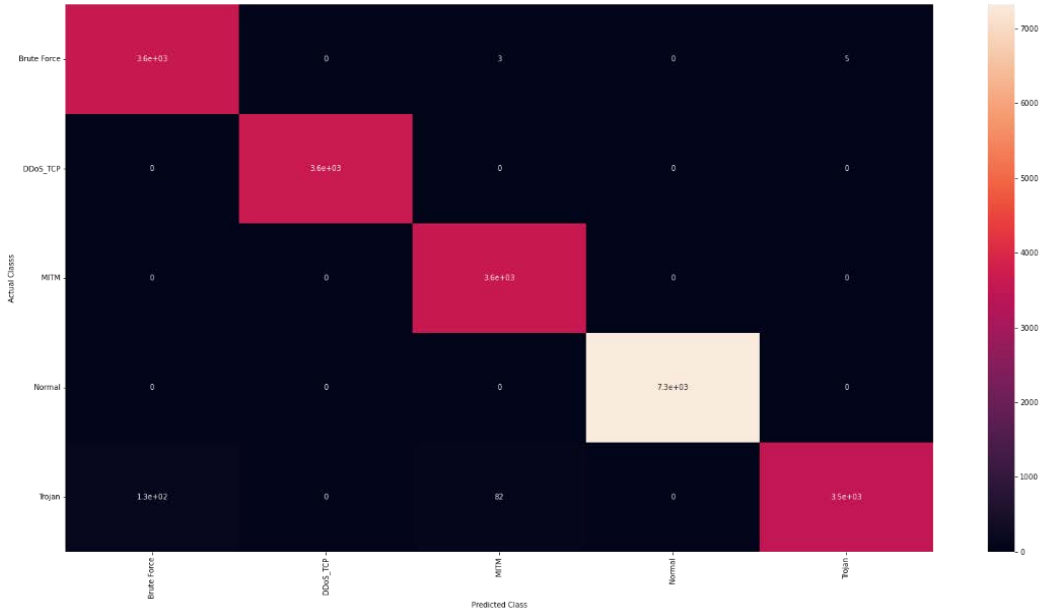


Figure 6.25 depicts the XGBoost model that used confusion matrices to calculate an accuracy of 98.985, a precision of 99.012, a recall of 98.985, and an F1 score of 98.978.

6.2.4.3 Extreme Gradient Boosting with t-SNE

In the XGBoost model development to classify normal and attack patterns, the t-SNE technique was used for dimensionality reduction and grid search for optimal hyperparameters. The t-SNE performed better with optimal accuracy compared to the DT model (see Section 6.3.2). The XGBoost model's confusion matrix using t-SNE and optimised hyperparameters is shown in Figure 6.26.

Figure 6.26

Confusion Matrix Results of Extreme Gradient Boosting with t-SNE.

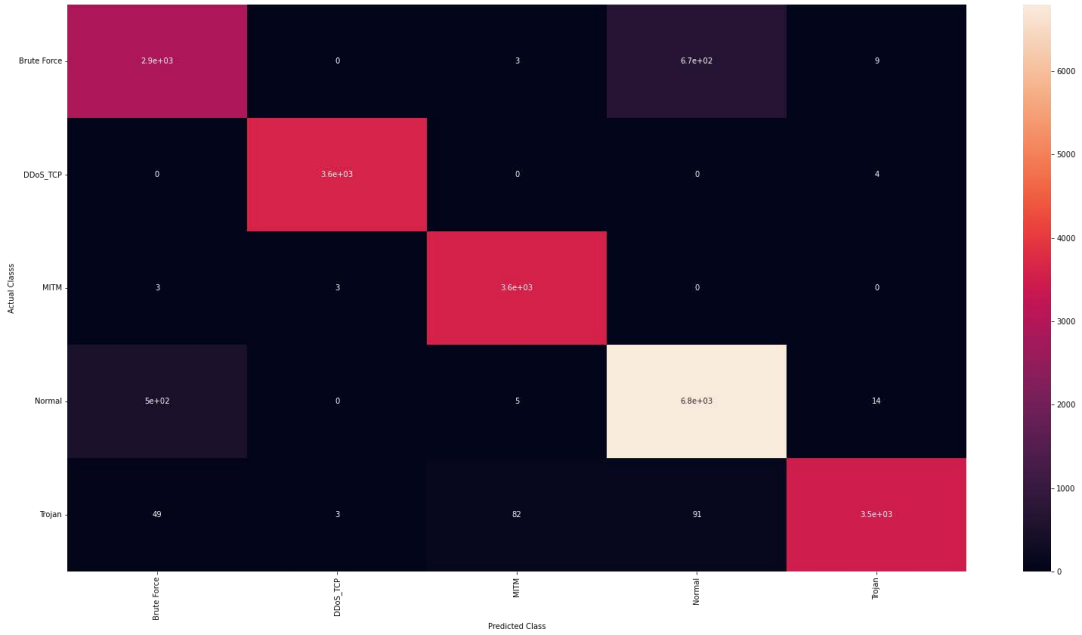


Figure 6.26 depicts the XGBoost model that used confusion matrices to compute performance metrics. An overall accuracy of 93.416, precision of 93.435, recall of 93.416, and F1 score of 93.398 was achieved in this test. The performance outcome has shown that t-SNE and GridSearchCV hyperparameter optimised the XGBoost model's performance and accurate detection of medjacking.

6.2.4.4 Extreme Gradient Boosting with UMAP

In the XGBoost with PCA and t-SNE, the performance of the model was optimal. This experiment examined the UMAP performance in the XGBoost model. The model's efficiency was assessed using different performance metrics. The XGBoost model's confusion matrix using UMAP and optimised hyperparameters is shown in Figure 6.27.

Figure 6.27

Confusion Matrix Results of Extreme Gradient Boosting with UMAP.

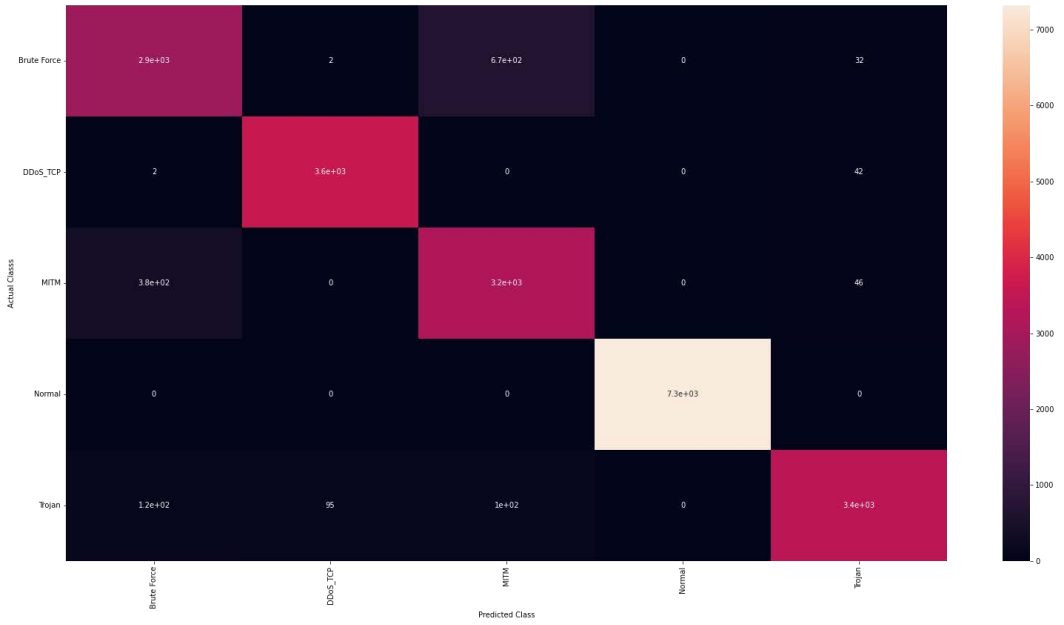


Figure 6.27 shows a five-by-five matrix illustrating the XGBoost confusion matrices with the actual class labels on one side and the predicted class labels on the other. An overall accuracy of 93.585, precision of 93.618, recall of 93.585, and F1 score of 93.578 was achieved in this test.

6.2.4.5 Extreme Gradient Boosting without Dimensionality Reduction

The XGBoost model was trained and tested with and without dimensionality reduction, and the differences in performance were negligible. Figure 6.28 depicts the outcome of the XGboost model without the dimensionality reduction confusion matrix.

Figure 6.28

Confusion Matrix Results of Extreme Gradient Boosting without Dimensionality Reduction.

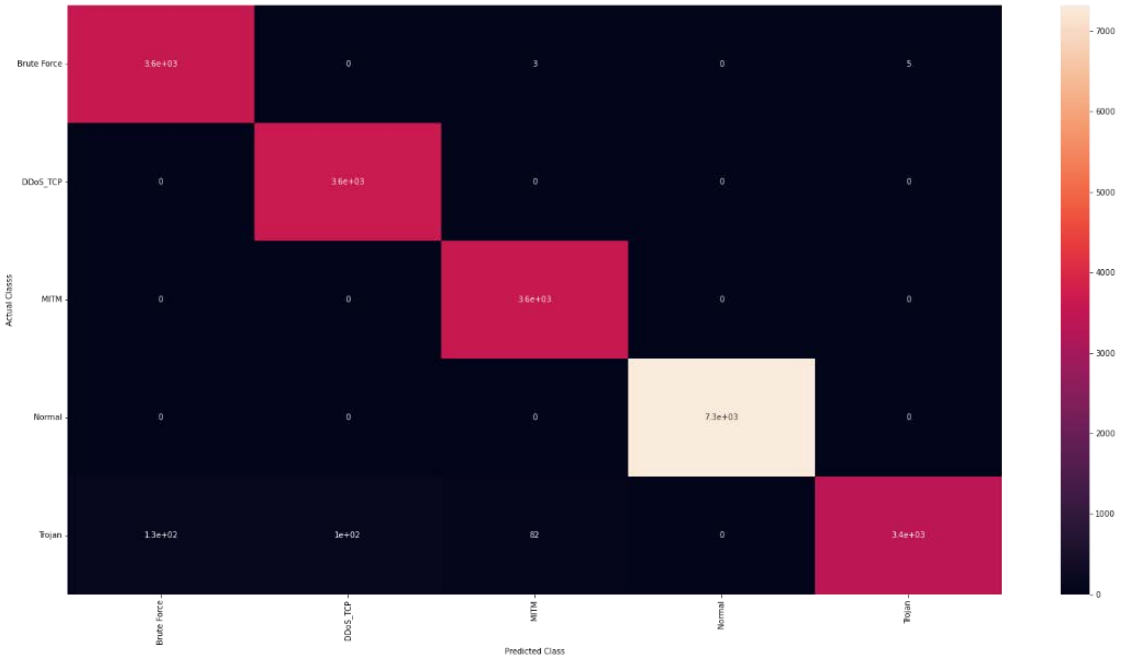


Figure 6.28 depicts the XGBoost model that used confusion matrices to compute an accuracy of 98.495, a precision of 98.536, a recall of 98.495, and an F1 score of 98.474. This test was successful in assessing the calculated metrics: accuracy, precision, recall, and an F1 score are presented in Table 6.4.

Table 6.4

Performance of the Extreme Gradient Boosting with PCA (10 and 14 components), t-SNE, and UMAP Dimensionality Reductions Techniques, and without Dimensionality Reduction for Medjacking Detection.

Model	Dimensionality Reduction	Accuracy	Precision	Recall	F1 Score
XGBoost	10 PCA Components	98.975	99.004	98.975	98.968
	14 PCA Components	98.985	99.012	98.985	98.978
	T-SNE	93.416	93.435	93.416	93.398

Model	Dimensionality Reduction	Accuracy	Precision	Recall	F1 Score
	UMAP	93.585	93.618	93.585	93.578
	None	98.495	98.536	98.495	98.474

Results from the performance evaluation demonstrated that the XGBoost model performed at its best and correctly recognised medjacking when using the PCA, t-SNE, UMAP techniques, and without dimensionality reduction.

6.3 The Classification Models' Performance

The performance of each MLP, DT, SVM, and XGBoost model was evaluated and compared for medjacking detection in the APS. Each model's effectiveness was evaluated based on different dimensionality reduction techniques and confusion matrices to identify various factors for signature medjacking detection. All models in this study, both with and without the application of dimensionality reduction techniques, were trained and tested, all evaluation scores are presented in Table 6.5.

Table 6.5

Performance of the Four ML Models with PCA (10 and 14 components), t-SNE, and UMAP Dimensionality Reductions Techniques, and without Dimensionality Reduction for Medjacking Detection.

Model	Dimensionality Reduction	Accuracy	Precision	Recall	F1 Score
MLP	10 PCA Components	79.142	89.868	79.142	78.262
	14 PCA Components	79.261	90.182	79.261	78.428
	t-SNE	90.138	90.413	90.138	90.028

Model	Dimensionality Reduction	Accuracy	Precision	Recall	F1 Score
	UMAP	86.809	86.972	86.809	86.600
	None	85.881	88.965	85.881	84.835
DT	10 PCA Components	95.459	95.722	95.459	95.421
	14 PCA Components	96.964	97.207	96.964	96.910
	t-SNE	47.970	46.733	47.970	46.792
	UMAP	62.961	69.664	62.961	63.519
	None	98.509	98.549	98.509	98.488
SVM	10 PCA Components	73.194	87.505	73.194	70.309
	14 PCA Components	73.925	89.076	73.925	71.444
	t-SNE	16.893	17.812	16.893	15.103
	UMAP	36.452	36.271	36.452	35.178
	None	82.923	87.384	82.923	81.589
XGBoost	10 PCA Components	98.975	99.004	98.975	98.968
	14 Components	98.985	99.012	98.985	98.978
	T-SNE	93.416	93.435	93.416	93.398
	UMAP	93.585	93.618	93.585	93.578
	None	98.495	98.536	98.495	98.474

Table 6.5 shows the performance analysis results of the classification models' ability to detect medjacking attacks in the APS. The MLP model had satisfactory accuracy scores for PCA (10 and 14 components), t-SNE, and UMAP compared to the DT and SVM models. The performance of the XGBoost model was compared with the three other models used in this

study. Compared to these models, the XGBoost model with all dimensionality reduction techniques and without dimensionality reduction predicted the medjacking attack with greater accuracy.

The DT model was outstanding with PCA (10 and 14 components) as well as without dimensionality reduction, but the DT model performed poorly with a t-SNE accuracy score of 47.970 and a UMAP accuracy score of 62.961. The DT model's performance when using t-SNE and UMAP for signature medjacking detection was not satisfied, and it could be considered unacceptable for medjacking detection. The SVM model performed with a satisfactory accuracy of 73.194 when using 10 PCA components and obtained an accuracy of 73.925 when using 14 PCA components. The SVM model performed better without dimensionality reduction with an accuracy of 82.923. However, the SVM model with t-SNE and UMAP was not able to predict medjacking attacks with very low accuracy scores. The SVM model's t-SNE accuracy score was 16.893, and the UMAP accuracy score was 36.452. These results show that the SVM model failed to detect medjacking; the only chance for the SVM model to detect medjacking was when using PCA and without dimensionality reduction.

6.3.1 Comparison of Classification Models

The results presented in Table 6.1 show the positive outcome of using ML and DL models to detect medjacking in the APS. This current study and literature show that the ML and DL models are more effective in predicting intrusion; however, the model preparation was critical. This preparation, consisting of hyperparameter tuning and different dimensionality reduction techniques used for comparison, has not be frequently discussed in current research literature. In terms of identifying whether the model is more effective in linear and nonlinear classification, the dimensionality reduction techniques could help understand model performance with classification tasks in cybersecurity. The linear dimensionality reduction algorithm PCA worked optimally in ML and DL models. However, hyperparameter tuning

added more values to identify optimal hyperparameters for each model. The comparison results presented in Table 6.1 demonstrated that the MLP and XGBoost models could detect medjacking with all dimensionality reduction techniques. In contrast, the DT and SVM models only performed well with PCA techniques, but t-SNE and UMAP demonstrated unsatisfactory results. To validate these results, the accuracy of this study is compared with other studies from the literature in Table 6.6. This comparison is based on the use of PCA in model development. All models are compared to understand the effectiveness of detecting attacks.

Table 6.6

Comparison of Machine Learning Models' Accuracy with the Principal Component Analysis.

Model	**Reference**	**Proposed System**	**Attacks**	**Dataset**	**Dataset instances**	**Accuracy**
MLP	This study	H-IDS	Brute force, DDoS, MITM, and Trojan	2022APS	109,360	79.142
	[575]	Intelligent botnet attack detection engine	Botnet	N-BaIoT	7062606	94.000
	[875]	IDS	DDoS, infiltration, brute force, and web	CICIDS2017	225,745	89.000
SVM	This study	H-IDS	Brute force, DDoS, MITM, and Trojan	2022APS	109,360	87.505
	[575]	Intelligent botnet attack detection engine	Botnet	N-BaIoT	7062606	90.000
DT	This study	H-IDS	Brute force, DDoS, MITM, and Trojan	2022APS	109,360	95.459
	[575]	Intelligent botnet attack	Botnet	N-BaIoT	7062606	79.000

Model	Reference	Proposed System	Attacks	Dataset	Dataset instances	Accuracy
		detection engine				
	[876]	IDS	DDoS	CICDDoS-2019	294,627	98.60
XGBoost	This study	H-IDS	Brute force, DDoS, MITM, and Trojan	2022APS	109,360	98.975
	[579]	IDS	Malware	WUSTL EHMS 2020, PE headers-Clamp	5213	98.660

Table 6.6 presents a comparison of the results obtained in the current study with results from the literature. The ML and DL models in this study, using hyperparameter tuning and different dimensionality reduction techniques, achieved a satisfactory level of detection accuracy when compared to existing literature. Although there are some minor differences in accuracy scores, these could be attributed to various factors. The primary factor is the utilisation of optimised hyperparameter tuning during the models' development. Additionally, the differences could be attributed to variations in the attack data.

In [575], Saif et al., proposed IDS to detect botnets, their system utilised Linear discriminant analysis and PCA for the dimensionality reduction phase, and models included NB, DT, RF, SVM, LR, CAN, MLP and Single-layer perceptron, the size of data was 849,234 instances. In another study [875], Mendonça et al., proposed IDS to detect DDoS, infiltration, brute force, and web attacks, PCA technique applied, four NB, SVM, RF and MLP models tested, the size of data was 2,830,743 instances.

DDoS attack detection was proposed in [876] by Kumari & Jain, the researchers employed PCA, Linear Discriminant Analysis (LDA) in dimensional reduction, and NB, DT, RF, AdaBoost, and LR models were trained, total numbers dataset instances was 294,627 [876].

The malware detection was proposed in [579] by Dhanya & Chitra, the researchers employed two datasets, total number dataset1 instances was 16,318 and dataset2 was 5213, XGBoost model was trained on both datasets.

The study's findings demonstrated a range of performance levels among the examined models when employing various dimensionality reduction techniques and a hyperparameter tuning approach. Although the MLP and XGBoost models with PCA have shown efficacy in detecting attacks as compared with the current literature, there is a lack of current research evaluating the effectiveness of t-SNE and UMAP in the field of cybersecurity for medical devices.

This study has found that DT and SVM models for medjacking detection did not perform optimally when using t-SNE and UNMAP dimensionality reduction techniques. However, when comparing the performance of the MLP and XGBoost models with t-SNE and UMAP, the MLP and XGBoost models performed well with these reduction techniques. These findings highlight the importance of carefully developing classification models in medjacking research. This study allows the research community to make the most appropriate techniques when developing the detection model for medjacking.

6.3.2 Justification of Machine Learning Model Accuracy in Literature

ML models used for classification tasks are contingent upon various factors, including the dataset, feature engineering, and the model's architecture. Although the accuracies vary, the majority of optimally set ML models attain an accuracy ranging from 70% to 95% for conventional classification challenges. As indicated in [877], a substantial number of the models demonstrate accuracy in identifying cybercrime, exhibiting accuracy rates ranging from 70% to 90% [877]. Related ML model's accuracy in [878], illustrates that a performance exceeding 70% is indicative of a highly effective model. Indeed, an accuracy metric ranging

from 70% to 90% is optimal. The accuracy ranges from 70% to 90% and is consistent with industry standards [878].

In [879], the researchers presented a filter-based feature-dropping method and utilised a range of predictive algorithms, encompassing DT, ANN, LR, K-Nearest Neighbour (KNN), and SVM. Their findings demonstrated the accuracies in some trained models, ranging from 88.13% to 90.85%. The comprehensive accuracy rates were recorded at 90.85% for DT, 84.4% for ANN, 77.64% for LR, 84.46% for KNN, and 60.89% for SVM. The accuracy rates for the multiclass classification were recorded as follows: 67.57% for DT, 77.51% for ANN, 65.29% for LR, 72.30% for KNN, and 53.95% for SVM. These evaluations were conducted utilising 19 features in the dataset [879].

In [880], researchers proposed IDS employing CNN to identify DoS attacks. Their findings indicated that the CNN achieved an accuracy of 99.87%, whereas the Naïve Bayes classifier attained an accuracy of 54.42% [880]. In [881], researchers developed the IDS employing ML approaches to identify malicious attacks. The study involved training eight ML models. The accuracy metrics were recorded as follows: 98.3 for RF, 98.3 for DT, 97.0 for XGBoost, 57.3 for Gaussian Naive Bayes, 93.4 for KNN, 56.8 for LR, 82.2 for Gradient Boosting, and 93.3 for AdaBoost [881].

Contingent upon the nature of the classification tasks, increased accuracy is frequently achievable through the preprocessing of datasets or the careful tuning of hyperparameters. Diverse ML models exhibit distinct architectures, resulting in varying levels of accuracy in their performance. Nevertheless, lowered accuracy may signify the model's failure for tasks or hyperparameters or features may not support the model's architecture. Hyperparameter tuning is crucial in enhancing ML models' performance by optimising the balance among model accuracy, generalisation, and computational efficiency [803-811]. The model's decision boundaries may be comprehensively optimised through meticulous testing hyperparameters to

identify even the most complex patterns indicating intrusions [812-815]. The process of tuning enhances precision and mitigates the incidence of false positives [816], a factor of paramount importance in detection systems, where the inability to identify intrusions or the increase of unwarranted alerts may present substantial risks.

Additional factors contributing to accuracy discrepancies may include the approach used for feature selection and the number of features utilised. According to [701], the feature selection process is essential in developing ML models for various reasons, as it significantly impacts the model's performance. The procedure entails identifying and selecting the most significant features from the dataset, while concurrently discarding irrelevant or superfluous features. According to [703], feature selection is the process of extracting a subset from a given feature set predicated on particular selection criteria, which identifies the pertinent features within the dataset.

In [882], the researcher demonstrated feature selection techniques, highlighting that several features directly influence the accuracy of the ML model to detect DDoS attacks. The researchers used various feature selection methodologies to train distinct ML models and calculated their accuracies. The SVM model used an information gain feature selection technique, utilising 20 selected features, and achieved an accuracy of 60.19%. In contrast, the correlation coefficient feature selection method, which incorporated 23 selected features, yielded an accuracy of 80.61%. The Chi-square feature selection method, employing 24 selected features, yielded an accuracy of 59.45%. In contrast, the forward selection feature selection technique, utilising 34 selected features, achieved an accuracy of 58.60%. The backward elimination feature selection method yielded a selection of 34 features, resulting in an accuracy of 59.62%. In contrast, the recursive elimination feature selection approach identified 28 features, achieving an accuracy of 89.18%. Lastly, the Lasso feature selection technique selected 24 features, resulting in an accuracy of 59.79%. A similar

approach was employed across various models; the accuracies of the Naive Bayes model, reliant upon diverse feature selections and a variety of features, were calculated at 85.08%, 63.79%, 48.77%, 84.79%, 83.42%, 92.12%, and 69.43%, respectively. In contrast, the accuracies of the RF model, which used different feature selection techniques and numbers of features numbers, were determined to be 99.92%, 99.91%, 99.91%, 99.91%, 99.94%, 99.97%, and 99.90%, respectively. The accuracies of the KNN model, derived from various feature selection techniques and differing quantities of features, were computed at 99.57%, 99.19%, 99.41%, 98.66%, 98.50%, 98.57%, and 99.54%, respectively. The accuracies of the DT model, evaluated across various feature selection techniques and differing quantities of features, were computed to be 99.88%, 99.83%, 99.85%, 99.80%, 99.86%, 99.80%, and 99.83%, respectively [882].

