INVESTIGATING THE SECURITY ISSUES OF IOT DEVICES USING MACHINE LEARNING TECHNIQUES

THESIS
submitted for the degree of
Doctor of Philosophy

by

Badeea Mahmoud AL Sukhni

School of Engineering, Technology and Design
Canterbury Christ Church University

October, 2024

Declaration

I, Badeea Al Sukhni, declare that:

- The work presented in this thesis, titled "Investigating the Security Issues of IoT Devices Using Machine Learning Techniques" is my own and embodies the results of my research during my period of registration.

- I have read and followed the University's Academic Integrity Policy and that the thesis does not breach copyright or other intellectual property rights of a third party. Where necessary I have gained permission to reproduce copyright materials.

- Any material which has been previously presented and accepted for the award of an academic qualification at this University or elsewhere is clearly identified in the thesis. • Where work is the product of collaboration the extent of the collaboration has been indicated.

Signed:

Dated: 9 October 2024

ACKNOWLEDGEMENTS

Starting with thanking God for providing me with the strength, wisdom, and resources to complete this thesis, I express my sincere gratitude. This accomplishment would not have been possible without His grace and mercy.

First, I would like to extend my gratitude to myself for all the hard work— the time, the pressure, the motivation, the ambition, and the vision that have come together to achieve this dissertation. Most importantly, I deeply appreciate the determination, courage, and patience that I have demonstrated throughout this journey.

I am deeply grateful to my first supervisor, Dr. Soumya Manna, my second supervisor and the chair, Dr. Leishi Zhang, and Dr. Jugal Dave as an external advisor. Your unwavering support, guidance, and encouragement throughout my research have been invaluable. Your expertise and mentorship have played an invaluable role in shaping my academic journey.

My heartfelt thanks go to the ETD School at CCCU for offering me the PhD scholarship and to its staff for their endless support. I would like to particularly thank Prof. Anne Nortcliffe (now at Wrexham University), Prof. Abdullahi Ahmed, Dr. Hany Hassanin, Dr. Adil Imam, Mr. Tim Jackson, Dr. Usman Abdullah, Dr. Atif Rasheed and Mrs. Sana Rahman. I would also like to extend my regards to the PhD students at CCCU—Merlin Kasirajan, Kiko Li, Mohammed Al-Alawi, Adedayo Olowolayemo, and Francis Okeke—with whom I shared many deep and often funny discussions, as well as experiences throughout our research journeys.

I would like to thank my beloved husband, Anas AL Kayed, who has been my source of strength throughout my journey. Your support—whether reminding me of tasks or standing by my side through every challenge—has been truly invaluable. Your confidence in my abilities has continually pushed me forward, and I am forever grateful for your constant presence.

I also deeply appreciate my family members and friends, specifically—Abdullah Namroqa, Bothaina Alsukhni, Naya and Tala Namroqa, Safa Jrwan, Noura Ibrahim, Majd Abuawad, Rania Kolaghassi, Natalie Aguilar, Zahra AlHumaidi, Ahed Abouda, Hala Zorkot, and Assef Hussein—

for their endless love, encouragement, and support. Your belief in me has been a continuous source of motivation, guiding me through the uncertainties of this journey.

I would like to express my deepest thanks to my parents, Mahmoud AL Sukhni and Khawla Obeidat, for everything you have provided me throughout my life. I am forever grateful for your faith in me, your investment in my education in Jordan and the UK, your constant support, encouragement, guidance, your presence whenever I needed you, and your unconditional love, and much more. Every accomplishment I have made or will achieve is because of you.

To my parents, Mahmoud Al Sukhni and Khawla Obeidat, who have dedicated their lives to us; my husband, Anas Al Kayed; my brothers, sisters, brothers-in-law, sisters-in-law, and my beloved nieces and nephews—you mean the world to me, and I dedicate this to you.

﴿ وَأَنْ لَيْسَ لِلْإِنسَانِ إِلَّا مَا سَعَى ۝ وَأَنَّ سَعْيَهُ سَوْفَ يُرَى ﴾

﴿And that there is not for human except that for which he strives ۝ And that his effort is going to be seen﴾

(Verses 39-40, Surah An-Najm, (The Qur'an))

- *Chosen with love by my parents ♡.*

ABSTRACT

The integration of the Internet of Things (IoT) across various sectors has notably increased vulnerability to sophisticated multilayer attacks, compromising multiple security layers and leading to significant breaches, including data loss, personal information theft, and financial losses. The existing research on multilayer IoT attacks faces gaps in real-world applicability due to reliance on outdated datasets and limited focus on adaptive, dynamic approaches to address multilayer vulnerabilities. Additionally, the complete reliance on automated processes without integrating human expertise in feature selection and weighting processes may affect the reliability of detection models. This thesis proposes a novel Semi-Automated Intrusion Detection System (SAIDS), integrating efficient feature selection, feature weighting, normalisation, visualisation, and human-machine interaction to enhance the detection and identification of multilayer attacks, thereby improving mitigation strategies.

This research contributes significantly to IoT security by highlighting the SAIDS framework's ability to efficiently detect and classify multilayer attacks in machine learning models optimising the computational process and extracting most significant features extracted out of dataset. By incorporating human expertise into the optimised feature analysis process, the proposed system enhances the reliability of detection models through binary (attack/no-attack) and multiclass classifications (UDP, ICMP, HTTP flood, MITM, TCP SYN, XSS, SQL injection, and Password cracking), thereby showing a potential for developing a robust foundation for future research in dynamic and adaptive security measures for IoT environments. These findings not only validate the practical applicability of SAIDS in real-world scenarios but also propose a standard framework for future IoT security enhancements using machine learning methods.

The SAIDS framework was evaluated using the Edge-IIoTset dataset, a recent IoT dataset. Additionally, it was evaluated on a dataset collected from the Cooja simulation platform running on the Contiki Operating System for simulated UDP flood attacks, as well as on real IoT devices, specifically an ARP poisoning attack on the Xiaomi Redmi Note 9S. Through this evaluation, the framework identified 13 significant features from the Edge-IIoTset dataset and seven significant features from the simulated environment dataset for the detection and classification of IoT multilayer attacks.

The research employs various machine learning models, with a focus on K-Nearest Neighbours (KNN), which outperformed other classifiers in terms of accuracy, precision, recall, and F1-score in binary classification and multiclass classification. It achieved a high accuracy rate of 99% in detecting normal traffic, TCP SYN, and ICMP flood, 97% in XSS, and 94% in HTTP flood, SQL injection, and password cracking attacks.

RESEARCH OUTPUTS FROM THIS THESIS

Journal Publications

- AL Sukhni, B. A., Manna, S. K., Dave, J. M., and Zhang, L., Extracting optimal number of features for Machine Learning models in Multilayer IoT Attacks'. Submitted to MDPI,Sensors.

- AL Sukhni, B. A., Manna, S. K., Dave, J. M., and Zhang, L., 'Evaluation of the Semi-Automated Intrusion Detection System (SAIDS) Against Multilayer IoT Attacks in Simulated and Real-World Environments'. Submitted to IEEE Security & Privacy.

Conference Publications

- AL Sukhni, B. A., Manna, S. K., Dave, J. M., and Zhang, L, 'Exploring Optimal Set of Features in Machine Learning for Improving IoT Multilayer Security', 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), Aveiro, Portugal, 2023, pp. 1-6. IEEE.

- AL Sukhni, B., Dave, J.M., Manna, S.K. and Zhang, L, 'Investigating the security issues of multi-layer IoT attacks using machine learning techniques', 2022 Human-Centered Cognitive Systems (HCCS), Shanghai, China, 2022, pp. 1-9. IEEE.

Book Chapter Publications

- AL Sukhni, B. A., Manna, S. K., Dave, J. M., and Zhang, L, 'Machine learning-based solutions for securing IoT systems against multilayer attacks', Communications in Computer and Information Science, pp. 140–153. Springer.

Poster Presentations

- AL Sukhni, B. A., Manna, S. K., Dave, J. M., and Zhang, L.,' Safeguarding IoMT: Semi-automated Intrusion Detection System (SAIDS) for Detecting Multilayer Attacks', Early career researchers session at the UK Government Security and Policing 2024 Exhibition,

in Farnborough International Exhibition and Conference Centre. March 2024. Available at: https://researchspace.canterbury.ac.uk/975y7/safeguarding-iomt-semi-automated.

- AL Sukhni, B., Manna, S., Dave, J. and Zhang, L., 'Investigating the security issues of multi-layer IoMT attacks using machine learning techniques'. (Poster presentation), Exploring Research and Development in the MedTech, Life Science and Healthcare sectors, Maidstone Innovation Centre, 9 Nov 2022. Available at: https://repository.canterbury.ac.uk/item/9315y/investigating-the-security-issues-of-multi-layer-iomt-attacks-using-machine-learning-techniques.

Conference Presentations

- AL Sukhni, B. A., Manna, S. K., Dave, J. M., and Zhang, L., 'Investigating Security Issues (Multilayer Attacks) on IoT Devices Using Machine Learning', Faculty Conference. CCCU, Canterbury, Feb 2024. https://researchspace.canterbury.ac.uk/9769w/investigating-security-issues-mu.

Awards

- First place winner at CCCU 3MT Competition.

Contents

ILLUSTRATIONS

Tables

ABBREVIATIONS

The following abbreviations are used in this:

ANN	Artificial Neural networks
ARP	Address Resolution Protocol
ASC	Attribute Selected Classifier
AUC	Area Under the Curve
BN	Bayesian Network
CART	Classification and Regression Tree
Chi²	Chi-Square
CML	Continuous Machine Learning
CNN	Convolutional Neural Network
CUMA	Combined UDPFlood and MITM Attacks
DAE	Deep Auto-Encoder
DDoS	Distributed Denial of Service Attack
DL	Deep Learning
DNN	Deep Neural Network
DNS	Domain Name System
DoS	Denial of Service Attack
DT	Decision Tree
DTE	Decision Tree Entropy
ECTFM	Ensemble Classification Utilising Traffic Flow Metrics
EHMS	Enhanced Healthcare Monitoring System
ELM	Extreme Learning Machine
F1	F1-score
FA	Firefly Algorithm
FCBF	Fast-Based-Correlation Feature
FF	Firefly Algorithm
GDPR	General Data Protection Regulation

GR	Gain Ratio
GRU	Gated Recurrent Unit
GWO	Gray Wolf Optimisation
H2H	Human-to-Human
H2M	Human-to-Machine
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDC	International Data Corporation
IDS	Intrusion Detection System
IG	Information Gain
IICSs	Internet Industrial Control Systems
IoMT	Internet of Medical Things
IoT	Internet of Things
IP	Internet Protocol
KNN	K-Nearest Neighbors
LEDEM	Learning-Driven Detection Mitigation
LMT	Logistic Model Trees
LR	Logistic Regression
LSTM	Long Short-Term Memory
M2M	Machine-to-Machine
MAGRU	Multi-Head Attention-Based Gated Recurrent Unit
MANET	Mobile Ad Hoc Networks
MBTCP	Modbus TCP
MI	Mutual Information
MITM	Man-In-The-Middle
ML	Machine Learning
MLP	Multi-Layer Perception
MSM	Model Selection Method
MQTT	Message Queue Telemetry Transport
NB	Naïve Bayes
PCA	Principal Component Analysis

Pr	Precision
PSO	Particle Swarm Optimisation
QDA	Quadratic discriminant analysis
R2L	Remote-to-Local
Rc	Recall
RF	Random Forest
RFID	Radio Frequency Identification
RNN	Recurrent Neural Network
SA	Statistical Aggregation
SAIDS	Semi-Automated Intrusion Detection System
SDELM	Semi-supervised Deep Extreme Learning Machine
SDN	Software-Defined Networking
SMO	Sequential Minimal Optimisation
SMOTE	Synthetic Minority Oversampling Technique
SQL	Structured query language
SVM	Support Vector Machine
TCP	Transmission Control Protocol
U2R	User-to-Root
UDP	User datagram protocol
VANETs	Vehicular Ad Hoc Networks
WSN	Wireless Sensor Network
XGBoost	Extreme Gradient Boosting
XSS	Cross-Site Scripting

1 INTRODUCTION TO IOT SECURITY AND MULTILAYER ATTACKS

1.1 OVERVIEW OF IOT SECURITY ATTACKS

The Internet of Things connects physical objects (sensors and actuators) in our environment to the internet, allowing them to communicate with cloud-based applications, and with users. Each object is assigned a unique IP address, enabling communication through both wired and wireless networks. Consequently, IoT facilitates a myriad of applications that enhance various aspects of life including fitness, leisure, transportation, energy, education, and healthcare—through affordable devices such as smart meters, wearables, and smartphones. According to the International Data Corporation (IDC), the number of IoT devices will reach 55.7 billion by 2025 (Hojlo, 2021). This exponential increase in IoT devices is due to the ubiquitous connectivity and decision-making capabilities of IoT devices (Anthi et al., 2019).

Researchers have categorised IoT architecture into several layers: three-layer, four-layer, and five-layer architectures. Butun, Osterberg and Song, (2020) defined a five-layer architecture including the physical, MAC, network, transport, and application layers. Malhotra et al. (2021); Hassija et al. (2019) proposed four-layer architectures. Specifically, Malhotra et al. (2021) included the perception, network, support, and application layers, while Hassija et al. (2019) included the sensing, network, middleware, and application layers. Additionally, researchers such as (Khanam et al. (2020); Tahsien, Karimipour and Spachos (2020); Al-Garadi et al. (2020) described a three-layer IoT architecture consisting of the perception/physical, network, and application layers. This research focuses on the three-layer architecture, consisting of the physical layer, the network layer, and the application layer, which is the standard architecture for IoT systems and the most widely cited one. The three-layer IoT architecture is illustrated in Figure 1.1.

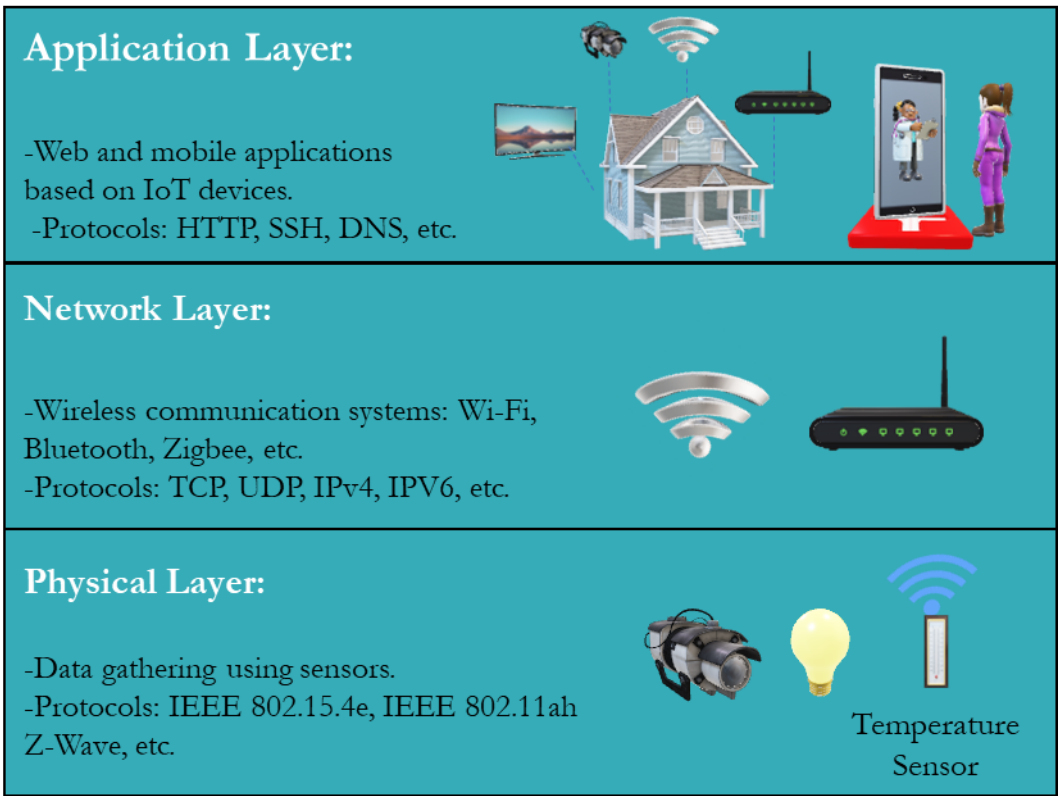


Figure 1.1. The IoT system's three-layer architecture (Tahsien, Karimipour and Spachos, 2020).

A. Physical Layer

The physical layer is the foundational layer of the IoT architecture. Its primary role is to sense, gather, and process information from the surrounding physical environment (Atlam and Wills, 2019). This layer is crucial as it generates a significant volume of IoT big data. Effective analysis of this data can facilitate the development of a context-aware IoT system. The physical layer utilises various types of sensors to monitor their surroundings, actuators to perform autonomous actions, Radio Frequency Identifications (RFIDs) to identify, track, and monitor IoT devices, and wireless sensor networks (WSNs) to provide essential sensing and communication services (Atlam and Wills, 2019).

IoT devices are often resource-constrained, they have limited memory space, and low computational capability, and processing capacity. As a result, to transmit the data obtained by the sensors, physical layer communication technologies with low energy consumption are required.

Examples of such communication protocols include IEEE 802.11ah, IEEE 802.15.4e, and Z-Wave (Tahsien, Karimipour and Spachos, 2020).

B. Network Layer

This IoT layer serves as the backbone for connecting and communicating IoT devices over the internet, where the centralised server is located. It is mainly utilised to transfer data and information between the physical and application layers using various technologies and protocols, including ZigBee, GSM, 2G,3G,4G,5G, LTE, Bluetooth, Wi-Fi, IPV4, IPV6, etc. Additionally, local clouds and servers in this layer store and process data, acting as middleware between the network and application layers (Tahsien, Karimipour and Spachos, 2020).

C. Application Layer

The application layer is the third layer in an IoT system, comprising several mobile and web applications that provide various services like smart home control, industrial automation, and healthcare monitoring. It serves as the point where users interact with the IoT system, making it easy for them to use these services. Common protocols used in this layer include HTTP, MQTT, and others, which help devices and applications communicate with each other and share information (Tahsien, Karimipour and Spachos, 2020).

A study by McKinsey highlights the economic impact of IoT, stating that by 2025, IoT could contribute between £155 billion and £270 billion per year to the global economy (Manyika et al., 2015). However, despite this significant contribution, the widespread adoption of IoT devices introduces critical challenges, particularly in terms of security and privacy. For instance, researchers from Kaspersky claim that in the first six months of 2021, there were 1.5 billion attacks on IoT devices that were vulnerable, compared to the 639 million in the previous six months.

As shown in Figure 1.2, which illustrates the likelihood of cyber security breaches experienced by UK businesses from 2017 to 2021, these incidents were not exclusively related to IoT. For large businesses, the rate of security breaches has fluctuated, beginning at 46% in 2017, decreasing to a low of 32% in 2019, then increasing again to 46% in 2020, and finally slightly decreasing to 39% in 2021. In contrast, the rate of security breaches for small businesses has dramatically increased

from 19% in 2018 to 26% by the end of 2021 (Department for Digital, Culture, Media & Sport, 2021). This highlights the need to increase the awareness of cyber security issues, the implementation of the General Data Protection Regulation (GDPR), and the adoption of more robust security measures for detecting and preventing cyberattacks.

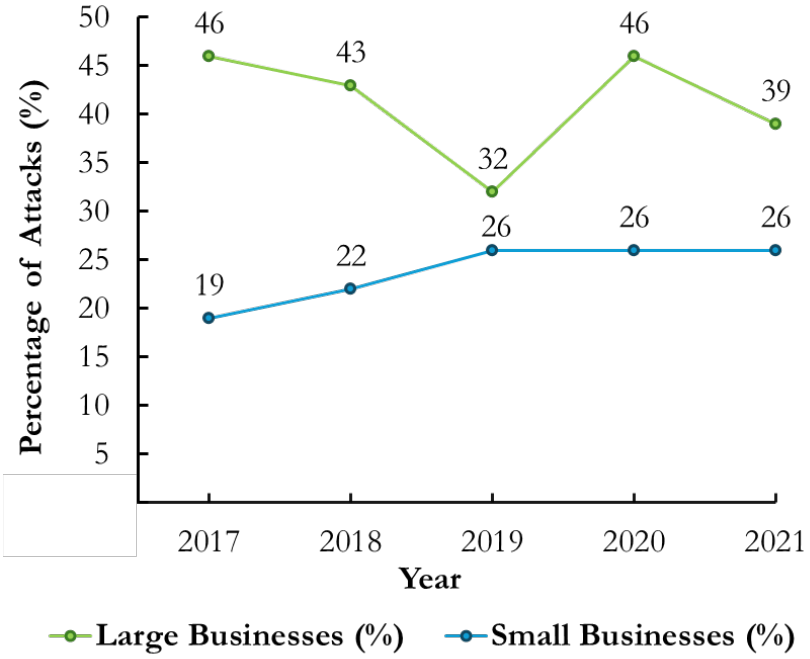


Figure 1.2. The increase of cyber-attacks in UK businesses over time (Department for Digital, Culture, Media & Sport, 2021).

Several factors make the IoT ecosystem a larger attack surface, rendering it an attractive target for cyber-attacks. The interconnected nature of IoT devices, as shown in Figure 1.3, means that compromising one device can potentially allow attackers to penetrate entire networks, leading to widespread security breaches. Moreover, the diverse nature of the IoT ecosystem, with devices from different manufacturers operating on varying standards and protocols, complicates the implementation of uniform security measures (Malan et al., 2020).

The IoT system relies on sensors installed in the surrounding environment that capture a variety of data, including users' behaviours, bank records, and other sensitive information, in addition to environmental data.

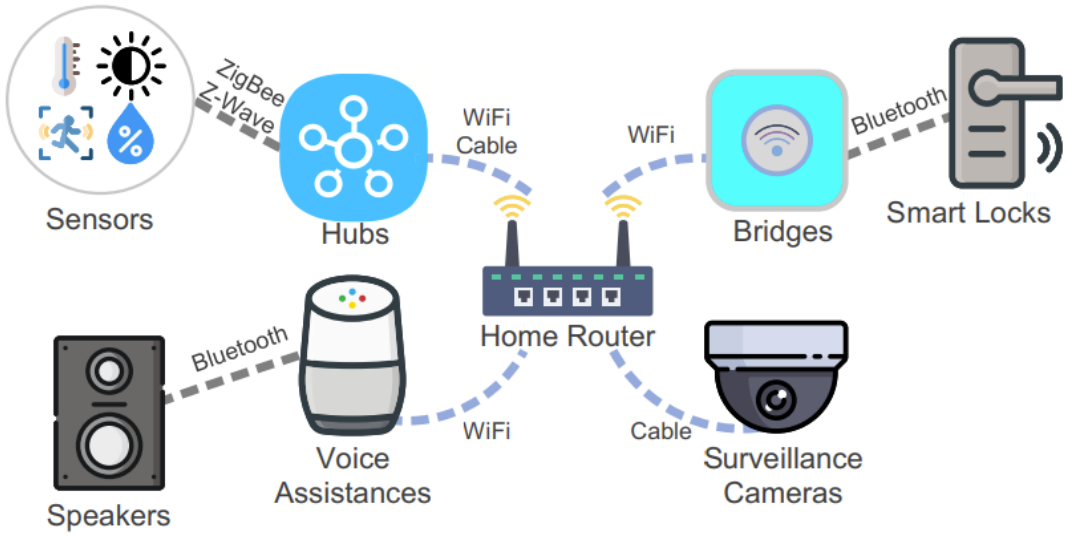


Figure 1.3. An example of interconnected IoT devices (Wan et al., 2020).

Furthermore, the resource and computational power limitations of IoT devices often make the implementation of robust security controls infeasible. Many IoT devices are designed with minimal security features, frequently lacking robust authentication and data encryption methods, prioritise convenience and cost over security (Malan et al., 2020). Additionally, estimations by the National Cyber Security Centre, UK, suggest that around 98% of IoT traffic is unencrypted (National Cyber Security Centre, 2024).

Several cyber-attacks have demonstrated the potential consequences of IoT security vulnerabilities. For instance, the Mirai botnet attack in 2016 exploited weak default passwords in IoT devices to create a massive botnet that launched disruptive Distributed Denial of Service (DDoS) attacks, leaving the US East Coast without internet. Similarly, IoT cyber-attacks launched on UK banks like Lloyds and Royal Bank of Scotland attempted to cause significant service disruptions by blocking access to 20 million UK accounts (Department for Science Innovation and Technology, 2024).

In May 2021, the Colonial Pipeline, a major fuel pipeline operator in the U.S., was targeted by the DarkSide ransomware group, which primarily used brute force password attacks, leading to significant operational disruptions. In response, Colonial Pipeline paid $4.4 million as ransom to regain control of their systems. Subsequently, the U.S. Department of Justice seized $2.3 million

from the DarkSide group at the time of recovery (Department of Justice, 2021; Beerman et al., 2023).

IoT-connected toys are prime targets for attacks that breach the privacy and safety of children. These devices often include cameras and microphones that can be exploited for surveillance, misuse, or unauthorised communication with children. Fitness devices collect highly personal data, such as health records and GPS locations, which are highly attractive to cybercriminals. Attacks targeting fitness devices aim to compromise personal information and even allow unauthorised tracking of individuals (Burton et al., 2021) .

1.2 TAXONOMY OF MULTILAYER IOT ATTACKS

On occasions of multilayer attacks, IoT devices can be affected from the sensor level to the cloud level, sometimes simultaneously, making it difficult to monitor and track due to the interconnectivity between layers. As mentioned, multilayer attacks can be harmful to IoT devices as they aim to exploit more than one layer of IoT architecture. For example, DDoS/DoS, MITM, cryptanalysis, eavesdropping and side channel attacks target the Machine-to-Machine (M2M), network, and cloud layers of the IoT system. Such attacks can lead to serious problems, such as losing control over important data and damaging the reputation and finances of those involved.

According to the IBM Security X-Force report for 2022, 74% of IoT attacks are caused by Mozi botnets launching MITM attacks (IBM Security, 2022). For example, according to the National Cyber Security Centre, UK, in 2020, a Russian hacking group leaked documents about a project aiming to create an IoT botnet inspired by the Mirai botnet, targeting security cameras and network video recorders to perform password attacks and grow the botnet. This botnet, once large enough, could launch powerful DDoS attacks, illustrating the significant threat of IoT vulnerabilities being exploited by both state and non-state actors (National Cyber Security Centre, 2022). Additionally, in 2014, hackers exploited a vulnerability in a Samsung smart refrigerator that enabled them to implement a MITM attack and steal users' Gmail credentials (Park, Chung and DeFranco, 2022). Moreover, according to the OWASP IoT Top 10 for 2018, the top vulnerability was weak, guessable, or hardcoded passwords, which make these devices easy targets for brute force attacks,

and the third vulnerability involved insecure ecosystem interfaces that result in implementation injection attacks such as Cross-Site Scripting (XSS) (OWASP, 2019).