According to the literature, many factors influence the accuracy of ML models. Various ML models yield distinct classification accuracies attributable to differences in their algorithmic architecture, capacity to manage complexity, techniques of feature engineering, dependence on hyperparameters, and competence in generalisation. The model selection aligns with the stated criteria to attain maximal efficacy. Comprehending these determinants enables researchers and industry professionals to effectively select, adjust, and implement models, thus ensuring accuracy and dependability in classification endeavours.

6.3.3 Justification of Dataset Sizes

In current literature, the variation in datasets, their size, and scope emerges from the specific features of the problems they tackle, the fields from which they are derived, and the objective of their study [666-668]. The variations are essential, as datasets are meticulously designed to align with distinct objectives, contextual environments, and resources during their research phase with available sources and tools [669-671]. Each dataset has been carefully designed to address a particular issue, with its architecture and magnitude contingent upon the

distinct requirements associated with that issue or objective. For example, features specific to a particular domain significantly influence the composition of datasets. The BoT-IoT dataset [568] primarily focused on identifying botnet attacks. The EHMS dataset [646] focused on a biometric gathering that includes electrocardiogram readings, blood oxygen saturation levels, temperature sensor data, and blood pressure metrics. The ECU-IoFT dataset [664] established drone-related Wi-Fi attacks to identify intrusions within the drone systems.

Datasets are derived from many sources, significantly impacting their size and configuration. The term heterogeneous data sources signifies that datasets may originate from diverse devices or environments. Within the context of the IoMT, data is generated by devices such as glucose monitors and heart rate trackers, each exhibiting distinct data generation rates. For instance, the DAD dataset [663], comprising 10 features and encompassing 101,583 instances, was generated to facilitate the development of cold aisle data centre characteristics and to gather relevant features. Temperature sensors were incorporated with Duplication, Interception, and Modification attack scenarios [663]. The ECU-IoFT dataset [664], comprising 10 features and encompassing 54,492 instances, was generated to facilitate the development of drone characteristics and identify attacks related to drone Wi-Fi. The exploitation of drone devices utilising the Tello API, alongside the techniques for WPA2-PSK Wi-Fi cracking and scenarios involving Wi-Fi deauthentication attacks scenario [664]. The IoT-23 dataset [661], comprising 19 features and encompassing 280,772,878 instances, was generated to facilitate the development of botnet features and to identify botnets. The gadgets utilised comprised a Philips Hue smart lamp, an Amazon Echo, and a Somfy smart door lock. The utilisation of malware attacks encompassed various entities, including Mirai, Torii, gagfyt, Kenjiro, Okiru, Hakai, IRCBoT, Hajime, Muhstik, and hide and seek Botnet detections [661]. The WID dataset [665] comprises 254 features and encompasses 6,526,404 instances, which were generated to develop Wi-Fi-related attributes to identify Wi-Fi-related intrusions. The

IoT devices were vulnerable to a variety of security threats, including Re)Assoc, Disas, Deauth, Rogue Access Points, SSh vulnerabilities, Kr00k, Krack, Botnet formations, SSDP exploitation, SQL Injection, Malware infiltration, Website spoofing, and Evil Twin attack scenarios [665].

Each dataset exhibits variations in size, features, and contextual scenarios. The aims associated with utilising datasets profoundly influence their size, features, and scenarios. Datasets developed for research enquiry demonstrate a particular objective to resolve or address. According to the literature, no singular dataset can address every domain-specific issue. For instance, the researcher in [883] stated that Deep Neural Networks, when trained on a singular dataset (source domain), exhibit poor performance on an alternative dataset (target domain) that, while different, possesses similar characteristics to the source domain. Domain adaptation endeavours to mitigate this issue and possesses significant potential for application in practical contexts, real-world situations, industrial implementations, and various data domains [883]. In another instance, within the context of a lung cancer detection dataset [884], the researcher stated the absence of studies concerning negative transfer between the source domain and the target domain. Theoretically, one can delineate the transfer learning problem by considering both the source and target datasets, which may exhibit high or low degrees of similarity. The negative transfer intensifies as the degree of similarity diminishes, resulting in an increased presence of unrelated samples within the source dataset. The transference of knowledge from disparate samples to the target model results in an adverse change in the model's performance [884].

In [652], the researchers undertook an exhaustive examination of the datasets accessible to the public. The researchers stated that the datasets utilised for examining network packets in commercial applications are not easily obtainable due to concerns related to privacy. Nevertheless, numerous publicly available datasets, such as DARPA, KDD, NSL-KDD, and ADFA-LD, are widely employed as reference points within the discipline. The researchers

conducted a comparative analysis of the extant datasets, demonstrating their inherent limitations. The principal constraint identified pertains to the comparative analysis of the datasets, namely DARPA 98, KDDCUP 99, CAIDA, NSL-KDD, ISCX 2012, ADFA-WD, ADFA-LD, and CICIDS2017, which lack any evidence of IoT activity [652]. According to the literature, individual datasets may not address every issue, and their scopes are unlikely to be uniform owing to variations in domain specifications and the intended aims of the research direction.

6.4 Results Limitations

This study conducted experiments using several models (ML and DL) and dimensionality reduction approaches to assess their effectiveness when developing the H-IDS. The approach was chosen to comprehensively assess the capabilities of different techniques (i.e., grid search) and identify the most accurate models. However, the wide range of approaches used in the study has shown the potential results. This study provides a rationale for the limitations to elucidate the nature of the consistency of results, as follows:

- The utilisation of several models (ML and DL) inherently resulted in inconsistency. Each model possessed unique advantages and disadvantages, affected by algorithmic intricacy, adaptability to hyperparameter adjustment, and appropriateness for the specific characteristics.
- The application of various dimensionality reduction techniques created an extra degree of inconsistency. PCA, t-SNE, and UMAP utilise distinct mechanisms to decrease the number of features while retaining crucial information. The inconsistency in the interpretation of the reduced feature set resulted in further inconsistencies in the results of different models.

- The relationship between dimensionality reduction and model performance was intricate. Each technique added advantages to some models, while others could not offer the same benefits. As an illustration, the MLP model achieved good accuracy assessments for PCA, t-SNE, and UMAP compared to the DT and SVM models. The XGBoost model exhibited better results when tested using all dimensionality reduction techniques. The relationship between the choice of dimensionality reduction technique and the resulting outcomes was noticeable, as it contributed a crucial role in the observed inconsistency in results.
- The grid search approach for hyperparameters has a notable effect on the consistency of the results. The grid search method effectively investigates each model's chosen range of parameters to determine the most effective hyperparameters. While the approach was comprehensive, it might have led to different levels of optimisation among models, depending on the range and specificity of the parameters used for fine-tuning. Some models may have reached their ideal configuration easier than others, resulting in differences in their comparative effectiveness.

The integration of several models, methods for dimensionality reduction, and the implementation of grid search to optimise hyperparameters ultimately influenced the consistency of results. This inconsistency results from the inherent disparities in model design, the intricate interplay between models and dimensionality reduction strategies, the dependence on hyperparameter adjustment, and the different features.

6.5 Hybrid Intrusion Detection System Findings

This study has developed H-IDS, the phases, and experimental results presented including a comparison of different studies. A list of phases, accomplished tasks, progress, comments, and findings are presented in Table 6.3.

Table 6.7

Hybrid Intrusion Detection System Findings.

Phase	Units	Accomplished	Progress	Comment	Finding
Requirements	Dataset, and H-IDS requirements	APS data, and attack data (brute force, DDoS, MITM), and tools.	The dataset was collected and processed.	The simulated APS prototype was developed. The proprietary device can be used in final production.	The 2022APS dataset was utilised for the next design phase.
Design	Flowchart	The details of the H-IDS components are provided.	Tested all components.	Employed open-source tools for cost-efficient purposes.	The licensed tools can be used to investigate further.
Implementation	Model preparation	Correlation, RFECV, dimensionality reduction, hyperparameter tuning, and trained five models.	MiniMaxScalar, labelencoder, RFECV, PCA t-SNE and UMAP, GridsearchCV, Autoencoder, MLP, DT, SVM and XGBoost were applied.	The unlabelled dataset was trained using autoencoders for anomaly detection, while the dataset was processed for the MLP, DT, SVM and XGBoost in signature detection	The most appropriate models investigated. These initial results can be used in the final development.
Testing	Performance evaluation	Feature selection based on RFECV with 5-fold, dimensionality reduction tested for each model, GirdsearchCV applied, and five models trained and tested.	RFECV, PCA, t-SNE, UMAP, GirdsearchCV, autoencoder, and all models' results were presented.	The anomaly model trained and tested. The signature models trained and tested.	All models can perform best when using PCA. The final production must not consider DT and SVM with t-SNE and UMAP.

Table 6.4 illustrates the final findings of the proposed H-IDS. In the first phase, all requirements were accomplished, including the normal and attack patterns dataset, tool, and other formal requirements for building H-IDS. In the second phase, the design of the H-IDS was created, which included the flowchart representing the preprocessing, feature engineering, model building, and analysis.

In the third phase, the models' implementation, including model selection and preparation, was completed. This implementation processed the dataset, identified correlation, applied the RFECV technique, applied PCA, t-SNE and UMAP, applied GirdsearchCV, and trained five models to detect the medjacking patterns. The implementation process trained the five modes to detect medjacking attacks. This study implemented the initial development of the H-IDS; however, the final product can build the integration of a hybrid approach; for this purpose, testing was required to compare each model's performance before investing in the final product development. The choice of models was tested for anomaly and signature detection. A hierarchical Fusion algorithm can be used for a hybrid approach. Hierarchical fusion is decision-making logic: if the autoencoder model detects the anomaly and then prioritises the classification outcome, this can be done in the final development.

In the fourth phase, testing was performed to compare the performance of each model. The experiments included anomaly detection based on each attack. The MLP and XGBoost can perform optimally with all dimensionality reduction techniques. However, DT and SVM can only perform better with PCA to detect medjacking. In signature detection, t-SNE and UMAP did not perform well in DT and SVM models.

6.6 Summary

Performance and detection accuracy were ideal for the H-IDS. Autoencoder detection of brute force attacks yielded a training accuracy of 80%–82% and a validation accuracy of

76%–88%. The DDoS attack findings indicate a training accuracy of 78%–82% and a validation accuracy of 70%–88%. The training accuracy for MITM attacks was 75–82% and the validation accuracy was 85%–94%. For Trojan attacks, the training accuracy was 75%–91% and the validation accuracy was 83%–92%. Different classification models were compared to determine medjacking detection efficacy. Compared to DT and SVM models, the MLP model with PCA, t-SNE, UMAP, and without dimensionality reduction predicted medjacking with satisfactory accuracy. With PCA, the DT model performed well, but with the t-SNE and UMAP reduction, it performed poorly (47.970 and 62.961, respectively). The SVM model for medjacking detection worked well with an accuracy of 73.194 while utilising PCA with an optimised hyperparameter, and worked well without dimensionality reduction. However, the SVM model with the t-SNE and UMAP techniques failed to predict medjacking assaults with a very low accuracy score of 16.893 and 36.452, respectively. This research compares the XGBoost model's performance to three other models. This study found that the XGBoost model with all-dimensionality reduction, and without dimensionality reduction predicted the medjacking attacks better than all other models.

Chapter 7. Conclusion

The research conducted in this thesis investigated the phenomenon of medjacking in APS devices; a comprehensive investigation has been conducted to examine medjacking to develop mitigation solution. The APS prototype was developed using a simulation approach, and a cybersecurity testbed was developed to exemplify medjacking attacks, ultimately, the dataset was collected. 2022APS Dataset is a novel dataset that includes the instances of APS normal and attack features. In this study, normal and attack patterns data were gathered, and the correlation of different features was conducted using the EDA technique to gain insights into the patterns. This research has developed the H-IDS to detect medjacking patterns in APS.

In the anomaly detection phase, an autoencoder was used to train on unlabelled 2022APS dataset consisting of four attacks and APS devices. This autoencoder method enabled the development of a compressed understanding of patterns. The autoencoder model's performance metrics were evaluated for Brute-force, DDoS, MITM, and Trojan attacks, the model predicted anomalies with accurate pattern reconstruction and low training and validation losses. The autoencoder exhibited consistent effectiveness in terms of accuracy and loss, thereby indicating its versatility in detecting anomalies.

In the signature detection phase, feature selection was used to choose a subset of relevant features, including correlation coefficient, variance threshold, and RFECV. Subsequently, PCA, t-SNE, and UMAP dimensionality reduction techniques were utilised to determine their optimal performance with medjacking classification. A grid search technique used for hyperparameters, and MLP, DT, SVM, and XGBoost models, were developed to detect medjacking. This research compared the performance of classification models with and without dimensionality reduction techniques. This research's quantitative and theoretical contribution is to assess the quantitative performance of different classification models'

effectiveness using the 2022APS dataset. Based on quantitative performance, this study presents the theoretical findings of each model.

The MLP model has performed exceptionally well with various dimensionality reduction techniques and without dimensionality reduction techniques. The DT model's performance was exceptional when PCA dimensionality reduction was applied with 10 and 14 components, and without dimensionality reduction, the performance was optimal. However, the DT model's performance was unsatisfactory when t-SNE and UMAP were used. The performance of the SVM model when using PCA with 10 and 14 components was satisfactory. However, the performance of the SVM model when using t-SNE and UMAP was not satisfactory, and as a result, the SVM model, when using t-SNE and UMAP, was unable to predict medjacking attacks; this indicates that the model failed to predict the medjacking instances. These SVM findings contribute to the research community quantitively and theoretically findings contribution. Another quantitative finding this study has found that the XGBoost model when using any of the dimensionality reduction techniques, could identify the medjacking patterns with good accuracy also, when using the XGBoost without dimensionally reduction, the model performed with satisfactory accuracy.

According to performance analysis, quantitively this study found that the MLP and XGBoost models, when using all dimensionality reductions, have shown exceptional performance. However, the DT and SVM models only identified medjacking patterns when the PCA with 10 and 14 components dimensionality reduction technique was employed. The signature detection approach proved to be extremely effective and perfectly tailored for automated medjacking pattern recognition. By utilising predetermined signatures, rapid identification becomes possible. Once these patterns are identified, the response to attacks can be promptly executed to effectively minimise the risk of medjacking attacks. By integrating anomaly- and signature detection methods, the limitations of each approach can be minimised.

This study has shown the feasibility of automated detection medjacking by integrating anomaly- and signature detection systems using ML and DL approaches. This proposed system can be further improved to help end users with early warning. However, early warning can only be attained in the final development of the software or application, so different actions can be performed depending on the software or predefined policies.

The significance of this study aimed to investigate weaknesses in an embedded APS. The wider problem of medjacking is the extent of embedded devices that are vulnerable simply because of data transfer requirements and current connectivity protocols. Automated detection methods can reduce medjacking risks and improve evidence-based detection, protect APS devices from medjacking.

7.1 Future Research

The investigation of medjacking in the APS and the development of the H-IDS discussed in this thesis provide a foundation for building and examining the effectiveness of the H-IDS from a practical viewpoint. Potential areas for future research aligned with this thesis are briefly described below.

This study has presented the details of APS, and the integration of digital technology to manage diabetes disease. These devices were not easily available to conduct the security assessment due to preoperative devices, which are a subset of non-free property with constraints and restrictions. Ethical permission was also required to conduct a thorough medjacking investigation. Therefore, this study proposes future research to conduct the medjacking investigation on APS preoperative devices. Future research involving preoperative devices should not include actual patients unless their participation is thoroughly deliberated with medical professionals.

This study provided the 2022APS dataset, the dataset consisted of the normal and attack patterns associated with medjacking. This study showed the correlation between normal and

attack patterns. To improve correlation, future research could apply this correlation strategy to compare results from simulated and preoperative devices. This study strongly recommends future research to collect a dataset from preoperative devices and investigate the correlation between normal and medjacking patterns.

Exploring the concept of dimensionality reduction problems in ML across different domains is complex, particularly in the development of ML and DL models for medjacking detection. This study utilised three algorithms for dimensionality reduction and performance of classification models were computed without dimensionality reduction techniques. The PCA algorithm demonstrated outstanding effectiveness in all four ML models analysed for medjacking detection, and without dimensionality redaction the performance of all models was optimal. However, t-SNE and UMAP have shown outstanding results specifically with the MLP and XGBoost models. To gain a deeper understanding of these three dimensionality reduction techniques, future research can utilise these findings and compare them with the dataset from the preoperative device. Another aspect worth exploring is the comparison of feature selection methods in the development of models.

The ML and DL models' comparisons, based on three dimensionality reductions in detecting four medjacking attacks, are proposed as a proof of concept. These models showed variations in accuracy depending on the dimensionality reduction technique. To investigate these variations further, it is recommended that these developed models be trained using preoperative devices to compare the differences. This thesis examined the most progressive and advanced ML and DL models; however, future investigation of the NB, LR, LightGBM, GRU, and GAN models with PCA, t-SNE, and UMAP dimensionality reduction is recommended for further research on detecting medjacking.

This research acknowledges the absence of real-world data pertaining to the specific issue, primarily related to ethical considerations, privacy concerns, and patient safety. To

mitigate this limitation, synthetic data was used for the training of the ML and DL models. Although synthetic data can replicate real-world scenarios and serve as a foundation for model development, the effectiveness of these models in the APS continues to be an important consideration. The prospective gap between synthetic and real-world data highlights the necessity of validating model effectiveness on real-world datasets, when accessible, to ensure both reliability and applicability. Furthermore, transfer learning emerges as an optimal strategy to tackle this challenge. Through the utilisation of pre-trained models within comparable domains, these models can be fine-tuned using constrained real-world data to enhance their generalisation and adaptability. Transfer learning can be one of the future directions, exploring the intersection of transfer learning and empirical validation can help as an explanation for the difference between synthetic training contexts and actual implementation circumstances.

Bibliography

1. Gybel Jensen, C., F. Gybel Jensen, and M.I. Loft, *Patients' Experiences With Digitalization in the Health Care System: Qualitative Interview Study.* Journal of Medical Internet Research, 2024. **26**: p. e47278.
2. Lapão, L.V., *The future of healthcare: the impact of digitalization on healthcare services performance.* The internet and health in Brazil: Challenges and trends, 2019: p. 435-449.
3. Wiig, S., et al., *What methods are used to promote patient and family involvement in healthcare regulation? A multiple case study across four countries.* BMC Health Services Research, 2020. **20**: p. 1-15.
4. Keenan, G., et al., *Challenges to nurses' efforts of retrieving, documenting, and communicating patient care information.* Journal of the American Medical Informatics Association, 2013. **20**(2): p. 245-251.
5. Hacker, K., *The burden of chronic disease.* Mayo Clinic Proceedings: Innovations, Quality & Outcomes, 2024. **8**(1): p. 112-119.
6. Strulik, H. and V. Grossmann, *The economics of aging with infectious and chronic diseases.* Economics & Human Biology, 2024. **52**: p. 101319.
7. Naik, N., et al., *Transforming healthcare through a digital revolution: a review of digital healthcare technologies and solutions.* Frontiers in digital health, 2022. **4**: p. 919985.
8. Akhtar, N., et al., *Efficacy and pitfalls of digital technologies in healthcare services: a systematic review of two decades.* Frontiers in public health, 2022. **10**: p. 869793.
9. Sikandar, H., et al., *Digital technologies in healthcare: A systematic review and bibliometric analysis.* 2022.
10. Senbekov, M., et al., *The recent progress and applications of digital technologies in healthcare: a review.* International journal of telemedicine and applications, 2020. **2020**(1): p. 8830200.
11. Ashfaq, Z., et al., *A review of enabling technologies for Internet of Medical Things (IoMT) Ecosystem.* Ain Shams Engineering Journal, 2022. **13**(4): p. 101660.
12. Huang, C., et al., *Internet of medical things: A systematic review.* Neurocomputing, 2023: p. 126719.
13. França, R.P., et al., *An Overview of the Internet of Medical Things and Its Modern Perspective.* Efficient Data Handling for Massive Internet of Medical Things: Healthcare Data Analytics, 2021: p. 1-23.
14. Al-Turjman, F., M.H. Nawaz, and U.D. Ulusar, *Intelligence in the Internet of Medical Things era: A systematic review of current and future trends.* Computer Communications, 2020. **150**: p. 644-660.
15. Shelke, Y., *IoMT and healthcare delivery in chronic diseases.* Advances in Telemedicine for Health Monitoring, 2020. **239**.
16. Mathkor, D.M., et al., *Multirole of the internet of medical things (IoMT) in biomedical systems for managing smart healthcare systems: An overview of current and future innovative trends.* Journal of infection and public health, 2024.
17. Siddiqui, S., A.A. Khan, and I. Dey, *Internet Technologies for Personalized Care*, in *Information and Communication Technology (ICT) Frameworks in Telehealth*. 2022, Springer. p. 173-189.
18. Sandulescu, V., et al., *Integrating IoMT and AI for Proactive Healthcare: Predictive Models and Emotion Detection in Neurodegenerative Diseases.* Algorithms, 2024. **17**(9): p. 376.

19. Ramalingam, S., et al. *Machine Learning Based Remote Health Care Data Monitoring and Prediction System Using IoMT*. in *2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 2024. IEEE.
20. Baseer, K., et al., *Healthcare diagnostics with an adaptive deep learning model integrated with the Internet of medical Things (IoMT) for predicting heart disease.* Biomedical Signal Processing and Control, 2024. **92**: p. 105988.
21. Rodríguez-Rodríguez, I., et al., *IoMT innovations in diabetes management: Predictive models using wearable data.* Expert Systems with Applications, 2024. **238**: p. 121994.
22. Gupta, S., et al., *Personal HealthCare of Things: A novel paradigm and futuristic approach.* CAAI Transactions on Intelligence Technology, 2023.
23. Keikhosrokiani, P., *IoT for enhanced decision-making in medical information systems: A systematic review.* Enhanced telemedicine and e-Health: Advanced IoT Enabled soft computing framework, 2021: p. 119-140.
24. Condry, M.W. and X.I. Quan, *Remote patient monitoring technologies and markets.* IEEE Engineering Management Review, 2023. **51**(3): p. 59-64.
25. Tan, S.Y., et al., *A systematic review of the impacts of remote patient monitoring (RPM) interventions on safety, adherence, quality-of-life and cost-related outcomes.* NPJ Digital Medicine, 2024. **7**(1): p. 192.
26. Su, C.-R., et al., *A novel framework for a remote patient monitoring (RPM) system with abnormality detection.* Health Policy and Technology, 2019. **8**(2): p. 157-170.
27. Boikanyo, K., et al., *Remote patient monitoring systems: Applications, architecture, and challenges.* Scientific African, 2023. **20**: p. e01638.
28. Anand, D., G. Singh, and V.K. Gupta, *Remote Patient Monitoring: An Overview of Technologies, Applications, and Challenges.* Handbook on Augmenting Telehealth Services, 2024: p. 213-232.
29. Bhatia, A. and T.M. Maddox, *Remote patient monitoring in heart failure: factors for clinical efficacy.* International Journal of Heart Failure, 2021. **3**(1): p. 31.
30. Kakria, P., N. Tripathi, and P. Kitipawang, *A real-time health monitoring system for remote cardiac patients using smartphone and wearable sensors.* International journal of telemedicine and applications, 2015. **2015**(1): p. 373474.
31. Malasinghe, L.P., N. Ramzan, and K. Dahal, *Remote patient monitoring: a comprehensive study.* Journal of Ambient Intelligence and Humanized Computing, 2019. **10**: p. 57-76.
32. Kok, C.L., et al. *Development and Evaluation of an IoT-Driven Auto-Infusion System with Advanced Monitoring and Alarm Functionalities*. in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2024. IEEE.
33. Recmanik, M., et al., *A Review of Patient Bed Sensors for Monitoring of Vital Signs.* Sensors, 2024. **24**(15): p. 4767.
34. Maddeh, M., et al., *Spatio-temporal cluster mapping system in smart beds for patient monitoring.* Sensors, 2023. **23**(10): p. 4614.
35. Ghozali, M.T., *Implementation of the IoT-based technology on patient medication adherence: A comprehensive bibliometric and systematic review.* Journal of Information and Communication Technology, 2023. **22**(4): p. 503-544.
36. Karagiannis, D., K. Mitsis, and K.S. Nikita, *Development of a low-power IoMT portable pillbox for medication adherence improvement and remote treatment adjustment.* Sensors, 2022. **22**(15): p. 5818.

37. Ismail, W. *Internet of medical things (IoMT) for patient healthcare monitoring system*. in *2019 IEEE 14th Malaysia international conference on communication (MICC)*. 2019. IEEE.
38. Laila, I., A. Arifin, and B. Armynah, *Internet of medical things (iomt)-based heart rate and body temperature monitoring system.* Indonesian Physical Review, 2022. **5**(1): p. 1-14.
39. Arthi, K. and B. Chidhambararajan, *Advanced Enabling Technologies of IoMT in Personalized Healthcare*, in *Technological Advancement in Internet of Medical Things and Blockchain for Personalized Healthcare*. 2024, CRC Press. p. 1-15.
40. Fuada, S., M. Särestöniemi, and M. Katz, *Analyzing the trends and global growth of energy harvesting for Implantable Medical Devices (IMDs) research—A bibliometric approach.* 2024.
41. Arun Ganesh, K., N. Sivakumaran, and S. Kumaravel. *Lower Limb Amputees Rehabilitation: IOT-Based Real-Time Human Movement*. in *Advances in Automation, Signal Processing, Instrumentation, and Control: Select Proceedings of i-CASIC 2020*. 2021. Springer.
42. Jeyaraman, M., A. Nallakumarasamy, and N. Jeyaraman, *Industry 5.0 in orthopaedics.* Indian Journal of Orthopaedics, 2022. **56**(10): p. 1694-1702.
43. Wilden, J., et al. *IoT based wearable smart insole*. in *2017 Global Wireless Summit (GWS)*. 2017. IEEE.
44. Rukmini, P.G., et al., *Recent Innovations in Footwear and the Role of Smart Footwear in Healthcare—A Survey.* Sensors, 2024. **24**(13): p. 4301.
45. Ghazi, R.F., J.S. Chiad, and F.M. Abdulghani, *Design and manufacturing a smart shoe for diabetic foot ulcer monitoring and prediction system using internet-of-things technology.* Journal of the Brazilian Society of Mechanical Sciences and Engineering, 2024. **46**(2): p. 83.
46. Rasouli, J.J., et al., *Artificial intelligence and robotics in spine surgery.* Global Spine Journal, 2021. **11**(4): p. 556-564.
47. Bommu, R., *Advancements in Medical Device Software: A Comprehensive Review of Emerging Technologies and Future Trends.* Journal of Engineering and Technology, 2022. **4**(2): p. 1− 8-1− 8.
48. Athanasopoulou, K., et al., *Artificial intelligence: the milestone in modern biomedical research.* BioMedInformatics, 2022. **2**(4): p. 727-744.
49. Rukasha, T., et al., *Evaluation of wearable electronics for epilepsy: A systematic review.* Electronics, 2020. **9**(6): p. 968.
50. Tang, J., et al., *Seizure detection using wearable sensors and machine learning: Setting a benchmark.* Epilepsia, 2021. **62**(8): p. 1807-1819.
51. Olsen, L.S., et al., *Wearables in real life: A qualitative study of experiences of people with epilepsy who use home seizure monitoring devices.* Epilepsy & Behavior, 2021. **125**: p. 108398.
52. Brinkmann, B.H., et al., *Seizure diaries and forecasting with wearables: epilepsy monitoring outside the clinic.* Frontiers in Neurology, 2021. **12**: p. 690404.
53. Hassan, S., E. Mwangi, and P.K. Kihato, *IoT based monitoring system for epileptic patients.* Heliyon, 2022. **8**(6).
54. Srinivas, P., et al. *Support Vector Machines Based Predictive Seizure Care using IoT-Wearable EEG Devices for Proactive Intervention in Epilepsy*. in *2024 2nd International Conference on Computer, Communication and Control (IC4)*. 2024. IEEE.

55. Raza, M., et al., *Intelligent IoT framework for indoor healthcare monitoring of Parkinson's disease patient.* IEEE Journal on Selected Areas in Communications, 2020. **39**(2): p. 593-602.
56. Giannakopoulou, K.-M., I. Roussaki, and K. Demestichas, *Internet of things technologies and machine learning methods for Parkinson's disease diagnosis, monitoring and management: a systematic review.* Sensors, 2022. **22**(5): p. 1799.
57. Romero, L.E., P. Chatterjee, and R.L. Armentano, *An IoT approach for integration of computational intelligence and wearable sensors for Parkinson's disease diagnosis and monitoring.* Health and Technology, 2016. **6**: p. 167-172.
58. d'Angelis, O., et al., *IoT architecture for continuous long term monitoring: Parkinson's Disease case study.* Internet of Things, 2022. **20**: p. 100614.
59. Islam, M., et al., *Internet of Things Device Capabilities, Architectures, Protocols, and Smart Applications in Healthcare Domain: A Review.* arXiv preprint arXiv:2204.05921, 2022.
60. Gatouillat, A., et al., *Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine.* IEEE internet of things journal, 2018. **5**(5): p. 3810-3822.
61. Kang, M., et al., *Recent patient health monitoring platforms incorporating internet of things-enabled smart devices.* International neurourology journal, 2018. **22**(Suppl 2): p. S76.
62. Koydemir, H.C. and A. Ozcan, *Wearable and implantable sensors for biomedical applications.* Annual Review of Analytical Chemistry, 2018. **11**: p. 127-146.
63. Khan, Y., et al., *Monitoring of vital signs with flexible and wearable medical devices.* Advanced Materials, 2016. **28**(22): p. 4373-4395.
64. Joyia, G.J., et al., *Internet of Medical Things (IOMT): applications, benefits and future challenges in healthcare domain.* J Commun, 2017. **12**(4): p. 240-247.
65. Rubí, J.N.S. and P.R.d.L. Gondim, *Interoperable Internet of Medical Things platform for e-Health applications.* International Journal of Distributed Sensor Networks, 2020. **16**(1): p. 1550147719889591.
66. Wei, K., et al., *Health monitoring based on internet of medical things: architecture, enabling technologies, and applications.* IEEE Access, 2020. **8**: p. 27468-27478.
67. Sundaravadivel, P., et al., *Everything you wanted to know about smart health care: Evaluating the different technologies and components of the internet of things for better health.* IEEE Consumer Electronics Magazine, 2017. **7**(1): p. 18-28.
68. Umpierrez, G.E. and D.C. Klonoff, *Diabetes technology update: use of insulin pumps and continuous glucose monitoring in the hospital.* Diabetes care, 2018. **41**(8): p. 1579-1589.
69. Arduser, L., *Impatient patients: A DIY usability approach in diabetes wearable technologies.* Communication Design Quarterly Review, 2018. **5**(4): p. 31-39.
70. Deshpande, S., et al., *Design and clinical evaluation of the Interoperable Artificial Pancreas System (iAPS) smartphone app: interoperable components with modular design for progressive artificial pancreas research and development.* Diabetes technology & therapeutics, 2019. **21**(1): p. 35-43.
71. Kesavadev, J., et al., *Evolution of insulin delivery devices: from syringes, pens, and pumps to DIY artificial pancreas.* Diabetes Therapy, 2020. **11**: p. 1251-1269.
72. Kesavadev, J., et al., *The do-it-yourself artificial pancreas: a comprehensive review.* Diabetes Therapy, 2020. **11**(6): p. 1217-1235.
73. Doyle-Delgado, K. and J.J. Chamberlain, *Use of Diabetes-Related Applications and Digital Health Tools by People With Diabetes and Their Health Care Providers.* Clinical Diabetes, 2020. **38**(5): p. 449-461.

74. Smith, A.A., R. Li, and Z.T.H. Tse, *Reshaping healthcare with wearable biosensors.* Scientific Reports, 2023. **13**(1): p. 4998.
75. Lau, C.-P., C.-W. Siu, and H.-F. Tse, *Future of implantable devices for cardiac rhythm management.* Circulation, 2014. **129**(7): p. 811-822.
76. Ricotti, L., et al., *Wearable and implantable pancreas substitutes.* Journal of Artificial Organs, 2013. **16**(1): p. 9-22.
77. Andreu-Perez, J., et al., *From wearable sensors to smart implants-–toward pervasive and personalized healthcare.* IEEE Transactions on Biomedical Engineering, 2015. **62**(12): p. 2750-2762.
78. Yaacoub, J.-P.A., et al., *Securing internet of medical things systems: limitations, issues and recommendations.* Future Generation Computer Systems, 2020. **105**: p. 581-606.
79. Somasundaram, R. and M. Thirugnanam, *Review of security challenges in healthcare internet of things.* Wireless Networks, 2021. **27**(8): p. 5503-5509.
80. Chacko, A. and T. Hayajneh, *Security and privacy issues with IoT in healthcare.* EAI Endorsed Transactions on Pervasive Health and Technology, 2018. **4**(14).
81. Thompson, B., et al., *An overview of safety issues on use of insulin pumps and continuous glucose monitoring systems in the hospital.* Current diabetes reports, 2018. **18**(10): p. 1-8.
82. Khera, M., *Think like a hacker: Insights on the latest attack vectors (and security controls) for medical device applications.* Journal of diabetes science and technology, 2017. **11**(2): p. 207-212.
83. Alsubaei, F., A. Abuhussein, and S. Shiva. *Security and privacy in the internet of medical things: taxonomy and risk assessment*. in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*. 2017. IEEE.
84. McMahon, E., et al. *Assessing medical device vulnerabilities on the Internet of Things*. in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2017. IEEE.
85. Aileni, R.M., et al., *Cybersecurity technologies for the internet of medical wearable devices (iomwd)*, in *Advances in Cyber Security Analytics and Decision Systems*. 2020, Springer, Cham. p. 117-140.
86. Radcliffe, J. *Hacking medical devices for fun and insulin: Breaking the human SCADA system*. in *Black Hat Conference presentation slides*. 2011.
87. Voelker, R., *Insulin Pumps Could Be Hacked.* Jama, 2019. **322**(5): p. 393-393.
88. Klonoff, D. and J. Han, *The first recall of a diabetes device because of cybersecurity risks*. 2019, SAGE Publications Sage CA: Los Angeles, CA.
89. Armstrong, D.G., et al., *Cybersecurity regulation of wireless devices for performance and assurance in the age of "medjacking".* Journal of diabetes science and technology, 2016. **10**(2): p. 435-438.
90. Reverberi, L. and D. Oswald. *Breaking (and fixing) a widely used continuous glucose monitoring system*. in *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*. 2017.
91. Britton, K.E. and J.D. Britton-Colonnese, *Privacy and security issues surrounding the protection of data generated by continuous glucose monitors.* Journal of diabetes science and technology, 2017. **11**(2): p. 216-219.
92. Klonoff, D.C., *Cybersecurity for connected diabetes devices.* Journal of diabetes science and technology, 2015. **9**(5): p. 1143-1147.
93. Beavers, J.L., M. Faulks, and J. Marchang. *Hacking NHS pacemakers: a feasibility study*. in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*. 2019. IEEE.

94. Baranchuk, A., et al., *Cybersecurity for cardiac implantable electronic devices: what should you know?* Journal of the American College of Cardiology, 2018. **71**(11): p. 1284-1288.
95. Primer, I.H.A., *IoT Hacking–A Primer.*
96. Chardenet, B., et al., *Exploring Vulnerabilities and Attack Vectors Targeting Pacemaker Devices in Healthcare.*
97. Kramer, D.B. and K. Fu, *Cybersecurity concerns and medical devices: lessons from a pacemaker advisory.* Jama, 2017. **318**(21): p. 2077-2078.
98. Puat, H.A.M. and N.A. Abd Rahman. *IoMT: a review of pacemaker vulnerabilities and security strategy*. in *Journal of Physics: Conference Series*. 2020. IOP Publishing.
99. Stergiopoulos, G., et al., *Process-Aware Attacks on Medication Control of Type-I Diabetics Using Infusion Pumps.* IEEE Systems Journal, 2023.
100. Finkle, J., *FDA warns of security flaw in Hospira infusion pumps.* Technol. News Reuters, 2015.
101. Zheng, G., et al., *Ideas and challenges for securing wireless implantable medical devices: A review.* IEEE Sensors Journal, 2016. **17**(3): p. 562-576.
102. Stergiopoulos, G., et al., *Process-Aware Attacks on Medication Control of Type-I Diabetics Using Infusion Pumps.* IEEE Systems Journal, 2023. **17**(2): p. 1831-1842.
103. Newaz, A., et al., *A survey on security and privacy issues in modern healthcare systems: Attacks and defenses.* arXiv preprint arXiv:2005.07359, 2020.
104. Webster, M., *Do No Harm: protecting connected medical devices, healthcare, and data from hackers and adversarial nation states*. 2021: John Wiley & Sons.
105. Somasundaram, R. and M. Thirugnanam, *Review of security challenges in healthcare internet of things.* Wireless Networks, 2020: p. 1-7.
106. Manikandan, A., et al., *Issues and challenges in security and privacy with e-Healthcare.* Internet of Things enabled Machine Learning for Biomedical Application, 2024: p. 222.
107. Li, N., et al., *A review of security issues and solutions for precision health in Internet-of-Medical-Things systems.* Security and Safety, 2023. **2**: p. 2022010.
108. Awotunde, J.B., et al., *AIoMT enabling real-time monitoring of healthcare systems: security and privacy considerations.* Handbook of Security and Privacy of AI-Enabled Healthcare Systems and Internet of Medical Things, 2024: p. 97-133.
109. Affia, A.-a.O., et al., *IoT health devices: exploring security risks in the connected landscape.* IoT, 2023. **4**(2): p. 150-182.
110. JOSE, S., *Report Identifies 162 Vulnerabilities, Potentially Exposing Patient Data, Disrupting Healthcare Operations, and Posing Risks to Patient Safety.* 2024.
111. Jimmy, F., *Emerging threats: The latest cybersecurity risks and the role of artificial intelligence in enhancing cybersecurity defenses.* Valley International Journal Digital Library, 2021: p. 564-574.
112. Reddy, A.R.P., *The Role of Artificial Intelligence in Proactive Cyber Threat Detection In Cloud Environments.* NeuroQuantology, 2021. **19**(12): p. 764-773.
113. Thapaliya, S. and A. Bokani, *Leveraging artificial intelligence for enhanced cybersecurity: insights and innovations.* SADGAMAYA, 2024. **1**(1): p. 46-52.
114. Mahboubi, A., et al., *Evolving techniques in cyber threat hunting: A systematic review.* Journal of Network and Computer Applications, 2024: p. 104004.
115. Sarker, I.H., *Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects.* Annals of Data Science, 2023. **10**(6): p. 1473-1498.

116. Sarker, I.H., et al., *Cybersecurity data science: an overview from machine learning perspective.* Journal of Big data, 2020. **7**: p. 1-29.
117. Shan, A. and S. Myeong, *Proactive Threat Hunting in Critical Infrastructure Protection through Hybrid Machine Learning Algorithm Application.* Sensors, 2024. **24**(15): p. 4888.
118. Kayhan, V., et al., *Unsupervised Threat Hunting using Continuous Bag-of-Terms-and-Time (CBoTT).* arXiv preprint arXiv:2403.10327, 2024.
119. Yazdinejad, A., et al., *Accurate threat hunting in industrial internet of things edge devices.* Digital Communications and Networks, 2023. **9**(5): p. 1123-1130.
120. Jahromi, A.N., H. Karimipour, and A. Dehghantanha, *An ensemble deep federated learning cyber-threat hunting model for Industrial Internet of Things.* Computer Communications, 2023. **198**: p. 108-116.
121. Al-Fuqaha, A., et al., *Internet of things: A survey on enabling technologies, protocols, and applications.* IEEE communications surveys & tutorials, 2015. **17**(4): p. 2347-2376.
122. Yang, Y., et al., *IoT Technologies and Applications*, in *Fog-Enabled Intelligent IoT Systems*. 2020, Springer. p. 1-37.
123. Oktian, Y.E., E.N. Witanto, and S.-G. Lee, *A conceptual architecture in decentralizing computing, storage, and networking aspect of IoT infrastructure.* IoT, 2021. **2**(2): p. 205-221.
124. Sun, L., et al., *Edge-cloud computing and artificial intelligence in internet of medical things: architecture, technology and application.* IEEE access, 2020. **8**: p. 101079-101092.
125. Dilibal, Ç. *Development of edge-IoMT computing architecture for smart healthcare monitoring platform*. in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. 2020. IEEE.
126. Ben Dhaou, I., et al., *Edge devices for internet of medical things: technologies, techniques, and implementation.* Electronics, 2021. **10**(17): p. 2104.
127. Gyamfi, E., J.A. Ansere, and M. Kamal, *Network Security System in Mobile Edge Computing-to-IoMT Networks Using Distributed Approach*, in *Security and Risk Analysis for Intelligent Edge Computing*. 2023, Springer. p. 171-191.
128. Subramaniam, E.V.D., et al., *Interoperable IoMT approach for remote diagnosis with privacy-preservation perspective in edge systems.* Sensors, 2023. **23**(17): p. 7474.
129. Cheruvu, S., et al., *Connectivity technologies for IoT.* Demystifying Internet of Things Security: Successful IoT Device/Edge and Platform Security Deployment, 2020: p. 347-411.
130. Buthelezi, B.E., et al., *ZigBee healthcare monitoring system for ambient assisted living environments.* International Journal of Communication Networks and Information Security, 2019. **11**(1): p. 85-92.
131. Tosi, J., et al., *Performance evaluation of bluetooth low energy: A systematic review.* Sensors, 2017. **17**(12): p. 2898.
132. Falcone, F., et al., *Future Wireless Communication Systems to Enable IoMT Services and Applications*, in *Smart and Secure Internet of Healthcare Things*. 2022, CRC Press. p. 137-158.
133. Razdan, S. and S. Sharma, *Internet of medical things (IoMT): Overview, emerging technologies, and case studies.* IETE technical review, 2022. **39**(4): p. 775-788.
134. Markova, V., et al. *Intelligent Architecture for Real-time Personal Health Monitoring in the Context of 3G/4G/5G Networks*. in *2021 29th National Conference with International Participation (TELECOM)*. 2021. IEEE.

135. Hassan, N., K.-L.A. Yau, and C. Wu, *Edge computing in 5G: A review.* IEEE Access, 2019. **7**: p. 127276-127289.
136. Sahoo, A.K., V. Kompally, and S.K. Udgata. *Wi-fi sensing based real-time activity detection in smart home environment*. in *2023 IEEE Applied Sensing Conference (APSCON)*. 2023. IEEE.
137. Xhafa, F., B. Kilic, and P. Krause, *Evaluation of IoT stream processing at edge computing layer for semantic data enrichment.* Future Generation Computer Systems, 2020. **105**: p. 730-736.
138. Zikria, Y.B., et al., *Internet of things (IoT): Operating system, applications and protocols design, and validation techniques*. 2018, Elsevier. p. 699-706.
139. Li, J.-Q., et al., *Industrial internet: A survey on the enabling technologies, applications, and challenges.* IEEE Communications Surveys & Tutorials, 2017. **19**(3): p. 1504-1526.
140. Badarinath, R. and V.V. Prabhu. *Advances in internet of things (IoT) in manufacturing*. in *Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing: IFIP WG 5.7 International Conference, APMS 2017, Hamburg, Germany, September 3-7, 2017, Proceedings, Part I*. 2017. Springer.
141. Alaa, M., et al., *A review of smart home applications based on Internet of Things.* Journal of network and computer applications, 2017. **97**: p. 48-65.
142. Choi, W., et al., *Smart home and internet of things: A bibliometric study.* Journal of Cleaner Production, 2021. **301**: p. 126908.
143. Ayan, O. and B. Turkay. *Smart thermostats for home automation systems and energy savings from smart thermostats*. in *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*. 2018. IEEE.
144. Hamdan, O., et al. *IoT-based interactive dual mode smart home automation*. in *2019 IEEE international conference on consumer electronics (ICCE)*. 2019. IEEE.
145. Sung, G.-M., et al., *Internet of Things–based smart home system using a virtualized cloud server and mobile phone app.* International Journal of Distributed Sensor Networks, 2019. **15**(9): p. 1550147719879354.
146. Ghubaish, A., et al., *Recent advances in the internet-of-medical-things (IoMT) systems security.* IEEE Internet of Things Journal, 2020. **8**(11): p. 8707-8718.
147. Cappon, G., et al., *Wearable continuous glucose monitoring sensors: a revolution in diabetes treatment.* Electronics, 2017. **6**(3): p. 65.
148. Cleveland, S.M. and M. Haddara. *IoT for Diabetics: A User Perspective*. in *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 3*. 2021. Springer.
149. Shin, D. and Y. Hwang, *Integrated acceptance and sustainability evaluation of Internet of Medical Things.* Internet Research, 2017.
150. Singh, R.P., et al., *Internet of Medical Things (IoMT) for orthopaedic in COVID-19 pandemic: Roles, challenges, and applications.* Journal of Clinical Orthopaedics and Trauma, 2020.
151. Yang, T., et al., *Combining point-of-care diagnostics and internet of medical things (IoMT) to combat the COVID-19 pandemic*. 2020, Multidisciplinary Digital Publishing Institute.
152. Asif-Ur-Rahman, M., et al., *Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things.* IEEE Internet of Things Journal, 2018. **6**(3): p. 4049-4062.
153. Farahani, B., F. Firouzi, and K. Chakrabarty, *Healthcare iot*, in *Intelligent Internet of Things*. 2020, Springer. p. 515-545.

154. Kulkarni, A. and S. Sathe, *Healthcare applications of the Internet of Things: A Review.* International Journal of Computer Science and Information Technologies, 2014. **5**(5): p. 6229-6232.
155. Qadri, Y.A., et al., *The future of healthcare internet of things: a survey of emerging technologies.* IEEE Communications Surveys & Tutorials, 2020. **22**(2): p. 1121-1167.
156. Rahmani, A.M., et al., *Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach.* Future Generation Computer Systems, 2018. **78**: p. 641-658.
157. Hiremath, S., G. Yang, and K. Mankodiya. *Wearable Internet of Things: Concept, architectural components and promises for person-centered healthcare*. in *2014 4th International Conference on Wireless Mobile Communication and Healthcare-Transforming Healthcare Through Innovations in Mobile and Wireless Technologies (MOBIHEALTH)*. 2014. IEEE.
158. IDF, *Diabetes facts and figures.* 2023.
159. AIHW, *Diabetes: Australian facts.* 2023.
160. DHAC, *New CGM device for Australians with diabetes.* 2022.
161. FDA, R.-t.s., *Tandem Diabetes Care, Inc. Recalls Version 2.7 of the Apple iOS t:connect Mobile App Used in Conjunction with t:slim X2 Insulin Pump with Control-IQ Technology Prompted by a Software Problem Leading to Pump Battery Depletion.* 2024.
162. CSBnetwork, *Faulty insulin pump tech led to hundreds of injuries, prompting app recall.* 2024.
163. Report, D.A., 2024.
164. Kuaban, G.S., et al., *Modelling of the energy depletion process and battery depletion attacks for battery-powered internet of things (iot) devices.* Sensors, 2023. **23**(13): p. 6183.
165. FDA, *FDA warns patients and health care providers about potential cybersecurity concerns with certain Medtronic insulin pumps.* 2019.
166. Konopik, J. and D. Blunck, *Development of an evidence-based conceptual model of the health care sector under digital transformation: integrative review.* Journal of medical Internet research, 2023. **25**: p. e41512.
167. Dionisio, M., et al., *The role of digital transformation in improving the efficacy of healthcare: A systematic review.* The Journal of High Technology Management Research, 2023. **34**(1): p. 100442.
168. Fahim, K.E., K. Kalinaki, and W. Shafik, *Electronic Devices in the Artificial Intelligence of the Internet of Medical Things (AIoMT)*, in *Handbook of Security and Privacy of AI-Enabled Healthcare Systems and Internet of Medical Things*. 2024, CRC Press. p. 41-62.
169. Shakeel, T., et al., *A survey on COVID-19 impact in the healthcare domain: worldwide market implementation, applications, security and privacy issues, challenges and future prospects.* Complex & intelligent systems, 2023. **9**(1): p. 1027-1058.
170. Islam, S.R., et al., *The internet of things for health care: a comprehensive survey.* IEEE access, 2015. **3**: p. 678-708.
171. Evans, J. and K. Ringrose, *From Fitbits to Pacemakers: Protecting Consumer Privacy and Security in the Healthtech Age.* Et Cetera, 2019. **68**(1): p. 1.
172. Bour, G., et al. *Security analysis of the internet of medical things (IoMT): Case study of the pacemaker ecosystem*. in *International Joint Conference on Biomedical Engineering Systems and Technologies*. 2022. Springer.