Since IoT systems are vulnerable to different types of attacks, several studies have been carried out to identify IoT attacks targeting each layer of the IoT architecture and discuss the possible security solutions to tackle these attacks. The surveys by Malhotra et al. (2021); Hassija et al. (2019); Tahsien, Karimipour and Spachos (2020) identified attacks targeting the IoT's four-layered architecture. Malhotra et al. (2021) highlighted that the physical layer consists of Eavesdropping, Jamming, Relay, Node Capture and Cloning attacks. The network layer is susceptible to MITM, routing, DDoS, and Sybil attacks. In the support layer, DoS and malicious insider attacks are common. The application layer is vulnerable to DoS, Phishing, Malicious Code Injection, and Session Hijacking attacks.

Tahsien, Karimipour and Spachos (2020) presented a detailed list of these attacks, including several types of active and passive attacks. The physical device/perception surface includes DoS, Eavesdropping, Jamming, Node Capture, Physical Attacks, and others. The network/transport surface is commonly targeted by DoS, Sybil, MITM, Eavesdropping, and Spoofing attacks, among others. At the cloud services surface, DoS, Session Hijacking, and Malicious and Insider Attacks are common at this layer. The web/application surface is vulnerable to DoS, Eavesdropping, Malicious Node attacks, and more. The survey by Hassija et al. (2019) divided the attacks into four layers: sensing layer, network layer, middleware layer, and application layer. Additionally, they discussed security concerns with the gateways that connect these layers. Moreover, the study focuses on addressing DDoS, Eavesdropping, and Spoofing.

Studies by Ahmad, R. and Alsmadi (2021); Kumar and Sharma (2022); Atlam and Wills (2019) addressed cyber-attacks on the three-layer IoT system. In Ahmad, R. and Alsmadi, (2021), the physical layer consists of side-channel attack, physical damage, node jamming, eavesdropping, etc. The network layer attacks are divided into encryption attacks (MITM, caching, spoofing, session hijacking, packet manipulation, cryptanalysis and RFID cloning), DoS/DDoS attacks (packet flooding, battery draining, SYN flood, botnet, ping of death, etc), routing attacks (sybil, wormhole/sinkhole, forwarding and nmap/port attack), and middleware attacks (brute-force, dictionary attack, message replay, etc). The application layer attacks are divided into malware attacks

(virus, ransomware, spyware, etc), privacy attacks (spear phishing, phishing, social engineering, etc), and code attacks (SQL injection, XSS, malicious script, session hijacking, etc).

In Kumar and Sharma (2022) taxonomy, the perception layer includes Node Capture, Replay, Botnet, Mirai, Eavesdropping, and Side Channel attacks. The network layer is vulnerable to DoS/DDoS, MITM, Sybil, Sinkhole, Spoofing, and Data Transit attacks. The application layer includes Phishing, Malicious Code Injection, Sniffing, Trust Management, and Policy Enforcement attacks. Atlam and Wills (2019), added an encryption attacks category alongside the three layers, which includes of cryptanalysis, side channel, and MITM attacks.

Moreover, Khanam et al., (2020); Mitrokotsa, Rieback and Tanenbaum (2010) added a category called encryption or multilayer or dimensional attacks alongside the three-layer IoT attacks. The new attack category in Khanam et al. (2020) is called multilayer/dimensional attacks and consists of DoS, side channel, MITM, and cryptanalysis attacks. Furthermore, Mitrokotsa, Rieback and Tanenbaum (2010) focused on RFID security, including attacks such as covert channels, crypto, traffic analysis, side channel, replay, and DoS attacks.

The above-mentioned studies focused on security attacks that target a single IoT layer, with only few studies touching on the topic of multilayer attacks such as Khanam et al. (2020); Mitrokotsa, Rieback and Tanenbaum (2010). However, all of them are limited to one or a subset of multilayer attacks. This highlights the necessity to have secure measures to protect against different types of multilayer attacks and this can be achieved by understanding the comprehensive taxonomy of multilayer attacks.

Figure 1.4 provides a comprehensive taxonomy of multilayer attacks within IoT systems, distinguishing them from the single-layer attacks associated with the Physical, Network, and Application layers. It focuses on the nature of multilayer attacks that span across these layers and describes their behavioural patterns. The red, blue, and green cells in this figure show an overview of single-layer IoT attacks that have been reported in the above literature. The grey cells show a taxonomy of multilayer IoT attacks.

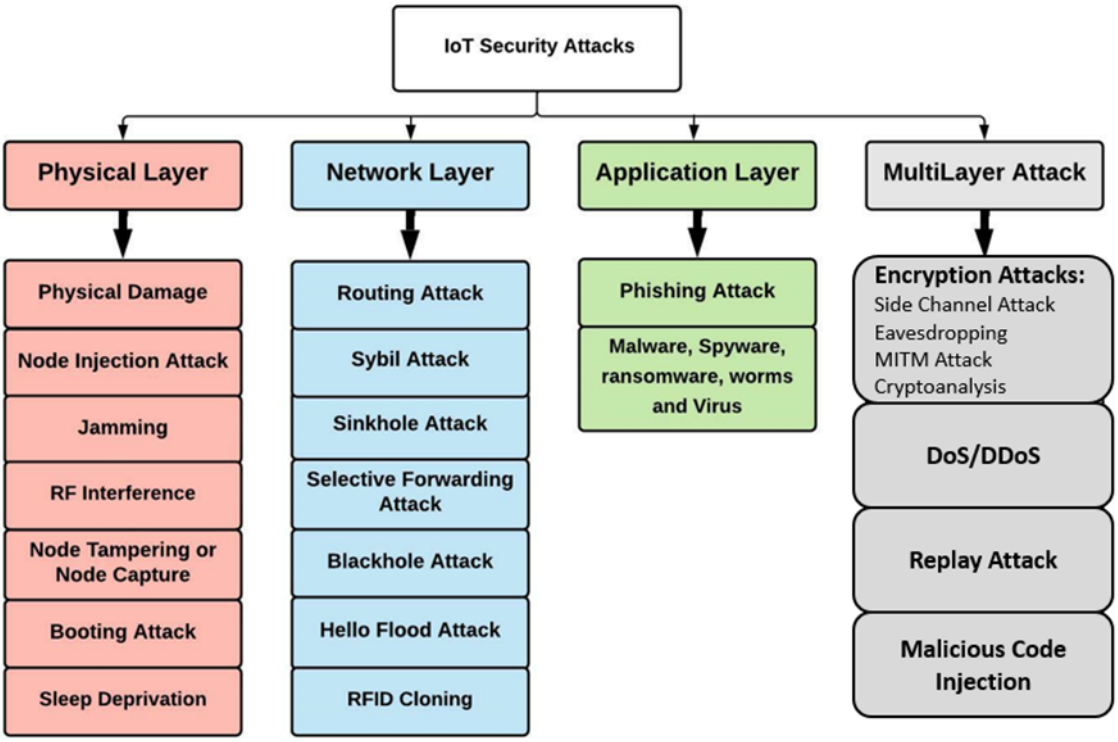


Figure 1.4. The three-layer IoT security attacks including multilayer attacks.

- **Physical Layer Attacks**

The IoT physical layer is vulnerable to a variety of security attacks. These attacks are described as follows:

Physical Damage: Adversaries can physically damage IoT devices, deactivating them and making IoT services inaccessible.

Node Injection Attack: This attack is considered a type of MITM attacks, where the intruder deploys a node between two legitimate IoT nodes to intercept the network traffic (Khanam et al., 2020; Atlam and Wills, 2019).

Jamming: This attack has significant consequences for IoT devices, as it can quickly drain device battery power by stopping data transfer and causing frequent retransmissions. It also has a negative impact on IoT networks by disrupting communication, draining energy resources, and reducing performance (Idrissi, Azizi and Moussaoui, 2020).

RF Interference: In this attack, an attacker uses an RFID tag to send noise signals across the radio frequencies used by RFIDs for communication. These noise signals interfere with RFID signals, reducing communication quality and affecting IoT sensor availability (Hassija et al., 2019; Tahsien, Karimipour and Spachos, 2020).

Node Tampering (or Node Capture): The attacker targets a sensor node, physically capturing and compromising it by attaching cables to its circuit board. As a result, the attacker may alter or remove sensitive information on the end node (Kumar and Sharma, 2022; Atlam and Wills, 2019).

Booting Attack: During the booting phase, IoT devices are vulnerable to various attacks, and attackers may target node devices while they are being restarted. This vulnerability exists because built-in security mechanisms are not enabled at that time (Hassija et al., 2019).

Sleep Deprivation: This is a type of DoS attack where the attacker keeps the nodes running continuously by sending an infinite number of seemingly legitimate requests. to This consumes the nodes' batteries, shortening their lifetime and leading to shutdown (Idrissi, Azizi and Moussaoui, 2020).

- **Network Layer Attacks**

As the backbone of the IoT system, the network layer is exposed to a range of security attacks, as detailed below:

Routing: This is a cyber-attack that effects the routing of messages. As a result, the attacker can alter, spoof, reroute, or drop packets at the network layer using such an attack (Mosenia and Jha, 2017).

Sybil Attack: In this attack, a single malicious node hides its original identity by claiming several identities and relocates throughout the network, causing the malicious node to remove the original nodes from the routing table (Tahsien, Karimipour and Spachos, 2020).

Sinkhole Attack: This is a routing attack in IoT systems where the attacker creates a flood of network traffic using fake routing information to disrupt connectivity (Kumar and Sharma, 2022).

Selective Forwarding Attack: This attack involves a malicious node selectively forwarding some packets while dropping others during transmission, thereby creating a gap in the IoT network (Tahsien, Karimipour and Spachos, 2020).

Blackhole Attack: The objective of this attack is to drop all normal packets encountered in the network. Initially, the intruder sends out malicious routing data to establish a direct path to the target. The sender then selects the malicious path for packet transmission and begins sending the packets after receiving a fake reply from the attacker. Then, the attacker begins to drop all packets routed through the malicious path.

Hello Flood Attack: In this attack, a malicious node transmits "HELLO PACKETS" to all other nodes in the network, claiming to be their neighbour. The attacker's goal is to exhaust the IoT network by sending a flood of route requests (Mosenia and Jha, 2017).

RFID Cloning: This is an attack on RFID tag where data is copied from one RFID tag to another. Although the two RFID tags contain identical data, the original RFID ID is not duplicated (Khanam et al., 2020).

- **Application Layer Attacks**

The most common IoT application layer attacks are listed below:

Phishing Attack: This is a type of social engineering attack that targets IoT users, using infected emails or phishing websites to obtain login credentials, such as passwords and credit card details.

Malware, Spyware, Ransomware, Worms and Viruses: The purpose of these attacks is to compromise the system's confidentiality. They are most commonly seen in the form of spam, Trojans, and other malware (Khanam et al., 2020).

- **MultiLayer Attacks**

Many attacks on IoT security are not limited to a single layer. Attacks that affect multiple layers, such as the physical, network, and application layer, are classified as multilayer attacks. These

attacks, including encryption attacks (side channel, eavesdropping, MITM, and cryptanalysis), DoS/DDoS, replay, and code injection attacks, are described in more detail below.

A. Encryption attacks

The intruder uses encryption attacks to compromise the three layers of the IoT system by breaking the encryption algorithms that safeguard the communication channels between IoT devices. Examples of such attacks include side channel, eavesdropping, cryptanalysis, and MITM attacks (Khanam et al., 2020; Deogirikar and Vidhate, 2017; Yang et al., 2017).

Side Channel Attack: In these attacks, the attackers target encryption devices in order to obtain encryption keys and steal sensitive information. Power consumption attacks, timing attacks, and electromagnetic attacks are examples of these attacks.

Eavesdropping: There are two ways to implement eavesdropping attacks whether in active or passive mode. In passive mode, the attacker listens to the data exchanged between two legitimate devices and gains the encryption key needed to decrypt confidential data, resulting in an invasion of users' privacy without their awareness. An example of active eavesdropping is the MITM attack, which will be discussed in the next point.

MITM Attack: This is an active attack, where the attacker acts as a router between two nodes exchanging sensitive information, allowing the attacker to capture the encryption key to decrypt and modify the data. As shown in Figure 1.5, a node injection attack is a type of MITM attack that compromises the IoT physical layer. Examples of MITM attacks that target the network layer include ARP poisoning, ICMP redirection, port stealing, DHCP spoofing. DNS spoofing and session hijacking target the IoT application layer.

Cryptanalysis: This attack is different from other encryption attacks, the attacker attempts to decrypt sensitive data without acquiring the encryption key. As shown in Figure 1.5, examples of cryptanalysis attacks include ciphertext-only attacks, known-plaintext attacks, chosen-plaintext attacks, and brute-force attacks such as FTP and SSH-patator.

B. Dos/DDoS Attack:

This attack involves overwhelming the target IoT device or network with a large amount of flood traffic by initiating a 3-way TCP handshake but not completing the final stage (ACK) to make the service unavailable for future requests, even from legitimate users. A botnet attack involves a group of IoT devices infected with malware, which can be used to launch DoS/DDoS attacks (Khanam et al., 2020). As shown in Figure 1.5, sleep deprivation and sensor data flooding are examples of DoS/DDoS attacks at the physical layer. Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP SYN), User Datagram Protocol (UDP) and Hello Floods compromise the network layer. Lastly, heartbleed, goldenEye, slowloris, slowhttptest, zero-day, Hypertext Transfer Protocol (HTTP), and Domain Name System (DNS), floods exploit the application layer (Ferrag et al., 2020).

C. Replay Attack:

An adversary compromises both the IoT physical and network layers using this attack to obtain sensitive information and mislead the receiving device. They do this by storing the transmitted data and rebroadcasting it later to one or more parties, which results in the exhaustion of system resources such as processors, batteries, and memory (Khanam et al., 2020; Kumar and Sharma, 2022).

D. Code Injection Attacks:

Cybercriminals target both the IoT physical and application layers with this attack. At the physical layer, the attacker physically injects malicious code, such as malware, into IoT nodes, forcing them to execute specific actions or even gain access to the IoT system (Atlam and Wills, 2019). As seen in Figure 1.5, SQL injection, XSS and malicious scripts are types of code injection attacks that can target the IoT application layer. For example, attackers may use XSS to inject malicious scripts into a trustworthy website, potentially compromising IoT applications. If an XSS attack is successful, it can lead to IoT account hijacking, revealing users' information and possibly crippling the IoT system (Malhotra et al., 2021).

- **Common Patterns in Multilayer Attacks**

There are common patterns in these multilayer attacks, as shown in Figure 1.5. For instance, eavesdropping, MITM and replay attacks are interception attacks targeting data confidentiality, where an unauthorised user gains access to sensitive data, IoT devices, and applications. In addition, DoS/DDoS attacks can be considered interruption attacks that compromise the availability of services by flooding the network with massive amount of traffic, causing disruption in IoT network operations. Furthermore, code injection attacks are fabrication attacks that target data integrity by creating illegitimate information within the IoT system. Cryptanalysis and side channel attacks exploit data confidentiality by compromising cryptographic algorithms to obtain sensitive information.

Moreover, these multilayer attacks share specific technical patterns across IoT layers. Attacks such as DoS/DDoS, replay, and sleep deprivation aim to exhaust system resources at different layers. For example, DoS/DDoS attacks flood the network and application layers with excessive requests, while sleep deprivation depletes battery life at the physical layer. These attacks lead to abnormal system behavior, such as high CPU/memory usage, excessive network traffic, or rapid battery depletion. Code injection attacks (e.g., node injection at the physical layer, SQL/XSS injection at the application layer) follow similar patterns by compromising IoT devices through the injection of malicious code into the system. Flooding attacks such as TCP SYN flood, ICMP flood, HTTP flood, UDP flood, Hello flood, and sensor data flooding overwhelm the IoT system with unnecessary requests or data. These attacks result in high traffic volumes, large packet sizes, or repeated communication attempts across various layers.

Attacks like MITM, eavesdropping, cryptanalysis, and side channel specifically target the cryptographic mechanisms that secure communication between IoT layers. These attacks exploit weaknesses in encryption processes, compromising confidentiality and integrity. Additionally, these multilayer attacks share common features in the protocols they exploit. For instance, HTTP flooding, session hijacking, and code injection all target vulnerabilities in HTTP traffic. Whether by overwhelming the server (HTTP flooding), intercepting session tokens (MITM), or injecting malicious input (code injection), they rely on manipulating or overloading HTTP requests.

Furthermore, attacks like TCP SYN flood, MITM, and replay share a common characteristic in their reliance on incomplete TCP handshakes. In SYN flooding, the attacker sends SYN packets without completing the connection, exhausting server resources. In MITM, the attacker intercepts the handshake to manipulate or eavesdrop on the session. Replay attacks manipulate session traffic by replaying or retransmitting TCP packets.

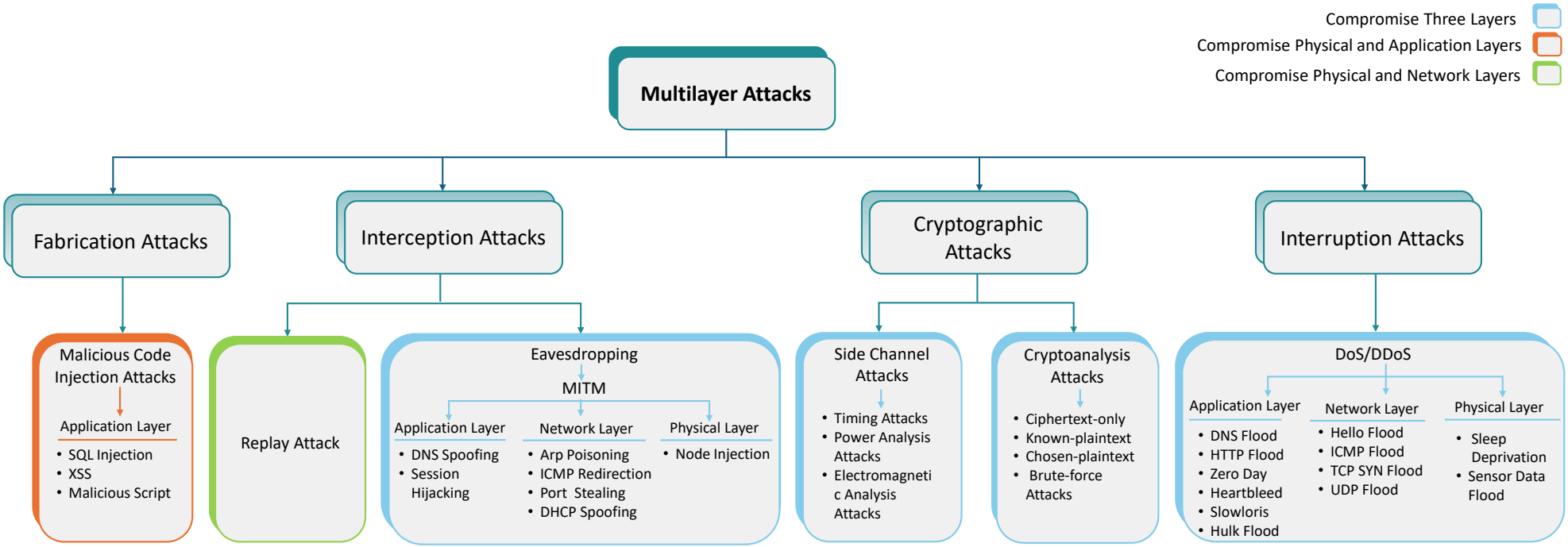


Figure 1.5. Classification and patterns of multilayer attacks in IoT systems.

1.3 TECHNOLOGIES TO DETECT IOT MULTILAYER ATTACKS

In 2024, the UK introduced new laws mandating that IoT devices adhere to minimum security standards. For example, manufacturers should not set default passwords that are common or simple, such as 'admin' or '12345'. They must publish clear contact details so that any bugs, vulnerabilities, or security issues can be promptly reported and addressed. Additionally, they are required to inform consumers about the minimum duration for which security updates will be provided for their devices (Department for Science Innovation and Technology, 2024).

- **Non-ML based Security Solutions**

Traditional security methods have been integral to the defence against IoT cyber threats. Among these, packet filtering and firewalls stand out. These techniques work by examining data packets moving in and out of a network and enforcing security rules to either allow or block traffic (Chaabouni et al., 2019). This helps prevent unauthorised access and malicious data from compromising the network.

Encryption is another critical traditional method, which helps maintain confidentiality and user privacy by employing cryptographic protocols to secure data during transit. Protocols such as Transport Layer Security (TLS), Secure Shell (SSH), and Hypertext Transfer Protocol Secure (HTTPS) ensure that data and code are encrypted before transmission, protecting them from interception and tampering. Encryption methods rely on using pairs of public and private keys to verify that data originates from legitimate sources and remains unaltered (Chaabouni et al., 2019).

Furthermore, there are several studies that have focused on detecting and mitigating multilayer attacks using other traditional techniques, such as distributed cross-layer approaches, behaviour-based anomaly detection techniques, and distributed mobile agents.

Studies by Bansal et al. (2011); Mahale et al. (2017); Bansal, Divya and Sofat (2010) have focused on detecting packet dropping, route misdirection, and Dos/DDoS attacks, in wireless mesh networks using cross-layer interaction techniques. For instance, Bansal et al. (2011) focused on detecting multilayer attacks in wireless mesh networks (WMNs). Packet dropping and route misdirection attacks target the routing layer, while shorter than Distributed Coordination Function Inter Frame Spacing (DIFS) time attacks, oversized Net Allocation Vector (NAV) attacks, and reduced backoff attacks target the MAC layer. The authors suggested a distributed cross-layer approach that utilises both network and MAC layer parameters for detection, as shown in Figure 1.6.

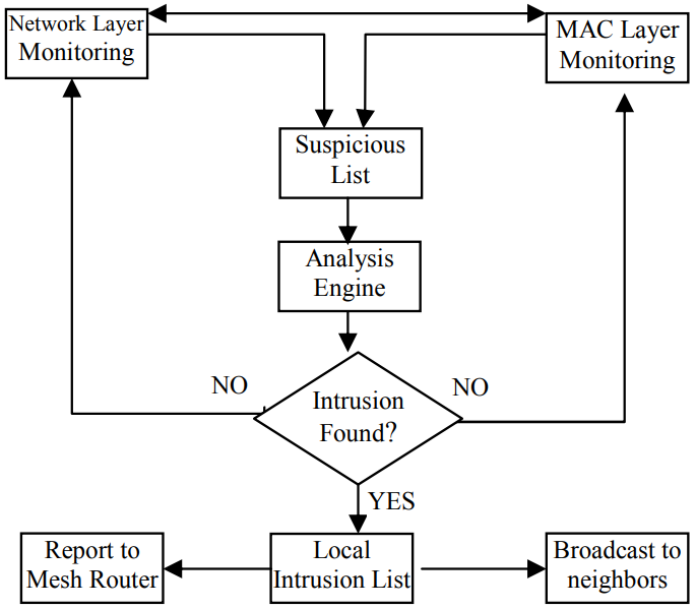


Figure 1.6. Framework for IoT IDS (Bansal, D., Sofat and Kumar, 2011).

Similarly, Bansal, Divya and Sofat (2010) proposed a framework that utilises cross-layer interactions for detecting DoS attacks, including collision attacks, packet dropping and misdirection attacks. Mahale et al. (2017) proposed an advanced cross-layer technique, which combines a device-driver packet filter and a remote firewall to mitigate DDoS attacks. Their approach dynamically adjusts network parameters in response to the attack. The authors use metrics such as the total number of packets, forwarded packets, and dropped packets to evaluate the effectiveness of their approach in filtering malicious traffic.

Other studies have focused on detecting multilayer attacks in mobile ad hoc networks. For example, Sodagudi and Rao (2014) focused on detecting DDoS attacks, specifically black hole attacks, and MITM attacks, specifically ARP cache poisoning attacks. The authors proposed behaviour-based anomaly detection techniques to detect these multilayer attacks. They developed MBHARP (Malicious Black hole attack with routing protocol), as shown in Figure 1.7, an approach to identify black holes, and the DL2MITM detection (Data link layer MITM attack) technique to detect MITM attacks.

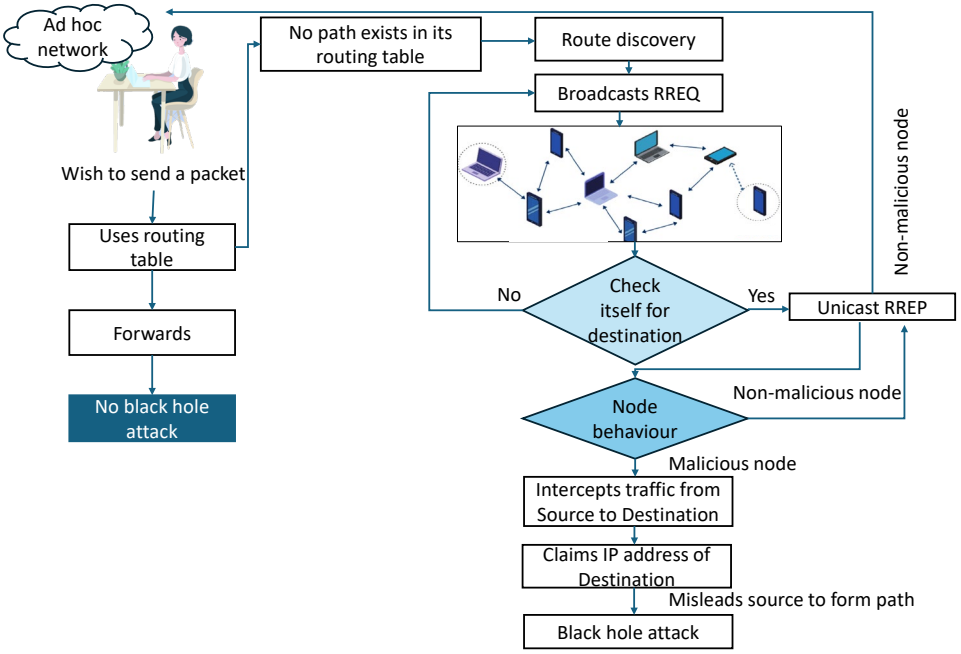


Figure 1.7. MBHARP framework (Sodagudi and Rao, 2014).

Mythili and Seetha (2021) proposed a novel approach called MPDDMA to detect multilayer packet dropping attacks using distributed mobile agents, which can detect such attacks in different layers and communicate with each other to identify the source of the attack. The authors used NS2 software to implement their approach. The metrics used to evaluate the MPDDMA approach include detection delay, detection accuracy, Packet Delivery Ratio (PDR), and total packets dropped. Based on these metrics, as the number of malicious nodes increases, the detection delay rises, detection accuracy drops from around 98% to above 80%, the PDR declines from 0.99 to 0.36, and the total packets dropped increases from 2 to 453 packets.