173. Dutta, S., *Striking a balance between usability and cyber-security in IoT devices*. 2017, Massachusetts Institute of Technology.
174. Chernyshev, M., S. Zeadally, and Z. Baig, *Healthcare data breaches: Implications for digital forensic readiness.* Journal of medical systems, 2019. **43**: p. 1-12.
175. Kwarteng, E. and M. Cebe, *A survey on security issues in modern Implantable Devices: Solutions and future issues.* Smart Health, 2022: p. 100295.
176. Flynn, T., et al. *Knock! Knock! Who is there? Investigating data leakage from a medical internet of things hijacking attack*. in *Proceedings of the 53rd Hawaii International Conference on System Sciences*. 2020.
177. Cook, S., M. Mehrnezhad, and E. Toreini, *Bluetooth security analysis of general and intimate health IoT devices and apps: the case of FemTech.* International Journal of Information Security, 2024: p. 1-21.
178. Thomasian, N.M. and E.Y. Adashi, *Cybersecurity in the internet of medical things.* Health Policy and Technology, 2021. **10**(3): p. 100549.
179. Koutras, D., et al., *Security in IoMT Communications: A Survey.* Sensors, 2020. **20**(17): p. 4828.
180. Zaldivar, D., A.T. Lo'Ai, and F. Muheidat. *Investigating the security threats on networked medical devices*. in *2020 10th annual computing and communication workshop and conference (CCWC)*. 2020. IEEE.
181. Newaz, A.I., et al., *A survey on security and privacy issues in modern healthcare systems: Attacks and defenses.* ACM Transactions on Computing for Healthcare, 2021. **2**(3): p. 1-44.
182. Straw, I., et al., *Simulation-based research for digital health pathologies: A multi-site mixed-methods study.* Digital Health, 2024. **10**: p. 20552076241247939.
183. Wani, R.U.Z., F. Thabit, and O. Can, *Security and privacy challenges, issues, and enhancing techniques for Internet of Medical Things: A systematic review.* Security and Privacy, 2024. **7**(5): p. e409.
184. Ghodsizad, T., *Internet of Medical Things with Considering of Artificial Intelligence.* International journal of sustainable applied science and engineering, 2024. **1**(1): p. 75-102.
185. Khatun, M.A., et al., *Machine Learning for Healthcare-IoT Security: A Review and Risk Mitigation.* IEEE Access, 2023.
186. Sun, Y., F.P.-W. Lo, and B. Lo, *Security and privacy for the internet of medical things enabled healthcare systems: A survey.* IEEE Access, 2019. **7**: p. 183339-183355.
187. Forouhi, N.G. and N.J. Wareham, *Epidemiology of diabetes.* Medicine, 2019. **47**(1): p. 22-27.
188. Egan, A.M. and S.F. Dinneen, *What is diabetes?* Medicine, 2019. **47**(1): p. 1-4.
189. Rasmussen, L., et al., *Diet and healthy lifestyle in the management of gestational diabetes mellitus.* Nutrients, 2020. **12**(10): p. 3050.
190. Association, A.D., *5. Lifestyle management: standards of medical care in diabetes-2019.* Diabetes care, 2019. **42**(Suppl 1): p. S46-S60.
191. Cotter, A.P., et al., *Internet interventions to support lifestyle modification for diabetes management: a systematic review of the evidence.* Journal of Diabetes and its Complications, 2014. **28**(2): p. 243-251.
192. Georga, E.I., et al., *Wearable systems and mobile applications for diabetes disease management.* Health and Technology, 2014. **4**(2): p. 101-112.
193. Lombana, D.A.B. and M. Di Bernardo, *Distributed PID control for consensus of homogeneous and heterogeneous networks.* IEEE Transactions on Control of Network Systems, 2014. **2**(2): p. 154-163.

194. Christofides, P.D., et al., *Distributed model predictive control: A tutorial review and future research directions.* Computers & Chemical Engineering, 2013. **51**: p. 21-41.
195. Wang, X. and M.D. Lemmon, *Event-triggering in distributed networked control systems.* IEEE Transactions on Automatic Control, 2010. **56**(3): p. 586-601.
196. Mubdir, B.A. and H.M.A. Bayram, *Adopting MQTT for a multi protocols IoMT system.* International Journal of Electrical and Computer Engineering, 2022. **12**(1): p. 834.
197. Vishnu, S., S.J. Ramson, and R. Jegan. *Internet of medical things (IoMT)-An overview*. in *2020 5th international conference on devices, circuits and systems (ICDCS)*. 2020. IEEE.
198. Naik, N. *Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP*. in *2017 IEEE international systems engineering symposium (ISSE)*. 2017. IEEE.
199. Billingsley, L., *Cybersmart: Protect the patient, protect the data.* Journal of Radiology Nursing, 2019. **38**(4): p. 261-263.
200. Klonoff, D. and J. Han, *The first recall of a diabetes device because of cybersecurity risks*. 2019, SAGE Publications Sage CA: Los Angeles, CA. p. 817-820.
201. US.FDA, *Medtronic Recalls Remote Controllers Used with Paradigm and 508 MiniMed Insulin Pumps for Potential Cybersecurity Risks.* 2021.
202. Finkle, J., *J&J warns diabetic patients: Insulin pump vulnerable to hacking.* Reuters. Published October, 2016. **4**.
203. Desai, B.C. *IoT: imminent ownership threat*. in *Proceedings of the 21st International Database Engineering & Applications Symposium*. 2017.
204. Alshalalfah, A.-L., G.B. Hamad, and O.A. Mohamed. *Towards system level security analysis of artificial pancreas via uppaal-smc*. in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2019. IEEE.
205. Best, J., *Could implanted medical devices be hacked?* Bmj, 2020. **368**.
206. Pham, C. and M. Diop. *Deploying low-cost, long-range innovative IoT with the WAZIUP IoT/gateway framework*. in *EWSN*. 2018.
207. Aceto, G., V. Persico, and A. Pescapé, *Industry 4.0 and Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0.* Journal of Industrial Information Integration, 2020: p. 100129.
208. Rajan Jeyaraj, P. and E.R.S. Nadar, *Smart-monitor: Patient monitoring system for IoT-based healthcare system using deep learning.* IETE Journal of Research, 2022. **68**(2): p. 1435-1442.
209. Ahmed, S.F., et al., *Insights into Internet of Medical Things (IoMT): Data fusion, security issues and potential solutions.* Information Fusion, 2024. **102**: p. 102060.
210. Osama, M., et al., *Internet of medical things and healthcare 4.0: Trends, requirements, challenges, and research directions.* Sensors, 2023. **23**(17): p. 7435.
211. Maleki Varnosfaderani, S. and M. Forouzanfar, *The role of AI in hospitals and clinics: transforming healthcare in the 21st century.* Bioengineering, 2024. **11**(4): p. 337.
212. Rahman, A., et al., *Machine learning and deep learning-based approach in smart healthcare: Recent advances, applications, challenges and opportunities.* AIMS Public Health, 2024. **11**(1): p. 58.
213. Sundar, R., et al. *Heart health prediction and classification: An IoMT and AI collaborative model*. in *MATEC Web of Conferences*. 2024. EDP Sciences.
214. Zhu, T., et al., *IoMT-enabled real-time blood glucose prediction with deep learning and edge computing.* IEEE Internet of Things Journal, 2022. **10**(5): p. 3706-3719.

215. Nilashi, M., et al., *A hybrid intelligent system for the prediction of Parkinson's Disease progression using machine learning techniques.* Biocybernetics and Biomedical Engineering, 2018. **38**(1): p. 1-15.
216. Vidya, B. and P. Sasikumar, *Gait based Parkinson's disease diagnosis and severity rating using multi-class support vector machine.* Applied Soft Computing, 2021. **113**: p. 107939.
217. Mahmoud, M.H., et al., *An automatic detection system of diabetic retinopathy using a hybrid inductive machine learning algorithm.* Personal and Ubiquitous Computing, 2023: p. 1-15.
218. Alyoubi, W.L., W.M. Shalash, and M.F. Abulkhair, *Diabetic retinopathy detection through deep learning techniques: A review.* Informatics in Medicine Unlocked, 2020. **20**: p. 100377.
219. Das, D., S.K. Biswas, and S. Bandyopadhyay, *A critical review on diagnosis of diabetic retinopathy using machine learning and deep learning.* Multimedia Tools and Applications, 2022. **81**(18): p. 25613-25655.
220. Gupta, S., S. Thakur, and A. Gupta, *Optimized hybrid machine learning approach for smartphone based diabetic retinopathy detection.* Multimedia Tools and Applications, 2022. **81**(10): p. 14475-14501.
221. Tymchenko, B., P. Marchenko, and D. Spodarets, *Deep learning approach to diabetic retinopathy detection.* arXiv preprint arXiv:2003.02261, 2020.
222. Alalhareth, M. and S.-C. Hong, *An Improved Mutual Information Feature Selection Technique for Intrusion Detection Systems in the Internet of Medical Things.* Sensors, 2023. **23**(10): p. 4971.
223. Yang, G., et al., *Interoperability and data storage in internet of multimedia things: investigating current trends, research challenges and future directions.* IEEE Access, 2020. **8**: p. 124382-124401.
224. Rana, B., Y. Singh, and P.K. Singh, *A systematic survey on internet of things: Energy efficiency and interoperability perspective.* Transactions on Emerging Telecommunications Technologies, 2021. **32**(8): p. e4166.
225. Nižetić, S., et al., *Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future.* Journal of cleaner production, 2020. **274**: p. 122877.
226. Ortiz, B.L., et al., *Data Preprocessing Techniques for AI and Machine Learning Readiness: Scoping Review of Wearable Sensor Data in Cancer Care.* JMIR mHealth and uHealth, 2024. **12**(1): p. e59587.
227. Al-amri, R., et al., *A review of machine learning and deep learning techniques for anomaly detection in IoT data.* Applied Sciences, 2021. **11**(12): p. 5320.
228. Aminizadeh, S., et al., *The applications of machine learning techniques in medical data processing based on distributed computing and the Internet of Things.* Computer methods and programs in biomedicine, 2023: p. 107745.
229. Mian, Z., K.L. Hermayer, and A. Jenkins, *Continuous glucose monitoring: review of an innovation in diabetes management.* The American journal of the medical sciences, 2019. **358**(5): p. 332-339.
230. Henkel, J., et al. *Ultra-low power and dependability for IoT devices (Invited paper for IoT technologies)*. in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. 2017. IEEE.
231. Singh, R., et al., *A review of bioresorbable implantable medical devices: Materials, fabrication, and implementation.* Advanced Healthcare Materials, 2020. **9**(18): p. 2000790.

232. Gerodimos, A., et al., *IoT: Communication protocols and security threats.* Internet of Things and Cyber-Physical Systems, 2023.
233. Anderson, K., O. Burford, and L. Emmerton, *Mobile health apps to facilitate self-care: a qualitative study of user experiences.* PloS one, 2016. **11**(5): p. e0156164.
234. Newaz, A.I., et al. *Healthguard: A machine learning-based security framework for smart healthcare systems*. in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. 2019. IEEE.
235. Ronquillo, J.G. and D.M. Zuckerman, *Software-related recalls of health information technology and other medical devices: Implications for FDA regulation of digital health.* The Milbank Quarterly, 2017. **95**(3): p. 535-553.
236. Trivedi, O.H., *Microservices-Based Approach for Healthcare Cybersecurity*. 2018, Florida Atlantic University.
237. Jiang, X., M. Lora, and S. Chattopadhyay, *An experimental analysis of security vulnerabilities in industrial IoT devices.* ACM Transactions on Internet Technology (TOIT), 2020. **20**(2): p. 1-24.
238. Ahmadi, H., et al., *The application of internet of things in healthcare: a systematic literature review and classification.* Universal Access in the Information Society, 2019: p. 1-33.
239. Ahmadi, H., et al., *The application of internet of things in healthcare: a systematic literature review and classification.* Universal Access in the Information Society, 2018: p. 1-33.
240. Somasundaram, R. and M. Thirugnanam, *Review of security challenges in healthcare internet of things.* WIRELESS NETWORKS, 2020.
241. Halperin, D., et al. *Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses*. in *2008 IEEE Symposium on Security and Privacy (sp 2008)*. 2008. IEEE.
242. Li, C., A. Raghunathan, and N.K. Jha. *Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system*. in *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*. 2011. IEEE.
243. Aldossari, M.Q. and A. Sidorova, *Consumer acceptance of Internet of Things (IoT): Smart home context.* Journal of Computer Information Systems, 2020. **60**(6): p. 507-517.
244. Čolaković, A. and M. Hadžialić, *Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues.* Computer networks, 2018. **144**: p. 17-39.
245. Ali, Z.H., H.A. Ali, and M.M. Badawy, *Internet of Things (IoT): definitions, challenges and recent research directions.* International Journal of Computer Applications, 2015. **128**(1): p. 37-47.
246. Abdul-Qawy, A.S., et al., *The internet of things (iot): An overview.* International Journal of Engineering Research and Applications, 2015. **5**(12): p. 71-82.
247. Atzori, L., A. Iera, and G. Morabito, *The internet of things: A survey.* Computer networks, 2010. **54**(15): p. 2787-2805.
248. Bello, O. and S. Zeadally, *Intelligent device-to-device communication in the internet of things.* IEEE Systems Journal, 2014. **10**(3): p. 1172-1182.
249. Mouha, R.A.R.A., *Internet of things (IoT).* Journal of Data Analysis and Information Processing, 2021. **9**(02): p. 77.
250. Özgür, L., et al. *An IoT based smart thermostat*. in *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*. 2018. IEEE.

251. Koritsoglou, K., et al. *Smart refrigeration equipment based on iot technology for reducing power consumption*. in *2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. 2022. IEEE.
252. Sharma, V. and D.V. Gadre. *Condition monitoring of industrial & commercial refrigeration systems using iot*. in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. 2021. IEEE.
253. Mourtzis, D., J. Angelopoulos, and N. Panopoulos, *Design and development of an IoT enabled platform for remote monitoring and predictive maintenance of industrial equipment.* Procedia Manufacturing, 2021. **54**: p. 166-171.
254. Khaled, A.E., *Internet of medical things (iomt): Overview, taxonomies, and classifications.* Journal of Computer and Communications, 2022. **10**(08): p. 64-89.
255. Vaidya, B. and H.T. Mouftah, *IoT applications and services for connected and autonomous electric vehicles.* Arabian Journal for Science and Engineering, 2020. **45**(4): p. 2559-2569.
256. Sachdev, S., et al., *Voice-controlled autonomous vehicle using IoT.* Procedia Computer Science, 2019. **160**: p. 712-717.
257. Bathla, G., et al., *Autonomous vehicles and intelligent automation: Applications, challenges, and opportunities.* Mobile Information Systems, 2022. **2022**(1): p. 7632892.
258. Biswas, A. and H.-C. Wang, *Autonomous vehicles enabled by the integration of IoT, edge intelligence, 5G, and blockchain.* Sensors, 2023. **23**(4): p. 1963.
259. Minovski, D., C. Åhlund, and K. Mitra, *Modeling quality of IoT experience in autonomous vehicles.* IEEE Internet of Things Journal, 2020. **7**(5): p. 3833-3849.
260. Philip, B.V., et al., *Distributed real-time IoT for autonomous vehicles.* IEEE Transactions on Industrial Informatics, 2018. **15**(2): p. 1131-1140.
261. Krasniqi, X. and E. Hajrizi, *Use of IoT technology to drive the automotive industry from connected to full autonomous vehicles.* IFAC-PapersOnLine, 2016. **49**(29): p. 269-274.
262. Passos, J., et al., *Wearables and Internet of Things (IoT) technologies for fitness assessment: a systematic review.* Sensors, 2021. **21**(16): p. 5418.
263. Yang, W., et al., *Narrowband wireless access for low-power massive internet of things: A bandwidth perspective.* IEEE wireless communications, 2017. **24**(3): p. 138-145.
264. Ikpehai, A., et al., *Low-power wide area network technologies for Internet-of-Things: A comparative review.* IEEE Internet of Things Journal, 2018. **6**(2): p. 2225-2240.
265. Nikoukar, A., et al., *Low-power wireless for the internet of things: Standards and applications.* IEEE Access, 2018. **6**: p. 67893-67926.
266. de Carvalho Silva, J., et al. *LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities*. in *2017 2nd International multidisciplinary conference on computer and energy science (SpliTech)*. 2017. IEEE.
267. Li, Y., et al., *Smart choice for the smart grid: Narrowband Internet of Things (NB-IoT).* IEEE Internet of Things Journal, 2017. **5**(3): p. 1505-1515.
268. Qadir, Q.M., et al., *Low power wide area networks: A survey of enabling technologies, applications and interoperability needs.* IEEE Access, 2018. **6**: p. 77454-77473.
269. Botta, A., et al., *Integration of cloud computing and internet of things: a survey.* Future generation computer systems, 2016. **56**: p. 684-700.

270. Firouzi, F., B. Farahani, and A. Marinšek, *The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT).* Information Systems, 2022. **107**: p. 101840.
271. Petrenko, A.S., et al. *The IIoT/IoT device control model based on narrow-band IoT (NB-IoT)*. in *2018 IEEE conference of Russian young researchers in electrical and electronic engineering (EIConRus)*. 2018. IEEE.
272. Lalhriatpuii, Ruchi, and V. Wasson. *Comprehensive Exploration of IoT Communication Protocol: CoAP, MQTT, HTTP, LoRaWAN and AMQP*. in *International Conference on Machine Learning Algorithms*. 2024. Springer.
273. Chakraborty, C. and J.J. Rodrigues, *A comprehensive review on device-to-device communication paradigm: trends, challenges and applications.* Wireless Personal Communications, 2020. **114**(1): p. 185-207.
274. Burhan, M., et al., *IoT elements, layered architectures and security issues: A comprehensive survey.* Sensors, 2018. **18**(9): p. 2796.
275. Rahmani, H., et al., *Next-generation IoT devices: Sustainable eco-friendly manufacturing, energy harvesting, and wireless connectivity.* IEEE Journal of Microwaves, 2023. **3**(1): p. 237-255.
276. Choo, K.-K.R., et al., *A Multidisciplinary Approach to Internet of Things (IoT) Cybersecurity and Risk Management*. 2020, Elsevier.
277. Conti, M., et al., *Internet of Things security and forensics: Challenges and opportunities.* 2018, Elsevier.
278. Mendhurwar, S. and R. Mishra, *Integration of social and IoT technologies: architectural framework for digital transformation and cyber security challenges.* Enterprise Information Systems, 2021. **15**(4): p. 565-584.
279. Aceto, G., V. Persico, and A. Pescapé, *Industry 4.0 and health: Internet of things, big data, and cloud computing for healthcare 4.0.* Journal of Industrial Information Integration, 2020. **18**: p. 100129.
280. Colton Hood, N.S., Cindy Manaoat Van, MHSA, Sarah E. Mossburg,, *Remote Patient Monitoring.* 2023. **The Patient Safety Network (PSNet) Collection**.
281. Pradeesh, E., et al., *Investigation on IoT based smart E-Inhaler integrated with mobile application.* Materials Today: Proceedings, 2022. **66**: p. 1082-1087.
282. Konstantinidis, E.I., et al. *Internet of things for an age-friendly healthcare*. in *MIE*. 2015.
283. Uhm, K.E., et al., *Effects of exercise intervention in breast cancer patients: is mobile health (mHealth) with pedometer more effective than conventional program using brochure?* Breast cancer research and treatment, 2017. **161**(3): p. 443-452.
284. Deshkar, S., R. Thanseeh, and V.G. Menon, *A review on IoT based m-Health systems for diabetes.* International Journal of Computer Science and Telecommunications, 2017. **8**(1): p. 13-18.
285. Ardito, C., et al. *Towards a Situation Awareness for eHealth in Ageing Society*. in *AIxAS@ AI* IA*. 2020.
286. Lofù, D., et al. *A situation awareness computational intelligent model for metabolic syndrome management*. in *2022 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*. 2022. IEEE.
287. Chen, T., et al., *Learning and management for Internet of Things: Accounting for adaptivity and scalability.* Proceedings of the IEEE, 2019. **107**(4): p. 778-796.
288. Pradyumna, G., et al., *Empowering healthcare with IoMT: Evolution, machine learning integration, security, and interoperability challenges.* IEEE Access, 2024.
289. Wu, W.H., et al., *MEDIC: Medical embedded device for individualized care.* Artificial intelligence in medicine, 2008. **42**(2): p. 137-152.

290. Perrier, A., et al., *Smart Diabetic Socks: Embedded device for diabetic foot prevention.* IRBM, 2014. **35**(2): p. 72-76.
291. Kulkarni, V.V., F. Seraj, and J.R. Rathikumar. *An embedded wearable device for monitoring diabetic foot ulcer parameters*. in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. 2020.
292. Wang, X., et al., *An empirical study of wearable technology acceptance in healthcare.* Industrial Management & Data Systems, 2015.
293. Gillinov, S., et al., *Variable accuracy of wearable heart rate monitors during aerobic exercise.* Med Sci Sports Exerc, 2017. **49**(8): p. 1697-1703.
294. Pinto, M.B. and A. Yagnik, *Fit for life: A content analysis of fitness tracker brands use of Facebook in social media marketing.* Journal of Brand Management, 2017. **24**(1): p. 49-67.
295. Topouchian, J., et al., *clinical accuracy of the Omron M3 comfort® and the Omron evolv® for self-blood pressure measurements in pregnancy and pre-eclampsia–validation according to the universal standard protocol.* Vascular health and risk management, 2018. **14**: p. 189.
296. Riposan-Taylor, A. and I.J. Taylor, *Personal connected devices for healthcare*, in *The Internet of Things for Smart Urban Ecosystems*. 2019, Springer. p. 333-361.
297. Aggidis, A.G., J.D. Newman, and G.A. Aggidis, *Investigating pipeline and state of the art blood glucose biosensors to formulate next steps.* Biosensors and Bioelectronics, 2015. **74**: p. 243-262.
298. Tran, J., R. Tran, and J.R. White, *Smartphone-based glucose monitors and applications in the management of diabetes: an overview of 10 salient "apps" and a novel smartphone-connected blood glucose monitor.* Clinical Diabetes, 2012. **30**(4): p. 173-178.
299. Santoso, F. and S.J. Redmond, *Indoor location-aware medical systems for smart homecare and telehealth monitoring: state-of-the-art.* Physiological measurement, 2015. **36**(10): p. R53.
300. McFarland, S., A. Coufopolous, and D. Lycett, *The effect of telehealth versus usual care for home-care patients with long-term conditions: a systematic review, meta-analysis and qualitative synthesis.* Journal of telemedicine and telecare, 2019: p. 1357633X19862956.
301. Saeed, N., M. Manzoor, and P. Khosravi, *An exploration of usability issues in telecare monitoring systems and possible solutions: a systematic literature review.* Disability and Rehabilitation: Assistive Technology, 2020. **15**(3): p. 271-281.
302. Pulver, A., R. Wei, and C. Mann, *Locating AED enabled medical drones to enhance cardiac arrest response times.* Prehospital Emergency Care, 2016. **20**(3): p. 378-389.
303. McCarthy, C.J. and R.N. Uppot, *Advances in virtual and augmented reality—exploring the role in health-care education.* Journal of Radiology Nursing, 2019. **38**(2): p. 104-105.
304. Munzer, B.W., et al., *Augmented reality in emergency medicine: a scoping review.* Journal of medical Internet research, 2019. **21**(4): p. e12368.
305. Templer, S., *Closed-Loop Insulin Delivery Systems: Past, Present, and Future Directions.* Frontiers in Endocrinology, 2022. **108**(7): p. 1614-1623.
306. Alsaleh, F.M., et al., *Insulin pumps: from inception to the present and toward the future.* Journal of clinical pharmacy and therapeutics, 2010. **35**(2): p. 127-138.
307. Schierloh, U., et al., *frontiers Frontiers in Endocrinology ORIGINAL RESEARCH published: 31 May 2022.* Technologies for diabetes, 2023: p. 48.

308. Dugan, J.A., et al., *Managing Diabetes in the Digital Age.* Physician Assistant Clinics, 2020. **5**(2): p. 177-190.
309. Ahmed, S.H., et al., *Do-It-Yourself (DIY) Artificial Pancreas Systems for Type 1 Diabetes: Perspectives of Two Adult Users, Parent of a User and Healthcare Professionals.* Advances in Therapy, 2020. **37**(9): p. 3929-3941.
310. Hng, T.M. and D. Burren, *Appearance of Do-It-Yourself closed-loop systems to manage type 1 diabetes.* Internal medicine journal, 2018. **48**(11): p. 1400-1404.
311. Joyce, M. and M. Lee, *A Patient-Designed Do-It-Yourself Mobile Technology System for Diabetes Promise and Challenges for a New Era in Medicine.* 2016.
312. Lee, J.M., E. Hirschfeld, and J. Wedding, *A patient-designed do-it-yourself mobile technology system for diabetes: promise and challenges for a new era in medicine.* Jama, 2016. **315**(14): p. 1447-1448.
313. Litchman, M.L., et al., *Twitter analysis of# OpenAPS DIY artificial pancreas technology use suggests improved A1C and quality of life.* Journal of diabetes science and technology, 2019. **13**(2): p. 164-170.
314. Litchman, M.L., et al., *Patient-Driven Diabetes Technologies: Sentiment and Personas of the# WeAreNotWaiting and# OpenAPS Movements.* Journal of Diabetes Science and Technology, 2020. **14**(6): p. 990-999.
315. Lewis, D., S. Leibrand, and O. Community, *Real-world use of open source artificial pancreas systems.* Journal of diabetes science and technology, 2016. **10**(6): p. 1411.
316. Russell, S.J., et al., *Outpatient glycemic control with a bionic pancreas in type 1 diabetes.* New England Journal of Medicine, 2014. **371**(4): p. 313-325.
317. JDRF, *FDA Grants Breakthrough Device Status: iLet Bionic Pancreas.* 2019.
318. Ekhlaspour, L., et al., *Feasibility studies of an insulin-only bionic pancreas in a home-use setting.* Journal of diabetes science and technology, 2019. **13**(6): p. 1001-1007.
319. Stone, M.P., et al., *Retrospective analysis of 3-month real-world glucose data after the MiniMed 670G system commercial launch.* Diabetes technology & therapeutics, 2018. **20**(10): p. 689-692.
320. Benhamou, P.Y., et al., *Customization of home closed-loop insulin delivery in adult patients with type 1 diabetes, assisted with structured remote monitoring: the pilot WP7 Diabeloop study.* Acta diabetologica, 2018. **55**: p. 549-556.
321. Silva, J.D., et al., *Real-world performance of the MiniMed™ 780G system: first report of outcomes from 4120 users.* Diabetes Technology & Therapeutics, 2022. **24**(2): p. 113-119.
322. O'Neill, S., *Update on technologies, medicines and treatments including Libre 3, Minimed 780G and Glucomen Day continuous glucose monitoring*. 2021.
323. FDA. *FDA warns patients and health care providers about potential cybersecurity concerns with certain Medtronic insulin pumps*. 2019 June 2019; Available from: https://www.fda.gov/news-events/press-announcements/fda-warns-patients-and-health-care-providers-about-potential-cybersecurity-concerns-certain.
324. Market, R., *Diabetes Devices Market Size, Share & Trends Analysis Report by Type (Blood Glucose Monitoring Devices, Insulin Delivery Devices), Distribution Channel (Hospital Pharmacies, Retail Pharmacies), End-use, Region, and Segment Forecasts, 2024-2030.* 2024.
325. Ekhlaspour, L., et al., *1063-P: First Test of the iLet, a Purpose-Built Bionic Pancreas Platform, in Children and Adolescents with Type 1 Diabetes*. 2019, American Diabetes Association.
326. Jafri, R.Z., et al., *77-OR: First human study testing the iLet, a purpose-built bionic pancreas platform*. 2019, Am Diabetes Assoc.

327. Hoskins, M., *Meet the Bigfoot Family and Their Homemade Closed Loop System.* 2016.
328. Bigfoot, *Bigfoot and Ypsomed announce agreement.* 2017.
329. Bigfoot, *FDA clearance Bigfoot Products.* 2020.
330. Quemerais, M.A., et al., *Preliminary evaluation of a new semi-closed-loop insulin therapy system over the prandial period in adult patients with type 1 diabetes: the WP6. 0 Diabeloop study.* Journal of diabetes science and technology, 2014. **8**(6): p. 1177-1184.
331. Whooley, S., *Roche, Diabeloop partner on insulin pump management.* 2020.
332. Dexcom, S.o., 2014.
333. Cobry, E.C., et al., *Review of the Omnipod® 5 Automated Glucose Control System Powered by Horizon™ for the treatment of Type 1 diabetes.* Therapeutic Delivery, 2020. **11**(8): p. 507-519.
334. DexCom, 2020.
335. DreaMed, 2015.
336. Nimri, R., et al., *Adjustment of Insulin Pump Settings in Type 1 Diabetes Management: Advisor Pro Device Compared to Physicians' Recommendations.* Journal of Diabetes Science and Technology, 2020: p. 1932296820965561.
337. Kunzmann, K., *Advisor Pro T1D Therapy Software Granted FDA Approval.* 2018.
338. Ginsberg, B.H., *Patch pumps for insulin.* Journal of diabetes science and technology, 2019. **13**(1): p. 27-33.
339. JDRF, *JDRF Partners with EOFlow.* 2017.
340. EoFlow, 2019.
341. Zisser, H.C., *The OmniPod Insulin Management System: the latest innovation in insulin pump therapy.* Diabetes Therapy, 2010. **1**(1): p. 10-24.
342. report, I.f., 2020.
343. Insulet, *Integrate the Dexcom G6 and Future G7 CGM into Insulet's Omnipod Horizon.* 2020.
344. Pisano, M., *Overview of insulin and non-insulin delivery devices in the treatment of diabetes.* Pharmacy and Therapeutics, 2014. **39**(12): p. 866.
345. Salsali, A. and M. Nathan, *A review of types 1 and 2 diabetes mellitus and their treatment with insulin.* American journal of therapeutics, 2006. **13**(4): p. 349-361.
346. Selam, J.-L., *Evolution of diabetes insulin delivery devices*. 2010, SAGE Publications Sage CA: Los Angeles, CA.
347. Lilly, I.R., *Lilly company announced $400 million investment.* 2019.
348. Hermanns, N., et al., *Trends in diabetes self-management education: where are we coming from and where are we going? A narrative review.* Diabetic Medicine, 2020. **37**(3): p. 436-447.
349. Lilly, *Lilly Diabetes and Ypsomed announced their collaboration to develop AID systems* 2020.
350. Acosta, G.J., et al., *USA Vs Europe: Who Is Leading the Diabetes Tech Race?* Current diabetes reports, 2019. **19**(11): p. 128.
351. Berget, C., et al., *A clinical review of the t: slim X2 insulin pump.* Expert Opinion on Drug Delivery, 2020: p. 1-13.
352. Blackstone, I.f., *A $337 million investment from Blackstone Life Sciences to manufacture diabetes technologies.* 2020.
353. Report, F., *MiniMed 670G and 770G System FDA apporval to use in young pediatric patients.* 2020.
354. Senseonics, *Senseonics Announces CE Mark Approval of the Eversense® CGM System.* 2016.

355. FDA, *Eversense Continuous Glucose Monitoring System.* 2018.
356. Senseonics, *Senseonics Announces Strategic Collaboration with Ascensia Diabetes Care.* 2020.
357. Control-IQ, *The FDA granted marketing authorization of the Control-IQ Technology controller to Tandem Diabetes.* 2019.
358. JDRF, *Tandem Diabetes Care Announces Partnership with JDRF to Develop Novel Dual-Chamber Infusion Pump.* 2013.
359. Tandem, *Tandem Diabetes Care and TypeZero Technologies Announce License Agreement* 2016.
360. Tandem, A.a., *Abbott and Tandem Diabetes Care Advance Development of Integrated Technologies for Future Automated Insulin Delivery Systems.* 2020.
361. Rosenmayr-Templeton, L., *An industry update: what is new in the field of therapeutic delivery?* Therapeutic Delivery, 2017. **8**(9): p. 729-735.
362. Stone, J.Y., N. Haviland, and T.S. Bailey, *Review of a commercially available hybrid closed-loop insulin-delivery system in the treatment of type 1 diabetes.* Therapeutic delivery, 2018. **9**(2): p. 77-87.
363. TypeZero, *Dexcom buys TypeZero.* 2018.
364. Sperling, K., et al., *EMA Grants Marketing Authorisation of mylife™ YpsoPump® for Use with NovoRapid® PumpCart®, Marking the Start of a Collaboration Between Novo Nordisk and Ypsomed.* 2016.
365. Ypsomed, *Ypsomed grows strongly in pharma customer business* 2020.
366. JDRF, *Ypsomed partners with JDRF*. 2018.
367. Hanaire, H., et al., *Efficacy of the Diabeloop closed-loop system to improve glycaemic control in patients with type 1 diabetes exposed to gastronomic dinners or to sustained physical exercise.* Diabetes, Obesity and Metabolism, 2020. **22**(3): p. 324-334.
368. Senseonics, *Senseonics Announces CE Mark Approval for Eversense® XL CGM System.* 2017.
369. Chen, C., et al., *Current and emerging technology for continuous glucose monitoring.* Sensors, 2017. **17**(1): p. 182.
370. Rodbard, D., *Continuous glucose monitoring: a review of successes, challenges, and opportunities.* Diabetes technology & therapeutics, 2016. **18**(S2): p. S2-3-S2-13.
371. Gia, T.N., et al., *IoT-based continuous glucose monitoring system: A feasibility study.* Procedia Computer Science, 2017. **109**: p. 327-334.
372. Bode, B., et al., *Alarms based on real-time sensor glucose values alert patients to hypo-and hyperglycemia: the guardian continuous monitoring system.* Diabetes technology & therapeutics, 2004. **6**(2): p. 105-113.
373. Group, D.R.i.C.N.S., *The accuracy of the Guardian® RT continuous glucose monitor in children with type 1 diabetes.* Diabetes technology & therapeutics, 2008. **10**(4): p. 266-272.
374. Kovatchev, B., et al., *Comparison of the numerical and clinical accuracy of four continuous glucose monitors.* Diabetes care, 2008. **31**(6): p. 1160-1164.
375. Blevins, T.C., *Professional continuous glucose monitoring in clinical practice 2010.* Journal of diabetes science and technology, 2010. **4**(2): p. 440-456.
376. Medtronic, *Medtronic Receives FDA Approval for Next Generation Professional Continuous Glucose Monitoring.* 2011.
377. Kovatchev, B.P., et al., *In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes*. 2009, SAGE Publications Sage CA: Los Angeles, CA.
378. Garcia, A., et al., *Dexcom G4AP: an advanced continuous glucose monitor for the artificial pancreas*. 2013, SAGE Publications Sage CA: Los Angeles, CA.

379. FDA, *FDA authorizes first fully interoperable Dexcom G6 continuous glucose monitoring system.* 2018.
380. Parkin, C.G., C. Graham, and J. Smolskis, *Continuous glucose monitoring use in type 1 diabetes: longitudinal analysis demonstrates meaningful improvements in HbA1c and reductions in health care utilization.* Journal of diabetes science and technology, 2017. **11**(3): p. 522-528.
381. Toschi, E. and H. Wolpert, *Utility of continuous glucose monitoring in type 1 and type 2 diabetes.* Endocrinology and Metabolism Clinics, 2016. **45**(4): p. 895-904.
382. Schmidt, S. and K. Nørgaard, *Bolus calculators.* Journal of diabetes science and technology, 2014. **8**(5): p. 1035-1041.
383. Walsh, J., R. Roberts, and T. Bailey, *Guidelines for insulin dosing in continuous subcutaneous insulin infusion using new formulas from a retrospective study of individuals with optimal glucose levels.* Journal of diabetes science and technology, 2010. **4**(5): p. 1174-1181.
384. Brzan, P.P., et al., *Mobile applications for control and self management of diabetes: a systematic review.* Journal of medical systems, 2016. **40**(9): p. 1-10.
385. Bally, L., H. Thabit, and R. Hovorka, *Glucose-responsive insulin delivery for type 1 diabetes: The artificial pancreas story.* International journal of pharmaceutics, 2018. **544**(2): p. 309-318.
386. Duckworth, W.C., C.D. Saudek, and R.R. Henry, *Why intraperitoneal delivery of insulin with implantable pumps in NIDDM?* Diabetes, 1992. **41**(6): p. 657-661.
387. Scaramuzza, A.E., et al., *Timing of bolus in children with type 1 diabetes using continuous subcutaneous insulin infusion (TiBoDi Study).* Diabetes technology & therapeutics, 2010. **12**(2): p. 149-152.
388. Raccah, D., et al., *Incremental value of continuous glucose monitoring when starting pump therapy in patients with poorly controlled type 1 diabetes: the RealTrend study.* Diabetes care, 2009. **32**(12): p. 2245-2250.
389. Medtronic, *Innovating for Life Since 1949.*
390. FDA, *Summary Of Safety And Effectiveness Data (SSED).* 2013.
391. FDA, *Medtronic Minimed 630G Insulin Pump.* 2016.
392. FDA, *Omnipod DASH™ Insulin Management System FDA clearance.* 2018.
393. Ly, T.T., et al., *Novel Bluetooth-enabled tubeless insulin pump: innovating pump therapy for patients in the digital age.* Journal of diabetes science and technology, 2019. **13**(1): p. 20-26.
394. Ramchandani, N. and R.A. Heptulla, *New technologies for diabetes: a review of the present and the future.* International journal of pediatric endocrinology, 2012. **2012**(1): p. 1-10.
395. FDA, *t:slim X2 Insulin Pump with Basal-IQ Technology.* 2018.
396. Accu-Chek, *Roche Diagnostics Accu-Chek Combo Insulin Pump System Now Available in United States.* 2012.
397. Ulbrich, S., et al., *Concept and Implementation of a Novel Patch Pump for Insulin Delivery.* Journal of diabetes science and technology, 2020. **14**(2): p. 324-327.
398. Kesavadev, J., et al., *Evolution of insulin delivery devices: from syringes, pens, and pumps to DIY artificial pancreas.* Diabetes Therapy, 2020. **11**(6): p. 1251-1269.
399. Dolan, B., *Cellnovo launches cellular-enabled insulin pump, usability trial.* 2012.
400. Hoskins, M., *Cellnovo Progressing on AP Dreams, But Where's First-Gen Pump for U.S.?* 2019.
401. Herzog, A.L., et al., *Survey about do-it-yourself closed loop systems in the treatment of diabetes in Germany.* Plos one, 2020. **15**(12): p. e0243465.

402. Markets, R.a., *Global Medical Devices Market Report 2023: Sector is Expected to Reach $695.2 Billion by 2031 at a CAGR of 6.2%.* 2023.
403. Groth, T., et al., *Wearable and implantable artificial kidney devices for end-stage kidney disease treatment: Current status and review.* Artificial Organs, 2023. **47**(4): p. 649-666.
404. Adhami, M., et al., *Drug loaded implantable devices to treat cardiovascular disease.* Expert Opinion on Drug Delivery, 2023. **20**(4): p. 507-522.
405. Coronel-Meneses, D., et al., *Strategies for surface coatings of implantable cardiac medical devices.* Frontiers in Bioengineering and Biotechnology, 2023. **11**: p. 1173260.
406. Schulder, M., et al., *Advances in technical aspects of deep brain stimulation surgery.* Stereotactic and functional neurosurgery, 2023. **101**(2): p. 112-134.
407. Wikina, S.B., *What caused the breach? An examination of use of information technology and health data breaches.* Perspectives in health information management, 2014. **11**(Fall).
408. Gabriel, M.H., et al., *Data breach locations, types, and associated characteristics among US hospitals.* Am J Manag Care, 2018. **24**(2): p. 78-84.
409. Aamodt, A. and M. Nygård, *Different roles and mutual dependencies of data, information, and knowledge—An AI perspective on their integration.* Data & Knowledge Engineering, 1995. **16**(3): p. 191-222.
410. de Assuncao, M.D., A. da Silva Veith, and R. Buyya, *Distributed data stream processing and edge computing: A survey on resource elasticity and future directions.* Journal of Network and Computer Applications, 2018. **103**: p. 1-17.
411. Zeydan, E. and J. Mangues-Bafalluy, *Recent advances in data engineering for networking.* IEEE Access, 2022. **10**: p. 34449-34496.
412. Gobinath, A., et al., *Internet of Medical Things (IoMT).* Smart Hospitals: 5G, 6G and Moving Beyond Connectivity, 2024: p. 91-105.
413. Ahmed, I., E. Balestrieri, and F. Lamonaca, *IoMT-based biomedical measurement systems for healthcare monitoring: A review.* Acta IMEKO, 2021. **10**(2): p. 174-184.
414. El-Saleh, A.A., et al., *The Internet of Medical Things (IoMT): opportunities and challenges.* Wireless Networks, 2024: p. 1-18.
415. Mamdiwar, S.D., et al., *Recent advances on IoT-assisted wearable sensor systems for healthcare monitoring.* Biosensors, 2021. **11**(10): p. 372.
416. Lee, H.-Y., et al., *Substantiation and Effectiveness of Remote Monitoring System Based on IoMT Using Portable ECG Device.* Bioengineering, 2024. **11**(8): p. 836.
417. Rubi, J. and A.J.A. Dhivya, *Wearable health monitoring systems using IoMT.* The Internet of Medical Things (IoMT) Healthcare Transformation, 2022: p. 225-246.
418. Suresh, S., et al., *Application of Body Sensor Networks in Health Care Using the Internet of Medical Things (IoMT)*, in *The Next Generation Innovation in IoT and Cloud Computing with Applications*. CRC Press. p. 131-156.
419. Adewole, K.S., et al., *Cloud-based IoMT framework for cardiovascular disease prediction and diagnosis in personalized E-health care*, in *Intelligent IoT systems in personalized health care*. 2021, Elsevier. p. 105-145.
420. Garg, N., et al., *Security in IoMT-driven smart healthcare: A comprehensive review and open challenges.* Security and Privacy, 2022. **5**(5): p. e235.
421. Hireche, R., H. Mansouri, and A.-S.K. Pathan, *Security and privacy management in Internet of Medical Things (IoMT): A synthesis.* Journal of cybersecurity and privacy, 2022. **2**(3): p. 640-661.
422. El-Moneim Kabel, S.A., et al., *Securing Internet-of-Medical-Things networks using cancellable ECG recognition.* Scientific Reports, 2024. **14**(1): p. 10871.

423. Messinis, S., et al., *Enhancing Internet of Medical Things security with artificial intelligence: A comprehensive review.* Computers in Biology and Medicine, 2024: p. 108036.
424. Yaacoub, J.-P.A., et al., *A survey on ethical hacking: issues and challenges.* arXiv preprint arXiv:2103.15072, 2021.
425. Rufai, A.U., et al., *Cyberattacks against Artificial Intelligence-Enabled Internet of Medical Things*, in *Handbook of Security and Privacy of AI-Enabled Healthcare Systems and Internet of Medical Things*. 2024, CRC Press. p. 191-216.
426. Mosenia, A. and N.K. Jha, *A comprehensive study of security of internet-of-things.* IEEE Transactions on emerging topics in computing, 2016. **5**(4): p. 586-602.
427. Bettayeb, M., et al. *IoT Testbed Security: Smart Socket and Smart Thermostat*. in *2019 IEEE Conference on Application, Information and Network Security (AINS)*. 2019. IEEE.
428. Lee, I., *Analyzing web descriptions of cybersecurity breaches in the healthcare provider sector: A content analytics research method.* Computers & Security, 2023. **129**: p. 103185.
429. Papaioannou, M., et al., *A survey on security threats and countermeasures in internet of medical things (IoMT).* Transactions on Emerging Telecommunications Technologies, 2022. **33**(6): p. e4049.
430. Almaiman, L. and N. Alqahtani. *Security and privacy on IoMT*. in *Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020) 12*. 2021. Springer.
431. Patel, A., et al., *Safeguarding the IoT: Taxonomy, security solutions, and future research opportunities.* Security and Privacy, 2024. **7**(2): p. e354.
432. Fereidouni, H., O. Fadeitcheva, and M. Zalai, *IoT and Man-in-the-Middle Attacks.* arXiv preprint arXiv:2308.02479, 2023.
433. Chua, J.A. and C. Pmp, *Cybersecurity in the healthcare industry.* Physician Leadership Journal, 2021. **8**(1).
434. Priestman, W., et al., *Phishing in healthcare organisations: Threats, mitigation and approaches.* BMJ health & care informatics, 2019. **26**(1).
435. Rani, L., *Security Issues and Defense Mechanism Using IoMT*, in *Artificial Intelligence Technology in Healthcare*. CRC Press. p. 211-242.
436. Harvey, P.J. *Introducing and facilitating internet of medical things (IoMT) research for undergraduate students and high school teachers*. in *2021 Fall ASEE Middle Atlantic Section Meeting*. 2021.
437. Rao, A., et al., *FIRE: A Finely Integrated Risk Evaluation Methodology for Life-Critical Embedded Systems.* Information, 2022. **13**(10): p. 487.
438. Salem, O., et al., *Man-in-the-Middle attack mitigation in internet of medical things.* IEEE Transactions on Industrial Informatics, 2021. **18**(3): p. 2053-2062.
439. Belkhouja, T., et al. *Light-weight solution to defend implantable medical devices against man-in-the-middle attack*. in *2018 IEEE Global Communications Conference (GLOBECOM)*. 2018. IEEE.
440. Kambourakis, G., C. Kolias, and A. Stavrou. *The mirai botnet and the iot zombie armies*. in *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. 2017. IEEE.
441. Kelly, C., et al. *Testing And Hardening IoT Devices Against the Mirai Botnet*. in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. 2020. IEEE.

442. Chen, J., et al. *Your iots are (not) mine: On the remote binding between iot devices and users*. in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2019. IEEE.
443. Marin, E., et al. *On the feasibility of cryptography for a wireless insulin pump system*. in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*. 2016.
444. Angrishi, K., *Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets.* arXiv preprint arXiv:1702.03681, 2017.
445. Antonakakis, M., et al. *Understanding the mirai botnet*. in *26th USENIX security symposium (USENIX Security 17)*. 2017.
446. Kolias, C., et al., *DDoS in the IoT: Mirai and other botnets.* Computer, 2017. **50**(7): p. 80-84.
447. Bertino, E. and N. Islam, *Botnets and internet of things security.* Computer, 2017. **50**(2): p. 76-79.
448. Anh, S.P. and Y. Nakamura, *A Baseline Investigation into the Evolution and Prevalence of Mirai and Hajime Utilizing a Network Telescope.* IEEE Access, 2024.
449. Wang, J., et al., *IoT-DeepSense: Behavioral Security Detection of IoT Devices Based on Firmware Virtualization and Deep Learning.* Security and Communication Networks, 2022. **2022**(1): p. 1443978.
450. Soliman, S.W., M.A. Sobh, and A.M. Bahaa-Eldin. *Taxonomy of malware analysis in the IoT*. in *2017 12th International Conference on Computer Engineering and Systems (ICCES)*. 2017. IEEE.
451. Rains, T., *Cybersecurity Threats, Malware Trends, and Strategies: Learn to mitigate exploits, malware, phishing, and other social engineering attacks*. 2020: Packt Publishing Ltd.
452. Abraham, S. and I. Chengalur-Smith, *An overview of social engineering malware: Trends, tactics, and implications.* Technology in Society, 2010. **32**(3): p. 183-196.
453. Anwar, A., et al. *A Recent Year On the Internet: Measuring and Understanding the Threats to Everyday Internet Devices*. in *Proceedings of the 38th Annual Computer Security Applications Conference*. 2022.
454. Wazid, M., et al., *IoMT malware detection approaches: analysis and research challenges.* IEEE Access, 2019. **7**: p. 182459-182476.
455. Alzahrani, F.A., M. Ahmad, and M.T.J. Ansari, *Towards design and development of security assessment framework for internet of medical things.* Applied Sciences, 2022. **12**(16): p. 8148.
456. Tyagi, V., et al., *Securing IoT Devices Against MITM and DoS Attacks: An Analysis.* Reshaping Intelligent Business and Industry: Convergence of AI and IoT at the Cutting Edge, 2024: p. 237-249.
457. Alaba, F.A., et al., *Internet of Things security: A survey.* Journal of Network and Computer Applications, 2017. **88**: p. 10-28.
458. Varga, P., et al. *Security threats and issues in automation IoT*. in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*. 2017. IEEE.
459. Aggarwal, E., et al. *CORGIDS: A correlation-based generic intrusion detection system*. in *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*. 2018.
460. Mihut, A., et al. *Reverse Engineering Approach to Validate the Implementation of Network Protocols in Operating Systems*. in *2024 23rd RoEduNet Conference: Networking in Education and Research (RoEduNet)*. 2024. IEEE.