Both Mythili and Seetha (2021); Mahale et al. (2017) used evaluation metrics to demonstrate the effectiveness of their approaches. Mythili and Seetha (2021) showed that their approach achieved detection accuracy between 80% and 98%. In contrast, Mahale et al. (2017), while reporting the number of packets dropped, did not address key metrics for evaluating IDS performance, such as detection rate, false positive rate, or false negative rate. The other researchers—Bansal et al. (2011); Bansal, Divya and Sofat (2010); Sodagudi and Rao (2014)—while providing valuable insights into the design of multilayer attack intrusion detection systems on wireless mesh networks (WMNs) and mobile ad hoc networks, did not offer an evaluation of their proposed solutions.

Despite the effectiveness of traditional security measures, they face significant challenges in IoT environments due to high false positive and false negative rates, and limited device processing power and storage, which restricts their ability to handle the computational demands of robust encryption and packet filtering. These methods are static, making them less responsive to dynamic multilayer attacks, as they require manual updates to adapt to new threats. While they are effective in NS2 simulations, they struggle with real-world complexities like interference and evolving attack patterns.

- **Machine learning Based Solutions**

To address the challenges of the traditional security methods, the application of machine learning has emerged as an essential advancement in enhancing IoT security. Machine Learning (ML) techniques offer significant advantages over traditional methods by providing the capability to learn from large datasets, identify complex patterns, reduce false alarms, operate efficiently on resource-constrained devices, detect IoT security attacks, and adapt to new threats in real-time.

The primary idea behind ML is to use datasets to train models, thereby improving their decision-making and predictive accuracy in identifying specific tasks. For instance, when a machine learning model identifies a malicious IP address trying to make a connection, it can detect the attack and block it, thereby preventing potential data breaches Sarker et al. (2020). Therefore, alongside the traditional approaches, the application of ML and Deep Learning (DL) in detecting IoT attacks has been demonstrated by researchers. Chen et al. (2020); Hussein et al. (2022); Kethineni and Pradeepini (2024) provide ML and DL models to tackle IoT attacks. For example, Chen et al.

(2020) proposed a machine learning-based system for detecting DDoS attacks in IoT environments. They utilised different techniques to identify several types of DDoS attacks, including sensor data flood in the physical layers and UDP flood, TCP SYN, and ICMP (Ping of death) in the network layer. Their approach involved IoT security authentication, rules of the SDN controller, and supervised learning models such as the decision tree. The authors employed manual feature selection based on the attack type. Their work demonstrates the potential of machine learning algorithms in accurately detecting various DDoS attacks with an accuracy of around 97% in IoT networks.

A study by Hussein et al. (2022) proposed an Intrusion Detection System based on a Random Forest machine learning classifier. Their model focused on detecting intrusions in IoT environments, such as MITM attacks, specifically ARP spoofing; DoS attacks, specifically UDP, HTTP, and SYN flooding; port scanning; and brute force attacks. The authors used the IoTID20 dataset to train their classifier and achieved accuracy ranges from 78.1% to 95.2% in identifying the attack types. In Kethineni and Pradeepini (2024), the authors proposed an Intrusion Detection System (IDS) based on the integration of a Convolutional Neural Network (CNN) with a Bidirectional Gated Recurrent Unit (Bi-GRU) to identify Distributed Denial of Service (DDoS) attacks in smart agriculture. The authors demonstrated the effectiveness of their proposed model using the ToN-IoT and APA-DDoS datasets, achieving high detection accuracy in identifying DDoS attacks.

In the existing state-of-art literature, studies focused on developing frameworks that incorporated machine learning models for single layer attacks identification, particularly DoS attacks, and have demonstrated high performance Chen et al. (2020); Hussein et al. (2022); Kethineni and Pradeepini (2024); Keserwani, et al. (2023); Tareq et al. (2022); Khacha et al. (2022); Al Nuaimi et al. (2023); Samin et al. (2023); Ullah et al. (2023); Ferrag et al. (2022). Those studies have not explored the detection and classification of multilayer IoT attacks, sometimes the overall training model is based on obsolete dataset which will make the IoT devices vulnerable to the new types of attacks. Besides, most of the studies have explored manual feature selection methods and a limited number of studies have investigated the tuning of the hyperparameters of the models used. Nonetheless, there remains a significant need for developing a robust and adaptive framework to detect a wider range of multilayer IoT attacks that exceed the constraints of current IDSs.

1.4 RESEARCH MOTIVATION

The rapid proliferation of IoT devices has introduced significant cybersecurity challenges, particularly with multilayer attacks that target multiple layers of IoT architecture simultaneously. These attacks include encryption attacks, distributed denial of service attacks, replay attacks, and code injection attacks.

The motivation for this research is closely aligned with the United Nations Sustainable Development Goal (SDG) 9: Industry, Innovation, and Infrastructure. This goal emphasises the importance of building resilient infrastructure, promoting inclusive and sustainable industrialisation, and fostering innovation. A critical aspect of achieving these objectives is ensuring the security and robustness of the infrastructures that support modern innovations and industrial systems.

Machine learning offers promising capabilities for enhancing cybersecurity by providing real-time detection and mitigation of complex threats. This research aims to leverage ML techniques to effectively identify and respond to multilayer IoT attacks. By improving the accuracy and efficiency of intrusion detection, this research contributes to building secure and resilient infrastructures, thereby supporting the broader objectives of SDG 9.

1.5 RESEARCH QUESTION

The following is the research question addressed by this study:

How can Machine Learning techniques be used effectively to detect Multilayer security attacks?

1.6 RESEARCH AIM AND OBJECTIVES

This aim of this study is to develop a Semi-Automated Intrusion Detection System (SAIDS) that integrates efficient feature selection, feature weighting, normalisation, visualisation, and human-machine teaming to detect and identify multilayer attacks, enhancing mitigation strategies.

The objectives of this research work are as follows:

Obj1: Investigate the existing machine learning algorithms and identify the current limitations of the standard framework for detecting IoT multilayer security attacks while optimising the number of most significant features in machine learning models through feature selection and weighting methods.

Obj2: Develop and optimise the parameters of SAIDS to enhance the accuracy of detecting multilayer IoT attacks.

Obj3: Performance testing of the proposed framework on simulated multilayer attacks and on real world to evaluate its effectiveness.

1.7 THESIS STRUCTURE

The thesis is organised into six comprehensive chapters as shown in Figure 1.8:

Chapter 2: State-of-the-Art Research on Multilayer Attacks in IoT Security - This chapter provides comprehensive review of related studies on the role of machine learning in enhancing IoT security, explores feature selection and weighting techniques, and relevant IoT datasets. Additionally, it identifies gaps in current research and highlights areas requiring further exploration and developments.

Chapter 3: Methodology for Advanced IoT Multilayer Attacks Detection - This chapter introduces the experimental tools and methodological framework used to develop the IoT multilayer detection and identification system. It details the processes involved in sourcing IoT security datasets, data preprocessing, feature selection, and weighting. Additionally, it discusses the use of machine learning models, the identification of optimal features, the integration of human insights into

the system, and key aspects of model training, including performance evaluation metrics.

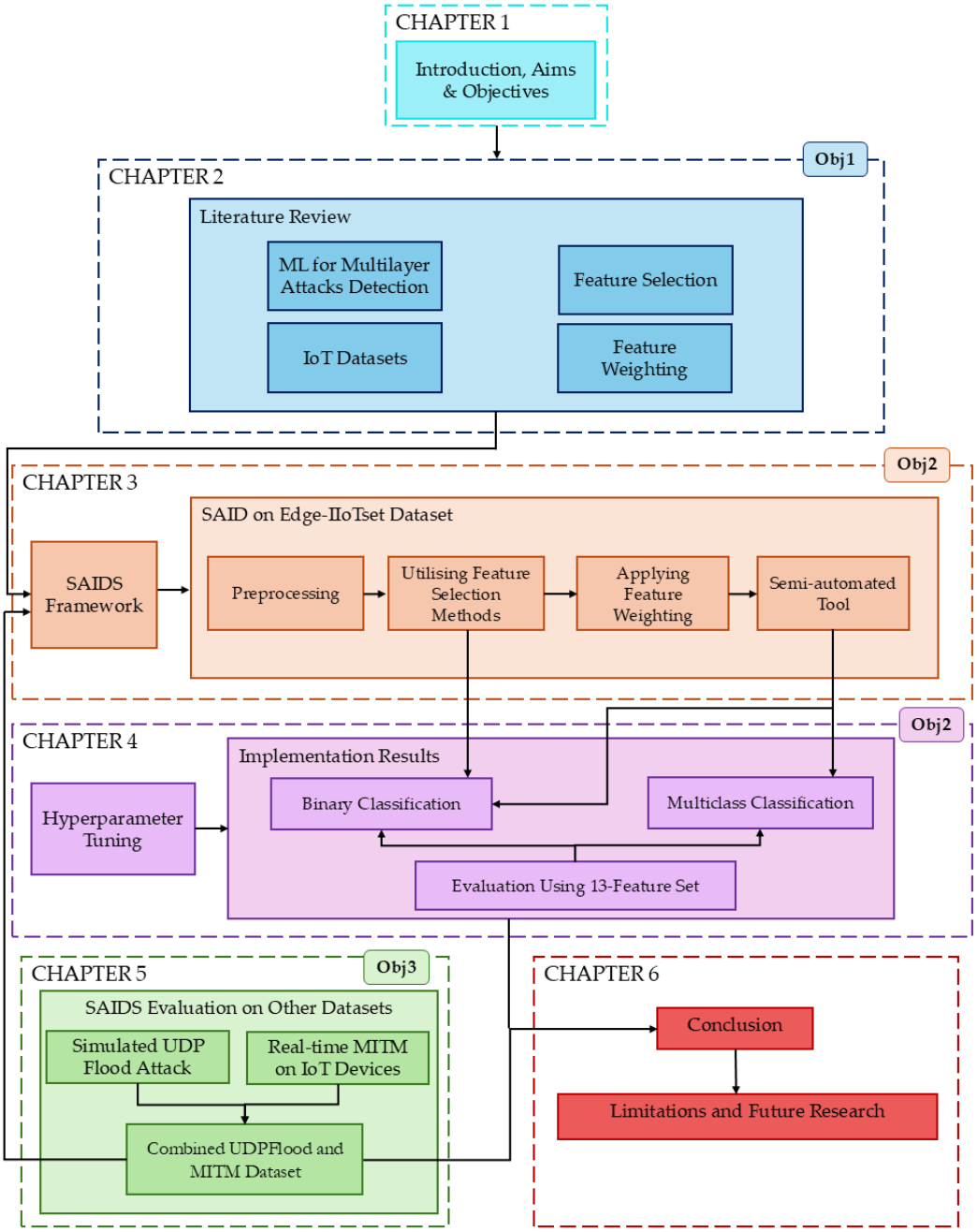


Figure 1.8. Diagram showing the thesis structure and chapter contents with the corresponding objective numbers.

Chapter 4: Experimental Results and Settings - It explores the development and application of the SAIDS, focusing on hyperparameter tuning of machine learning models and summarising the implementation results.

Chapter 5: Evaluation of the Proposed SAIDS Across Diverse Datasets - It examines the effectiveness of the SAIDS through simulated and real-time attack scenarios. This chapter discusses the robustness and adaptability of SAIDS across different IoT environments.

Chapter 6: Conclusion and Future Work – This chapter summarises the thesis, highlighting the key contributions and insights derived from the research. This chapter also discusses the research implications and proposes potential future research directions.

2 State-of-Art Research on Multilayer Attacks in IoT Security

As shown in Figure 2.1, this chapter provides a comprehensive review of the literature on the role of ML in detecting multilayer IoT attacks. It also examines the datasets used to train ML models, as well as feature selection and weighting techniques. The chapter concludes by identifying gaps in existing research and highlighting the need for flexible ML models and comprehensive computational frameworks to address these multilayer IoT security challenges.

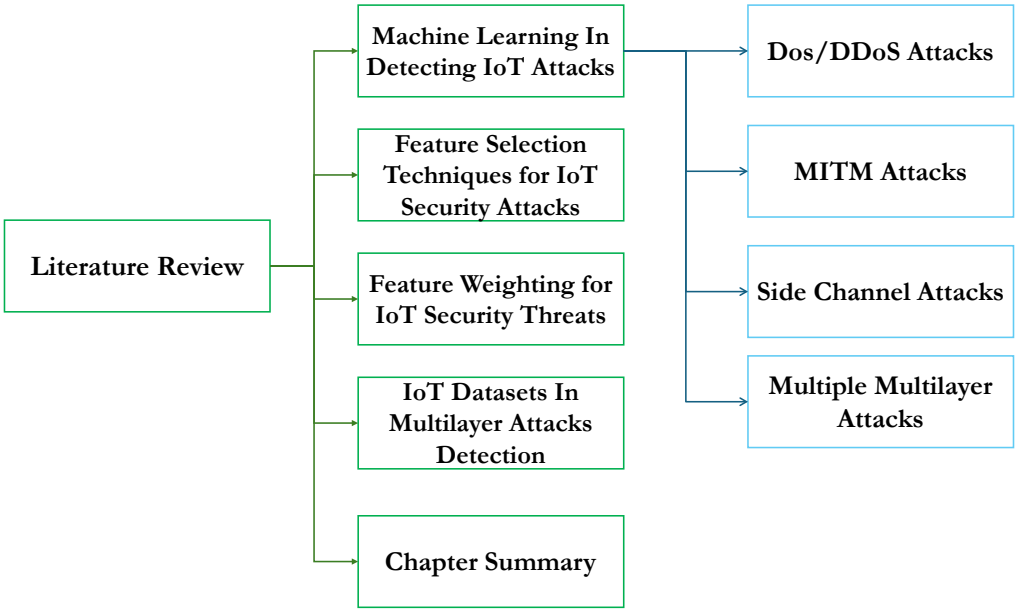


Figure 2.1. Structure of the literature review chapter and its sections.

2.1 Role of Machine Learning in Enhancing IoT Security

This section provides an overview of existing datasets used in Intrusion Detection Systems for detecting IoT cyber-attacks, distinguishing between datasets created from IoT devices and those that are not. It also reviews various ML techniques that have been proposed for detecting multilayer attacks to secure IoT systems, ensuring the confidentiality, integrity, and availability of data, as well as protecting user privacy. For a full analysis and summary of the existing studies, refer to Appendix 1.

2.1.1 Existing Datasets for IoT Intrusion Detection Systems

Table 2.1 provides a comprehensive overview of existing datasets used particularly in Intrusion Detection Systems for detecting IoT cyber-attacks. As shown in the table, widely used datasets in network security research, such as KDDCUP 1999, NSL-KDD, UNSW-NB15, CICIDS2017, and CICDDoS2019, are not specific to IoT systems. In contrast, the BoT-IoT, ToN-IoT, Edge-IIoTset, and BoTNeT-IoT datasets are specifically designed for IoT systems, containing the unique characteristics of IoT traffic.

Table 2.1. Summary of datasets used in IoT intrusion detection systems.

Dataset		Year	IoT Specific	Total Features
KDDCUP 99	Vibhute et al. (2024)	1999	No	41
NSL-KDD	Aljawarneh et al. (2018)	2009	No	43
UNSW-NB15	Ahmad, Z. et al. (2021)	2015	No	49
CICIDS2017	Salman et al. (2022)	2017	No	80
BoT-IoT	Peterson et al. (2021)	2018	Yes	45
BoTNeT-IoT	Belkacem (2024)	2018	Yes	23
CICDDoS2019	Rehman et al. (2021)	2019	No	86
ToN-IoT	Alsaedi et al. (2020)	2020	Yes	44
Edge-IIoTset	Ferrag et al. (2022)	2022	Yes	61

The BoT-IoT dataset, introduced in 2018, contains 45 features and includes data on DoS and DDoS attacks from multilayer attacks. Similarly, the BoTNeT-IoT dataset, also released in 2018, is

limited to only two botnet attacks (Mirai and Gafgyt) and includes just 23 features. On the other hand, the ToN-IoT dataset, introduced in 2020, represents a more comprehensive IoT dataset with 44 features, designed to capture a broad range of IoT traffic and various types of multilayer attacks. Lastly, the Edge-IIoTset dataset, released in 2022, is particularly focused on Industrial IoT (IIoT) environments, providing 61 features that address the complexity of real IoT device traffic and include most of the multilayer attacks.

2.1.2 Dos/DDoS Attacks Detection Using ML

The detection of DoS/DDoS attacks has been extensively studied, employing various machine learning and deep learning techniques, as shown in Table 2.2. In Priya et al. (2020) , the authors applied three classification algorithms: K-Nearest Neighbors (KNN), Random Forest (RF), Naïve Bayes (NB) for the detection of DDoS attacks. They created their own dataset and utilised two features, delta time and packet size, to distinguish DDoS packets. The proposed approach can detect various types of DDoS attack in an IoT network with a detection accuracy of 98.5%. However, as they used a simulation tool (hping3) to generate DDoS attacks and trained their model on detecting these attacks, their model is limited to detecting attacks produced by hping3 and may not be capable of detecting attacks generated by other tools or real IoT devices. Ravi and Shalinie (2020) proposed a Learning-Driven Detection Mitigation (LEDEM) model that employs a Semi-supervised Deep Extreme Learning Machine (SDELM) to detect DDoS attacks on IoT servers. The researchers created their own dataset and used the UNB-ISCX dataset to evaluate their model and compared the results with cutting-edge solutions. The proposed model achieved a reported accuracy of 96.28%. However, the model detects the DDoS attacks according to the training data and was unable to detect untrained attacks.

Table 2.2. Attacks, layers, datasets, ML algorithms, and features considered in reviewed studies.

Ref.	Attacks and Layers	Dataset	ML Algorithm	Features	Accuracy
Priya et al. (2020)	DDoS attacks on network layer	Own dataset	NB, KNN, RF	2	98.50%
Ravi and Shalinie, (2020)	DDoS attacks on network layer	Primary: own dataset, Secondary: UNBISCX	semi-supervised ML	155	96.28%
Rehman et al. (2021)	DDoS attacks on network layer	CICDDoS2019	GRU, NB and SMO	-	99.69%
Moustafa, et al. (2019)	Botnet attacks on application layer	UNSW-NB15, NIMS botnet	NB, DT and ANN	36	98.29% – 99.54%
Sangodoyin et al. (2021)	DDoS flooding attacks on network layer	Own dataset	QDA, NB, KNN, and CART	-	98%
Chkirbene et al. (2020)	DDoS attacks on network and application layer	UNSW-NB15	DT and CART algorithms	13	95.37%
Doshi, et al. (2018)	DDoS attacks on the network layer	Own dataset	KNN, SVM, DT, RF, and DNN	5	99%
Chen et al. (2020)	DDoS attacks on Multilayer	Own dataset	DT	-	99.98%
Salman et al. (2022)	DoS attacks on network layer	Primary: own dataset, secondary: CICIDS2017	DT, RF, and deep learning models	39	97%
Hady et al. (2020)	MITM attack on network layer	Own dataset	RF, DNN, SVM, and ANN	34	92.44%
Mukhtar et al. (2020)	Side channel attacks	Own datasets	CNN	800	67%
Gad, et al. (2021)	Attacks on network and application layers	ToN-IoT	Logistic Regression, NB, DT, SVM, KNN, RF, AdaBoost, XGBoost	20	98.60%
Makkar et al. (2021)	Spam attacks on application layer	REFIT smart home	Bayesian Generalised LM, Boosted LM, xgboost, Generalised LM, and bagged model	15	79.8% – 91.8%
Zolanvari et al. (2019)	Injection attacks on application layer	Own datasets	SVM, KNN, NB, RF, DT, LR, and ANN.	23	F1-score: 96.81%

Similarly, Rambabu and Venkatram (2021) developed a ML model to identify DDoS attacks on IoT networks called Ensemble Classification Using Traffic Flow Metrics (ECTFM). Their approach used the KNN algorithm to reduce the high dimensionality of the training data by partitioning it into multiple clusters. The authors used the DEFCON, ADFA, LBNL, KYOTO, and CICIDS2017 datasets as inputs to their algorithm and achieved 95% accuracy. Rehman et al.

(2021) proposed a model to detect DDoS called DIDDOS (Detection and Identification of Distributed Denial of Service). The authors utilised the Gated Recurrent Unit (GRU) a type of Recurrent Neural Network (RNN), Sequential Minimal Optimisation (SMO), and NB. The CICDDoS2019 dataset was used to evaluate the proposed system, and the reported accuracy of the GRU model is 99.69%. The key strength of their model is its ability to detect new attacks that did not occur in the training dataset.

In the study by Al-Yaseen, Othman and Nazri (2017), the authors proposed a multi-level hybrid Intrusion Detection System (IDS) that employs a Support Vector Machine (SVM) and Extreme Learning Machine (ELM) to enhance the detection of DoS, User-to-Root (U2R), and Remote-to-Local (R2L) attacks. To improve the IDS performance, a modified K-means approach was used to reduce the size of the training dataset, balance the dataset for training the SVM and ELM, as well as reduce classifier training time. The suggested model increased overall performance and achieved an accuracy of 95.75% with a false alarm rate of 1.87%, according to experimental results on the KDD CUP 1999 dataset.

Moreover, Moustafa, Turnbull and Choo (2019) developed an ensemble learning algorithm called AdaBoost. The model integrates NB, Decision Tree (DT) and Artificial Neural networks (ANN) algorithms to identify botnet attacks that compromise the protocols of IoT networks. The authors used the UNSW-NB15 and NIMS botnet datasets, focusing on three network protocols: DNS, HTTP, and Message Queue Telemetry Transport (MQTT). The ensemble model achieved a detection rate between 95.25% and 99.86% and a false positive rate between 0.01% and 0.72% for DNS and HTTP data sources. However, their model may not be able to identify malicious attacks such as zero-day attacks and newer threats because it assumes that the incoming packets conform to the MQTT, DNS, and HTTP protocols.

Studies by Bagaa et al., 2020; Sangodoyin et al. (2021) propose ML frameworks for quickly detecting and preventing security attacks such as DDoS in SDN environments. Bagaa et al. (2020) built their novel security framework based on the NSL-KDD dataset and evaluated it in a smart building scenario. Also, they employed a one-class SVM algorithm for detecting anomalous behaviours: DDoS, Probe, U2R and R2L, their proposed model achieved a high detection accuracy of 99.71%.

Despite the high accuracy of detection, their model may not be suitable for modern IoT networks as they used an outdated dataset.

Sangodoyin et al. (2021) built their dataset in a simulated environment and covers HTTP, UDP and TCP flooding attacks that were launched using Low Orbit Ion Cannon. Their SDN system consists of 16 hosts and 10 OpenFlow switches, connected via a 100 Mbps link. Quadratic discriminant analysis (QDA), NB, KNN, and CART are the ML algorithms used in their study. Although all the ML algorithms were highly effective at detecting and classifying DDoS flooding attacks, CART showed the best performance with a reported accuracy of 98%. However, the training time, which is around 12.4 ms, is considered high for real-time IoT systems (Zainudin, Ahakonye et al., 2023).

Chkirbene et al., (2020) proposed a hybrid anomaly-based IDS that combines CART and RF algorithms. For feature selection, the RF technique was employed to reduce the dataset's dimensions to the most important attributes. Different IoT attack classes, such as reconnaissance, DoS, wormhole, and backdoor, were identified using the CART classifier. The proposed system was tested using the UNSW-NB15 dataset, and the results showed that the system achieved an accuracy of 95.37%. Despite its high detection rate, the model was unable to identify various attacks due to a lack of training data for specific types of attacks, including DoS, wormhole, shellcode, and backdoor.

Moreover, the three studies below have been conducted on detecting three different types of DDoS attacks: TCP SYN flooding, UDP flooding, and HTTP flooding. The first study was conducted by Doshi, Apthorpe and Feamster (2018). The authors proposed a binary classification model for distinguishing between benign and malicious DDoS attacks. They also employed five IoT-specific network behaviours (such as packet size, regular time intervals between packets, protocols used, bandwidth, and a limited number of endpoints) to guide feature selection. Additionally, they deployed a variety of ML algorithms, including KNN, SVM, DT, RF, and Deep Neural Networks (DNN) algorithms to detect DDoS attacks, including HTTP GET flood, TCP SYN flood, and UDP flood. The authors created their own dataset using Raspberry Pi v3 and several IoT devices, such as a YI home camera, a Belkin smart switch and a blood pressure monitor. They then tested

the deep learning and machine learning models on the generated dataset and achieved a detection accuracy higher than 99%.

In the study by Cvitic et al. (2022), the authors proposed a DDoS detection model based on four IoT device classes (very high, high, medium, and low level of traffic predictability) using a boosting technique for Logistic Model Trees (LMT), which is a combination of Logistic Regression (LR) and DT. The authors built four primary datasets (C1DDOS, C2DDOS, C3DDOS, and C4DDOS) by connecting 41 smart home devices and introducing the three mentioned DDoS attacks. A secondary dataset from the University of New South Wales, consisting of 28 IoT devices and containing normal and legitimate traffic, was also used Sivanathan et al., (2019). Their model achieved an accuracy between 99.92%–99.99%.

Lastly, Salman et al. (2022) created their own dataset using data collected from seven IoT devices. The authors utilised DT, RF, and DL models for identifying IoT devices and detecting the above-mentioned three DDoS attacks. They also used the CICIDS2017 dataset for non-IoT devices with abnormal traffic, which contains several types of attacks: web attack, infiltration, brute force, DoS, port scan, DDoS, heartbleed, and botnet. The results showed that the RF classifier outperformed other algorithms, achieving a device-type identification accuracy of 94.5%, traffic-type classification accuracy of 93.5%, and abnormal traffic detection accuracy of 97%. However, their approach may not be able to identify new devices.

2.1.3 MITM Attacks Detection Using ML

Several studies have addressed the detection of MITM attack using different ML models, as shown in Table 2.2. In Saharkhizan et al., (2020), the authors proposed an enhanced DL model for intrusion detection by combining a Decision tree with a Long Short-Term Module (LSTM). They used a simulated dataset by Frazão et al., (2019), which includes four cyberattack categories: MITM, Ping DDoS, TCP SYN DoS attacks, and Modbus query flood attacks. The proposed mechanism's accuracy reached 99%. However, the detection complexity of the proposed model is high.

In the study by Hady et al. (2020), the authors built a real-time testbed called Enhanced Healthcare Monitoring System (EHMS) in which several sensors were placed on a patient's body to monitor

the patient's biometrics and collect network flow measurements, which were then forwarded to a remote server for further analysis and actions. The authors used four machine learning algorithms for MITM attack detection: RF, DNN, SVM, and ANN. According to their findings, the SVM algorithm outperformed the others, with an accuracy of around 92.44 %.

2.1.4 Side Channel Attacks Detection

Furthermore, three additional studies have addressed side channel attacks using ML algorithms. Illuri (2021) integrated a ML model with chaotic logistic maps for detecting side channel attacks. The authors utilised ELMs to distinguish between correct and incorrect sub-keys, making side channel attacks impossible. The suggested model was able to detect side-channel attacks with 95% accuracy. The sensitivity analysis showed a result of 99.7% after applying the chaotic methodology to medical image datasets, such as MRI, mammogram, and diabetic retinopathy images.

Mukhtar et al. (2020) proposed a Convolutional Neural Network (CNN) model for detecting side channel attacks. The suggested CNN model uses Principal Component Analysis (PCA) as a pre-processing step to select the best features, and the Synthetic Minority Oversampling Technique (SMOTE) was utilised to balance the data. The suggested approach is less computationally complex than existing deep learning-based models and performs more efficiently in terms of time. However, the reported accuracy of the model is only 67%.

Mukhtar et al. (2019) used ML algorithms to protect systems from side channel attacks by focusing on recovering secret key information from leaked power signals. The authors achieved this by employing the double-and-add-always algorithm to encrypt the data using a secret key. Classification algorithms such as NB, SVM, RF, and Multi-Layer Perception (MLP) were analysed with the ECC datasets, and the model achieved an accuracy of 90%.

2.1.5 Other Multilayer Attacks Detection

Various studies have focused on the detection of other multilayer attacks, presented in Figure 2.2, by leveraging different machine learning models.

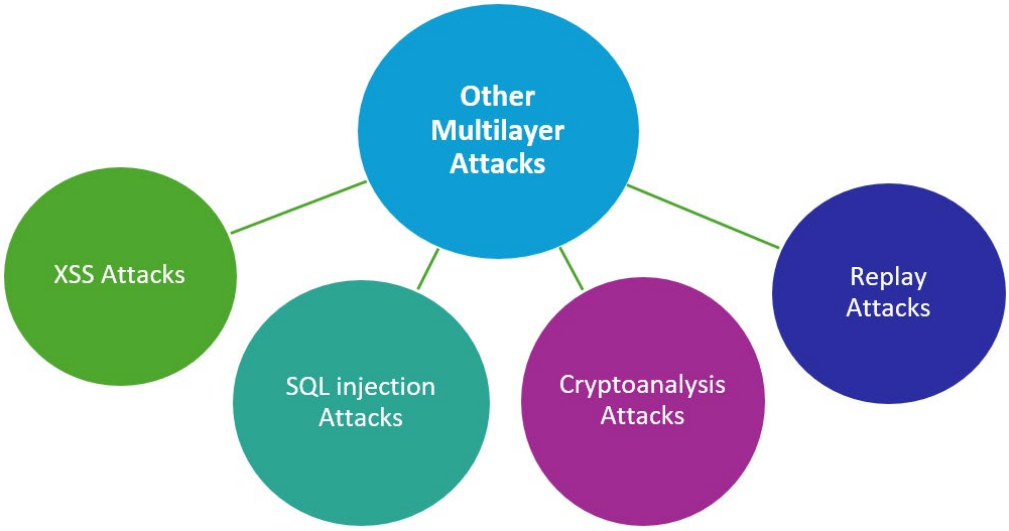


Figure 2.2. Other multilayer attacks.

In Gad, Nashat and Barkat (2021), the authors illustrated a ML-based IDS for Vehicular Ad Hoc Networks (VANETs) based on the ToN-IoT dataset, which is an updated version of the NSL-KDD dataset. The ToN-IoT contains the most recent attack types, such as MITM, DoS, DDoS, ransomware, password cracking attack, scanning, injection, backdoor, and XSS. To address the class imbalance problem, the authors used SMOTE and employed the Chi-Square technique for feature selection. They also compared the performance of several ML algorithms, including LR, NB, DT, SVM, KNN, RF, AdaBoost and Extreme Gradient Boosting (XGBoost). The results revealed that the XGBoost classifier outperformed the other algorithms with an accuracy of 98.60%.

(Makkar et al. (2021) proposed an ML framework for identifying web spam attacks in IoT devices. Five ML models (Bayesian Generalised Linear Model, Boosted Linear Model, XGboost, Generalised Linear Model, and bagged model) were evaluated in their method using a variety of input features, such as Gain Ratio (GR) and Information Gain (IG), and symmetrical uncertainty. The authors validated their proposed framework using the REFIT smart home dataset. The

accuracy of the proposed method ranged from 79.8% to 91.8% for the five machine learning models. Zolanvari et al. (2019) proposed a Machine learning anomaly-based Intrusion Detection System to detect several Industrial IoT attacks, such as command injection, backdoor, and SQL injection attacks, using seven algorithms, including DT, SVM, NB, LR, RF, KNN and ANN. The system's results showed that the RF classifier outperformed the others, achieving an F-measure value of 96.81%.

Moreover, two studies conducted by Anthi et al. (2019); Sarkar et al. (2021) addressed three multilayer attacks: replay, DoS/DDoS, and MITM attacks. Anthi et al. (2019) proposed a three-layer anomaly-based IDS for smart home networks of IoT devices, using nine supervised classification algorithms (NB, J48, Zero R, Bayesian Network (BN), One R, LR, SVM, MLP, and RF) to detect IoT network attacks, such as DoS, MITM, replay, spoofing and reconnaissance. The system's performance revealed that the J48 classifier outperformed the others in terms of F-measure, with values ranging from 90% and 98%. Sarkar et al. (2021) proposed a comprehensive technique for securing IoT devices based on a nature-inspired Gravitational Search-guided artificial neural key. Artificial neural network synchronisation is employed in their approach to develop a neural key exchange protocol for cryptographic purposes between two IoT devices over a public channel. The values $\alpha 1$, $\alpha 2$, $\alpha 3$, β, and γ were calculated to enable resistance against attacks, including session hijacking, impersonation, replay, brute force, DDoS, Geometric, MITM, and social engineering. The findings revealed that if these attacks are carried out, there is a 0% chance of success.

2.2 Feature Selection Techniques for IoT Security Attacks

The importance of feature selection is critical in machine learning models, particularly because real-world data often contains noise, including irrelevant information about the problem. By employing feature selection, the computational load for training those machine learning models with a large number of features is not only reduced, but noise in the data can also be filtered out, ensuring that the models focus on the most relevant information. As shown in Table 2.3, researchers have

explored different wrapper and filter feature selection methods such as the bijective soft set technique, correlation, fast-based-correlation feature (FCBF) algorithm, Information Gain (IG), and Gain Ratio (GR), which have proven beneficial in accurately identifying botnet, DoS, DDoS, and MITM attacks in IoT networks.

The bijective soft set technique manages uncertainty and reduces dimensionality through bijective mappings. Correlation identifies features with the strongest linear relationship to the target, while the FCBF, a correlation-based approach, selects the optimal set of features by retaining those with high correlation to the class label and removing less important ones. It uses symmetrical uncertainty to assess how much each feature contributes to prediction. Mutual information evaluates the dependence between variables, selecting features that share the most information with the target. Information Gain measures a feature's ability to improve predictive accuracy by reducing uncertainty, though it can sometimes favour features with many categories. Gain Ratio improves on Information Gain by normalising it, ensuring that features with more categories do not receive unfair preference, leading to a more balanced feature selection process Shafiq et al. (2020); Egea et al. (2018).

Table 2.3. Overview of the used feature selection methods.

Feature Selection Methods
• Bijective Soft Set Technique. • Correlation. • fast-based-correlation feature algorithm. • Mutual Information. • Gain Ratio. • Information Gain.

As shown in Table 2.4, Shafiq et al. (2020) focused on detecting botnet attacks in IoT networks through effective feature selection. They employed a wrapper feature selection method called the bijective soft set technique and introduced a new feature selection metric called CorrACC, where several machine learning classifiers were utilised to analyse the approaches using the BoT-IoT dataset. The decision tree and Random Forest classifiers proved to be effective by selecting seven significant features for identifying botnet attacks in IoT traffic, with an accuracy rate of over 95%.

Additionally, Su, He and Wu, (2022) assessed multiple machine learning algorithms such as DT, RF and gradient boosting algorithm to predict IoT network attacks. Their research focused on identifying three types of IoT attacks: MITM, DoS, and scan attacks. To select the most effective 10 features associated with these attacks from the IoT2020 dataset, a correlation technique (with a threshold of 0.6) was used. It was found that the decision tree algorithm was generally the most accurate classifier, but the random forest algorithm had better AUC scores. The classifiers achieved an accuracy between 87% and 91% in detecting MITM attacks, and 99% in detecting DoS attacks. Similarly, Egea et al. (2018) introduced FCBFiP, a novel feature selection method based on the modification of the fast-based-correlation feature (FCBF) algorithm. The aim was to detect DoS attacks using features from the KDD99 dataset. This technique divided the feature space into fragments to enhance correlation and improve machine learning applications. The experimental results showed the effectiveness of FCBFiP FCBFiP in reducing the 41 features into 20, achieving an F1 score of 99% in identifying DoS and other cyber-attacks on IoT devices, while also showing improvements in execution time.

Table 2.4. Summary of Feature Selection Techniques in IoT Threat Detection.

Ref.	Focus of Study	Dataset Used	Feature Selection Method	Number of Features	Accuracy
Shafiq et al. (2020)	Detecting botnet attacks in IoT networks	BoT-IoT	Bijective soft set technique, DT and RF	7	>95%
Su, He and Wu (2022)	Predicting MITM, DoS, and scan attacks in IoT networks	IoT2020	Correlation	10	87%-91% (MITM), 99% (DoS)
Egea et al. (2018)	Detecting DoS attacks	KDD99, CNAE-9, LSVT voice	FCBF algorithm modification (FCBFiP)	20	F1-score: 99%
Alalhareth and Hong (2023)	Detecting various cyberattacks in IoMT networks	WUSTL-EHMS-2020	Enhanced MI	10	92% - 94%
Nimbalkar and Kshirsagar (2021)	Detecting DoS and DDoS attacks in IoT networks	BoT-IoT & KDD Cup 1999	GR and IG	16 & 19	99%
Albulayhi et al. (2022)	Detecting DDoS and DoS attacks	NSL-KDD & IoTID20	GR and IG	11 & 28 (IoTID20), 15 & 25 (NSL-KDD)	99.98%

The research by Alalhareth and Hong (2023) enhanced the Mutual Information (MI) feature selection method for detecting various cyberattacks in IoT networks, specifically targeting the Internet of Medical Things (IoMT). Using the WUSTL-EHMS-2020 dataset, which includes patients' biometric data and spoofing attacks, the authors demonstrated the effectiveness of their method. They reduced the number of features from 44 to 10 and employed several machine learning classifiers, including SVM, LR, RF, DT, and LSTM, to evaluate the selected features. By using only the selected 10 features, these models achieved an accuracy between 92% and 94% in detecting the spoofing attack.

The research done by Nimbalkar and Kshirsagar, (2021); Albulayhi et al. (2022) applied Gain Ratio and Information Gain as feature selection methods for detecting DoS and DDoS attacks in IoT networks. In Nimbalkar and Kshirsagar, (2021), the authors used only 16 features for detecting the mentioned attacks, and their system achieved an impressive 99% accuracy and detection rate on the BoT-IoT dataset. Similarly, on the KDD Cup 1999 dataset, with the use of 19 features, the system achieved the same high accuracy and detection rates.

Similarly, Albulayhi et al., (2022) introduced a new feature selection approach that integrates gain ratio and information gain, utilising mathematical techniques such as intersection and union rules. Their approach was used to identify relevant features from the NSL-KDD and IoTID20 datasets for detecting DDoS and DoS attacks. They evaluated their approach using four machine learning algorithms: DT, KNN, Bagging, and ANN. The results showed that their approach resulted in 11 and 28 relevant features from the IoTID20 dataset and 15 and 25 relevant features from the NSL-KDD dataset. Furthermore, the classification accuracy achieved by their approach was high, around 99.98%.

Although these existing feature selection methods can extract significant features from datasets, it remains unclear which is the most effective. Also, each feature selection method has its pros and cons, even the accuracy of those methods invariably depends on the training dataset. The aim of this research is to ascertain the optimal number of significant features, irrespective of all feature selection methods used, by incorporating multiple feature selection methods in the decision-making process.

2.3 METHODS OF FEATURE WEIGHTING IN DETECTING IOT THREATS

To scrutinise significant features through feature selection methods, employing feature weighting proves valuable by assigning scores/weights to each feature, indicating its significance in detecting IoT attacks within a dataset. As shown in Table 2.5, research has been carried out to develop advanced feature weighting methods for phishing site detection in smart cities.

Table 2.5. Summary of feature weighting methods in IoT threat detection.

Ref.	Method	Dataset Used	Accuracy
Sujatha et al. (2023)	GWO and Firefly Algorithm	Own dataset	95.75%
Swathi et al. (2023)	Particle Swarm Optimisation	Own dataset	93%
Khan, Sohail and Nazir (2022)	Statistical Aggregation and Multi-objective Optimisation	Not specified	85%
Subramani and Selvi (2023)	Rule-based techniques and Multi-Objective Particle Swarm Optimisation	KDD '99 Cup and CIDD	F1-score: 94.51% (DoS)

Sujatha et al. (2023) presented a novel feature weighting method using hybrid bio-inspired algorithms, specifically Gray Wolf Optimisation (GWO) and Firefly Algorithm (FF). This technique significantly enhances the performance of an ANN used for classification, demonstrating a detection accuracy of 95.75%. In contrast, the Particle Swarm Optimisation (PSO)-based feature weighting method developed by Swathi et al. (2023) achieved 95% accuracy during training and 93% accuracy in detection during testing, along with an impressive 98.4% accuracy rate in locating untrustworthy sites. Both studies underscore the effectiveness of employing feature weighting techniques in the domain of cybersecurity, particularly in the detection of phishing sites in smart city environments. Complementing these studies, research by Khan, Sohail and Nazir (2022) focuses on IoT device security, employing a statistical aggregation (SA) and multi-objective optimisation method (based on the ratio analysis) for feature weighting of security authentication features. This method demonstrates a substantial accuracy improvement, reaching 85%.

Moreover, the study conducted in Subramani and Selvi (2023) explains a combination of rule-based techniques and Multi-Objective Particle Swarm Optimisation for feature selection. They also

enhance attack detection in IoT-based wireless sensor networks by employing an advanced Multiclass Support Vector Machines classifier. To validate the effectiveness of their approach, the authors conducted experiments using the KDD '99 Cup and CIDD datasets, showcasing that their methodology not only enhances intruder detection accuracy but also effectively reduces false-positive rates. By using this approach, the authors achieved an F1-score of 94.51% in detecting DoS attacks.

The above-mentioned studies primarily focus on specific attacks on specific applications (e.g., phishing site detection in smart cities) or outdated datasets (e.g., KDD '99 and CIDD). There is a gap in research regarding the applicability of these feature selection and weighting methods across diverse IoT environments and attack vectors, especially for multilayer attacks. Additionally, there is a notable absence of discussion on incorporating human expertise in the loop of feature selection and weighting processes, which could enhance the interpretability and reliability of the detection models, especially in complex scenarios where automated methods might struggle. Despite the severe impact of these multilayer attacks on various IoT devices, there is currently no comprehensive framework for their detection and mitigation.

2.4 Overview of IoT Datasets Utilised in Multilayer Attack Research

A dataset in the context of cybersecurity refers to a collection of data specifically used to train, test, and evaluate IDS using ML and DL models. Such datasets are critical for detecting cybersecurity attacks on various systems, including IoT infrastructures. However, the reliance on these datasets arises from the impracticality of using real network traffic data publicly due to privacy concerns. Consequently, there is a continuous need to generate new datasets that can effectively reflect the dynamic and evolving landscape of IoT security threats, ensuring that detection models remain effective under varied and realistic conditions.

The most frequently utilised datasets in research, aimed at detecting both single and multilayer IoT attacks, include NSL-KDD, BoT-IoT, UNSW-NB15, ToN-IoT and Edge-IIoTset.

- **NSL-KDD Dataset**

The NSL-KDD has a total of 25,192 records (13,449 instances are normal and 11,743 instances are abnormal traffic) with 41 features, and the attack category consists of Probe, DoS, R2L and U2R (Aljawarneh, Aldwairi and Yassein, 2018). As discussed in section 2.1, authors such as Bagaa et al. (2020); Albulayhi et al. (2022) have utilised this dataset.

Furthermore, Aljawarneh, Aldwairi and Yassein (2018) developed a hybrid intrusion detection model using the NSL-KDD dataset to address both binary and multiclass classification problems. The study's primary objective was to create a system capable of efficiently estimating the intrusion scope threshold degree by utilising the most relevant features from network transaction data. The 41 features in the dataset were reduced to 8 significant features using the vote algorithm and information gain technique. The proposed model integrates several machine learning algorithms to detect DDoS, Probe, R2L and U2R attacks, including NB, J48, Random Tree, REP Tree, Meta Pagging, Decision Stump, and AdaBoostM1 classifiers. The experimental results demonstrated the model's efficacy, achieving a high accuracy rate of 99.81% for binary classification and 98.56% for multiclass classification.

Similarly, Liang et al. (2019); Tang et al. (2016) employed the NSL-KDD dataset in their approaches. Liang et al. (2019) proposed a multi-agent intrusion detection system for IoT networks, based on blockchain and DNN. All communications between agents are recorded on the blockchain, which secures the system against attacks such as DDoS, Probe, U2R, and R2L, with a DNN utilised for intrusion detection. The DNN model achieved an accuracy rate of 98%. Additionally, Tang et al. (2016) proposed an anomaly-based intrusion detection system that employs DNN to detect DDoS attacks in network traffic. And the authors utilised only 6 of the dataset's 41 features, achieving an accuracy rate of 75.75%.

In other studies, AL-Hawawreh, Moustafa and Sitnikova, (2018); Tama, Comuzzi and Rhee (2019) utilised the NSL-KDD dataset in conjunction with the UNSW-NB15 dataset in their models for detecting Backdoors, DoS, Reconnaissance, Worms, DDoS, Probe, R2L and U2R attacks. AL-Hawawreh et al. (2018) developed a deep learning-based anomaly detection model for Internet Industrial Control Systems (IICSs). The suggested model used both a Deep Auto-Encoder (DAE)

and a DNN algorithm, achieving an exceptionally high performance with a 99% detection rate and a 1.8% false positive rate. Tama et al. (2019) developed a two-level classifier and hybrid feature selection strategy for IDS in IoT. The model integrates the Rotation Forest and bagging algorithms as a two-level classifier, while particle swarm optimisation, genetic, and ant colony algorithms were employed in a hybrid feature selection technique. The proposed model achieved an accuracy rate of 85.8%.

- **BoT-IoT Dataset**

The BoT-IoT dataset involves a total of 73 million records, of which 9,543 instances are normal and the remainder represent abnormal traffic. It includes three dependent attributes (attack/normal traffic, attack category, and attack subcategory) and 43 independent attributes (invalid features, standard features, and calculated features) grouped into three categories. The attack categories include information theft, reconnaissance, DoS, and DDoS, while the attack subcategories include data theft, OS fingerprinting, keylogging, TCP, HTTP, UDP, and service scanning (Peterson, Leevy and Khoshgoftaar, 2021).

Shafiq et al. (2020); Nimbalkar and Kshirsagar (2021) utilised the BoT-IoT dataset in their proposed models as discussed in section 2.2. Furthermore, Ferrag et al. (2020) used the BoT-IoT dataset along with the CICIDS2017 dataset in their proposed intrusion detection system for IoT networks, called RDTIDS, which is located in the fog computing layer to detect DDoS attacks. In their model, they employed the REP Tree, and JRip algorithm classifiers to feed the Forest PA classifier. Using the BoT-IoT dataset and the CICIDS2017 dataset, the proposed model achieved high accuracies of 96.995% and 96.665%, detection rates of 95.175% and 94.475%, and false alarm rates of 1.120% and 1.145%, respectively.

Finally, Alhowaide, Alsmadi and Tang (2021) used the BoT-IoT dataset along with NSL-KDD, UNSW-NB15, and BoTNetIoT datasets to train and evaluate their proposed ensemble model for securing IoT networks, called the Model Selection Method (MSM). The proposed model utilised a total of 15 different classifiers, including both ensemble and traditional classifiers. The authors also utilised 5-fold cross-validation, which divides the datasets into 80% for training and 20% for testing resulting in model accuracies ranging from 93% to 100%.

- **UNSW-NB15 Dataset**

The UNSW-NB15 dataset includes 82,332 records in the testing dataset and 75,341 records in the training dataset, covering both normal and abnormal traffic. It also includes 49 features and nine attack categories, including DoS, shellcode, backdoors, reconnaissance, fuzzers, Analysis, generic, exploits, and worms Ahmad, Z. et al. (2021). As discussed in section 2.1, and the current section 2.4, authors such as Moustafa, Turnbull and Choo (2019); Chkirbene et al. (2020); AL-Hawawreh, Moustafa and Sitnikova (2018); Alhowaide, Alsmadi and Tang (2021) have utilised the UNSW-NB15 dataset.

(Ahmad, M. et al. (2021) utilised the UNSW-NB15 dataset, proposing feature clusters specifically focused on Flow, MQTT, and TCP protocols. By clustering features, they effectively addressed common issues such as data imbalance, overfitting, and the curse of dimensionality. They applied supervised ML algorithms, including RF, SVM, and ANN to these clusters. Using these algorithms, they extracted 18 TCP features, 13 Flow & MQTT features, and 11 top features selected from both TCP and Flow & MQTT. Their proposed model reported classification accuracies of 96.96% with Flow and MQTT features, 91.4% with TCP features, and 97.54% with the top features.

- **ToN-IoT Dataset**

The ToN-IoT dataset has 223,390,21 records of normal and attacks data, with 44 features. It also has two categories class label and attack categories such as MITM, DoS, DDoS, ransomware, password cracking attack, scanning, injection, backdoor, and XSS. Several researchers, including (Kethineni and Pradeepini (2024); Gad, Nashat and Barkat (2021) have used this dataset as discussed in Sections 1.3 and 2.1.

Moreover, Alotaibi and Ilyas (2023) utilised the ToN-IoT dataset to train their proposed model, which integrates four machine learning models—Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbor—into two ensemble techniques: stacking and voting. This methodology significantly improved the effectiveness of the Intrusion Detection System, achieving a remarkable accuracy rate of 98.63%.

- **Edge-IIoTset Dataset**

The Edge-IIoTset dataset was created in 2022 by Ferrag et al. (2022) and is applicable for IoT and industrial Internet of things (IIoT) applications. It includes data related to 14 different attack types, generated from more than 10 types of IoT devices, including sensors and detectors. These attacks include Backdoor Attack, DDoS HTTP Flood Attack, DDoS ICMP Flood Attack, DDoS TCP SYN Flood Attack, DDoS UDP Flood Attack, MITM Attack, OS Fingerprinting Attack, Password Attack, Port Scanning Attack, Ransomware Attack, SQL Injection Attack, Uploading Attack, Vulnerability Scanner Attack, and XSS Attack. The dataset consists of 20,952,648 records across five attack categories: injection, malware, DoS and DDoS, MITM attacks, and information gathering. Additionally, it includes 63 features Ferrag et al. (2022). The dataset has been cited in multiple studies, including works by Keserwani, Aggarwal and Chauhan (2023); Tareq et al. (2022); Khacha et al. (2022); Al Nuaimi et al. (2023); Samin et al. (2023); Ullah et al. (2023); Ferrag et al. (2022).

The paper by Keserwani, Aggarwal and Chauhan (2023) presents an approach to detect and classify attacks on IoT networks. Their research utilises DT, RF, and ensemble techniques, specifically CatBoost and XGBoost, trained on the Edge-IIoTset Dataset. Through manual feature selection, they managed to reduce the features to 20 significant features. The results showed that XGBoost is the most effective model, achieving an accuracy rate of 97.99%. Tareq et al. (2022) utilised two CNN models, DenseNet and Inception Time, for detecting IoT cyber-attacks through a multi-class classification approach. These models were tested on the Edge-IIoTset, ToN-IoT, and UNSW-NB15 datasets. The authors used all 63 features of the Edge-IIoTset dataset and implemented hyperparameter tuning on the models used. Additionally, when employing the Edge-IIoTset dataset with the Inception Time approach, the highest accuracy achieved was 94.94%.

Moreover, Khacha et al. (2022) proposed a model that combines CNN and LSTM techniques to efficiently detect and classify cyber-attacks in IIoT networks. The proposed CNN-LSTM model was trained on the Edge-IIoTset dataset, with hyperparameter tuning implemented to optimise performance. Notably, the model achieved high accuracies in binary and multi-class classification, demonstrating its effectiveness in distinguishing between various types of cyber threats. Al Nuaimi et al. (2023) evaluated the performance of six different algorithms—J48, PART, BayesNets,

AdaBoost, LogitBoost, and Attribute Selected Classifier (ASC)—on the Edge-IIoTset dataset for binary-class and multi-class intrusion detection in IoT/IIoT systems. Automated techniques from WEKA were employed for hyperparameter tuning. The models used achieved an accuracy higher than 85.40%.

Samin et al. (2023) evaluated Naïve Bayes and Decision Tree classifiers on the Edge-IIoTset to detect IoT cyber-attacks. Through manual feature selection, they reduced the number of significant features in the dataset to 46. The experimental results demonstrated that the DT classifier outperformed the NB model, achieving an accuracy of 72%, while NB achieved a lower accuracy of 47%. Ullah et al. (2023) introduced a multi-head attention-based gated recurrent unit (MAGRU) to detect malicious activities in IIoT environments. The MAGRU was evaluated using two datasets: Edge-IIoTset and MQTTset. The authors used the SMOTE technique for data balancing and the extremely gradient boosting (XGBoost) model for feature selection in their framework, filtering out significant features with an importance score greater than zero. This process resulted in 31 features from the Edge-IIoTset and 20 features from the MQTTset. The proposed model achieved high results in precision, recall, F1-score, and accuracy across both datasets.

Ferrag et al. (2022), who introduced the Edge-IIoTset dataset, used centralised learning and federated learning to evaluate their effectiveness in detecting and identifying malicious activities in IoT and IIoT environments. In centralised learning, they employed SVM, DT, KNN, RF, and DNN for binary-class and multiple-class (6 and 15 classes) classification. The authors reduced the 63 features to 46 significant features using manual approaches and implemented hyperparameter tuning on the models used. The results show that the models achieved a high accuracy rate for binary-class classifiers. However, the multi-class classification results were considerably lower, ranging from 67.11% to 85.62% using DT, RF, SVM, and KNN, with the highest being 96.01% for DNN, and between 92% to 96% in Federated learning.

As shown in Table 2.6, the NSL-KDD, UNSW-NB15 and BoT-IoT datasets are often used to train models for multilayer DDoS attack detection. Other available public datasets such as CICDDoS2019, CICIDS2017, KDD CUP 1999, BoTNetIoT, NIMS botnet, MQTTset, APA-DDoS, ToN-IoT and Edge-IIoTset are used for the detection of DoS, DDoS, MITM, and other

attacks. However, not all of these datasets are related to IoT environments. Table 2.7 presents a brief analysis of these datasets.

Table 2.6. NSL-KDD and BoT-IoT dataset.