461. Tauhid, A., et al., *A survey on security analysis of machine learning-oriented hardware and software intellectual property.* High-Confidence Computing, 2023. **3**(2): p. 100114.
462. Nurgaliyev, A. and H. Wang. *Analysis of reverse engineering*. in *2023 15th International Conference on Advanced Computational Intelligence (ICACI)*. 2023. IEEE.
463. Astillo, P.V., et al., *SMDAps: A Specification-based Misbehavior Detection System for Implantable Devices in Artificial Pancreas System.* Journal of Internet Technology, 2021. **22**(1): p. 1-11.
464. Zheng, G., et al., *Fingerprint access control for wireless insulin pump systems using cancelable delaunay triangulations.* IEEE Access, 2019. **7**: p. 75629-75641.
465. Reverberi, L. and D. Oswald. *Breaking (and fixing) a widely used continuous glucose monitoring system*. in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*. 2017.
466. Nozaripour, R. and K. Khorasani. *Replay Cyberattack Detection in an IoT-based Healthcare System using RFID Sensors*. in *2023 IEEE 13th International Conference on RFID Technology and Applications (RFID-TA)*. 2023. IEEE.
467. Singh, S., et al., *Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions.* Journal of Ambient Intelligence and Humanized Computing, 2017: p. 1-18.
468. Surendran, S., A. Nassef, and B.D. Beheshti. *A survey of cryptographic algorithms for IoT devices*. in *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. 2018. IEEE.
469. Radhakrishnan, I., S. Jadon, and P.B. Honnavalli, *Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices.* Sensors, 2024. **24**(12): p. 4008.
470. Priyadarshani, R., et al., *Jamming Intrusions in Extreme Bandwidth Communication: A Comprehensive Overview.* arXiv preprint arXiv:2403.19868, 2024.
471. Zafir, E.I., et al., *Enhancing security of internet of robotic things: a review of recent trends, practices, and recommendations with encryption and blockchain techniques.* Internet of Things, 2024: p. 101357.
472. de Curtò, J., et al., *Enhancing Communication Security in Drones Using QRNG in Frequency Hopping Spread Spectrum.* Future Internet, 2024. **16**(11): p. 412.
473. Ntshabele, K., et al., *A trusted security key management server in LoRaWAN: Modelling and analysis.* Journal of Sensor and Actuator Networks, 2022. **11**(3): p. 52.
474. Ali, U., et al., *Enhanced lightweight and secure certificateless authentication scheme (ELWSCAS) for internet of things environment.* Internet of Things, 2023. **24**: p. 100923.
475. Chen, Q. *The Application of AES Algorithm in Intelligent Wireless Network Information Security Protection System*. in *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)*. 2024. IEEE.
476. Abdelkader, S., et al., *Securing modern power systems: Implementing comprehensive strategies to enhance resilience and reliability against cyber-attacks.* Results in engineering, 2024: p. 102647.
477. Zhang, Z., M. Li, and L. Xie, *Data-driven replay attack detection for unknown cyber-physical systems.* Information Sciences, 2024. **670**: p. 120562.
478. Volkova, A., et al., *Security challenges in control network protocols: A survey.* IEEE Communications Surveys & Tutorials, 2018. **21**(1): p. 619-639.
479. Meyer, U. and S. Wetzel. *A man-in-the-middle attack on UMTS*. in *Proceedings of the 3rd ACM workshop on Wireless security*. 2004.

480. Hei, X., et al., *Patient infusion pattern based access control schemes for wireless insulin pump system.* IEEE Transactions on parallel and Distributed Systems, 2014. **26**(11): p. 3108-3121.
481. Fawaz, K., K.-H. Kim, and K.G. Shin. *Protecting privacy of {BLE} device users*. in *25th USENIX Security Symposium (USENIX Security 16)*. 2016.
482. Chauhan, J., et al. *Characterization of early smartwatch apps*. in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 2016. IEEE.
483. Hassija, V., et al., *Security issues in implantable medical devices: Fact or fiction?* Sustainable Cities and Society, 2021. **66**: p. 102552.
484. Ibrahim, M., A. Al-Wadi, and R. Elhafiz, *Security Analysis for Smart Healthcare Systems.* Sensors, 2024. **24**(11): p. 3375.
485. Kanneboina, A. and G. Sundaram, *Improving security performance of Internet of Medical Things using hybrid metaheuristic model.* Multimedia Tools and Applications, 2024: p. 1-26.
486. Kabir, M.R. and S. Ray. *DT-IoMT: A Digital Twin Reference Model for Secure Internet of Medical Things*. in *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. 2024. IEEE.
487. Hei, X., et al. *PIPAC: Patient infusion pattern based access control scheme for wireless insulin pump system*. in *2013 Proceedings IEEE INFOCOM*. 2013. IEEE.
488. Qu, Y., et al., *Privacy of things: Emerging challenges and opportunities in wireless internet of things.* IEEE Wireless Communications, 2018. **25**(6): p. 91-97.
489. Ahmad, U., et al. *Securing insulin pump system using deep learning and gesture recognition*. in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. 2018. IEEE.
490. Waheed, N., et al., *Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures.* ACM Computing Surveys (CSUR), 2020. **53**(6): p. 1-37.
491. Thompson, B., et al., *An Overview of Safety Issues on Use of Insulin Pumps and Continuous Glucose Monitoring Systems in the Hospital.* Current diabetes reports, 2018. **18**(10): p. 81.
492. Billingsley, L., *Cybersmart: Protect the Patient, Protect the Data.* Journal of Radiology Nursing, 2019.
493. Millard, W.B., *Where bits and bytes meet flesh and blood: Hospital responses to malware attacks.* Annals of Emergency Medicine, 2017. **70**(3): p. A17-A21.
494. Zhao, J., et al. *A visible light channel based access control scheme for wireless insulin pump systems*. in *2018 IEEE International Conference on Communications (ICC)*. 2018. IEEE.
495. US-FDA, *Cybersecurity Vulnerabilities of Hospira Symbiq Infusion System: FDA Safety Communication.* Silver Spring, Maryland (www. fda. gov/MedicalDevices/Safety/AlertsandNotices/ucm456815. htm), 2015.
496. Park, Y., et al. *This ain't your dose: Sensor spoofing attack on medical infusion pump*. in *10th USENIX workshop on offensive technologies (WOOT 16)*. 2016.
497. Da Silva, M.M., *Cable and wireless networks: Theory and practice*. 2018: CRC Press.
498. Islam, M. and S. Jin, *An overview research on wireless communication network.* Networks, 2019. **5**(1): p. 19-28.

499. A Rida, J.F. and B. Abood, *Survey of Improved Performance Radio Frequency Channels in Wireless Communication System.* International Journal of Civil Engineering and Technology, 2019. **10**(12): p. 70-83.
500. Alobaidy, H.A., et al., *Wireless transmissions, propagation and channel modelling for IoT technologies: Applications and challenges.* IEEE Access, 2022. **10**: p. 24095-24131.
501. Kocakulak, M. and I. Butun. *An overview of Wireless Sensor Networks towards internet of things*. in *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*. 2017. Ieee.
502. Gupta, B.B. and M. Quamara, *An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols.* Concurrency and Computation: Practice and Experience, 2020. **32**(21): p. e4946.
503. Gubbi, J., et al., *Internet of Things (IoT): A vision, architectural elements, and future directions.* Future generation computer systems, 2013. **29**(7): p. 1645-1660.
504. Bonaventure, O., *Computer Networking: Principles, Protocols and Practice.* 2011.
505. Cowley, J., *Communications and networking: an introduction*. 2012: Springer Science & Business Media.
506. Panek, C., *Networking fundamentals*. 2019: John Wiley & Sons.
507. Akyildiz, I.F. and J.M. Jornet, *The internet of nano-things.* IEEE Wireless Communications, 2010. **17**(6): p. 58-63.
508. Akyildiz, I.F. and J.M. Jornet, *Electromagnetic wireless nanosensor networks.* Nano Communication Networks, 2010. **1**(1): p. 3-19.
509. Du, K.-L. and M.N. Swamy, *Wireless communication systems: from RF subsystems to 4G enabling technologies*. 2010: Cambridge University Press.
510. Rappaport, T.S., *Wireless communications: principles and practice*. 2024: Cambridge University Press.
511. Garg, V., *Wireless communications & networking*. 2010: Elsevier.
512. Zohourian, A., et al., *IoT-PRIDS: Leveraging packet representations for intrusion detection in IoT networks.* Computers & Security, 2024. **146**: p. 104034.
513. Victor, P., et al., *IoT malware: An attribute-based taxonomy, detection mechanisms and challenges.* Peer-to-peer Networking and Applications, 2023. **16**(3): p. 1380-1431.
514. Pavica, C., et al., *A feedbackcontrolled optimization approach to minimize ransomware propagation in internet of things networks.* 2024.
515. Portalatin, M., et al. *Data Analytics for Cyber Risk Analysis Utilizing Cyber Incident Datasets*. in *2021 Systems and Information Engineering Design Symposium (SIEDS)*. 2021. IEEE.
516. Kumar, A. and I. Sharma. *Identifying patterns in common vulnerabilities and exposures databases with exploratory data analysis*. in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*. 2022. IEEE.
517. Camacho, J., R.A. Rodríguez-Gómez, and E. Saccenti, *Group-wise principal component analysis for exploratory data analysis.* Journal of Computational and Graphical Statistics, 2017. **26**(3): p. 501-512.
518. Pires, S. and C. Mascarenhas. *Cyber Threat Analysis Using Pearson and Spearman Correlation Via Exploratory Data Analysis*. in *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*. 2023. IEEE.
519. Atlam, H.F., et al., *Internet of things forensics: A review.* Internet of Things, 2020. **11**: p. 100220.

520. Mangino, A., M.S. Pour, and E. Bou-Harb, *Internet-scale insecurity of consumer internet of things: An empirical measurements perspective.* ACM Transactions on Management Information Systems (TMIS), 2020. **11**(4): p. 1-24.
521. Singh, P., et al., *Using log analytics and process mining to enable self-healing in the Internet of Things.* Environment Systems and Decisions, 2022. **42**(2): p. 234-250.
522. Rafiuzzaman, M., et al. *Failure prediction in the internet of things due to memory exhaustion*. in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. 2019.
523. Shafik, W., *Machine Learning Techniques for Multicriteria Decision-Making*, in *Multi-Criteria Decision-Making and Optimum Design with Machine Learning*. 2025, CRC Press. p. 165-194.
524. Zeydan, E., S.S. Arslan, and M. Liyanage, *Managing Distributed Machine Learning Lifecycle for Healthcare Data in the Cloud.* IEEE Access, 2024.
525. Gupta, D., M. Bhatia, and A. Kumar, *Resolving data overload and latency issues in multivariate time-series IoMT data for mental health monitoring.* IEEE Sensors Journal, 2021. **21**(22): p. 25421-25428.
526. Mansouri, S., S. Boulares, and S. Chabchoub, *Machine Learning for Early Diabetes Detection and Diagnosis.*
527. Vishal, P. and P.J. IR. *Predictive Analytics Model for Heart Disease Prediction with Personalized Recommendation Systems using Machine Learning Techniques*. in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2024. IEEE.
528. Salehi, W., et al., *An approach to binary classification of Alzheimer's disease using LSTM.* Bioengineering, 2023. **10**(8): p. 950.
529. Arun, S., E.R. Sykes, and S. Tanbeer, *RemoteHealthConnect: Innovating patient monitoring with wearable technology and custom visualization.* Digital health, 2024. **10**: p. 20552076241300748.
530. Otapo, A.T., et al., *Prediction and Detection of Terminal Diseases Using Internet of Medical Things: A Review.* arXiv preprint arXiv:2410.00034, 2024.
531. Motwani, A., P.K. Shukla, and M. Pawar, *Ubiquitous and smart healthcare monitoring frameworks based on machine learning: A comprehensive review.* Artificial Intelligence in Medicine, 2022. **134**: p. 102431.
532. Moghadam, M.P., et al., *Impact of artificial intelligence in nursing for geriatric clinical care for chronic diseases: A systematic literature review.* IEEE Access, 2024.
533. AboulEla, S., et al., *Navigating the Cyber Threat Landscape: An In-Depth Analysis of Attack Detection within IoT Ecosystems.* AI, 2024. **5**(2): p. 704-732.
534. Khan, I.A., et al., *XSRU-IoMT: Explainable simple recurrent units for threat detection in Internet of Medical Things networks.* Future generation computer systems, 2022. **127**: p. 181-193.
535. Shukla, P.K., et al., *Attaining an IoMT-based health monitoring and prediction: a hybrid hierarchical deep learning model and metaheuristic algorithm.* Neural Computing and Applications, 2023: p. 1-18.
536. Kokilavani, T., et al., *Machine Learning for Decision Support Systems in IoMT*, in *Technological Advancement in Internet of Medical Things and Blockchain for Personalized Healthcare*. CRC Press. p. 37-61.
537. Yousuff, M., et al., *Deep Learning Interpretation of Biomedical Data in IOMT and Cyber-Physical Systems*, in *Cyber-Physical Systems for Industrial Transformation*. 2023, CRC Press. p. 235-258.

538. Rahnenführer, J., et al., *Statistical analysis of high-dimensional biomedical data: a gentle introduction to analytical goals, common approaches and challenges.* BMC medicine, 2023. **21**(1): p. 182.
539. Zafeiropoulos, N., et al., *Graph Neural Networks for Parkinson's Disease Monitoring and Alerting.* Sensors, 2023. **23**(21): p. 8936.
540. Ali, H., et al., *Review of Time Domain Electronic Medical Record Taxonomies in the Application of Machine Learning. Electronics 2023, 12, 554*. 2023.
541. Chaithra, N., et al., *Internet of Medical Things with Artificial Intelligence for Improved Healthcare Systems*, in *Smart Healthcare Systems*. 2024, CRC Press. p. 18-32.
542. Petreska, A. and D. Slavkovska, *Artificial Intelligence and Machine Learning Algorithms in Modern Cardiology.* South East European Journal of Cardiology, 2024. **5**: p. 17-25.
543. Gupta, A. and M. Rathi. *Disentangling the Hidden Patterns of Heart Disease: A Factor Analysis and Machine Learning Approach for Early Detection and Prevention*. in *Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing*. 2023.
544. Tarek, Z., et al., *An optimized model based on deep learning and gated recurrent unit for COVID-19 death prediction.* Biomimetics, 2023. **8**(7): p. 552.
545. Sabry, F., et al., *Machine learning for healthcare wearable devices: the big picture.* Journal of Healthcare Engineering, 2022. **2022**(1): p. 4653923.
546. Zhang, G., et al., *TSDroid: A novel android malware detection framework based on temporal & spatial metrics in IoMT.* ACM Transactions on Sensor Networks, 2023. **19**(3): p. 1-23.
547. Sitara, K. and V. Manu, *Security Concerns and Forensic Aspects of IoHT*, in *Smart and Secure Internet of Healthcare Things*. 2022, CRC Press. p. 113-126.
548. Alanazi, H.K., A. Abd El-Aziz, and H. Hamdi, *Securing IoT Devices in e-Health using Machine Learning Techniques.* International Journal of Advanced Computer Science and Applications, 2023. **14**(9).
549. Navaz, A.N., et al., *Trends, technologies, and key challenges in smart and connected healthcare.* Ieee Access, 2021. **9**: p. 74044-74067.
550. Hossain, M.T., et al. *Cyberattacks Classification on Internet of Medical Things Using Information Gain Feature Selection and Machine Learning*. in *2024 Advances in Science and Engineering Technology International Conferences (ASET)*. 2024. IEEE.
551. Rahman, M. and H. Jahankhani, *Security vulnerabilities in existing security mechanisms for iomt and potential solutions for mitigating cyber-attacks.* Information security technologies for controlling pandemics, 2021: p. 307-334.
552. Ibrahim, D.R. and M.Y. Thanoun. *IoMT Availability Threats Attacks and Solution*. in *2024 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI)*. 2024. IEEE.
553. Kolokotronis, N., et al. *An intelligent platform for threat assessment and cyber-attack mitigation in IoMT ecosystems*. in *2022 IEEE Globecom Workshops (GC Wkshps)*. 2022. IEEE.
554. Reji, A., et al. *Anomaly detection for the internet-of-medical-things*. in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2023. IEEE.
555. Jayaraj, I.A., et al., *Detecting and Localizing Wireless Spoofing Attacks on the Internet of Medical Things.* Journal of Sensor and Actuator Networks, 2024. **13**(6): p. 72.

556. Karagiannis, S., et al., *Chidroid: A Mobile Android Application for Log Collection and Security Analysis in Healthcare and IoMT.* Applied Sciences, 2023. **13**(5): p. 3061.
557. Guerra, L. and L. Gonçalves, *Proactive Cybersecurity tailoring through deception techniques.*
558. Neshenko, N., et al., *Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations.* IEEE Communications Surveys & Tutorials, 2019. **21**(3): p. 2702-2733.
559. Xu, S., *Cybersecurity dynamics: A foundation for the science of cybersecurity.* Proactive and dynamic network defense, 2019: p. 1-31.
560. Christen, M., B. Gordijn, and M. Loi, *The ethics of cybersecurity*. 2020: Springer Nature.
561. Ventre, D., *Artificial Intelligence, Cybersecurity and Cyber Defence*. 2020: John Wiley & Sons.
562. JONES, J., A. IONIŢĂ, and I.-C. MIHAI. *AI and IoT Mapping and the Transition to an Interconnected Cyber Defence and Intelligence Capabilities*. in *International Conference on Cybersecurity and Cybercrime*. 2022.
563. Anastopoulos, V. and D. Giovannelli, *Automated/Autonomous Incident Response*. 2022.
564. Si-Ahmed, A., M.A. Al-Garadi, and N. Boustia, *Survey of Machine Learning based intrusion detection methods for Internet of Medical Things.* Applied Soft Computing, 2023. **140**: p. 110227.
565. Talukder, M.A., et al., *MLSTL-WSN: machine learning-based intrusion detection using SMOTETomek in WSNs.* International Journal of Information Security, 2024. **23**(3): p. 2139-2158.
566. Franklin, E. and B. Pranggono, *Anomaly-Based Intrusion Detection System for the Internet of Medical Things.* IJID (International Journal on Informatics for Development). **12**(2): p. 374-383.
567. Sun, Z., et al., *Optimized machine learning enabled intrusion detection 2 system for internet of medical things.* Franklin Open, 2024. **6**: p. 100056.
568. Koroniotis, N., et al., *Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset.* Future Generation Computer Systems, 2019. **100**: p. 779-796.
569. Azam, Z., M.M. Islam, and M.N. Huda, *Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree.* IEEE Access, 2023.
570. Albin Ahmed, A., et al., *Android Ransomware Detection Using Supervised Machine Learning Techniques Based on Traffic Analysis.* Sensors, 2023. **24**(1): p. 189.
571. Hasan, M., et al., *Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches.* Internet of Things, 2019. **7**: p. 100059.
572. Leevy, J.L., et al., *Detecting cybersecurity attacks across different network features and learners.* Journal of Big Data, 2021. **8**: p. 1-29.
573. Gangavarapu, T., C. Jaidhar, and B. Chanduka, *Applicability of machine learning in spam and phishing email filtering: review and approaches.* Artificial Intelligence Review, 2020. **53**(7): p. 5019-5081.
574. Feng, W., et al. *A support vector machine based naive Bayes algorithm for spam filtering*. in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*. 2016. IEEE.

575. Saif, S., N. Yasmin, and S. Biswas, *Feature engineering based performance analysis of ML and DL algorithms for Botnet attack detection in IoMT.* International Journal of System Assurance Engineering and Management, 2023. **14**(Suppl 1): p. 512-522.
576. Kalash, M., et al. *Malware classification with deep convolutional neural networks*. in *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)*. 2018. IEEE.
577. Yan, B. and G. Han, *LA-GRU: Building combined intrusion detection model based on imbalanced learning and gated recurrent unit neural network.* security and communication networks, 2018. **2018**.
578. Al-kahtani, M.S., et al., *Intrusion detection in the Internet of Things using fusion of GRU-LSTM deep learning model.* Intelligent Automation & Soft Computing, 2023. **37**(2).
579. Dhanya, L. and R. Chitra, *A novel autoencoder based feature independent GA optimised XGBoost classifier for IoMT malware detection.* Expert Systems with Applications, 2024. **237**: p. 121618.
580. Douiba, M., et al., *An improved anomaly detection model for IoT security using decision tree and gradient boosting.* The Journal of Supercomputing, 2023. **79**(3): p. 3392-3411.
581. Hernandez-Jaimes, M.L., et al., *Artificial intelligence for IoMT security: A review of intrusion detection systems, attacks, datasets and Cloud-Fog-Edge architectures.* Internet of Things, 2023: p. 100887.
582. Seh, A.H., et al. *Healthcare data breaches: insights and implications*. in *Healthcare*. 2020. MDPI.
583. Manikandan, J. and R.K. Choudhary, *Fortifying Medical IoT Systems: A Comprehensive Analysis of Advanced AI Techniques for Network Security.* Available at SSRN 4999651, 2024.
584. Rani, S., et al., *SmartHealth: An intelligent framework to secure IoMT service applications using machine learning.* ICT Express, 2024. **10**(2): p. 425-430.
585. Zhou, A. and S. Piramuthu. *Smart IoMT Applications in Senior Healthcare: Balancing Functionality, Security, and Privacy Challenges*. in *2024 Ninth International Conference On Mobile And Secure Services (MobiSecServ)*. 2024. IEEE.
586. Bajpayi, P., S. Sharma, and M.S. Gaur. *AI Driven IoT Healthcare Devices Security Vulnerability Management*. in *2024 2nd International Conference on Disruptive Technologies (ICDT)*. 2024. IEEE.
587. Alevizos, L. and M. Dekker, *Towards an AI-Enhanced Cyber Threat Intelligence Processing Pipeline.* Electronics, 2024. **13**(11): p. 2021.
588. Alrubayyi, H., et al., *Security Threats and Promising Solutions Arising from the Intersection of AI and IoT: A Study of IoMT and IoET Applications.* Future Internet, 2024. **16**(3): p. 85.
589. Nasayreh, A., et al., *Automated Detection of Cyber Attacks in Healthcare Systems: A Novel Scheme with Advanced Feature Extraction and Classification.* Computers & Security, 2024: p. 104288.
590. Saheed, Y.K. and M.O. Arowolo, *Efficient cyber attack detection on the internet of medical things-smart environment based on deep recurrent neural network and machine learning algorithms.* IEEE Access, 2021. **9**: p. 161546-161554.
591. Manimaran, M., et al., *Implementing Machine Learning-based Autonomic Cyber Defense for IoT-enabled Healthcare Devices.* Journal of Artificial Intelligence and Technology, 2023. **3**(4): p. 162-172.

592. Ksibi, S., F. Jaidi, and A. Bouhoula. *Cyber-risk management within IOMT: A context-aware agent-based framework for a reliable e-health system*. in *The 23rd International Conference on Information Integration and Web Intelligence*. 2021.
593. Kilincer, I.F., et al., *Automated detection of cybersecurity attacks in healthcare systems with recursive feature elimination and multilayer perceptron optimization.* Biocybernetics and Biomedical Engineering, 2023. **43**(1): p. 30-41.
594. Dutta, P.E., H. Neog, and N. Medhi. *Health monitoring in Internet of Medical Things (IoMT) using machine learning (ML) approaches*. in *2021 IEEE Globecom Workshops (GC Wkshps)*. 2021. IEEE.
595. Tomer, V., S. Sharma, and M. Davis, *Resilience in the Internet of Medical Things: A Review and Case Study.* Future Internet, 2024. **16**(11): p. 430.
596. Salmi, S., L. Oughdir, and A. El Affar. *Defending Against App-Layer DDoS: Advanced Machine Learning Security*. in *2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*. 2024. IEEE.
597. Khokhar, R.H., et al., *A Survey on Supply Chain Management: Exploring Physical and Cyber Security Challenges, Threats, Critical Applications, and Innovative Technologies.* International Journal of Supply and Operations Management, 2024.
598. Zachos, G., et al., *An anomaly-based intrusion detection system for internet of medical things networks.* Electronics, 2021. **10**(21): p. 2562.
599. Mohri, M., A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. 2018: MIT press.
600. Thamilarasu, G., A. Odesile, and A. Hoang, *An intrusion detection system for internet of medical things.* IEEE Access, 2020. **8**: p. 181560-181576.
601. Goodfellow, I., Y. Bengio, and A. Courville, *Deep learning*. 2016: MIT press.
602. Chen, C., et al., *Deep learning on computational-resource-limited platforms: a survey.* Mobile Information Systems, 2020. **2020**: p. 1-19.
603. Si-Ahmed, A., M.A. Al-Garadi, and N. Boustia, *Survey of Machine Learning Based Intrusion Detection Methods for Internet of Medical Things.* arXiv preprint arXiv:2202.09657, 2022.
604. Alpaydin, E., *Introduction to machine learning*. 2020: MIT press.
605. Adi, E., et al., *Machine learning and data analytics for the IoT.* Neural computing and applications, 2020. **32**: p. 16205-16233.
606. Al-Garadi, M.A., et al., *A survey of machine and deep learning methods for internet of things (IoT) security.* IEEE Communications Surveys & Tutorials, 2020. **22**(3): p. 1646-1685.
607. Zhu, J., et al., *Research on data security detection algorithm in IoT based on K-means.* Scalable Computing: Practice and Experience, 2021. **22**(2): p. 149–159-149–159.
608. Alhowaide, A., I. Alsmadi, and J. Tang. *PCA, Random-forest and pearson correlation for dimensionality reduction in IoT IDS.* in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. 2020. IEEE.
609. Zelaya, C.V.G. *Towards explaining the effects of data preprocessing on machine learning*. in *2019 IEEE 35th international conference on data engineering (ICDE)*. 2019. IEEE.
610. Berry, M.W., A. Mohamed, and B.W. Yap, *Supervised and unsupervised learning for data science*. 2019: Springer.
611. Van Engelen, J.E. and H.H. Hoos, *A survey on semi-supervised learning.* Machine learning, 2020. **109**(2): p. 373-440.