Ref.	ML Techniques	IoT Attacks	Other Datasets
NSL-KDD dataset			
Aljawarneh et al. (2018)	NB, J48, Random Tree, REP Tree, Meta Pagging, Decision Stump, AdaBoost	DDoS, Probe, R2L, U2R	—
Liang et al. (2019)	DNN	DDoS, Probe, R2L, U2R	—
Tang et al. (2016)	DNN	DDoS	—
AL-Hawawreh et al. (2018)	DAE and a DNN	Backdoors, DoS, Reconnaissance, Worms, DDoS, Probe, R2L, U2R	UNSW-NB15
Tama et al. (2019)	Rotation Forest and Bagging	Backdoors, DoS, Reconnaissance, Worms, DDoS, Probe, R2L, U2R	UNSW-NB15
BoT-IoT dataset			
Shafiq et al. (2020)	NB, RF, DT, BN, RF	DDoS	—
Nimbalkar et al. (2021)	JRip	DDoS	KDD Cup 1999
Ferrag et al. (2020)	REP Tree, Jrip, Forest PA	DDoS/DoS, slowloris, Port Scan, Slowhttptest, GoldenEye, Heartbleed, Infiltration	CICIDS2017
Alhowaide et al. (2021)	15 Different Classifiers	DDoS and zero-day	NSL-KDD, UNSW-NB15, BoTNetIoT
UNSW-NB15 dataset			
Ahmad et al. (2021)	RF, SVM, ANN	reconnaissance, DoS, wormhole, backdoor	—
Moustafa et al. (2019)	NB, DT and ANN	botnets	NIMS botnet
Chkirbene et al. (2020)	CART and RF	reconnaissance, DoS, wormhole, backdoor	—
ToN-IoT dataset			
Alotaibi and Ilyas (2023)	RF, DT, Logistic Regression, and KNN	MITM,DoS,DDoS, ransomware, password cracking, scanning, XSS injection, backdoor	—
Kethineni and Pradeepini (2024)	CNN and Bi-GRU	DDoS	APA-DDoS
Gad et al. (2021)	LR, NB, DT, SVM, KNN, RF, AdaBoost, XGBoost	VANETs Cyber attacks	NSL-KDD dataset
Edge-IIoTset dataset			
Keserwani et al. (2023)	DT, RF, CatBoost, XGBoost	IoT/IIoT Cyber attacks	—
Tareq et al. (2022)	DenseNet and Inception Time	IoT/IIoT Cyber attacks	ToN-IoT, and UNSW-NB15
Khacha et al. (2022)	CNN and LSTM	IoT/IIoT Cyber attacks	—
Al Nuaimi et al. (2023)	J48, PART, BayesNets, AdaBoost, LogitBoost, Attribute Selected	IoT/IIoT Cyber attacks	—
Samin et al. (2023)	DT, NB	IoT/IIoT Cyber attacks	—
Ullah et al. (2023)	XGBoost, multi-head attention, and gated recurrent	IoT/IIoT Cyber attacks	MQTTset
Ferrag et al. (2022)	SVM, DT, KNN, RF, ANN, and federated learning	Injection, malware, DoS, DDoS, MITM, information gathering	—

As shown in Table 2.7, datasets like KDDCUP 99, NSL-KDD, UNSW-NB15, BoT-IoT, BoTNeT-IoT, and CICDDoS2019 are limited to only one type of multilayer attack, specifically DoS/DDoS. In contrast, datasets such as CICIDS2017, ToN-IoT, and Edge-IIoTset cover a broader range of IoT attacks, including more sophisticated multilayer threats like SQL Injection, XSS, and MITM.

Table 2.7. Analysis of datasets used for detecting IoT attacks.

Dataset	Total Number of Attacks	Multi-layer Attacks
KDDCUP 99	4	DoS
NSL-KDD	4	DoS
UNSW-NB15	9	DoS
CICIDS2017	14	DoS, XSS, SQL Injection
BoT-IoT	10	DoS, DDoS
BoTNeT-IoT	2	DoS, DDoS
CICDDoS2019	12	DDoS
ToN-IoT	9	DoS/DDoS, Password Cracking Injection, XSS, MITM
Edge-IIoTset	14	DoS/DDoS, SQL Injection, XSS, MITM, Password Cracking

2.5 CHAPTER SUMMARY

As the current trend of the industrial revolution is based on Industry 4.0, the applications of IoT, smart automation, and web 3.0 have been diversified and driven by AI as a backbone. Each layer of the IoT architecture is now exposed to different types of security threats. The future of the IoT relies on a robust security framework, as protecting IoT systems from these attacks is a challenging task, especially when those attacks can affect multiple layers.

The literature review presented various types of security attacks and provided a taxonomy of attacks that can compromise multiple layers of IoT systems. Developed technologies and computational frameworks have been reviewed to tackle these attacks, and datasets have been investigated for training and evaluating ML models in terms of their features and the types of attacks they can be used for. Despite the advantages of recent ML approaches and the high detection accuracy in detecting multilayer attacks, gaps and challenges still exist, as identified through this research.

- **Benchmark Dataset**

One challenge in the detection of multilayer IoT security attacks is the lack of benchmark datasets. Many researchers have had to create their own datasets for training and testing their models, making it difficult to compare the effectiveness of models proposed by different researchers, as well as view the datasets and the extracted features. While the majority of the ML models reported in the literature have achieved good accuracy figures, it is challenging to verify whether the models will perform well on other datasets.

Very few datasets that include both normal traffic and multilayer attacks have been made publicly available. However, these are outdated and do not accurately represent the characteristics of today's traffic, as seen in newer devices and network services. Even more recent databases rarely contain any IoT traffic. It is essential to obtain datasets with traffic produced by IoT devices to develop anomaly detection systems that can profile legitimate traffic for IoT devices.

Furthermore, available datasets often have a limited number of features relevant to one or a subset of security attacks. For example, the NSL-KDD, BoT-IoT and UNSW-NB15 datasets have been commonly used to identify multilayer IoT attacks such as DDoS attacks, but it is unclear whether the data contains all the relevant features for other types of multilayer attacks. Creating and publishing benchmark datasets would benefit the entire research community.

- **Feature Extraction on Semi-structured Data**

ML algorithms require structured data as input, so the data has to be transformed into a structured table before being input into ML algorithms. This is challenging because IoT data often contains both structured and semi-structured information. Unlike structured data, where each attribute can be regarded as a feature for analysis, it is non-trivial to identify meaningful features from semi-structured information and transform the relevant data into structured columns. Developing effective computational methods that can identify and extract meaningful or relevant features from the semi-structured information is one of the keys to the success of intrusion detection. Human expert knowledge and machine-learned knowledge can both be helpful in the feature identification process.

- **Flexible ML Models for the Detection of Different Types of Attacks**

Some of ML models are less flexible in detecting new attacks, new types of attacks, or attacks targeting new devices. Additionally, the detection complexity and training time for some models are high for real-time IoT systems.

Another characteristic of multilayer IoT attacks is their complexity. A large variety of attacks exist, each of which may be relevant to a specific set of features that may or may not exist in the data. There is a need for a one-size-fits-all solution that works effectively in detecting all possible multilayer IoT attacks. A possible solution to this problem is to develop an approach that first helps identifies key features and patterns that characterise different types of attacks. By then identifying the common features between these attacks, such information can be used as prior knowledge before training ML models for intrusion detection. This approach enables the development of flexible machine learning models that are not only created to detect the studied multilayer attacks but also robust enough to adapt to new types of these multilayer attacks by focusing on these common features.

- **Computational Framework**

Like many other knowledge discovery applications, IoT intrusion detection involves several stages, including focusing, data processing, data transformation, modelling, and evaluation. Most existing IDS rely on pre-processed data and pre-trained models, and therefore do not include components to cover all the above-mentioned stages. Many of them lack flexibility in feature selection, feature weighting, scaling, selection of distance metrics, tuning hyperparameters, handling real-time threats, and adjusting the models according to evolving changes. A generic computational framework that covers every stage of the knowledge discovery process with flexible options will provide a more systematic and effective solution to the problem.

Although these existing feature selection methods can extract significant features from datasets, it remains unclear which is the most effective. Also, each feature selection method has its pros and cons, even the accuracy of those methods invariably depends on the training dataset. The aim of this research is to ascertain the optimal number of significant features, irrespective of all feature

selection methods used, by incorporating multiple feature selection methods in the decision-making process.

The above-mentioned studies primarily focus on specific attacks on specific applications (e.g., phishing site detection in smart cities) or outdated datasets (e.g., KDD '99 and CIDD). There is a gap in research regarding the generalisability of these feature selection and weighting methods across diverse IoT environments and attack vectors, especially for multilayer attacks. Additionally, there is a notable absence of discussion on incorporating human expertise in the loop of feature selection and weighting processes, which could enhance the interpretability and reliability of the detection models, especially in complex scenarios where automated methods might struggle. Despite the severe impact of these multilayer attacks on various IoT devices, there is currently no comprehensive framework for their detection and mitigation.

In this research, the latest and most comprehensive benchmark IoT cybersecurity dataset Edge-IIoTset dataset is utilised. While previous studies have employed this dataset to detect intrusions in IoT and industrial IoT systems Keserwani et al.(2023); Tareq et al, (2022); Khacha et al., 2022; Al Nuaimi et al. (2023); Samin et al. (2023); Ullah et al. (2023); Ferrag et al. (2022) these studies have not extensively explored the detection and classification of multilayer attacks, nor have they extensively applied feature weighting techniques. Instead, most studies have explored manual feature selection methods, with only one exception employing XGBoost as the feature selection method. Additionally, a limited number of studies have investigated the tuning of the hyperparameters of the models used.

3 METHODOLOGY FOR ADVANCED IOT MULTILAYER ATTACKS DETECTION

This chapter presents the comprehensive methodology employed in developing the IoT multilayer detection and identification system. As shown in Figure 3.1, it covers the experimental tools used, the structural framework of methodology, and its specific implementation using the Edge-IIoTset dataset.

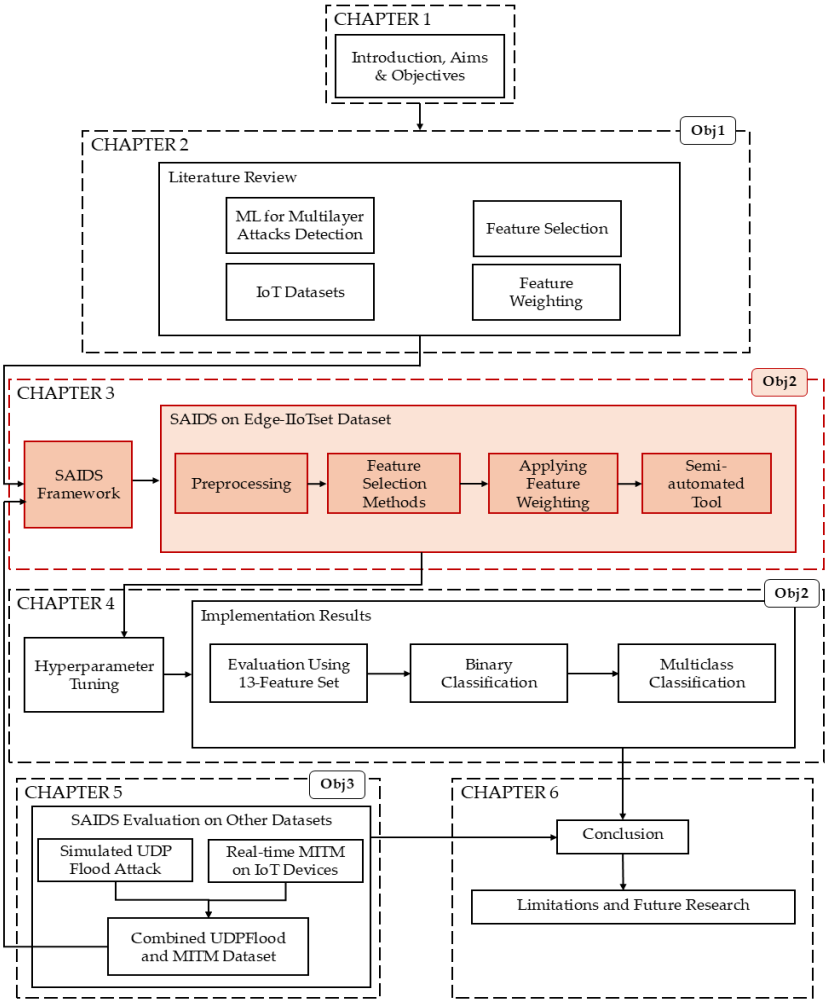


Figure 3.1. Thesis structure showing Chapter 3's placement within the overall project.

3.1 METHODOLOGICAL FRAMEWORK

This section discusses the procedural methodology for developing the IoT multilayer SAIDS system. This includes incorporating systematic integration of feature selection, feature weighting, and a semi-automated approach in the overall framework where human expertise and machine learning algorithms work together, as illustrated in Figure 3.2. The term "semi-automated" reflects the system's use of both machine learning algorithms and human insight, combining automation's efficiency with expert feedback to ensure accuracy and relevance. This approach enhances the overall efficacy of the IoT multilayer detection system through the strengths of both human judgment and algorithmic precision.

To support the development and evaluation of the SAIDS, Python 3 on Google Colab, a cloud-based Jupyter Notebook environment, was selected. This platform was chosen for its robust capabilities in machine learning, data processing, feature selection, feature weighting, and visualisation, all of which are critical factors for the successful implementation of the proposed framework.

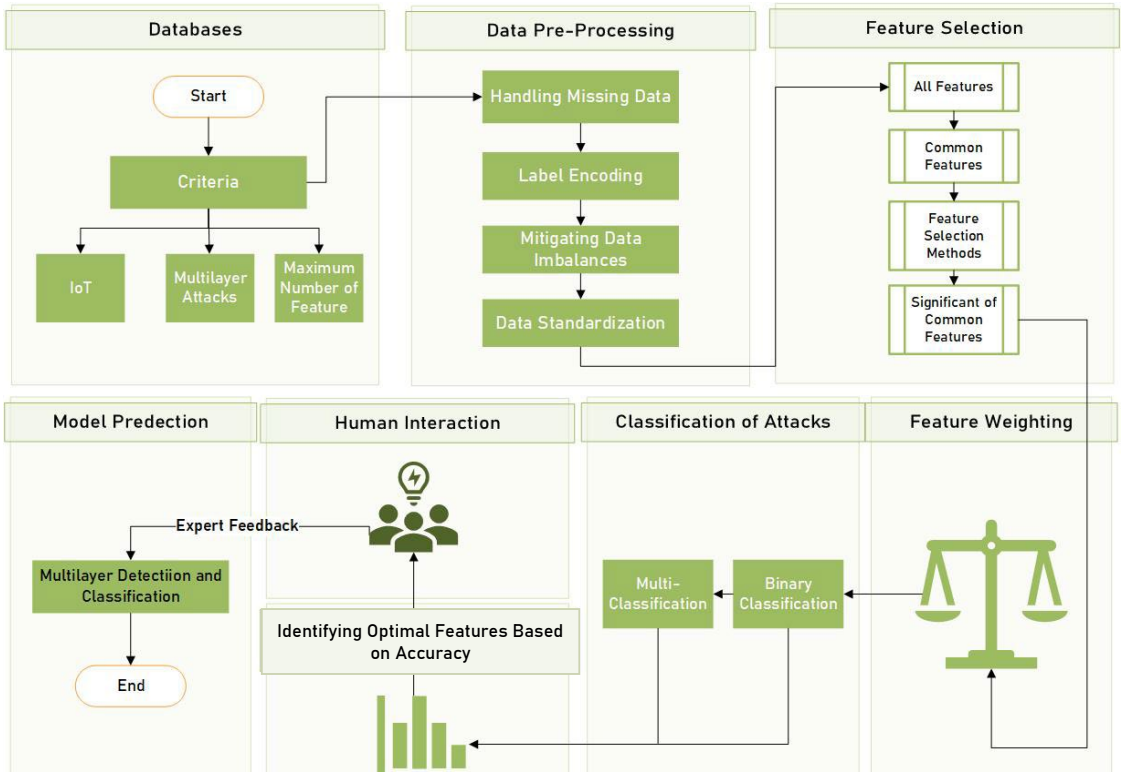


Figure 3.2. Semi-automated intrusion detection system (SAIDS).

A. Datasets Selection

The methodology starts with selecting the data from various sources while considering specific criteria. These criteria include IoT-specific recent datasets, datasets related to multilayer attacks, and considerations for the maximum number of features that can be processed by ML algorithms.

B. Data Pre-Processing

Pre-processing is critical to transform the raw dataset for further analysis, ensuring that algorithms have quality data to learn from. This stage includes handling missing data and converting categorical data into a numerical format understandable by machine learning algorithms through label encoding. Additionally, it transforms semi-structured IoT data into structured data by utilising human expert knowledge to extract meaningful features from raw data. It also involves standardising data to a common scale across all features, typically resulting in a dataset with a mean of zero and a standard deviation of one or normalising the data to a range between 0 and 1. Both techniques help in improving the performance and speed of machine learning algorithms. Finally, it involves mitigating data imbalances to avoid bias towards a particular class, which is common in IoT security where attacks are rarer than normal events.

C. Feature Selection

After pre-processing, the next step is to filter the significant features to predict the target variable from the dataset. Irrelevant or redundant features may increase computational complexity, and sometimes negatively impact the model's performance. The process begins with the identification of features common to multilayer attacks. Subsequently, various feature selection methods are applied to determine the most significant of common features.

The approach of identifying common features between multilayer attacks has been selected in the SAIDS framework because, as mentioned in section 1.2, multilayer attacks share several common characteristics. For example, multiple attack types, such as **HTTP Flooding**, **Session Hijacking**, and **Code Injection**, exploit vulnerabilities in the **HTTP protocol**, while attacks like **SYN Floods** and **MITM** attacks target the incomplete **TCP handshake**. By focusing on these shared features, the SAIDS framework can detect a range of attack types that exhibit similar underlying behaviors.

This not only improves the efficiency of the detection system but also reduces the computational complexity, as the model is able to concentrate on the most relevant and impactful features.

D. Feature Weighting

During this stage, weights are assigned to each of the selected features, aiding the machine learning algorithms to prioritise the most significant features throughout the learning process.

E. Classification of Attacks

Machine Learning models were used to help in identifying the most important features for both classification tasks, namely; binary and multiclass classifications. This stage is essential for the development of the semi-automated tool.

F. Identifying Optimal Features Based on Accuracy

All these stages, starting from data pre-processing to classification, are bundled as a package. A semi-automated tool is created for visualising the impact of sequentially adding top-weighted features into ML classifiers. This tool aids in guiding the selection of the most significant features that contribute to a higher accuracy rate. The visualisations assist the feature selection for both binary classification and multiclass classification.

G. Human Interaction

The inclusion of expert feedback in the SAIDS by integrating human and machine learning approaches, where cybersecurity experts evaluate the semi-automated tool's output and provide feedback. This feedback is essential for the models' enhancement, as it can be used to adjust the feature selection process, and model parameters, or to interpret the results from the ML algorithms.

H. Models Predictions

The proposed framework incorporates two classification tasks: binary classification to distinguish between normal IoT traffic or malicious attacks, and multi-classification, for predicting multiple types of IoT attacks. If the IoT traffic is flagged as malicious to multilayer attacks, the system

further investigates to identify the type of multilayer attacks through multiclass classification. Also, the system is designed to easily integrate and add classifiers as needed. For the specific use case in this study, classifiers such as Decision Tree, K-Nearest Neighbors, Naive Bayes, Random Forest, and Artificial Neural Network were utilised as they are suitable for this prototype.

3.2 Implementation of SAIDS on the Edge-IIoTset Dataset

This section describes the practical application of the proposed methodology for the detection and classification of multilayer attacks within IoT networks using the Edge-IIoTset dataset, as described in Figure 3.3. The rationale behind selecting this dataset is that it is the latest and most comprehensive benchmark IoT cybersecurity dataset, including most of the multilayer attacks from real traffic of IoT devices.

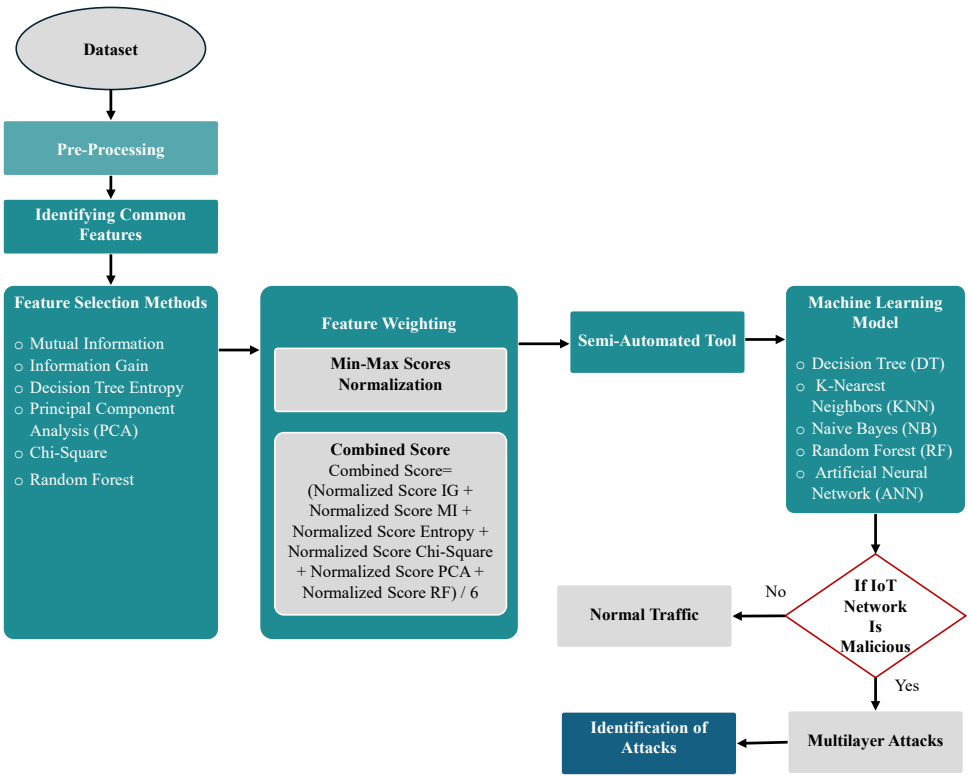


Figure 3.3. Implementation of SAIDS to Edge-IIoTset dataset.

The following pseudocode explains the Implementation of SAIDS to Edge-IIoTset dataset in more detail. Refer to Appendix 2 for more information on the tools used.

```
Algorithm: SAIDS
Input:    Dataset D
          Feature Set F = {f1, f2, ..., fM}
          Top Features to Select T
Output:   Optimal Features F_opt
          Best ML Model M_best
Begin:    Step 1: Preprocess (D)
          Step 2: F_common = IdentifyCommonFeatures(D)
          Step 3: Initialise containers:
                  F_ranked = {}
                  F_normalised = {}
                  Scores_Sum = {}
                  F_combined = {}
          Step 4: Feature Selection and Normalisation:
                  for each fs_method in [MI, IG, DTE, Chi², PCA, RF]:
                  F_scores = FeatureSelection (D, F_common, method=fs_method)
                  F_ranked[fs_method] = SortFeatures (F_scores, descending=True)
                  F_normalised[fs_method] = Normalise(F_scores)
                    for each feature in F_normalised[fs_method]:
                    if feature not in Scores_Sum:
                       Scores_Sum[feature] = F_normalised[fs_method] [feature]
                    else:
                            Scores_Sum[feature] += F_normalised[fs_method] [feature]
          Step 5: Calculate Combined Scores:
                  for feature in Scores_Sum:
                    F_combined[feature] = Scores_Sum[feature] / len(F_normalised)
          Step 6: Sort Features by Combined Scores:
                  F_sorted = SortFeatures (F_combined, descending=True)
                  F_weighted = F_sorted
          Step 7: Model Training and Selection:
                  M_best_acc = 0
                  F_opt = []
                  M_best = None
                  for N in range (1, T+1):
                    F_subset = F_weighted [: N]
                    for M in [DT, KNN, NB, RF, ANN]:
                          M_tuned = HyperparameterOptimisation (M, F_subset)
                          M_trained = Train (M_tuned, F_subset)
                          M_metrics = Test (M_trained, F_subset)
                          if M_metrics['accuracy'] > M_best_acc:
                                M_best_acc = M_metrics['accuracy']
                                M_best = M_trained
                                F_opt = F_subset
                           end if
                         end for
                  end for
          Step 8: VisualiseImpact(F_opt)
          Step 9: HumanExpertReview (F_opt, M_best)
          Step 10: FinaliseModel (M_best, F_opt)
                  Return (F_opt, M_best)
End
```

3.2.1 Data Pre-processing

This stage focuses on preparing the Edge-IoTset dataset for analysis. As discussed in section 2.4, the Edge-IIoTset dataset has a total of 63 features, including the "Attack_label" and "Attack_type" features. These features are presented in Table 3.1. The following steps outline the data pre-processing procedure.

Table 3.1. List of 63-features of the Edge-IIoTset dataset.

N.	Feature Name	Description	N.	Feature Name	Description
1	frame.time	Arrival Time	33	tcp.payload	TCP payload
2	ip.src_host	IP Source Host	34	tcp.seq	TCP Sequence Number
3	ip.dst_host	IP Destination Host	35	tcp.srcport	TCP Source Port
4	arp.dst.proto_ipv4	ARP Target IP address	36	udp.port	UDP Source or Destination Port
5	arp.opcode	ARP Opcode	37	udp.stream	UDP Stream index
6	arp.hw.size	ARP Hardware size	38	udp.time_delta	UDP Time since previous frame
7	arp.src.proto_ipv4	ARP Sender IP address	39	dns.qry.name	DNS Name
8	icmp.checksum	ICMP Checksum	40	dns.qry.name_len	DNS Name Length
9	icmp.seq_le	ICMP Sequence Number	41	dns.qry.qu	DNS "QU" question
10	icmp.transmit_timestamp	ICMP Transmit Timestamp	42	dns.qry.type	DNS Type
11	icmp.unused	ICMP Unused	43	dns.retransmission	DNS Retransmission
12	http.file_data	HTTP File Data	44	dns.retransmit_request	DNS query retransmission
13	http.content_length	HTTP Content length	45	dns.retransmit_request_in	DNS Retransmitted request
14	http.request.uri.query	HTTP Request URI Query	46	mqtt.conack.flags	MQTT Acknowledge Flags
15	http.request.method	HTTP Request Method	47	mqtt.conflag.cleansess	MQTT Clean Session Flag
16	http.referer	HTTP Referer	48	mqtt.conflags	MQTT Connect Flags
17	http.request.full_uri	HTTP Full request URI	49	mqtt.hdrflags	MQTT Header Flags
18	http.request.version	HTTP Request Version	50	mqtt.len	MQTT Msg Len
19	http.response	HTTP Response	51	mqtt.msg_decoded_as	MQTT Message decoded as
20	http.tls_port	HTTP Unencrypted HTTP protocol detected over encrypted port	52	mqtt.msg	MQTT Message
21	tcp.ack	TCP Acknowledgment Number	53	mqtt.msgtype	MQTT Message Type
22	tcp.ack_raw	TCP Acknowledgment number (raw)	54	mqtt.proto_len	MQTT Protocol Name Length
23	tcp.checksum	TCP Checksum	55	mqtt.protoname	MQTT Protocol Name
24	tcp.connection.fin	TCP Connection finish (FIN)	56	mqtt.topic	MQTT Topic
25	tcp.connection.rst	TCP Connection reset (RST)	57	mqtt.topic_len	MQTT Topic Length
26	tcp.connection.syn	TCP Connection establish request (SYN)	58	mqtt.ver	MQTT Version
27	tcp.connection.synack	TCP Connection establish acknowledge	59	mbtcp.len	Modbus Length
28	tcp.dstport	TCP Destination Port	60	mbtcp.trans_id	Modbus Transaction Identifier
29	tcp.flags	TCP Flags	61	mbtcp.unit_id	Modbus Unit Identifier
30	tcp.flags.ack	TCP Acknowledgment	62	Attack_label	0 indicates normal and 1 indicates attacks
31	tcp.len	TCP Segment Len	63	Attack_type	Attack categories

A) Renaming Features

The "Attack_label" feature includes both normal and abnormal traffic (attacks), presented as 0 and 1 respectively in the dataset. This feature was renamed into "Label".

B) Feature Modification

The "frametime" feature was split into two separate attributes: "frame.time_WithoutIP" and "frame.time_WithIP". This division was essential because the original attribute contained both IP addresses and timestamps and it increased the total number of features in the dataset from 63 to 64.

- "frame.time_WithoutIP" includes only the timestamps, with the IP addresses replaced with 1. The timestamps were converted into integers using the datetime package.

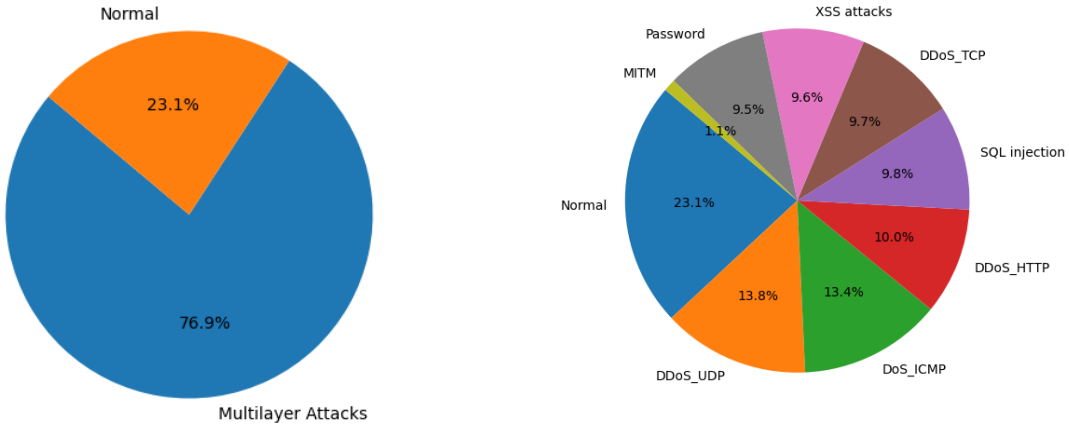
- "frame.time_WithIP" had 0 values replaced with 1, 6 values replaced with 2, timestamps replaced with 3, and IP addresses converted into integers using the Excel Add-in (ip2location-ip-conversion).

C) Categorical Data Encoding

Label encoder which is inspired by (Rashid et al., 2023) was used to encode the categorical variables.

- The "Attack_type" feature, which includes 14 different types of IoT cyber-attacks; however, the aim of this research is to focus on multilayer-related attacks. This means narrowing down the analysis to eight specific attacks and then encoding them. The eight multilayer attacks are: DDoS_TCP (DDoS TCP SYN Flood), DDoS_UDP, DDoS_HTTP, DoS_ICMP, MITM (ARP and DNS Spoofing), Password (Password Cracking), SQL injection, and XSS attacks. The distribution of the dataset's traffic can be seen in Figure 3.4a and 3.4b.

- DDoS_TCP was encoded as 1, DDoS_UDP as 2, DDoS_HTTP as 3, DDoS_ICMP as 4, SQL injection as 5, XSS attacks as 6, MITM as 7, and Password cracking as 8.



(a) **(b)**

Figure 3.4. Distribution of traffic (a) normal and multilayer attacks (b) normal and attack types.

- In "http.request.method", values (Get, Trace, Post) were encoded into 1, 2, and 3, respectively.

- In "tcp.srcport", values (_googlecast_tcp.local, Desktop_UHF0SF2, Desktop_UHF0SF2.local) were encoded into 1, 2, and 3, respectively.

- In "mqtt.msg", values (32322e, 32342e, 32332e) were encoded into 1, 2, and 3, respectively.

- String values like (http/1.1) in "http.request.version" were replaced with 1, and (MQTT) in "mqtt.protoname" and (Temperature_and_humidity) in "mqtt.topic" were also replaced with 1.

- The "http.request.uri.query" feature values like (id=%28SELECT%20, id=2%20, id=3%20, etc.,) were replaced with 1, 2, 3, etc.

- URLs in the "http.request.full_uri" feature, such as (http://192.168.0.128/dvwa/vulnerabilities/), were replaced with 1, 2, 3, etc.

- String values in the "tcp.options" feature starting with (0101050a, 0101080, etc.,) were converted into 1, 2, 3, etc.

- String values in the "tcp.payload" feature starting with 0, 1, 2, etc., were converted into 100, 101, 102, etc.

- The "http.file_data" feature values like (400 Bad Request, 404 Not Found, etc.,) were converted into 1, 2, 3, etc.

D) Handling IP Addresses

IP addresses in features like "ip.src_host", "ip.dst_host", "arp.dst.proto_ipv4", "arp.src.proto_ipv4", and "http.referer" were converted into numbers using the Excel Add-in (ip2location-ip-conversion).

E) Removing Unnecessary Features

Upon investigation, 10 features (icmp.transmit_timestamp, icmp.unused, http.tls_port, dns.qry.type, dns.retransmission, dns.retransmit_request_in, mqtt.msg_decoded_as, mbtcp.len, mbtcp.trans_id, and mbtcp.unit_id) were found to have zero values in the entire dataset and were therefore removed. Including such features would introduce noise, leading to potential overfitting and reduced model performance. Additionally, removing these features improves computational efficiency and simplifies the dataset, making it easier to manage. This resulted in a reduction of the total features from 64 to 54, including the Label and Attack_type in the Edge-IIoTset dataset.

F) Dataset Splitting

To evaluate the effectiveness of the classification models, the dataset was divided into a training set and a testing set, with 70% of the data allocated for training and 30% for testing, as recommended by (Samin et al., 2023; Ullah et al., 2023).

G) Standardisation and Balancing

The Z-score method was chosen for standardisation to ensure consistency across all features. Standardising the dataset using Z-scores transforms the data in each feature to have a mean of 0 and a standard deviation of 1. This is important for ML algorithms that are sensitive to the scale of data, such as k-nearest neighbours model (Dini et al., 2023). By standardising the features, we ensure that each feature contributes equally to the analysis, preventing features with larger scales from dominating the model.

The Synthetic Minority Over-sampling Technique (SMOTE) was applied to address the issue of imbalanced data distribution, as it is widely used by researchers, as discussed in section 2.1. It increases the representation of the minority class, resulting in a more balanced dataset and enhancing the model's ability to learn from all classes, thereby improving overall performance and robustness (Samin et al., 2023; Maghrabi, 2024).

3.2.2 Feature Selection

This stage includes the identification of common features between multilayer attacks and the utilisation of various feature selection methods on the common features.

3.2.2.1 Identifying common features

This stage aims to identify the features that are commonly found in multilayer attacks. As shown in Figure 3.5, this approach begins by iterating over the "attack_type" feature to separate the data based on the nature of the network activity. Following the categorisation, a list of attributes corresponding to each type of attack was compiled. These features have diverse characteristics, such as traffic volume, packet size, and distinct behavioural patterns, which help distinguish malicious traffic from normal traffic. Then, the frequency of each attribute's occurrence was counted. This quantification step aids in assessing the distribution of features associated with each attack type. Attributes that appear more frequently suggest a pattern that may be characteristic of

a particular form of attack. From the final 54 features of the original 64 features (including "label" and "Attack_type"), 34 features have been identified as common features, as shown in Table 3.2.

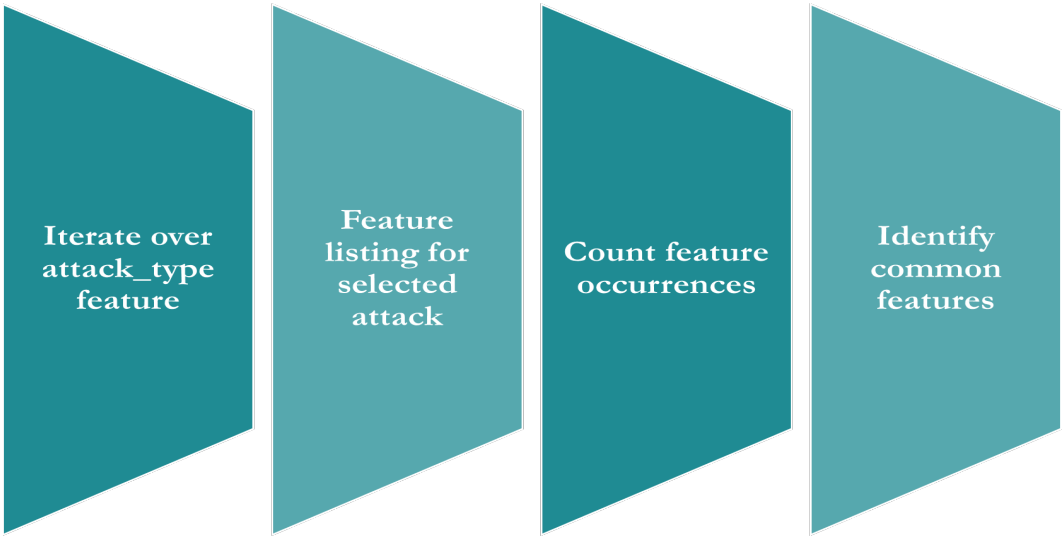


Figure 3.5. Identifying common features between multilayer attacks.

Table 3.2. Thirty-four common features between multilayer attacks.

No.	Feature Name	No.	Feature Name	No.	Feature Name
1	frame.time_WithoutIP	13	arp.dst.proto_ipv4	25	tcp.connection.rst
2	frame.time_WithIP	14	arp.opcode	26	tcp.connection.syn
3	ip.src_host	15	arp.hw.size	27	tcp.connection.synac
4	ip.dst_host	16	arp.src.proto_ipv4	28	tcp.dstport
5	tcp.len	17	http.request.method	29	tcp.flags
6	tcp.options	18	http.request.full_uri	30	tcp.flags.ack
7	tcp.payload	19	http.request.version	31	udp.stream
8	tcp.seq	20	http.response	32	http.file_data
9	tcp.srcport	21	tcp.ack	33	http.content_length
10	udp.port	22	tcp.ack_raw	34	icmp.seq_le
11	udp.time_delta	23	tcp.checksum		
12	dns.qry.name	24	tcp.connection.fin		

3.2.2.2 Integration of Feature Selection Methods

Feature selection plays a crucial role in enhancing the performance of ML models, mitigating the risk of overfitting, and speeding up the training process. By identifying and employing the most

relevant features, models can be trained more effectively to detect complex patterns associated with multilayer attacks in IoT systems (Al Sukhni et al., 2023).

Six feature selection methods were employed in this research to provide a diverse set of techniques for analysing the strength of the relationship between each of the 34 common features of multilayer IoT attacks and the target variable "label". These methods include Mutual Information (MI), Information Gain (IG), Decision Tree Entropy (DTE), Principal Component Analysis (PCA), Chi-Square (Chi^2), and Random Forest (RF). As demonstrated in Chapter 2, these methods have been extensively employed by researchers for IDS due to their effectiveness in handling diverse data types, including both numerical and categorical variables. Additionally, their ability to reduce the dimensionality of the feature space enhances the efficiency of the classification algorithm (Göcs and Johanyák, 2023).

A. Mutual Information (MI)

Mutual information is a measure of information between two random variables and is helpful in detecting both linear and non-linear relationships between variables. The mutual information between two random variables X and Y is calculated using Equation (1) (Sulaiman and Labadin, 2015):

$$MI(X,Y) = \sum p\ (x,y) \log \frac{p(x,y)}{p(x)p(\mathrm{y})} \qquad (1)$$

Where:

- MI(X, Y) represents the mutual information between two variables X and Y.
- p(x, y) is the joint probability for x ∈ X and y ∈Y.

Based on the MI scores in Figure 3.6, the features "frame.time_WithoutIP" and "tcp.dstport" show the strongest relationship with the target variable "label". To accurately decide and select the insignificant features, this research utilised a permutation test as suggested by (François, Wertz and Verleysen, 2006), with the number of permutations set to 1000 to calculate the p-values. The permutation test is a statistical method that assesses the significance of the mutual information score by comparing it to scores derived from data generated under the null hypothesis (where the features and the target variable are independent).

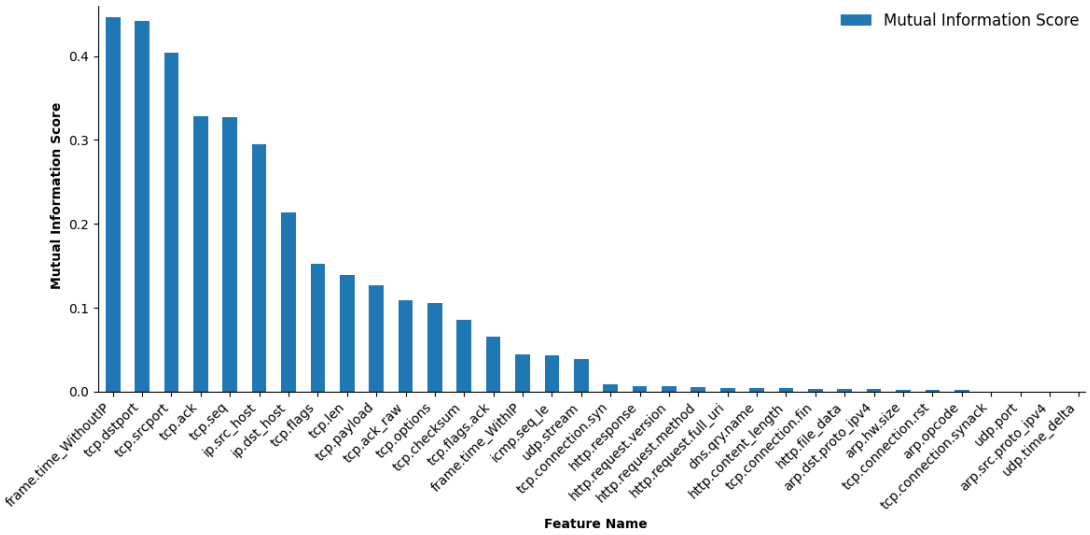


Figure 3.6. Mutual information scores of features for the target variable.

Figure 3.7 presents the 34 common features with their p-values. Features with p-values greater than or equal to 0.05 were considered to be irrelevant and excluded from further analysis. This threshold was chosen because the p-values of eight features were higher than their mutual information scores, as shown in Table 3.3, indicating that their relationship with the target "label" variable was not significant. As a result, features such as "'tcp.connection.synack", "arp.opcode", "udp.time_delta", "arp.src.proto_ipv4", "udp.port", "http.file_data", "tcp.connection.fin", and "arp.hw.size" were excluded. This approach of implementing the permutation test resulted in reducing the significant features to 26.

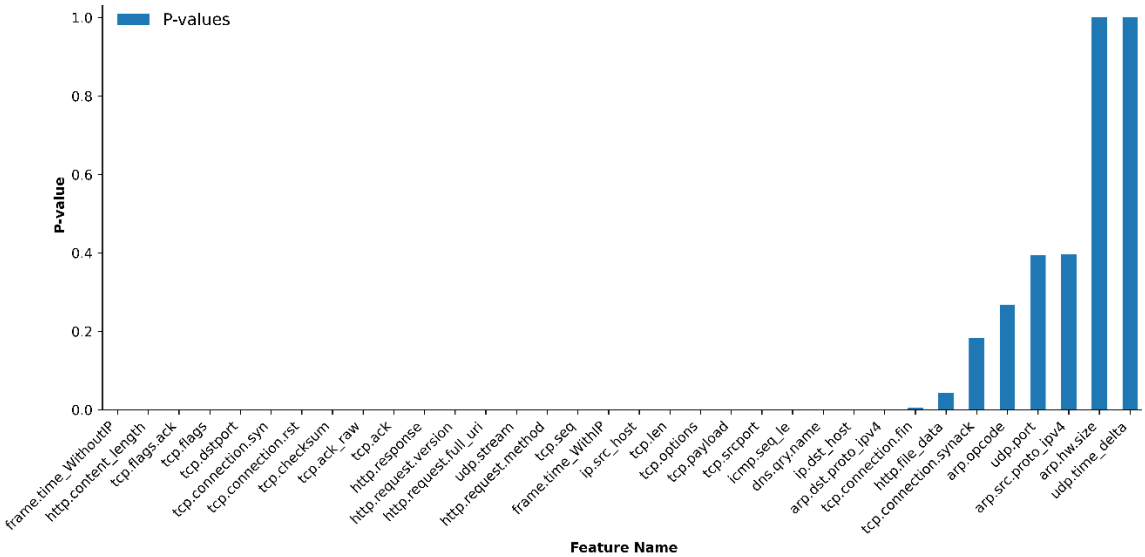


Figure 3.7. P-values of features for significance testing with the target variable.

Table 3.1. Feature selection decisions based on mutual information scores and p-values.

Feature Name	Mutual Information Score	P-Value	Decision
frame.time_WithoutIP	0.446368	0	Keep
tcp.dstport	0.441615	0	Keep
tcp.srcport	0.403821	0	Keep
tcp.ack	0.328572	0	Keep
tcp.seq	0.32686	0	Keep
ip.src_host	0.294443	0	Keep
ip.dst_host	0.214121	0	Keep
tcp.flags	0.152636	0	Keep
tcp.len	0.139283	0	Keep
tcp.payload	0.12627	0	Keep
tcp.ack_raw	0.109191	0	Keep
tcp.options	0.105654	0	Keep
tcp.checksum	0.085841	0	Keep
tcp.flags.ack	0.065424	0	Keep
frame.time_WithIP	0.044193	0	Keep
icmp.seq_le	0.043087	0	Keep
udp.stream	0.038415	0	Keep
tcp.connection.syn	0.009105	0	Keep
http.response	0.006775	0	Keep
http.request.version	0.006665	0	Keep
http.request.method	0.005857	0	Keep
http.request.full_uri	0.004189	0	Keep
dns.qry.name	0.004078	0	Keep
http.content_length	0.003954	0	Keep
tcp.connection.fin	0.00332	0.005	Exclude
http.file_data	0.002897	0.043	Exclude
arp.dst.proto_ipv4	0.002617	0	Keep
arp.hw.size	0.002185	1	Exclude
tcp.connection.rst	0.001981	0	Keep
arp.opcode	0.001852	0.267	Exclude
tcp.connection.synack	0.000222	0.183	Exclude
udp.port	0.00009	0.395	Exclude
arp.src.proto_ipv4	0	0.397	Exclude
udp.time_delta	0	1	Exclude

B. Information Gain (IG)

Information Gain quantifies how much information each feature provides for predicting the target. Features that provide more information have a higher information gain value and can be considered significant, while those with lower values can be eliminated. This research utilised the InfoGainAttributeEval attribute evaluator, alongside the ranker search method in Weka, to consider all possible subsets of features, evaluate them based on their information gain with respect to the target variable "Label", and then list the results in rank order based on their information gain scores. The formula for calculating IG is as follows (Kurniabudi et al., 2020).

$$IG(H,C) = Entropy(H) - \sum_{Values(C)} \frac{|Hv|}{|H|} Entropy(Hv) \qquad (2)$$

Where:

- C represents the feature, while H stands for the dataset.
- v represents a potential value for feature C, and Values(C) is the collection of all possible values for feature C.
- |Hv| is the count of samples associated with the value v.
- |H| indicates the total number of samples in the dataset.
- Entropy (Hv) is entropy calculated for the samples of v.

Entropy(H) is measured using the following formula, where c represents the count of values within the classification class and Pi represents the number of samples for class i (Kurniabudi et al., 2020).

$$Entropy(H) = \sum_{i}^{c} -P_i log_2 P_i \qquad (3)$$

Figure 3.8 shows that "arp.src.proto_ipv4", "arp.opcode", and "arp.hw.size" have information gain scores of zero, indicating that they do not provide any information that can be used to classify the data. The remaining 31 features are considered the most significant for the classification task.

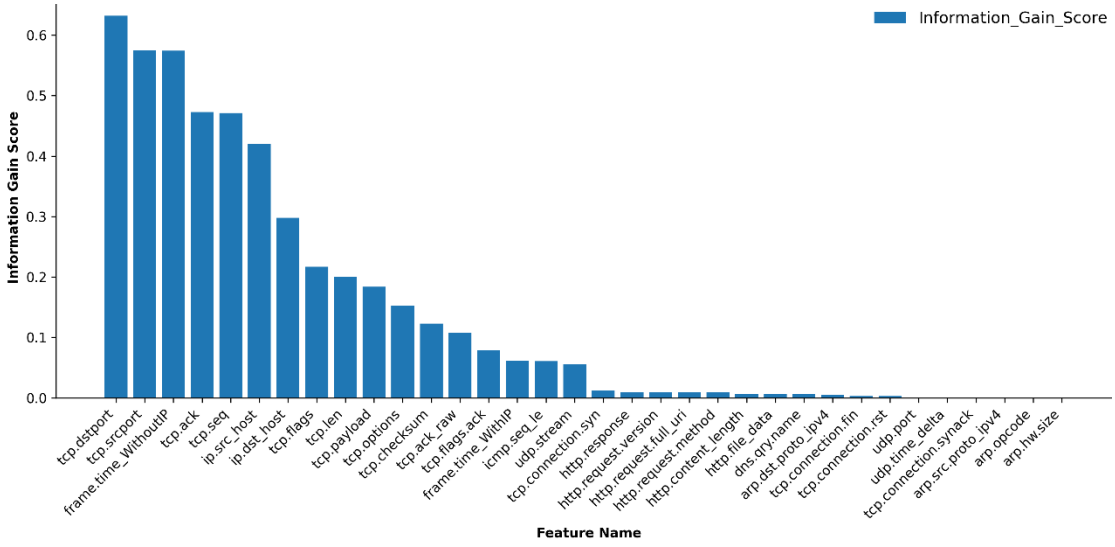


Figure 3.8. Information gain scores of features for the target variable.

C. Decision Tree Entropy (DTE)

DTE is one of the feature selection methods utilised in this research. This method involves training a Decision Tree classifier using the entropy criterion to assess the importance of each feature. The

results in Figure 3.9 show that only 7 out of 34 features are considered significant using this technique, with "tcp.srcport" and "tcp.dstport" having relatively higher entropy scores. Additionally, "ip.src_host", "frame.time_WithoutIP", "frame.time_WithIP", "udp.time_delta", and "ip.dst_host" suggest an essential role in the classification process.

It is important to note that the remaining features with an entropy score of 0.000000 had no impact on the decision-making process of the classifier and do not contribute to distinguishing between different classes.

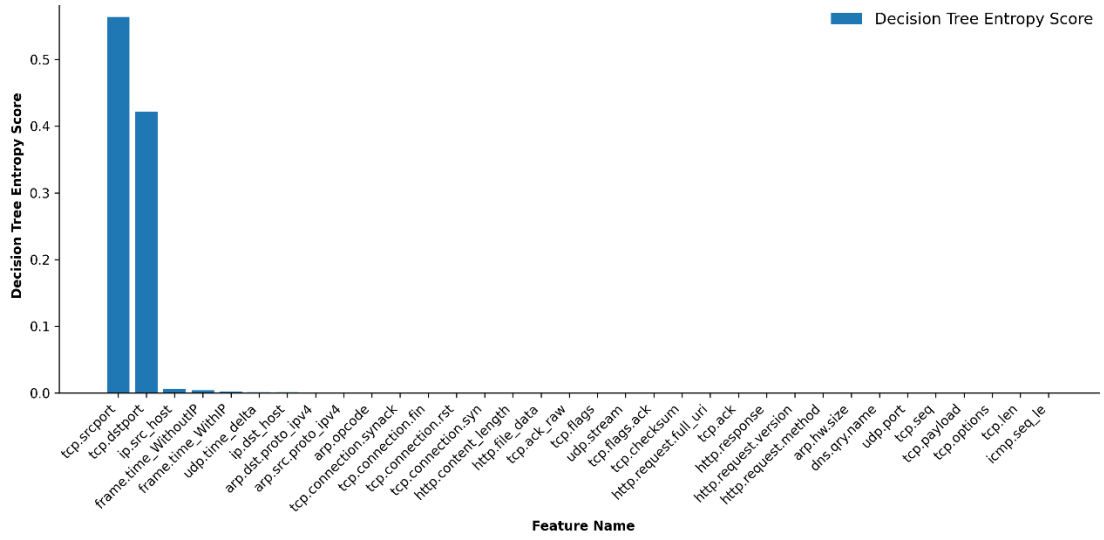


Figure 3.9. Decision tree entropy scores of features for the target variable.

D. Chi-Square (Chi²)

Chi-square is another method used in this research to determine the statistical relationship between each feature and the target variable. This method involves calculating the chi-square scores for each feature. Ranking the features based on these scores allows researchers to identify the features that have a stronger association with the target variable "Label". A greater Chi² score indicates a stronger significance between the feature and the target variable. The following formula is used to calculate the Chi² scores (Hurtik, Molek and Perfilieva, 2020).

$$Chi^2 = \sum \frac{(O - E)^2}{E} \qquad (4)$$

Where:

- O is the observed value and E is the expected value.

The results of Chi² in Figure 3.10 show that all features are important, with higher Chi² scores indicating a stronger connection to the target. For example, "frame.time_WithIP" and "tcp.ack_raw" have higher Chi² scores and demonstrate a stronger relationship with the target variable "Label".