612. Sagel, A., et al., *Knowledge as Invariance--History and Perspectives of Knowledge-augmented Machine Learning.* arXiv preprint arXiv:2012.11406, 2020.
613. Zhang, L., et al., *Hierarchical deep-learning neural networks: finite elements and beyond.* Computational Mechanics, 2021. **67**: p. 207-230.
614. Curtis, F.E. and K. Scheinberg, *Optimization methods for supervised machine learning: From linear models to deep learning*, in *Leading Developments from INFORMS Communities*. 2017, INFORMS. p. 89-114.
615. Huang, J., et al. *Unsupervised deep learning by neighbourhood discovery*. in *International Conference on Machine Learning*. 2019. PMLR.
616. Ouali, Y., C. Hudelot, and M. Tami, *An overview of deep semi-supervised learning.* arXiv preprint arXiv:2006.05278, 2020.
617. Li, Y., *Deep reinforcement learning: An overview.* arXiv preprint arXiv:1701.07274, 2017.
618. Shu, Y. and Y. Xu. *End-to-end captcha recognition using deep CNN-RNN network*. in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. 2019. IEEE.
619. Kopp, M., M. Nikl, and M. Holena. *Breaking CAPTCHAs with Convolutional Neural Networks*. in *ITAT*. 2017.
620. Hassan, H.A., et al., *Intrusion Detection Systems for the Internet of Thing: A Survey Study.* Wireless Personal Communications, 2023. **128**(4): p. 2753-2778.
621. Khraisat, A., et al., *Survey of intrusion detection systems: techniques, datasets and challenges.* Cybersecurity, 2019. **2**(1): p. 1-22.
622. Heidari, A. and M.A. Jabraeil Jamali, *Internet of Things intrusion detection systems: A comprehensive review and future directions.* Cluster Computing, 2023. **26**(6): p. 3753-3780.
623. Ioulianou, P., et al., *A signature-based intrusion detection system for the internet of things.* Information and Communication Technology Form, 2018.
624. Molina-Coronado, B., et al., *Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process.* IEEE Transactions on Network and Service Management, 2020. **17**(4): p. 2451-2479.
625. Patcha, A. and J.-M. Park, *An overview of anomaly detection techniques: Existing solutions and latest technological trends.* Computer networks, 2007. **51**(12): p. 3448-3470.
626. Rathore, H., et al., *A novel deep learning strategy for classifying different attack patterns for deep brain implants.* IEEE Access, 2019. **7**: p. 24154-24164.
627. Kintzlinger, M., et al., *CardiWall: a trusted firewall for the detection of malicious clinical programming of cardiac implantable electronic devices.* IEEE Access, 2020. **8**: p. 48123-48140.
628. Rathore, H., et al. *DLRT: Deep learning approach for reliable diabetic treatment*. in *GLOBECOM 2017-2017 IEEE global communications conference*. 2017. IEEE.
629. Maulud, D. and A.M. Abdulazeez, *A review on linear regression comprehensive in machine learning.* Journal of Applied Science and Technology Trends, 2020. **1**(4): p. 140-147.
630. Begli, M., F. Derakhshan, and H. Karimipour. *A layered intrusion detection system for critical infrastructure using machine learning*. in *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*. 2019. IEEE.
631. Schneble, W. and G. Thamilarasu. *Attack detection using federated learning in medical cyber-physical systems*. in *Proc. 28th Int. Conf. Comput. Commun. Netw.(ICCCN)*. 2019.

632. Alani, M.M., A. Mashatan, and A. Miri. *Explainable Ensemble-Based Detection of Cyber Attacks on Internet of Medical Things*. in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. 2023. IEEE.
633. Raschka, S. and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. 2019: Packt publishing ltd.
634. Latendresse, J., et al., *An Exploratory Study on Machine Learning Model Management.* ACM Transactions on Software Engineering and Methodology, 2024.
635. Zhang, R.F. and R.J. Urbanowicz. *A scikit-learn compatible learning classifier system*. in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*. 2020.
636. Raschka, S., J. Patterson, and C. Nolet, *Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence.* Information, 2020. **11**(4): p. 193.
637. Nguyen, G., et al., *Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey.* Artificial Intelligence Review, 2019. **52**: p. 77-124.
638. Faska, Z., et al., *A robust and consistent stack generalized ensemble-learning framework for image segmentation.* Journal of Engineering and Applied Science, 2023. **70**(1): p. 74.
639. Sharma, A., N. Patel, and R. Gupta, *Enhancing Customer Lifetime Value Prediction Using Random Forests and Neural Network Ensemble Methods.* European Advanced AI Journal, 2022. **11**(8).
640. Boutahir, M.K., et al., *Meta-learning guided weight optimization for enhanced solar radiation forecasting and sustainable energy management with VotingRegressor.* Sustainability, 2024. **16**(13): p. 5505.
641. Alsolami, T., B. Alsharif, and M. Ilyas, *Enhancing cybersecurity in healthcare: Evaluating ensemble learning models for intrusion detection in the internet of medical things.* Sensors, 2024. **24**(18): p. 5937.
642. Mousa'B, M.S., et al., *An explainable ensemble deep learning approach for intrusion detection in industrial Internet of Things.* IEEE Access, 2023. **11**: p. 115047-115061.
643. Bibers, I., O. Arreche, and M. Abdallah, *A Comprehensive Comparative Study of Individual ML Models and Ensemble Strategies for Network Intrusion Detection Systems.* arXiv preprint arXiv:2410.15597, 2024.
644. Hossain, M.A., et al., *Towards superior android ransomware detection: An ensemble machine learning perspective.* Cyber Security and Applications, 2025. **3**: p. 100076.
645. Peppes, N., et al., *Performance of machine learning-based multi-model voting ensemble methods for network threat detection in agriculture 4.0.* Sensors, 2021. **21**(22): p. 7475.
646. Hady, A.A., et al., *Intrusion detection system for healthcare systems using medical and network data: A comparison study.* IEEE Access, 2020. **8**: p. 106576-106584.
647. Faysal, J.A., et al. *XGB-RF: A hybrid machine learning approach for IoT intrusion detection*. in *Telecom*. 2022. MDPI.
648. Essa, M.S. and S.K. Guirguis, *Evaluation of Tree-Based Machine Learning Algorithms for Network Intrusion Detection in the Internet of Things.* IT Professional, 2023. **25**(5): p. 45-56.
649. Ibrahimi, K. and H. Benaddi. *Improving the ids for bot-iot dataset-based machine learning classifiers*. in *2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet)*. 2022. IEEE.

650. Samdekar, R., S. Ghosh, and K. Srinivas. *Efficiency enhancement of intrusion detection in iot based on machine learning through bioinspire*. in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. 2021. IEEE.
651. Sanju, P., *Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks.* Journal of Engineering Research, 2023. **11**(4): p. 356-361.
652. Khraisat, A. and A. Alazab, *A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges.* Cybersecurity, 2021. **4**: p. 1-27.
653. Stiawan, D., et al., *CICIDS-2017 dataset feature analysis with information gain for anomaly detection.* IEEE Access, 2020. **8**: p. 132911-132921.
654. Roopak, M., G.Y. Tian, and J. Chambers. *Deep learning models for cyber security in IoT networks*. in *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*. 2019. IEEE.
655. Meidan, Y., et al., *N-baiot—network-based detection of iot botnet attacks using deep autoencoders.* IEEE Pervasive Computing, 2018. **17**(3): p. 12-22.
656. Moustafa, N., et al., *Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions.* IEEE Communications Surveys & Tutorials, 2023. **25**(3): p. 1775-1807.
657. Hassini, K., et al., *An end-to-end learning approach for enhancing intrusion detection in Industrial-Internet of Things.* Knowledge-Based Systems, 2024. **294**: p. 111785.
658. Khanan, A., et al., *From Bytes to Insights: A Systematic Literature Review on Unraveling IDS Datasets for Enhanced Cybersecurity Understanding.* IEEE Access, 2024.
659. Dowling, S., M. Schukat, and H. Melvin. *A ZigBee honeypot to assess IoT cyberattack behaviour*. in *2017 28th Irish signals and systems conference (ISSC)*. 2017. IEEE.
660. Bezerra, V.H., et al. *Providing IoT host-based datasets for intrusion detection research∗.* in *Anais do XVIII Simpósio Brasileiro de Segurança da Informaçao e de Sistemas Computacionais*. 2018. SBC.
661. Parmisano, A., S. Garcia, and M.J. Erquiaga, *A labeled dataset with malicious and benign iot network traffic.* Stratosphere Laboratory: Praha, Czech Republic, 2020.
662. Al-Hadhrami, Y. and F.K. Hussain, *Real time dataset generation framework for intrusion detection systems in IoT.* Future Generation Computer Systems, 2020. **108**: p. 414-423.
663. Vigoya, L., et al., *Annotated dataset for anomaly detection in a data center with IoT sensors.* Sensors, 2020. **20**(13): p. 3745.
664. Ahmed, M., et al., *ECU-IoFT: A dataset for analysing cyber-attacks on Internet of Flying Things.* Applied Sciences, 2022. **12**(4): p. 1990.
665. Thabit, F., et al., *Enhanced an Intrusion Detection System for IoT networks through machine learning techniques: an examination utilizing the AWID dataset.* Cogent Engineering, 2024. **11**(1): p. 2378603.
666. Ball, J.E., D.T. Anderson, and C.S. Chan, *Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community.* Journal of applied remote sensing, 2017. **11**(4): p. 042609-042609.
667. L'heureux, A., et al., *Machine learning with big data: Challenges and approaches.* Ieee Access, 2017. **5**: p. 7776-7797.
668. Mehrabi, N., et al., *A survey on bias and fairness in machine learning.* ACM computing surveys (CSUR), 2021. **54**(6): p. 1-35.

669. Shahria, M.T. and M.H. Rahman, *Activities of Daily Living Object Dataset: Advancing Assistive Robotic Manipulation with a Tailored Dataset.* Sensors, 2024. **24**(23): p. 7566.
670. El-Madafri, I., M. Peña, and N. Olmedo-Torre, *Dual-Dataset Deep Learning for Improved Forest Fire Detection: A Novel Hierarchical Domain-Adaptive Learning Approach.* Mathematics, 2024. **12**(4): p. 534.
671. Theodorakopoulos, L., A. Theodoropoulou, and Y. Stamatiou, *A State-of-the-Art Review in Big Data Management Engineering: Real-Life Case Studies, Challenges, and Future Research Directions.* Eng, 2024. **5**(3): p. 1266-1297.
672. Boralkar, R.R. and S.S. Kulkarni. *IoT Based Smart Agriculture System Using ESP32.* in *2024 4th Interdisciplinary Conference on Electrics and Computer (INTCEC)*. 2024. IEEE.
673. Dzahir, M.A.S.M. and K.S. Chia. *Evaluating the Energy Consumption of ESP32 Microcontroller for Real-Time MQTT IoT-Based Monitoring System*. in *2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*. 2023. IEEE.
674. Najeh, H., C. Lohr, and B. Leduc, *Real-Time Human Activity Recognition on Embedded Equipment: A Comparative Study.* Applied Sciences, 2024. **14**(6): p. 2377.
675. Park, E. and K.C. Chan. *Designing a Testbed to Assess Secure Control of Cyber-Physical Systems.* in *2019 International Conference on Mechatronics, Robotics and Systems Engineering (MoRSE)*. 2019. IEEE.
676. Gemmer, D.D., et al. *A Scalable Cyber Security Framework for the Experimentation of DDoS Attacks of Things*. in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. 2023. IEEE.
677. Xavier, G., C. Novo, and R. Morla, *Tweaking Metasploit to Evade Encrypted C2 Traffic Detection.* arXiv preprint arXiv:2209.00943, 2022.
678. Osueke, T.T., *Detecting Brute-Force Attack in IoT Device using Network Flow Data*. 2018, Dublin, National College of Ireland.
679. Shamsuzzaman, H., et al., *Cybersecurity Risk Mitigation in Industrial Control Systems Analyzing Physical Hybrid And Virtual Test Bed Applications.* Academic Journal on Artificial Intelligence, Machine Learning, Data Science and Management Information Systems, 2024. **1**(01): p. 19-39.
680. Abid, A., F. Jemili, and O. Korbaa, *Real-time data fusion for intrusion detection in industrial control systems based on cloud computing and big data techniques.* Cluster Computing, 2024. **27**(2): p. 2217-2238.
681. Kampourakis, V., V. Gkioulos, and S. Katsikas, *A systematic literature review on wireless security testbeds in the cyber-physical realm.* Computers & Security, 2023: p. 103383.
682. Siboni, S., et al., *Security testbed for Internet-of-Things devices.* IEEE transactions on reliability, 2019. **68**(1): p. 23-44.
683. Zhou, X., et al., *Review on testing of cyber physical systems: Methods and testbeds.* IEEE Access, 2018. **6**: p. 52179-52194.
684. Gomez, J., et al., *A survey on network simulators, emulators, and testbeds used for research and education.* Computer Networks, 2023. **237**: p. 110054.
685. Chernyshev, M., et al., *Internet of things (iot): Research, simulators, and testbeds.* IEEE Internet of Things Journal, 2017. **5**(3): p. 1637-1647.
686. Jayaraj, I.A., et al., *A systematic review of radio frequency threats in IOMT.* Journal of Sensor and Actuator Networks, 2022. **11**(4): p. 62.

687. Mahmood, S., H.N. Nguyen, and S.A. Shaikh, *Automotive cybersecurity testing: Survey of testbeds and methods.* Digital Transformation, Cyber Security and Resilience of Modern Societies, 2021: p. 219-243.
688. Rao, A.S., B.V. Vardhan, and H. Shaik. *Role of exploratory data analysis in data science*. in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*. 2021. IEEE.
689. Rajest, S.S. and R. Regin, *In-depth Exploratory Data Analysis of Global Surface Temperature: Uncovering Patterns, Anomalies, and Long-term Trends in Climate Data.* Central Asian Journal of Theoretical and Applied Science, 2024. **5**(6): p. 563-578.
690. Sen, A., N. Thakur, and P. Akilandeswari, *Predicting heart attacks: Risk factors analysis using exploratory data analysis and visualization*, in *Computer Science Engineering*. 2024, CRC Press. p. 712-724.
691. Nobre, J., E.S. Pires, and A. Reis, *Anomaly Detection in Microservice-Based Systems.* Applied Sciences, 2023. **13**(13): p. 7891.
692. Sadare, O.O., A.M. Azeez, and A.A. Opatunji, *Phishing Attack and Defense: An Exploratory Data Analytics of Uniform Resource Locators for Cybersecurity.*
693. Rabzelj, M., et al., *Cyberattack graph modeling for visual analytics.* IEEE Access, 2023. **11**: p. 86910-86944.
694. Sharma, N. and N.S. Yadav. *Ensemble learning based classification of UNSW-NB15 dataset using exploratory data analysis*. in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. 2021. IEEE.
695. Yamany, B., et al., *A holistic approach to ransomware classification: Leveraging static and dynamic analysis with visualization.* Information, 2024. **15**(1): p. 46.
696. Saeed, S., et al., *A systematic literature review on cyber threat intelligence for organizational cybersecurity resilience.* Sensors, 2023. **23**(16): p. 7273.
697. Rana, A., et al., *Heatmap-Based Deep Learning Model for Network Attacks Classification.* SN Computer Science, 2024. **5**(8): p. 1113.
698. Liu, Y., et al., *Daily activity feature selection in smart homes based on pearson correlation coefficient.* Neural Processing Letters, 2020. **51**(2): p. 1771-1787.
699. Saranya, G. and A. Pravin, *A novel feature selection approach with integrated feature sensitivity and feature correlation for improved prediction of heart disease.* Journal of Ambient Intelligence and Humanized Computing, 2023. **14**(9): p. 12005-12019.
700. Pudjihartono, N., et al., *A review of feature selection methods for machine learning-based disease risk prediction.* Frontiers in Bioinformatics, 2022. **2**: p. 927312.
701. Li, J., et al., *Feature selection: A data perspective.* ACM computing surveys (CSUR), 2017. **50**(6): p. 1-45.
702. Song, L., P. Langfelder, and S. Horvath, *Comparison of co-expression measures: mutual information, correlation, and model based indices.* BMC bioinformatics, 2012. **13**: p. 1-21.
703. Cai, J., et al., *Feature selection in machine learning: A new perspective.* Neurocomputing, 2018. **300**: p. 70-79.
704. Yang, L. and A. Shami, *Ids-ml: An open source code for intrusion detection system development using machine learning.* Software Impacts, 2022. **14**: p. 100446.
705. Kadiyala, P. and K.A. Kumar, *A Deep Learning–Based Malware and Intrusion Detection Framework.* Convergence of Deep Learning In Cyber-IoT Systems and Security, 2022: p. 367-380.
706. Sadia, H., et al., *Intrusion Detection System for Wireless Sensor Networks: A Machine Learning based Approach.* IEEE Access, 2024.

707. Lazzarini, R., H. Tianfield, and V. Charissis, *A stacking ensemble of deep learning models for IoT intrusion detection.* Knowledge-Based Systems, 2023. **279**: p. 110941.
708. Osa, E., P.E. Orukpe, and U. Iruansi, *Design and implementation of a deep neural network approach for intrusion detection systems.* e-Prime-Advances in Electrical Engineering, Electronics and Energy, 2024. **7**: p. 100434.
709. Fedorchenko, E., E. Novikova, and A. Shulepov, *Comparative review of the intrusion detection systems based on federated learning: Advantages and open challenges.* Algorithms, 2022. **15**(7): p. 247.
710. Purushotam Naidu, K., et al., *Intrusion Detection System Using Gated Recurrent Neural Network.* International Journal of Software Computing and Testing, 2024. **10**(1): p. 1-8p.
711. Raj, R.K., et al., *An empirical approach to understanding data science and engineering education*, in *Proceedings of the working group reports on innovation and technology in computer science education*. 2019. p. 73-87.
712. Chatterjee, A., et al., *Exploring online public survey lifestyle datasets with statistical analysis, machine learning and semantic ontology.* Scientific Reports, 2024. **14**(1): p. 24190.
713. Lima, M.G., et al., *Impacts of Data Preprocessing and Hyperparameter Optimization on the Performance of Machine Learning Models Applied to Intrusion Detection Systems.* arXiv preprint arXiv:2407.11105, 2024.
714. Talukder, M.A., et al., *Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction.* Journal of big data, 2024. **11**(1): p. 33.
715. Chua, T.-H. and I. Salam, *Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection Using Progressive Dataset.* Symmetry, 2023. **15**(6): p. 1251.
716. Musthafa, M.B., et al., *Optimizing iot intrusion detection using balanced class distribution, feature selection, and ensemble machine learning techniques.* Sensors, 2024. **24**(13): p. 4293.
717. Parihar, M. and C. Fung. *IDS with deep learning techniques*. in *2023 7th Cyber Security in Networking Conference (CSNet)*. 2023. IEEE.
718. Hosseini, S. and S.R. Sardo, *Data mining tools-a case study for network intrusion detection.* Multimedia Tools and Applications, 2021. **80**(4): p. 4999-5019.
719. Hossain, Z., et al. *Network Intrusion Detection using Machine Learning Approaches*. in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. 2021. IEEE.
720. Tufan, E., C. Tezcan, and C. Acartürk, *Anomaly-based intrusion detection by machine learning: A case study on probing attacks to an institutional network.* IEEE Access, 2021. **9**: p. 50078-50092.
721. Dwibedi, S., M. Pujari, and W. Sun. *A comparative study on contemporary intrusion detection datasets for machine learning research*. in *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2020. IEEE.
722. Jayant, P., et al. *Intrusion Detection in Network Traffic Using LSTM and Deep Learning*. in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2024. IEEE.
723. Mynuddin, M., et al. *Automatic Network Intrusion Detection System Using Machine Learning and Deep Learning*. in *2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*. 2024. IEEE.

724. Susilo, B. and R.F. Sari. *Intrusion detection in software defined network using deep learning approach*. in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. 2021. IEEE.
725. Okey, O.D., et al., *BoostedEnML: Efficient technique for detecting cyberattacks in IoT systems using boosted ensemble machine learning.* Sensors, 2022. **22**(19): p. 7409.
726. Weng, Y. and J. Wu, *Fortifying the global data fortress: a multidimensional examination of cyber security indexes and data protection measures across 193 nations.* International Journal of Frontiers in Engineering Technology, 2024. **6**(2): p. 13-28.
727. Almansoori, A., et al. *Analysis of cybercrime on social media platforms and its challenges*. in *The International Conference on Artificial Intelligence and Computer Vision*. 2021. Springer.
728. Zhou, C. and R.C. Paffenroth. *Anomaly detection with robust deep autoencoders*. in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017.
729. Kim, S., W. Jo, and T. Shon, *APAD: Autoencoder-based payload anomaly detection for industrial IoE.* Applied Soft Computing, 2020. **88**: p. 106017.
730. Tabassum, T., O. Toker, and M.R. Khalghani, *Cyber–physical anomaly detection for inverter-based microgrid using autoencoder neural network.* Applied Energy, 2024. **355**: p. 122283.
731. Pinto, A., et al., *Enhancing Critical Infrastructure Security: Unsupervised Learning Approaches for Anomaly Detection.* International Journal of Computational Intelligence Systems, 2024. **17**(1): p. 236.
732. Wintjen, M., *Practical Data Analysis Using Jupyter Notebook: Learn how to speak the language of data by extracting useful and actionable insights using Python*. 2020: Packt Publishing Ltd.
733. De Santana, T.L., et al., *Bug Analysis in Jupyter Notebook Projects: An Empirical Study.* ACM Transactions on Software Engineering and Methodology, 2024. **33**(4): p. 1-34.
734. Ahrén, B. and G. Pacini, *Glucose effectiveness: Lessons from studies on insulin-independent glucose clearance in mice.* Journal of Diabetes Investigation, 2021. **12**(5): p. 675-685.
735. Bergman, R.N., *Origins and History of the Minimal Model of Glucose Regulation.* Frontiers in Endocrinology, 2021. **11**: p. 1151.
736. Patel, B.J., et al., *Comparison and correlation of glucose levels in serum and saliva of both diabetic and non-diabetic patients.* Journal of international oral health: JIOH, 2015. **7**(8): p. 70.
737. Krinsley, J.S. and J.-C. Preiser, *Time in blood glucose range 70 to 140 mg/dl> 80% is strongly associated with increased survival in non-diabetic critically ill adults.* Critical care, 2015. **19**(1): p. 1-11.
738. Kwok, R., et al., *Evaluation of NovoRapid infusion as a treatment option in the management of diabetic ketoacidosis.* Internal medicine journal, 2017. **47**(11): p. 1317-1320.
739. MFMER, *Diabetes Screening according to Mayo Foundation for Medical Education and Research (MFMER).* 2021.
740. Hu, H., D. Gu, and M. Brady, *A modular computing architecture for autonomous robots.* Microprocessors and Microsystems, 1998. **21**(6): p. 349-361.