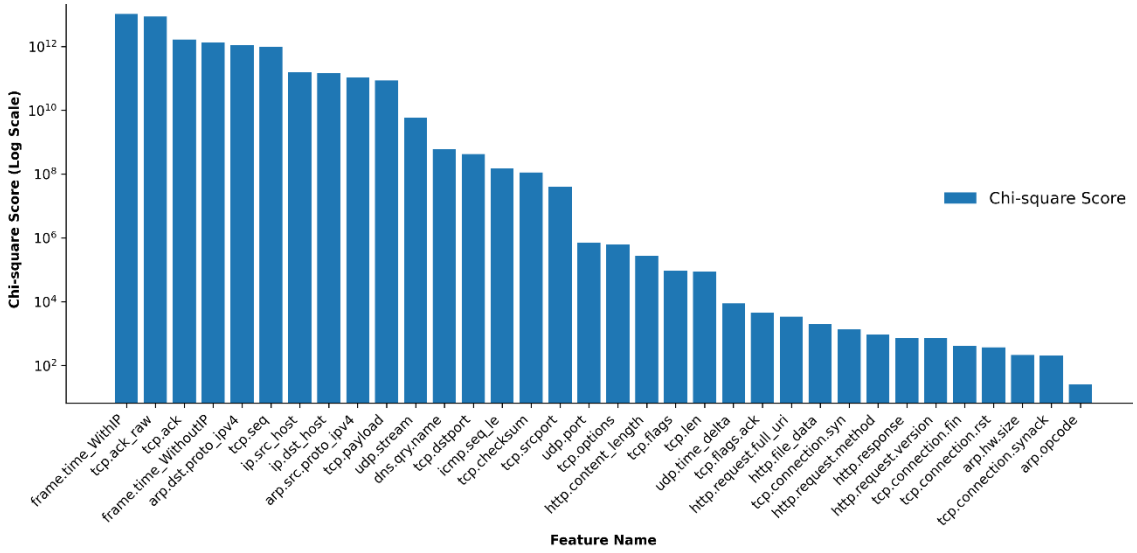


Figure 3.10. Chi-square scores of features for the target variable.

E. Principal Component Analysis (PCA)

Principal Component Analysis is employed in this research to decrease the dimensionality of the feature space in order to improve the efficiency of the classification algorithms. PCA scores are calculated using the covariance matrix (Bolboacă et al., 2011). A visualisation of the feature names and their PCA scores using a bar chart is conducted, as shown in Figure 3.11(Bolboacă et al., 2011) Figure 3.11 shows that the feature "frame.time_WithoutIP" has the highest PCA score of around 0.142, followed by 'frame.time_WithIP', which has a score of 0.09. These two features are the most influential in the reduced feature space. On the other hand, features with lower PCA scores, closer to zero, have less impact on the overall variance in the dataset. For example, "icmp.seq_le" has a PCA score of 3.118535e-18, which is extremely close to zero, indicating that it does not contribute significantly to the target variable "Label" and is excluded. However, the remaining 33 out of 34 features are significant and provide useful information for classification.

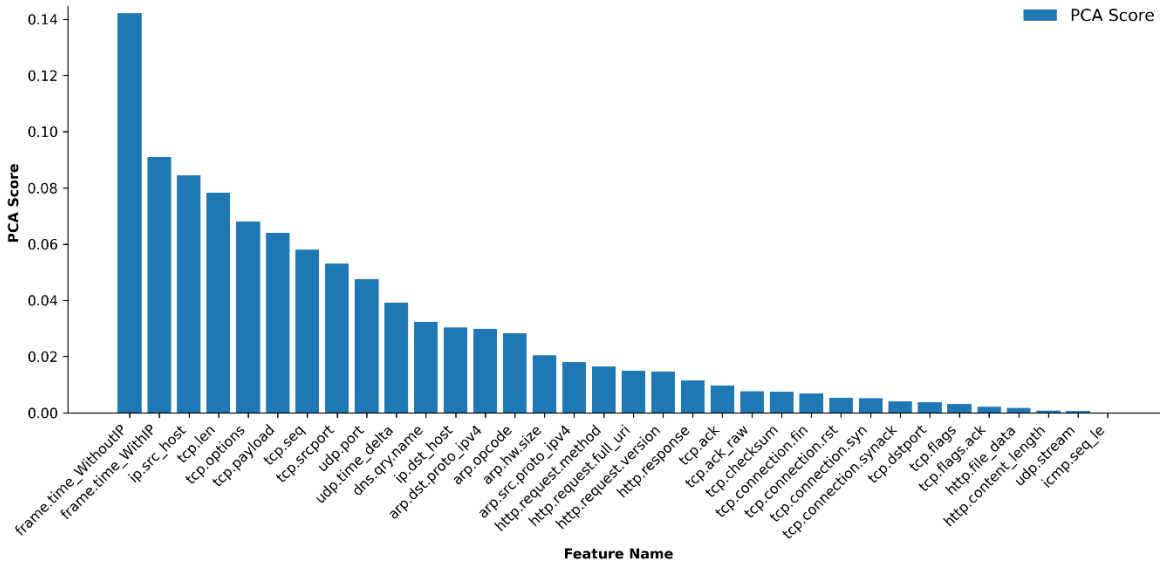


Figure 3.11. PCA scores of features for dimensionality reduction.

F. Random Forest

Finally, the Random Forest method, which integrates feature importance as part of its algorithm, is utilised in this research to evaluate the importance of each feature in predicting the target variable "Label". This feature selection method is effective due to its ability to handle large datasets with a diverse range of data types, including both numerical and categorical variables. The results, as shown in Figure 3.12, indicate that the highest Random Forest scores are for the features "tcp.dstport" and "tcp.srcport", both with a score around 0.2, suggesting that they are significant to the prediction of the target variable. On the other hand, the seven features "arp.src.proto_ipv4", "udp.port", "http.request.version", "arp.hw.size", "dns.qry.name", "http.content_length", "udp.time_delta" have scores of zero, indicating that they do not contribute meaningfully to the classification task. These seven features are excluded, leaving 27 out of the 34 features as significant.

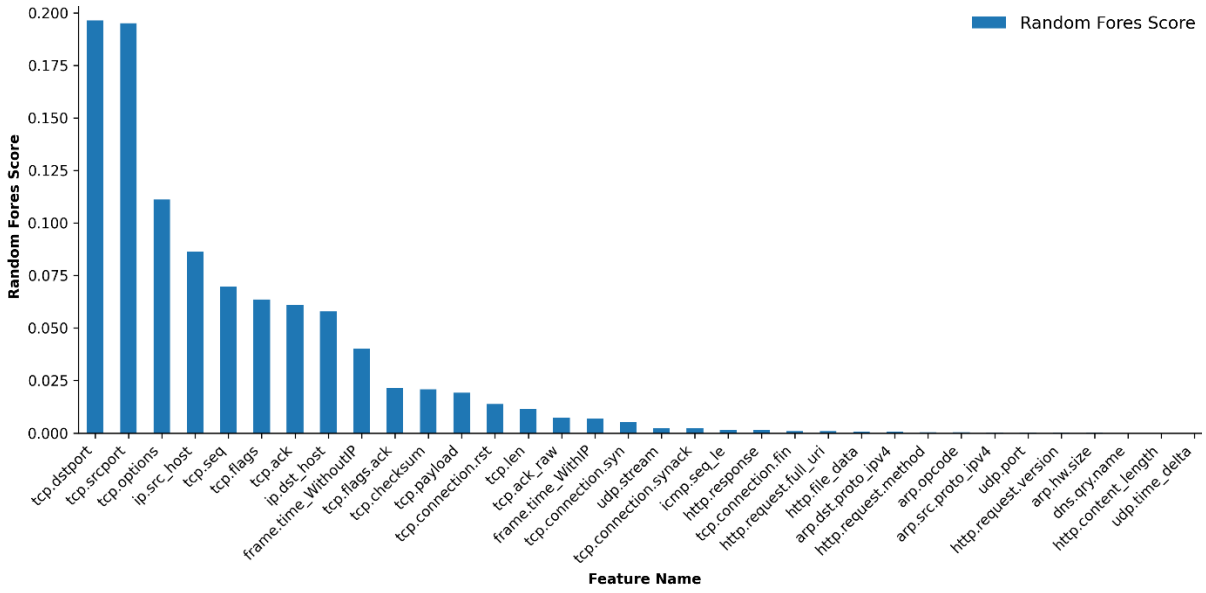


Figure 3.12. Random forest feature scores for predicting the target variable.

3.2.3 Feature Weighting

Since each feature selection method has its strengths and weaknesses, this research incorporates the benefits of the six feature selection methods discussed in section 3.3.2 and combines their scores to identify multilayer attacks. The proposed approach aligns with the ensemble feature selection approach presented by (Göcs and Johanyák, 2023), which strengthens the advantages of each method while mitigating their weaknesses. By combining the scores of various ranking methods, weights can be assigned to the features to make the final feature selection more robust and less influenced by any single ranking method.

To assign weights to the features, the scores obtained from the six feature selection methods are first normalised using the Min-Max normalisation technique to ensure comparability. Min-Max normalisation is chosen following the approach of (Zainudin, Akter et al., 2023; Alalhareth and Hong, 2023). The normalisation formula is as follows:

$$\text{Normalised Score} = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (5)$$

Where:

- X is the original score value.
- max (X) is the maximum value of the score.
- min(X) is the minimum value of the score.

The normalised scores from all feature selection methods are then combined to calculate a final score for each feature. The combined score (arithmetic mean) is calculated using the following formula, which averages the normalised scores from the feature selection methods:

$$\text{Combined Score} = \sum_{n=1}^{n=N} \frac{Xn}{n} \quad (6)$$

Where:

- X is the normalised score of each feature selection method and n is the number of feature selection methods used.

As a result, features are ranked based on their combined scores, with higher scores indicating greater importance. Lastly, each feature is assigned a weight equal to its combined score.

Table 3.2 and Figure 3.13 present the sorted weights of the features based on their importance. They show that "tcp.srcport" has the highest weight of around 0.7, followed by "tcp.dstport" at around 0.62 and "frame.time_WithoutIP" at 0.54. This feature weighting is a strategic step, allowing machine learning models to prioritise features that are consistently identified as significant across six feature selection techniques.

Table 3.2. Feature weights analysis based on their importance.

Feature Name	Feature Weights	Feature Name	Feature Weights
tcp.srcport	0.6968	udp.time_delta	0.0464
tcp.dstport	0.6273	dns.qry.name	0.0417
frame.time_WithoutIP	0.5413	arp.opcode	0.0334
ip.src_host	0.3976	icmp.seq_le	0.033
tcp.seq	0.3893	udp.stream	0.0326
tcp.ack	0.337	http.request.method	0.0242
frame.time_WithIP	0.3123	arp.hw.size	0.0239
tcp.options	0.254	arp.src.proto_ipv4	0.023
ip.dst_host	0.2453	http.request.full_uri	0.0221
tcp.ack_raw	0.2232	http.request.version	0.0218
tcp.len	0.206	tcp.connection.rst	0.0191
tcp.payload	0.1888	http.response	0.0191
tcp.flags	0.1714	tcp.connection.syn	0.0176
tcp.checksum	0.0906	tcp.connection.fin	0.0106
tcp.flags.ack	0.0662	tcp.connection.synack	0.0078
udp.port	0.0561	http.file_data	0.0048
arp.dst.proto_ipv4	0.0558	http.content_length	0.0036

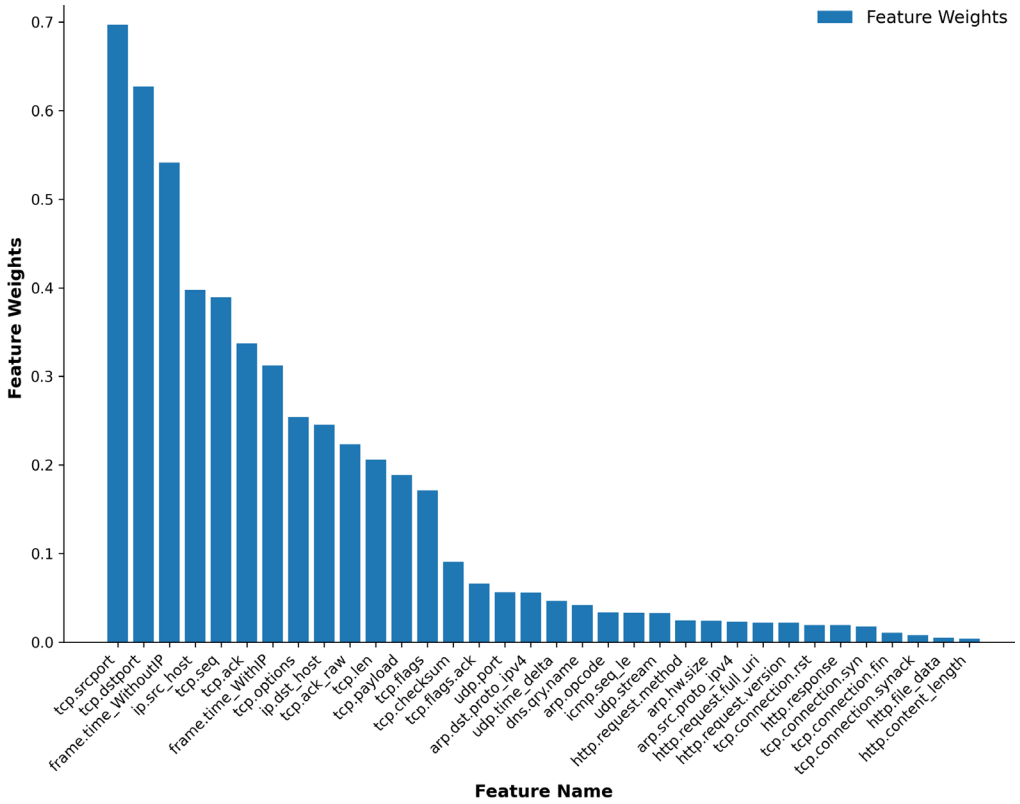


Figure 3.13. Feature weights analysis based on their importance.

3.2.4 Machine Learning Models and Performance Evaluation Metrics

Five machine learning classification algorithms are employed in SAIDS to evaluate the selected features from the "Edge-IIoTset" dataset in detecting multilayer IoT attacks. These algorithms include Decision Tree (DT), K-Nearest Neighbors (KNN), Naive Bayes (NB), Random Forest (RF), and Artificial Neural Network (ANN). These models are chosen for this research due to their effectiveness in detecting IoT security attacks, as discussed in Chapter 2, and their performance results based on empirical testing. The utilisation of diverse models enables the comparison of results across multiple models, ensuring the robustness and applicability of the findings, and identifying feature combinations that enhance the models' accuracy.

This research consists of two classification stages: binary and multiclass classifications. The binary classification stage is used to classify the network traffic as normal or as multilayer IoT attack categories. If the traffic is flagged as abnormal (multilayer IoT attacks), the system proceeds to the second stage, the multiclass classification, to identify the specific type of multilayer attacks.

Several evaluation metrics are used to full understand and evaluate the effectiveness of the classification algorithms across various attack types. These metrics include accuracy, Precision (Pr), Recall (Rc), F1-score (F1), and AUC for both training and testing datasets. The utilisation of the AUC metric serves (Albulayhi et al., 2022) to assess the presence of overfitting and underfitting in machine learning models. This approach is essential because a model prone to overfitting may fail to detect new types of attacks that were not present in the training data, while an underfitting model may not detect even the known attacks.

3.2.5 Semi-automated tool for Identifying Optimal Features

After feature weighting, a semi-automatic feature selection tool is developed to visualise the test accuracy of the machine learning algorithms. This tool adds the top-weighted features one after another for both binary and multiclass classification to identify the most significant features.

Figures 3.14, 3.15, 3.16, and 3.17 illustrate the model's testing accuracy using various feature sets, ranging from 1 to 34, and a separate set of 62 from the Edge-IIoTset dataset.

- **Binary classification**

Figure 3.14 presents the testing accuracies for ML models (DT, KNN, NB, RF, and ANN) for binary classification. Each model's performance varies across different feature sets, with a colour scale from red to green indicating accuracy levels. Red represents lower accuracy, while green represents higher accuracy. Among these models, the KNN model consistently outperforms the others in this task. For the full results of the testing accuracies for ML models for binary classification, refer to Appendix 3.

The detailed performance of KNN is shown in Figure 3.15, which highlights the binary classification results. Initially, the KNN model achieves an accuracy of 91.05% for the first feature set. As additional features are incorporated, the accuracy significantly improved, reaching nearly perfect performance (around 100%) for feature sets 2 to 13. This suggests that these features are highly informative for the binary classification task. However, after 13-feature set, there is a noticeable decline in accuracy to approximately 98%. This reduction could be due to the introduction of noise or irrelevant information from the additional features, which can confuse the model and decrease its performance.

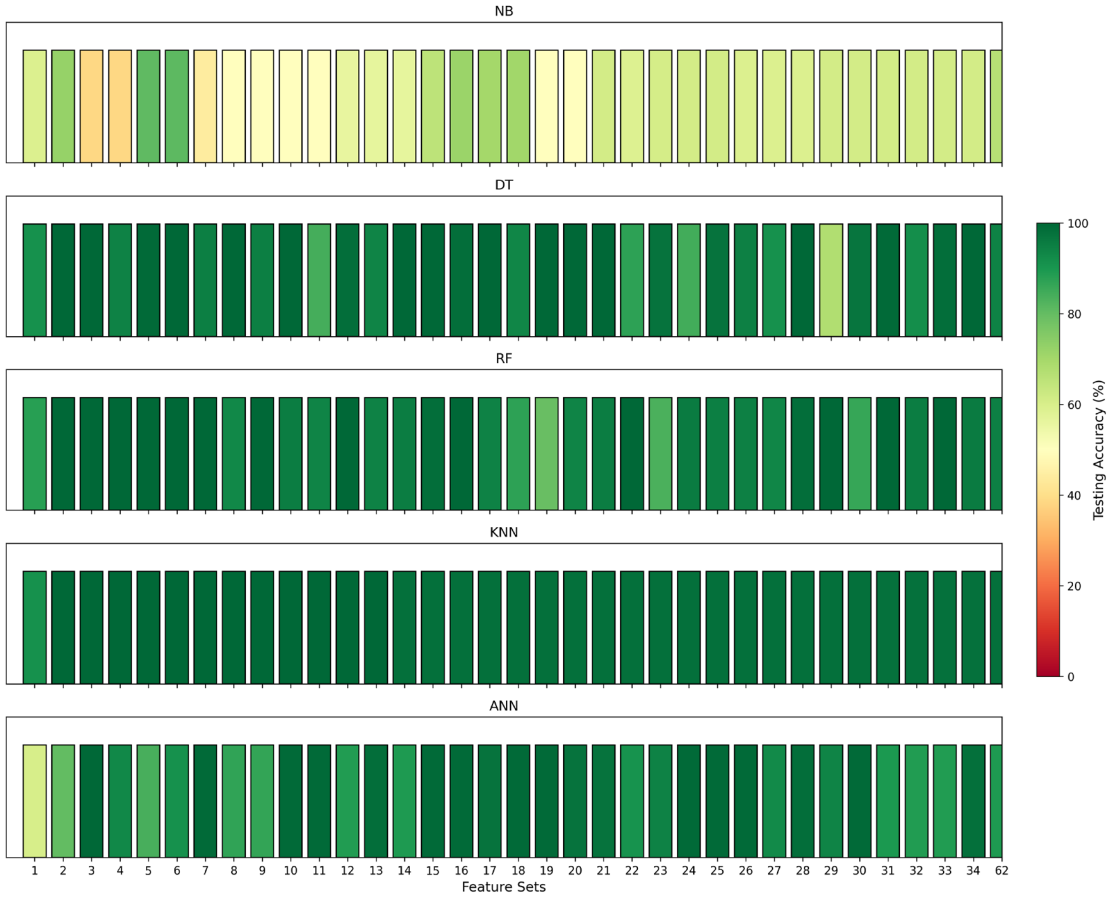


Figure 3.14. Model accuracies for binary classification across different feature sets.

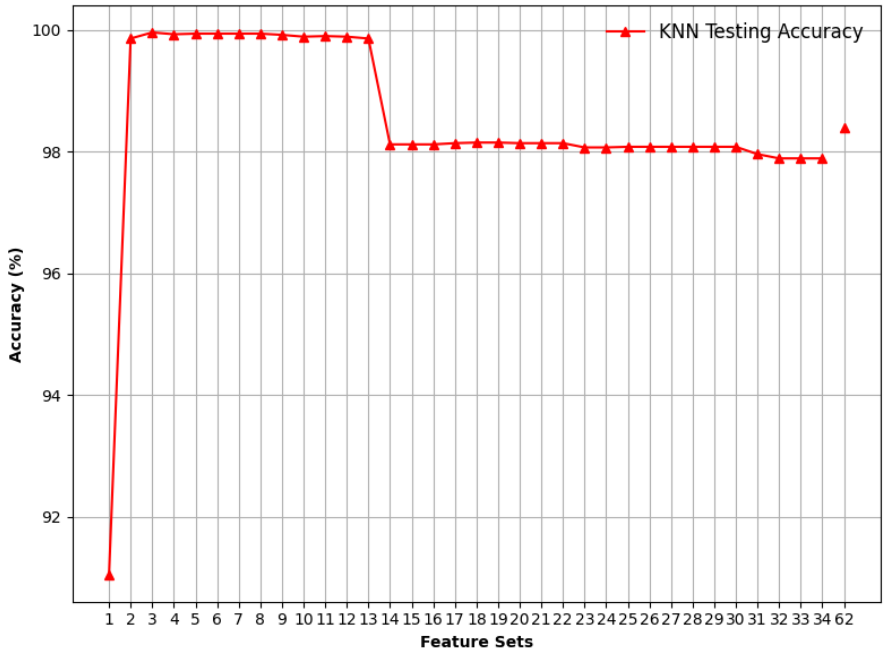


Figure 3.15. Visualising binary classification using KNN model.

- **Multiclass Classification**

Figure 3.16 mirrors the structure of Figure 3.14 but focuses on multiclass classification. It demonstrates that the KNN model again outperforms the other models, achieving high accuracy. For the full results of the testing accuracies for ML models for multiclass classification, refer to Appendix 3.

To delve deeper into the KNN model's performance in multiclass classification, Figure 3.17 is presented. The testing accuracy starts at a low of 52% with only one feature and increases significantly as more features are added, reaching a peak at the 9-feature set with an accuracy of around 96%. After the 9-feature set, there is a notable drop in accuracy at feature set 10. A slight recovery is seen at the 13-feature set with an accuracy of 90%, followed by another drop. From feature set 14 onward, the accuracy stabilises within the mid-85% range, with minor increases but no return to the peak levels seen with feature sets 9 or 13. These critical feature sets, 9 and 13, are detailed in Tables 3.5 and 3.6, respectively.

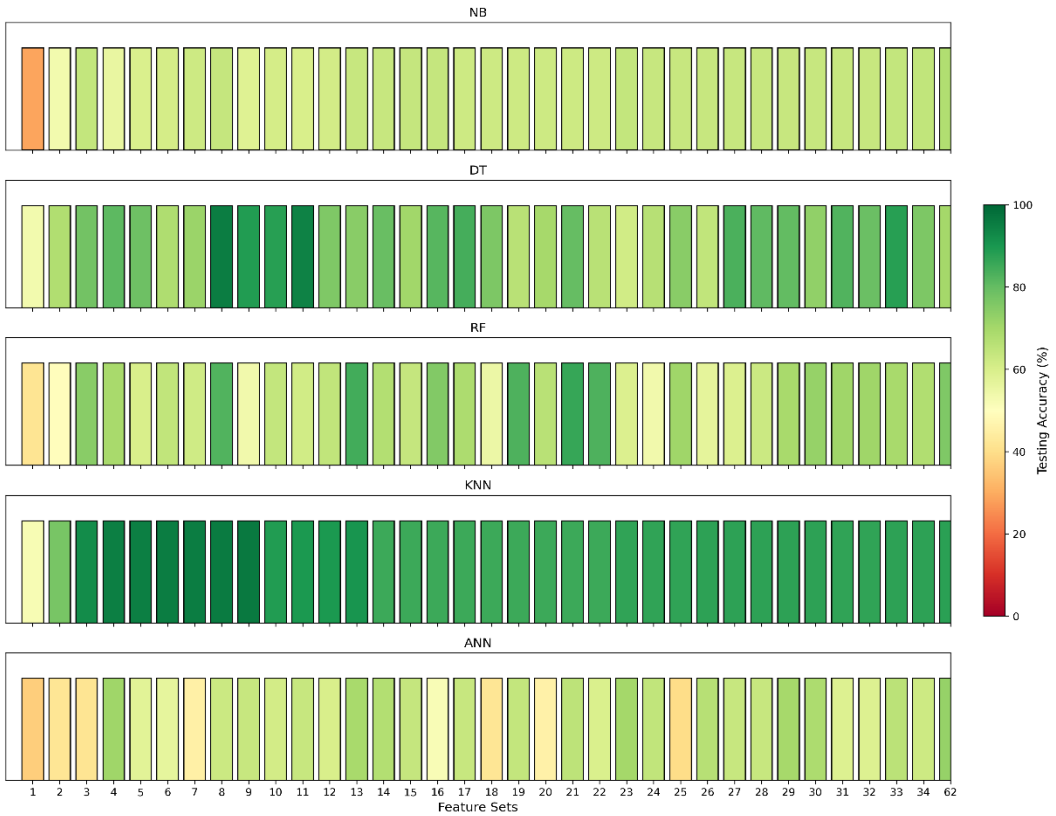


Figure 3.16. Model accuracies for multiclass classification across different feature sets.

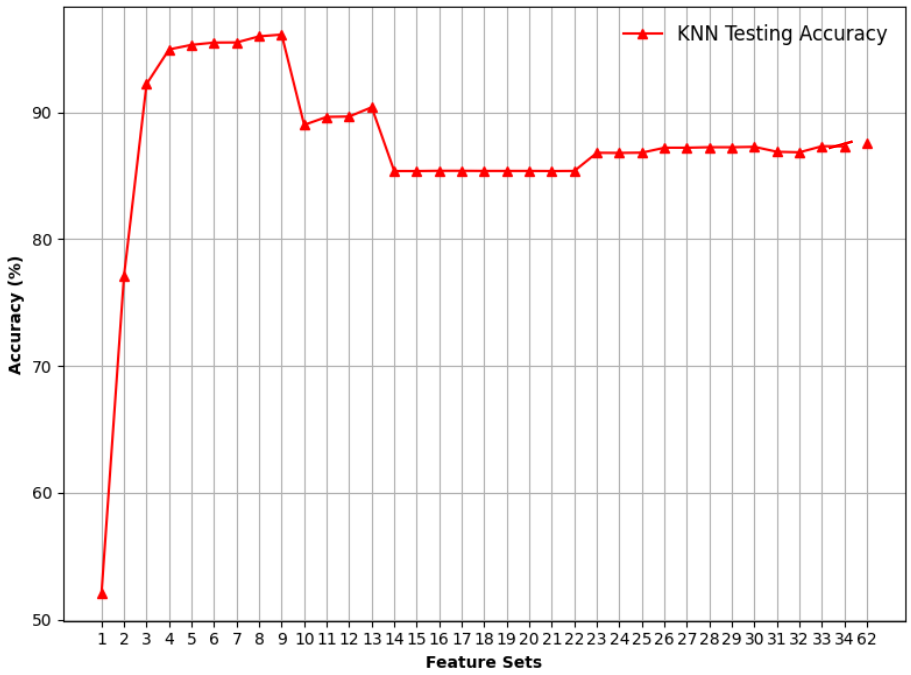


Figure 3.17. Visualising multiclass classification using KNN model.