741. Farritor, S., et al. *A systems-level modular design approach to field robotics*. in *Proceedings of IEEE International Conference on Robotics and Automation*. 1996. IEEE.
742. Kumar, S., et al., *Modular design and decentralized control of the recupera exoskeleton for stroke rehabilitation.* Applied Sciences, 2019. **9**(4): p. 626.
743. Garcia, S. and C.T. Trinh, *Modular design: implementing proven engineering principles in biotechnology.* Biotechnology Advances, 2019. **37**(7): p. 107403.
744. Edmonds, N., D. Stark, and J. Davis. *MASS: modular architecture for sensor systems*. in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* 2005. IEEE.
745. Davis, D. and B.J. Burry Mark, *Untangling parametric schemata: enhancing collaboration through modular programming.* 2011.
746. Zhou, Y. and J. Bi, *Clickp4: Towards modular programming of p4*, in *Proceedings of the SIGCOMM Posters and Demos*. 2017. p. 100-102.
747. Coronado, E., F. Mastrogiovanni, and G. Venture. *Development of intelligent behaviors for social robots via user-friendly and modular programming tools*. in *2018 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*. 2018. IEEE.
748. Waraga, O.A., et al., *Design and implementation of automated IoT security testbed.* Computers & Security, 2020. **88**: p. 101648.
749. Sachidananda, V., et al. *Let the cat out of the bag: A holistic approach towards security analysis of the internet of things*. in *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security*. 2017.
750. Liu, M., et al., *Host-based intrusion detection system with system calls: Review and future trends.* ACM Computing Surveys (CSUR), 2018. **51**(5): p. 1-36.
751. Kennedy, D., et al., *Metasploit: the penetration tester's guide*. 2011: No Starch Press.
752. Rahalkar, S., *Metasploit for Beginners*. 2017: Packt Publishing Ltd.
753. Jaswal, N., *Mastering Metasploit: Take your penetration testing and IT security skills to a whole new level with the secrets of Metasploit*. 2018: Packt Publishing Ltd.
754. Li, W., et al., *Securing MQTT Ecosystem: Exploring Vulnerabilities, Mitigations, and Future Trajectories.* IEEE Access, 2024.
755. Abeles, D. and M. Zioni, *Mqtt-pwn documentation.* 2018.
756. Gupta, A. and L.S. Sharma, *Detecting attacks in high-speed networks: Issues and solutions.* Information Security Journal: A Global Perspective, 2020. **29**(2): p. 51-61.
757. Sousa, B.F.L.M., et al. *An intrusion detection system for denial of service attack detection in internet of things*. in *Proceedings of the second international conference on internet of things, data and cloud computing*. 2017.
758. Chaabouni, N., et al., *Network intrusion detection for IoT security based on learning techniques.* IEEE Communications Surveys & Tutorials, 2019. **21**(3): p. 2671-2701.
759. Ahmed, M., et al., *ECU-IoHT: A dataset for analyzing cyberattacks in Internet of Health Things.* Ad Hoc Networks, 2021. **122**: p. 102621.
760. Li, Y., et al., *Backdoor learning: A survey.* IEEE Transactions on Neural Networks and Learning Systems, 2022.
761. Allouzi, M.A. and J.I. Khan, *Identifying and modeling security threats for IoMT edge network using Markov chain and common vulnerability scoring system (CVSS).* arXiv preprint arXiv:2104.11580, 2021.
762. Kumar, P., G.P. Gupta, and R. Tripathi, *An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks.* Computer Communications, 2021. **166**: p. 110-124.
763. Schober, P., C. Boer, and L.A. Schwarte, *Correlation coefficients: appropriate use and interpretation.* Anesthesia & analgesia, 2018. **126**(5): p. 1763-1768.

764. Baak, M., et al., *A new correlation coefficient between categorical, ordinal and interval variables with Pearson characteristics.* Computational Statistics & Data Analysis, 2020. **152**: p. 107043.
765. Akoglu, H., *User's guide to correlation coefficients.* Turkish journal of emergency medicine, 2018. **18**(3): p. 91-93.
766. Norouzian, R. and L. Plonsky, *Correlation and simple linear regression in applied linguistics.* The Palgrave handbook of applied linguistics research methodology, 2018: p. 395-421.
767. Pandey, S., *Principles of correlation and regression analysis.* Journal of the practice of cardiovascular sciences, 2020. **6**(1): p. 7-11.
768. Peng, J., K. Cai, and X. Jin, *High concurrency massive data collection algorithm for IoMT applications.* Computer Communications, 2020. **157**: p. 402-409.
769. Ibaida, A., A. Abuadbba, and N. Chilamkurti, *Privacy-preserving compression model for efficient IoMT ECG sharing.* Computer Communications, 2021. **166**: p. 1-8.
770. Ali, N.J., et al., *An intelligent approach for enhancing the Quality of Service in IoMT based on 5G.* Periodicals of Engineering and Natural Sciences (PEN), 2023. **11**(3): p. 58-67.
771. Fu, T., et al., *Correlation research of phase angle variation and coating performance by means of Pearson's correlation coefficient.* Progress in Organic Coatings, 2020. **139**: p. 105459.
772. ABOUNAIMA, M.C., et al. *The pearson correlation coefficient applied to compare multi-criteria methods: case the ranking problematic*. in *2020 1st international conference on innovative research in applied science, engineering and technology (IRASET)*. 2020. IEEE.
773. Zhi, X., et al. *Research on the Pearson correlation coefficient evaluation method of analog signal in the process of unit peak load regulation*. in *2017 13th ieee international conference on electronic measurement & instruments (icemi)*. 2017. IEEE.
774. Siska, C. and K. Kechris, *Differential correlation for sequencing data.* BMC research notes, 2017. **10**: p. 1-9.
775. Umar, I.H., H. Lin, and J.I. Hassan, *Transforming Landslide Prediction: A Novel Approach Combining Numerical Methods and Advanced Correlation Analysis in Slope Stability Investigation.* Applied Sciences, 2024. **14**(9): p. 3685.
776. Treder, M.S., R. Codrai, and K.A. Tsvetanov, *Quality assessment of anatomical MRI images from generative adversarial networks: Human assessment and image quality metrics.* Journal of Neuroscience Methods, 2022. **374**: p. 109579.
777. Ye, G., et al., *Machine learning-based continuous intracranial pressure prediction for traumatic injury patients.* IEEE Journal of Translational Engineering in Health and Medicine, 2022. **10**: p. 1-8.
778. Muñoz-Pichardo, J.M., et al., *Multiple ordinal correlation based on Kendall's Tau measure: A proposal.* Mathematics, 2021. **9**(14): p. 1616.
779. de Raadt, A., et al., *A comparison of reliability coefficients for ordinal rating scales.* Journal of Classification, 2021: p. 1-25.
780. Kursa, M.B., *Kendall transformation brings a robust categorical representation of ordinal data.* Scientific Reports, 2022. **12**(1): p. 8341.
781. Hill, C., et al., *Comparing programming languages for data analytics: Accuracy of estimation in Python and R.* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2024: p. e1531.
782. Berjawi, O., et al., *Cyberattacks detection through behavior analysis of internet traffic.* Procedia Computer Science, 2023. **224**: p. 52-59.

783. Andresini, G., et al., *Multi-channel deep feature learning for intrusion detection.* IEEE Access, 2020. **8**: p. 53346-53359.
784. Mallampati, S.B. and H. Seetha, *Enhancing Intrusion Detection with Explainable AI: A Transparent Approach to Network Security.* Cybernetics and Information Technologies, 2024. **24**(1): p. 98-117.
785. Li, Y., et al. *Power terminal anomaly monitoring technology based on autoencoder and multi-layer perceptron*. in *Proceedings of the 2024 3rd International Conference on Networks, Communications and Information Technology*. 2024.
786. Sarhan, M., et al., *Feature extraction for machine learning-based intrusion detection in IoT networks.* Digital Communications and Networks, 2022.
787. Al Lail, M., A. Garcia, and S. Olivo, *Machine learning for network intrusion detection—a comparative study.* Future Internet, 2023. **15**(7): p. 243.
788. Gouda, H.A., M.A. Ahmed, and M.I. Roushdy, *Optimizing anomaly-based attack detection using classification machine learning.* Neural Computing and Applications, 2024. **36**(6): p. 3239-3257.
789. Chimphlee, S. and W. Chimphlee, *Machine learning to improve the performance of anomaly-based network intrusion detection in big data.* Indonesian Journal of Electrical Engineering and Computer Science, 2023. **30**(2): p. 1106-1119.
790. Huda, S.A., et al. *Evaluation of Machine Learning Models for Detecting Network-Based Intrusions*. in *Proceedings of the 11th International Conference on Smart Media and Applications (SMA 2022), Gunsan, Republic of Korea*. 2021.
791. Sharma, V., *A study on data scaling methods for machine learning.* International Journal for Global Academic & Scientific Research, 2022. **1**(1): p. 31-42.
792. Al-Mimi, H., et al., *An Enhanced Intrusion Detection System for Protecting HTTP Services from Attacks.* International Journal of Advances in Soft Computing & Its Applications, 2023. **15**(3).
793. Kaushik, B., et al., *Performance evaluation of learning models for intrusion detection system using feature selection.* Journal of Computer Virology and Hacking Techniques, 2023. **19**(4): p. 529-548.
794. Hidayat, I., M.Z. Ali, and A. Arshad, *Machine learning-based intrusion detection system: an experimental comparison.* Journal of Computational and Cognitive Engineering, 2023. **2**(2): p. 88-97.
795. Alrayes, F.S., et al., *Intrusion Detection in IoT Systems Using Denoising Autoencoder.* IEEE Access, 2024.
796. Santos, K.C., R.S. Miani, and F. de Oliveira Silva, *Evaluating the Impact of Data Preprocessing Techniques on the Performance of Intrusion Detection Systems.* Journal of Network and Systems Management, 2024. **32**(2): p. 36.
797. NERLIKAR, P., et al., *Analysis of intrusion detection using machine learning techniques.* International Journal of Computer Networks and Communications Security, 2020. **8**(10): p. 84-93.
798. She, X. and Y. Sekiya. *A convolutional autoencoder based method with smote for cyber intrusion detection*. in *2021 IEEE International Conference on Big Data (Big Data)*. 2021. IEEE.
799. Al-Imran, M. and S.H. Ripon, *Network intrusion detection: an analytical assessment using deep learning and state-of-the-art machine learning models.* International Journal of Computational Intelligence Systems, 2021. **14**(1): p. 200.
800. Reddy, L.N., S. Butakov, and P. Zavarsky. *Applying ML Algorithms to improve traffic classification in Intrusion Detection Systems*. in *2020 IEEE 19th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. 2020. IEEE.

801. Sah, G., S. Banerjee, and S. Singh, *Intrusion detection system over real-time data traffic using machine learning methods with feature selection approaches.* International Journal of Information Security, 2023. **22**(1): p. 1-27.
802. CVE-2020-15802, *National Vulnerability Database.* 2020.
803. Feurer, M. and F. Hutter, *Hyperparameter optimization.* Automated machine learning: Methods, systems, challenges, 2019: p. 3-33.
804. Raiaan, M.A.K., et al., *A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks.* Decision Analytics Journal, 2024: p. 100470.
805. Yu, T. and H. Zhu, *Hyper-parameter optimization: A review of algorithms and applications.* arXiv preprint arXiv:2003.05689, 2020.
806. Khan, M.A., et al. *Machine Learning-based Test Case Prioritization using Hyperparameter Optimization*. in *Proceedings of the 5th ACM/IEEE International Conference on Automation of Software Test (AST 2024)*. 2024.
807. Nematzadeh, S., et al., *Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases.* Computational biology and chemistry, 2022. **97**: p. 107619.
808. Karl, F., et al., *Multi-Objective Hyperparameter Optimization--An Overview.* arXiv preprint arXiv:2206.07438, 2022.
809. Vanitha, K., S.S. Sree, and S. Guluwadi, *Deep learning ensemble approach with explainable AI for lung and colon cancer classification using advanced hyperparameter tuning.* BMC Medical Informatics and Decision Making, 2024. **24**(1): p. 222.
810. Asif, D., et al., *Enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization.* Algorithms, 2023. **16**(6): p. 308.
811. Alif, M.A.R., *Enhancing diabetic retinopathy diagnosis: A lightweight cnn architecture for efficient exudate detection in retinal fundus images.* arXiv preprint arXiv:2408.06784, 2024.
812. Zakariah, M., et al., *Intrusion Detection System with Customized Machine Learning Techniques for NSL-KDD Dataset.* Computers, Materials & Continua, 2023. **77**(3).
813. Abdelmoumin, G., D.B. Rawat, and A. Rahman, *On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things.* IEEE Internet of Things Journal, 2021. **9**(6): p. 4280-4290.
814. Sabeel, U., et al., *Building an intrusion detection system to detect atypical cyberattack flows.* IEEE Access, 2021. **9**: p. 94352-94370.
815. Tran, N., et al., *Hyper-parameter optimization in classification: To-do or not-to-do.* Pattern Recognition, 2020. **103**: p. 107245.
816. Alibrahim, H. and S.A. Ludwig. *Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization*. in *2021 IEEE Congress on Evolutionary Computation (CEC)*. 2021. IEEE.
817. El-Hassani, F.Z., et al., *A new optimization model for MLP hyperparameter tuning: modeling and resolution by real-coded genetic algorithm.* Neural Processing Letters, 2024. **56**(2): p. 105.
818. Saranyaraj, D., M. Manikandan, and S. Maheswari, *A deep convolutional neural network for the early detection of breast carcinoma with respect to hyper-parameter tuning.* Multimedia Tools and Applications, 2020. **79**(15-16): p. 11013-11038.
819. Lee, H., et al., *Predicting Liquid Natural Gas Consumption via the Multilayer Perceptron Algorithm Using Bayesian Hyperparameter Autotuning.* Energies, 2024. **17**(10): p. 2290.

820. Mantovani, R.G., et al., *An empirical study on hyperparameter tuning of decision trees.* arXiv preprint arXiv:1812.02207, 2018.
821. Gomes Mantovani, R., et al., *Better trees: an empirical study on hyperparameter tuning of classification decision tree induction algorithms.* Data Mining and Knowledge Discovery, 2024: p. 1-53.
822. Guido, R., M.C. Groccia, and D. Conforti, *A hyper-parameter tuning approach for cost-sensitive support vector machine classifiers.* Soft Computing, 2023. **27**(18): p. 12863-12881.
823. Tsirikoglou, P., et al., *A hyperparameters selection technique for support vector regression models.* Applied Soft Computing, 2017. **61**: p. 139-148.
824. Villalobos-Arias, L., et al. *Evaluating hyper-parameter tuning using random search in support vector machines for software effort estimation*. in *Proceedings of the 16th ACM international conference on predictive models and data analytics in software engineering*. 2020.
825. Batchu, R.K. and H. Seetha, *A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning.* Computer Networks, 2021. **200**: p. 108498.
826. Mamidanna, S.K., C. Reddy, and A. Gujju. *Detecting an insider threat and analysis of XGBoost using hyperparameter tuning*. in *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. 2022. IEEE.
827. Jiyad, Z.M., et al. *DDoS Attack Classification Leveraging Data Balancing and Hyperparameter Tuning Approach Using Ensemble Machine Learning with XAI*. in *2024 Third International Conference on Power, Control and Computing Technologies (ICPC2T)*. 2024. IEEE.
828. Mokbal, F.M.M., et al., *XGBXSS: an extreme gradient boosting detection framework for cross-site scripting attacks based on hybrid feature selection approach and parameters optimization.* Journal of Information Security and Applications, 2021. **58**: p. 102813.
829. Ghatasheh, N., I. Altaharwa, and K. Aldebei, *Modified genetic algorithm for feature selection and hyper parameter optimization: case of XGBoost in spam prediction.* IEEE Access, 2022. **10**: p. 84365-84383.
830. Mane, S., M. Kulkarni, and S. Gupta, *Hyperparameter Optimization for Indoor Localization in Wi-Fi IoT Application.* Wireless Personal Communications, 2024: p. 1-29.
831. Ali, Y.A., et al., *Hyperparameter search for machine learning algorithms for optimizing the computational complexity.* Processes, 2023. **11**(2): p. 349.
832. Anggoro, D.A. and S.S. Mukti, *Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure.* International Journal of Intelligent Engineering & Systems, 2021. **14**(6).
833. Metz, L., et al., *Using a thousand optimization tasks to learn hyperparameter search strategies.* arXiv preprint arXiv:2002.11887, 2020.
834. Wu, J., et al., *Hyperparameter optimization for machine learning models based on Bayesian optimization.* Journal of Electronic Science and Technology, 2019. **17**(1): p. 26-40.
835. Wang, J., J. Xu, and X. Wang, *Combination of hyperband and Bayesian optimization for hyperparameter optimization in deep learning.* arXiv preprint arXiv:1801.01596, 2018.

836. Catillo, M., A. Pecchia, and U. Villano, *Successful intrusion detection with a single deep autoencoder: theory and practice.* Software Quality Journal, 2024. **32**(1): p. 95-123.
837. Xu, W., J. He, and Y. Shu. *DeepHealth: Deep representation learning with autoencoders for healthcare prediction*. in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2020. IEEE.
838. Alfeo, A.L., et al., *Using an autoencoder in the design of an anomaly detector for smart manufacturing.* Pattern Recognition Letters, 2020. **136**: p. 272-278.
839. Sattarov, T., D. Herurkar, and J. Hees, *Explaining anomalies using denoising autoencoders for financial tabular data.* arXiv preprint arXiv:2209.10658, 2022.
840. Ferraro, A., V. La Gatta, and M. Postiglione. *Empowering Network Security with Autoencoders*. in *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*. 2024. IEEE.
841. Mena, P., R. Borrelli, and L. Kerby, *Detecting Anomalies in Simulated Nuclear Data Using Autoencoders.* Nuclear Technology, 2024. **210**(1): p. 112-125.
842. Ikotun, A.M., et al., *K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data.* Information Sciences, 2023. **622**: p. 178-210.
843. Baldi, P. *Autoencoders, unsupervised learning, and deep architectures*. in *Proceedings of ICML workshop on unsupervised and transfer learning*. 2012. JMLR Workshop and Conference Proceedings.
844. Zhou, C., Y. Jia, and M. Motani, *Optimizing autoencoders for learning deep representations from health data.* IEEE journal of biomedical and health informatics, 2018. **23**(1): p. 103-111.
845. Tschannen, M., O. Bachem, and M. Lucic, *Recent advances in autoencoder-based representation learning.* arXiv preprint arXiv:1812.05069, 2018.
846. Hwang, W.S., J.G. Shon, and J.S. Park, *Web session hijacking defense technique using user information.* Human-centric Computing and Information Sciences, 2022. **12**: p. 16.
847. Mirian, A., et al. *Hack for hire: Exploring the emerging market for account hijacking*. in *The World Wide Web Conference*. 2019.
848. Baitha, A.K. and S. Vinod, *Session hijacking and prevention technique.* Int. J. Eng. Technol, 2018. **7**(2.6): p. 193-198.
849. Haider, W., *Developing reliable anomaly detection system for critical hosts: a proactive defense paradigm*. 2018, UNSW Sydney.
850. Xu, W., et al., *Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset.* IEEE Access, 2021. **9**: p. 140136-140146.
851. Said Elsayed, M., et al. *Network anomaly detection using LSTM based autoencoder*. in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. 2020.
852. Delibasoglu, I. and F. Heintz. *Time Series Anomaly Detection Leveraging MSE Feedback with AutoEncoder and RNN*. in *31st International Symposium on Temporal Representation and Reasoning (TIME 2024)*. 2024. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
853. Zhang, C., et al., *Autoencoder in autoencoder networks.* IEEE transactions on neural networks and learning systems, 2022. **35**(2): p. 2263-2275.
854. Kim, J.-C. and K. Chung, *Multi-modal stacked denoising autoencoder for handling missing data in healthcare big data.* IEEE access, 2020. **8**: p. 104933-104943.
855. Aslan, Ö., et al., *A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions.* Electronics, 2023. **12**(6): p. 1333.

856. Maleki, S., S. Maleki, and N.R. Jennings, *Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering.* Applied Soft Computing, 2021. **108**: p. 107443.
857. Takahashi, H., et al., *Deep Positive-Unlabeled Anomaly Detection for Contaminated Unlabeled Data.* arXiv preprint arXiv:2405.18929, 2024.
858. Cheng, Z., et al., *Improved autoencoder for unsupervised anomaly detection.* International Journal of Intelligent Systems, 2021. **36**(12): p. 7103-7125.
859. Yokkampon, U., et al., *Robust unsupervised anomaly detection with variational autoencoder in multivariate time series data.* IEEE Access, 2022. **10**: p. 57835-57849.
860. Mehmood, I., et al., *Managing data diversity on the internet of medical things (IoMT).* International Journal of Information Technology and Computer Science, 2020. **12**(6): p. 49-56.
861. Manimurugan, S., et al., *Two-stage classification model for the prediction of heart disease using IoMT and artificial intelligence.* Sensors, 2022. **22**(2): p. 476.
862. Ordway-West, E., P. Parveen, and A. Henslee. *Autoencoder evaluation and hyper-parameter tuning in an unsupervised setting*. in *2018 IEEE international congress on big data (BigData congress)*. 2018. IEEE.
863. Wu, X., et al., *A hybrid classification autoencoder for semi-supervised fault diagnosis in rotating machinery.* Mechanical Systems and Signal Processing, 2021. **149**: p. 107327.
864. Nayak, D.R., et al., *A deep autoencoder approach for detection of brain tumor images.* Computers and Electrical Engineering, 2022. **102**: p. 108238.
865. Pathirage, C.S.N., et al., *Structural damage identification based on autoencoder neural networks and deep learning.* Engineering structures, 2018. **172**: p. 13-28.
866. Yaser, A.L., H.M. Mousa, and M. Hussein, *Improved DDoS detection utilizing deep neural networks and feedforward neural networks as autoencoder.* Future Internet, 2022. **14**(8): p. 240.
867. Ko, J.U., et al., *A new auto-encoder-based dynamic threshold to reduce false alarm rate for anomaly detection of steam turbines.* Expert Systems with Applications, 2022. **189**: p. 116094.
868. Li, Z., W. Lin, and Y. Zhang. *Real-time drive-by bridge damage detection using deep auto-encoder*. in *Structures*. 2023. Elsevier.
869. El Zein, M., et al. *A deep learning framework for denoising mri images using autoencoders*. in *2023 5th international conference on bio-engineering for smart technologies (BioSMART)*. 2023. IEEE.
870. Abbas, M.F.B. and T. Srikanthan. *Low-complexity signature-based malware detection for IoT devices*. in *Applications and Techniques in Information Security: 8th International Conference, ATIS 2017, Auckland, New Zealand, July 6–7, 2017, Proceedings*. 2017. Springer.
871. Bahador, M.B., M. Abadi, and A. Tajoddin, *HLMD: a signature-based approach to hardware-level behavioral malware detection and classification.* The Journal of Supercomputing, 2019. **75**: p. 5551-5582.
872. Agoramoorthy, M., et al. *An Analysis of Signature-Based Components in Hybrid Intrusion Detection Systems*. in *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*. 2023. IEEE.
873. Schrötter, M., A. Niemann, and B. Schnor, *A Comparison of Neural-Network-Based Intrusion Detection against Signature-Based Detection in IoT Networks.* Information, 2024. **15**(3): p. 164.

874. Kaur, S. and M. Singh, *Hybrid intrusion detection and signature generation using deep recurrent neural networks.* Neural Computing and Applications, 2020. **32**(12): p. 7859-7877.
875. Mendonça, R.V., et al., *Intrusion detection system based on fast hierarchical deep convolutional neural network.* IEEE Access, 2021. **9**: p. 61024-61034.
876. Kumari, P. and A.K. Jain, *Timely detection of DDoS attacks in IoT with dimensionality reduction.* Cluster Computing, 2024: p. 1-19.
877. Aljarboua, E.F., M.B.M. Din, and A.A. Bakar. *Cyber-Crime Detection: Experimental Techniques Comparison Analysis*. in *2022 International Visualization, Informatics and Technology Conference (IVIT)*. 2022. IEEE.
878. Barkved, K., *How To Know if Your Machine Learning Model Has Good Performance.* 2022.
879. Kasongo, S.M. and Y. Sun, *Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset.* Journal of Big Data, 2020. **7**(1): p. 105.
880. Nguyen, S.-N., et al. *Design and implementation of intrusion detection system using convolutional neural network for DoS detection*. in *Proceedings of the 2nd international conference on machine learning and soft computing*. 2018.
881. Alalwany, E. and I. Mahgoub, *Classification of normal and malicious traffic based on an ensemble of machine learning for a vehicle can-network.* Sensors, 2022. **22**(23): p. 9195.
882. Nadeem, M.W., et al., *DDoS Detection in SDN using Machine Learning Techniques.* Computers, Materials & Continua, 2022. **71**(1).
883. Singhal, P., et al., *Domain adaptation: challenges, methods, datasets, and applications.* IEEE access, 2023. **11**: p. 6973-7020.
884. Chui, K.T., et al., *Multiround transfer learning and modified generative adversarial network for lung cancer detection.* International Journal of Intelligent Systems, 2023. **2023**(1): p. 6376275.