Table 3.3. The 13-Feature set for Detecting and Identifying Multilayer Attacks.

No.	Feature Name	No.	Feature Name
1	frame.time_WithoutIP	8	tcp.seq
2	frame.time_WithIP	9	tcp.srcport
3	ip.src_host	10	tcp.ack
4	ip.dst_host	11	tcp.ack_raw
5	tcp.len	12	tcp.dstport
6	tcp.options	13	tcp.flags
7	tcp.payload		

Table 3.4. The 9-feature set for detecting and identifying multilayer attacks.

No.	Feature Name	No.	Feature Name
1	frame.time_WithoutIP	6	tcp.options
2	frame.time_WithIP	7	tcp.payload
3	ip.src_host	8	tcp.seq
4	ip.dst_host	9	tcp.srcport
5	tcp.len		

3.3 CHAPTER SUMMARY

This chapter presents a robust methodology for developing a Semi-Automated Intrusion Detection System (SAIDS) aimed at identifying multilayer IoT attacks. By integrating a range of experimental tools and methodologies, including advanced machine learning techniques, comprehensive feature selection, and weighting strategies, the research ensures high detection accuracy and efficiency. The methodological framework leverages the strengths of both human expertise and machine learning algorithms, enhancing the overall effectiveness of the intrusion detection system.

The successful implementation on the Edge-IIoTset dataset further validates the proposed approach, demonstrating its capability to address the complexity and diversity of multilayer IoT threats effectively. This dataset, known for its comprehensive representation of IoT cybersecurity threats, provided a solid foundation for testing and fine-tuning the SAIDS.

This implementation includes:

- **Feature Selection Methods**: MI, IG, DTE, Chi^2, and RF.
- **Machine Learning Models**: DT, KNN, NB, RF, and ANN.
- **Evaluation Metrics**: Accuracy, Precision, Recall, F1-score, and Area Under the Curve.

The feature selection process for the binary classification effectively reduces the total number of features from 62 to 34-common features. Then feature selection models reduce the 34-common features to enhance model performance. Figure 3.18 visualises the reduced features of each feature selection method, providing a clear representation of the effectiveness of each method in extracting the informative features.

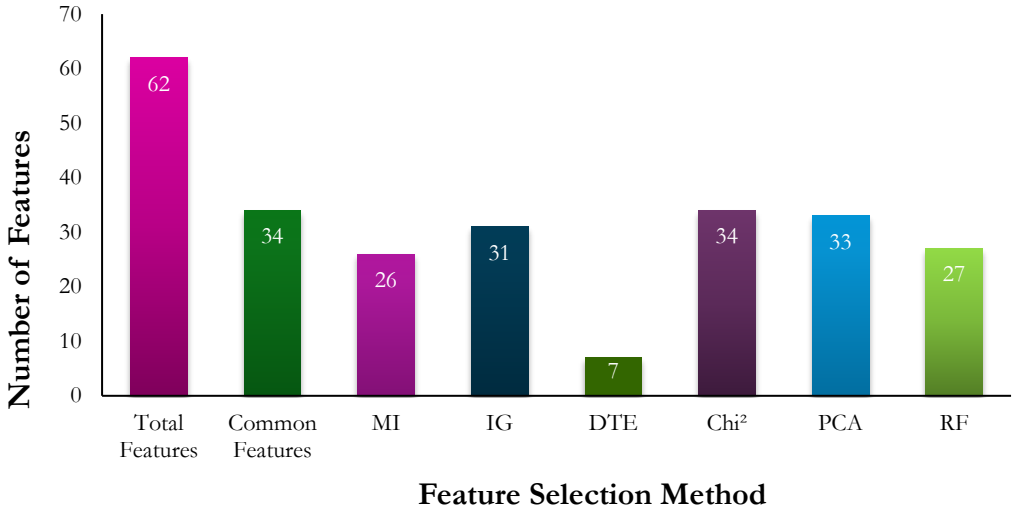


Figure 3.18. Feature selection methods and their reduced features.

In the feature weighting step, features are assigned weights based on their importance across the six feature selection methods. The highest-weighted features are: tcp.srcport, tcp.dstport, and frame.time_WithoutIP.

The KNN model consistently outperforms the other models in both binary and multiclass classification tasks when using the semi-automated tool. In the KNN model, the 13-feature set in the binary classification marks a critical threshold where the KNN model's accuracy slightly declines, indicating the introduction of potentially noisy features. Additionally, feature set 9 shows the highest accuracy for multiclass classification, suggesting a highly effective but possibly oversimplified model. The slight reduction in accuracy at 13-feature set might indicate that additional features could potentially improve the model's ability to identify a broader spectrum of multilayer attacks.

To enhance the evaluation of 13-feature set for binary classification and feature sets 13 and 9 for multiclass classification, a thorough analysis is implemented in the next chapter. This involves comparing different evaluation metrics for each class, specifically precision, recall, F1-score, and AUC for both training and testing datasets. This comprehensive assessment is crucial for a full understanding of the model's effectiveness across various attack types.

4 EXPERIMENTAL RESULTS AND SETTINGS

As shown in Figure 4.1, this chapter presents the implementation results, showcasing the performance of five fine-tuned machine learning models—RF, DT, KNN, ANN, and NB classifiers—in classifying IoT network traffic into normal and multilayer attacks (identifying the specific types of attacks) by employing SAIDS on the Edge-IIoTset dataset. It also details the hyperparameter tuning process for the ML training models.

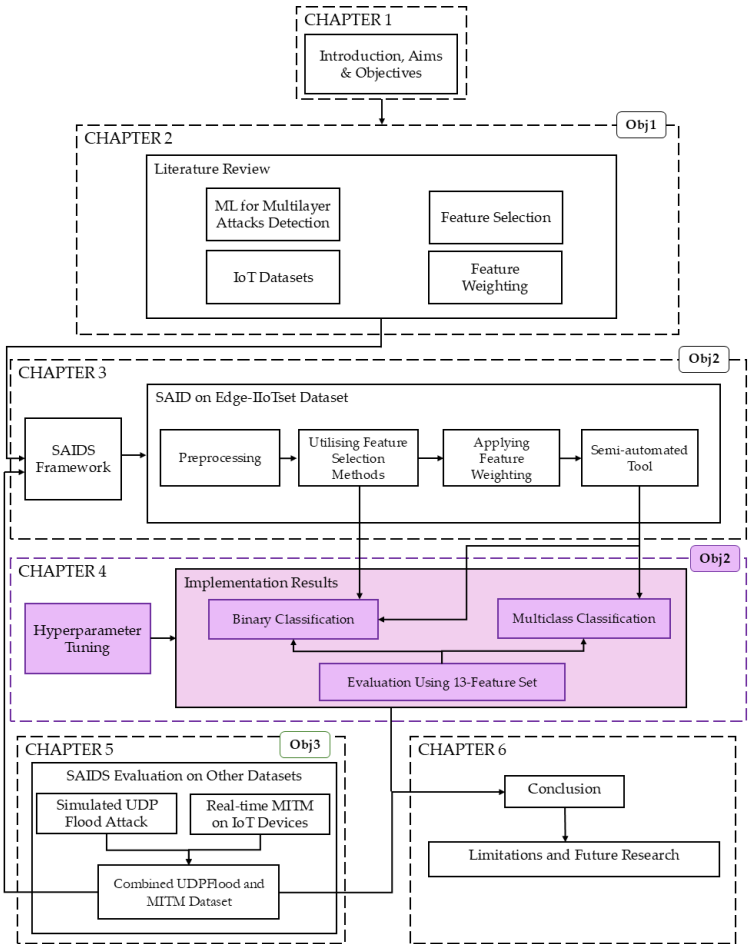


Figure 4.1. Thesis structure showing Chapter 4's placement within the overall project.

4.1 HYPERPARAMETER TUNING OF MACHINE LEARNING MODELS

Hyperparameter tuning is a critical step in optimising machine learning models to ensure their best performance, particularly in complex tasks like detecting multilayer attacks in IoT networks. In this research, hyperparameter tuning is employed to enhance the models used in the Semi-Automated Intrusion Detection System (SAIDS). Initially, Grid Search was employed for tuning the hyperparameters, but due to its time-consuming nature, the approach was shifted to Random Search. This decision is in line with the observations made by (Ali et al., 2023), who highlighted the slow performance of Grid Search. Random Search offers a faster alternative by sampling a fixed number of parameter settings from the specified distributions. This method significantly reduces computational time while still effectively identifying the optimal hyperparameters, thus improving the models' accuracy and reducing error rates (Ali et al., 2023). The random search is conducted using the RandomizedSearchCV class from the scikit-learn library. Upon completion of the search process, the best parameters are extracted from the RandomizedSearchCV object, as presented below.

Table 4.1. Hyperparameters tuning for DT, RF, KNN, ANN, and NB models.

Model	Parameters	Settings
DT	Criterion	entropy
	Maximum Depth	5
	Minimum Samples Split	10
	Maximum Features	sqrt
	Minimum Samples Leaf	4
RF	Criterion	gini
	Maximum Depth	10
	Number of Estimators	10
KNN	Number of Neighbors	5
	Power Parameter (p)	1
	Distance Metric	manhattan
ANN	Activation Function	ReLU
	Optimiser	adam
	Evaluation Metrics	accuracy
	Number of Epochs	10
	Batch Size	32
	Loss Function (Binary)	binary_crossentropy
	Loss Function (Multiclass)	categorical_crossentropy

4.2 IMPLEMENTATION RESULTS

This section provides a comprehensive evaluation of the five ML models using their optimised hyperparameters. The analysis begins with detecting multilayer attacks through binary classification based on the accuracy, using both the full set of 62 features and a reduced set of 34 common features from the Edge-IIoTset dataset and the most significant features identified by each feature selection method applied to the reduced 34 common features dataset. This is followed by a detailed assessments of the full set of 62 features, 34 common features, and feature sets 13 and 9, as identified in section 3.3.5, utilising precision, recall, F1-score, and AUC for both training and testing datasets for multiclass classification to identify the specific types of multilayer attacks. The goal is to determine the most effective feature combinations for accurate and robust multilayer attack classification.

4.2.1 Feature Selection Outputs for Detecting Multilayer Attacks

From the data shown in Table 4.2 of the binary classification based on accuracy, utilising a reduced feature set of 34, it is evident that there is a significant increase in accuracy for the DT model, which increased to 99.87% with the 34 features from 94.3% with all 62 features. The RF classifier also experienced a gain in performance, improving to 95.78% accuracy from 94.58%. The ANN classifier also showed a notable increase in accuracy, up to 86.41% from 76.1% when the feature set was reduced. This demonstrates that a reduced set of 34 features can enhance the predictive capabilities of machine learning models for IoT network traffic by eliminating noise and irrelevant data.

Also, from the analysis of feature selection methods (MI, IG, DTE, PCA, Chi^2, and RF), it is apparent that the models achieve high accuracy rates when employing the feature selection method MI with the permutation test. This suggests that using MI for feature selection, with just 26 features, is an effective strategy for enhancing the accuracy of ML models. Following MI is IG with 31 features, which also shows good performance. Both the Chi^2 method and the utilisation of all 34 features achieved good results; this is because the Chi^2 method considers all 34 features to be

relevant and significant, resulting in their effective performance. However, the DTE method is the least effective in feature selection. Within the MI feature selection technique, the ANN model demonstrated the highest accuracy at 99.83%. The KNN model follows closely with an accuracy of 97.95%, and the DT model achieves a third-best accuracy of 97.10%. On the other hand, the RF model shows slightly lower performance with an accuracy of 95.55%, and the NB model records the lowest accuracy at 61.18%.

Table 4.2. Comparative accuracy analysis of ML models using different feature selection methods for binary classification. *Values for MI, highlighted in yellow, indicate the best feature selection method. MI is tested with 26 significant features, IG with 31 features, DTE with 7 features, Chi² with 34 features, PCA with 33 features, and FR with 27 features.

	FS Methods							
Machine Learning Model	**MI**	**IG**	**Chi²**	**PCA**	**RF**	**DTE**	**All 34**	**All 62**
DT	**97.10**	85.74	99.87	99.87	94.32	71.03	99.87	94.3
RF	**95.55**	98.41	95.78	98.46	84.9	99.84	95.78	94.58
KNN	**97.95**	97.89	97.89	84.84	97.88	99.93	97.89	98.4
ANN	**99.83**	98.88	86.41	80.56	92.25	92.92	86.41	76.1
NB	**61.18**	61.19	61.22	43.96	61.31	38.48	61.22	66.77

4.2.2 Semi-automated tool Outputs for Identifying Multilayer Attacks

4.2.2.1 62-Features for Identifying IoT Multilayer Attacks

In the initial evaluation using all 62 features for classifying multilayer IoT attack types, as shown in Table 4.3, it is found that KNN and RF algorithms exhibit the highest overall performance with high precision, recall, F1-scores, testing accuracy, AUC (both training and Testing) ranging from 99% to 100%. As shown in Figure 4.2, KNN performs well in detecting DDoS_TCP, DDoS_UDP, DDoS_ICMP, and MITM, while RF performs particularly well in classifying DDoS_UDP, DDoS_ICMP, and Password attacks. NB also shows strong performance, especially for DDoS and MITM attacks, with consistent high accuracy (ranging from 80% to 100%) and AUC scores (76% to 100%). While DT and ANN demonstrate good overall accuracy (75% to 100%) and AUC (75% to 1.00).

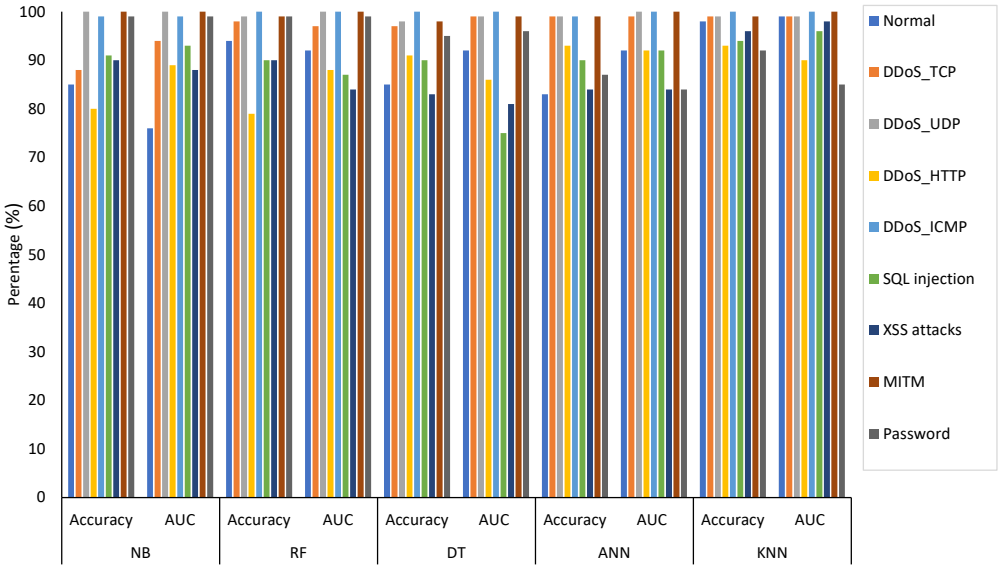


Figure 4.2. Performance Analysis of ML algorithms using 62-features.

Table 4.3. Performance of ML algorithms for multilayer IoT attack classification using 62-features. *Yellow highlights indicate worst predictions.

Alg	Metric	Normal	DDoS_TCP	DDoS_UDP	DDoS_HTTP	DDoS_ICMP	SQL injection	XSS attacks	MITM	Password
NB	Pr	0.99	0.46	1.00	0.34	0.96	*0.69*	*0.75*	1.00	1.00
	Rc	0.38	1.00	1.00	0.98	1.00	*0.15*	*0.05*	1.00	0.99
	f1	0.54	0.63	1.00	0.50	0.98	*0.25*	*0.09*	1.00	1.00
	Acc.	0.85	0.88	1.00	0.80	0.99	0.91	0.90	1.00	0.99
	AUC Training	0.76	0.96	1.00	0.88	0.99	0.94	0.89	1.00	0.99
	AUC Testing	0.76	0.94	1.00	0.89	0.99	0.93	0.88	1.00	0.99
RF	Pr	0.92	0.89	1.00	0.32	1.00	*0.00*	*0.00*	0.66	1.00
	Rc	0.85	0.91	1.00	0.99	1.00	*0.00*	*0.00*	1.00	1.00
	f1	0.88	0.90	1.00	0.49	1.00	*0.00*	*0.00*	0.79	1.00
	Acc.	0.94	0.98	0.99	0.79	1.00	0.90	0.90	0.99	0.99
	AUC Training	0.92	0.98	1.00	0.87	1.00	0.86	0.83	1.00	0.99
	AUC Testing	0.92	0.97	1.00	0.88	1.00	0.87	0.84	1.00	0.99
DT	Pr	0.63	0.88	1.00	0.59	1.00	*0.00*	0.28	0.46	1.00
	Rc	0.94	0.88	0.90	0.34	1.00	*0.00*	0.48	1.00	0.49
	f1	0.75	0.88	0.95	0.43	1.00	*0.00*	0.35	0.63	0.66
	Acc.	0.85	0.97	0.98	0.91	1.00	0.90	0.83	0.98	0.95
	AUC Training	0.90	0.99	0.99	0.85	1.00	0.81	0.82	0.99	0.97
	AUC Testing	0.92	0.99	0.99	0.86	1.00	0.75	0.81	0.99	0.96
ANN	Pr	0.78	0.81	1.00	0.42	0.84	*0.01*	0.26	1.00	0.35
	Rc	0.50	0.99	0.99	0.70	1.00	*0.00*	0.54	1.00	0.22
	f1	0.61	0.89	0.99	0.53	0.91	*0.00*	0.35	1.00	0.27
	Acc.	0.83	0.99	0.99	0.93	0.99	0.90	0.84	0.99	0.87
	AUC Training	0.94	0.99	1.00	0.92	1.00	0.92	0.89	1.00	0.88
	AUC Testing	0.92	0.99	1.00	0.92	1.00	0.92	0.84	1.00	0.84
KNN	Pr	0.97	1.00	1.00	0.70	1.00	0.66	0.75	0.99	0.68
	Rc	0.95	1.00	1.00	0.69	1.00	0.80	`0.93	1.00	0.41
	f1	0.96	1.00	1.00	0.69	1.00	0.72	0.83	1.00	0.51
	Acc.	0.98	0.99	0.99	0.93	1.00	0.94	0.96	0.99	0.92
	AUC Training	0.99	1.00	1.00	0.98	1.00	0.99	0.99	1.00	0.98
	AUC Testing	0.99	0.99	0.99	0.90	1.00	0.96	0.98	1.00	0.85

4.2.2.2 34-Common Features for Identifying IoT Multilayer Attacks

A similar evaluation is conducted using the 34 common features for identifying multilayer IoT attacks, as illustrated in Table 4.4. KNN and RF continue to exhibit the highest accuracy in detecting specific attacks, similar to the initial evaluation using 62 features.

Table 4.4. Performance of ML algorithms for multilayer IoT attack classification using 34-common features. *Yellow highlights indicate worst predictions.

Alg	Metric	Normal	DDoS_TCP	DDoS_UDP	DDoS_HTTP	DDoS_ICMP	SQL injection	XSS attacks	MITM	Password
NB	Pr	0.99	0.41	1.00	0.33	0.96	*0.64*	*0.78*	1.00	1.00
	Rc	0.22	1.00	1.00	0.97	1.00	*0.17*	*0.07*	1.00	0.99
	f1	0.36	0.58	1.00	0.49	0.98	*0.27*	*0.13*	1.00	1.00
	Acc.	0.81	0.86	1.00	0.79	0.99	0.90	0.91	1.00	0.99
	AUC Training	0.71	0.96	1.00	0.90	0.99	0.94	0.88	1.00	0.99
	AUC Testing	0.71	0.93	1.00	0.92	0.99	0.92	0.87	1.00	0.99
RF	Pr	0.69	0.52	1.00	0.50	0.99	0.33	*0.00*	0.69	1.00
	Rc	0.54	0.60	1.00	0.27	1.00	0.92	*0.00*	1.00	1.00
	f1	0.60	0.55	1.00	0.35	1.00	0.49	*0.00*	0.81	1.00
	Acc.	0.83	0.90	0.99	0.90	0.99	0.81	0.90	0.99	0.99
	AUC Training	0.86	0.89	1.00	0.87	1.00	0.85	0.82	0.99	0.99
	AUC Testing	0.86	0.87	1.00	0.87	1.00	0.85	0.78	0.99	0.99
DT	Pr	0.74	1.00	1.00	0.50	1.00	0.46	0.54	1.00	1.00
	Rc	0.92	0.31	1.00	0.27	1.00	0.89	0.38	1.00	1.00
	f1	0.89	0.47	1.00	0.35	1.00	0.60	0.44	1.00	1.00
	Acc.	0.90	0.93	1.00	0.90	1.00	0.88	0.91	1.00	0.99
	AUC Training	0.94	0.96	1.00	0.90	1.00	0.92	0.90	1.00	1.00
	AUC Testing	0.92	0.93	1.00	0.90	1.00	0.93	0.90	1.00	1.00
ANN	Pr	0.67	0.68	1.00	0.23	0.99	0.53	*0.18*	0.96	*0.38*
	Rc	0.57	0.99	1.00	0.23	1.00	0.72	*0.22*	1.00	*0.08*
	f1	0.62	0.81	1.00	0.23	1.00	0.61	*0.20*	0.98	*0.13*
	Acc.	0.83	0.95	0.99	0.84	0.99	0.90	0.82	0.99	0.90
	AUC Training	0.92	0.99	1.00	0.83	1.00	0.93	0.90	1.00	0.85
	AUC Testing	0.92	0.99	1.00	0.83	0.99	0.93	0.85	1.00	0.84
KNN	Pr	0.97	1.00	1.00	0.68	1.00	0.66	0.75	0.99	0.68
	Rc	0.94	1.00	1.00	0.68	1.00	0.80	`0.92	1.00	0.41
	f1	0.96	1.00	1.00	0.68	1.00	0.72	0.83	1.00	0.51
	Acc.	0.97	0.99	0.99	0.93	1.00	0.94	0.96	0.99	0.92
	AUC Training	0.99	1.00	1.00	0.98	1.00	0.98	0.99	1.00	0.98
	AUC Testing	0.99	0.99	0.99	0.90	0.99	0.96	0.98	1.00	0.85

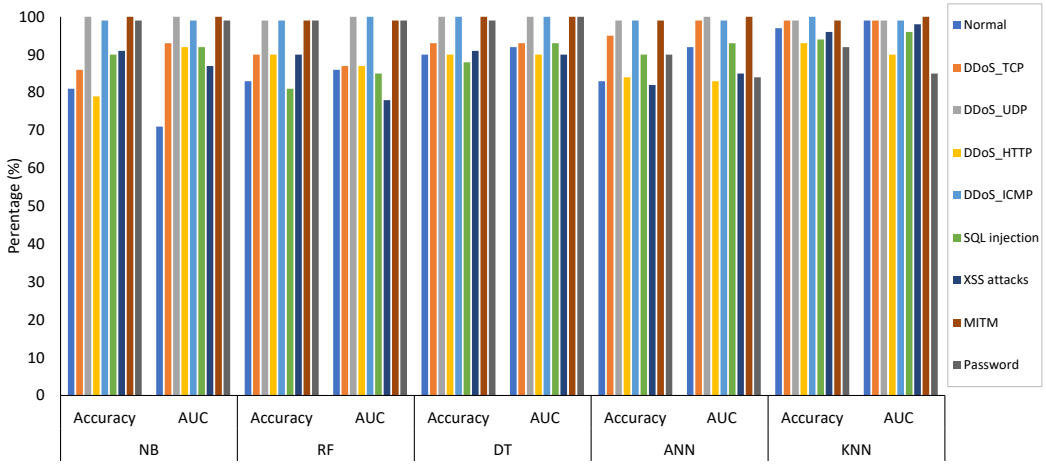


Figure 4.3. Performance Analysis of ML algorithms using 34-features.

- **Performance Analysis with 62 Features**

As shown in Figure 4.4, it is found that RF, DT, and ANN models fail to distinguish SQL injection attacks, with precision, recall, and F1-scores for these attacks as low as zero. The RF model also fails to detect XSS attacks, exhibiting similarly low values for precision, recall, and F1-scores. NB shows poor performance in detecting both XSS attacks and SQL injection attacks, further highlighting the need for optimisation or alternative approaches to enhance detection accuracy for these specific types of attacks.

- **Performance Analysis with 34 Features**

In this evaluation, as shown in Figure 4.5, it is discovered that the RF model fails to detect XSS attacks, as evidenced by zero values in precision, recall, and F-measure. Both NB and ANN demonstrate lower performance in detecting XSS attacks. Additionally, NB struggles with detecting SQL injection attacks, and ANN shows lower performance in detecting password attacks. These observations indicate that the models still face challenges in identifying specific multilayer IoT attacks even with a reduced feature set.

These performance issues with the RF, NB, DT and ANN models in detecting specific multilayer IoT attacks (XSS, SQL injection, and password attacks) using both the full 62 features and the reduced 34 common features suggest a need for refinement.

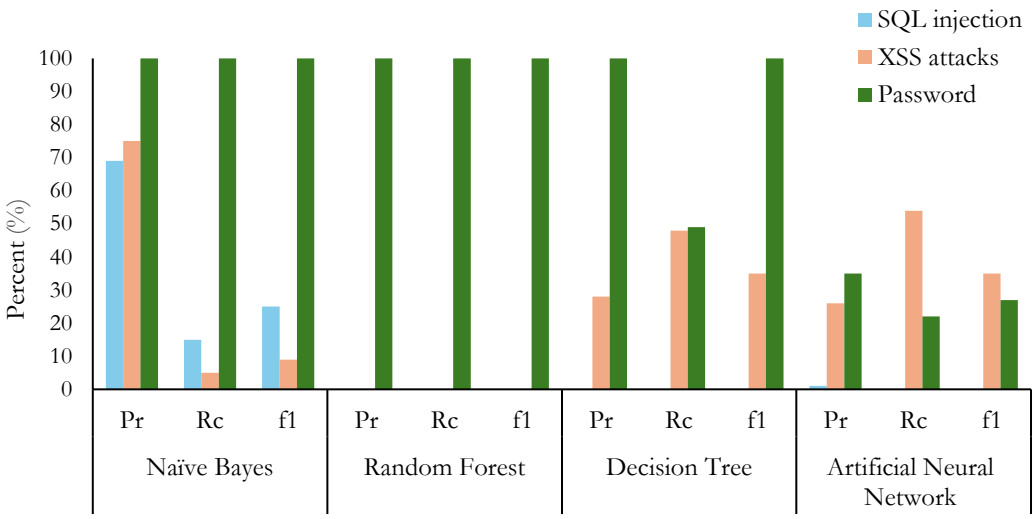


Figure 4.4. Performance analysis of ML models on all 62-features for multilayer attack identification.

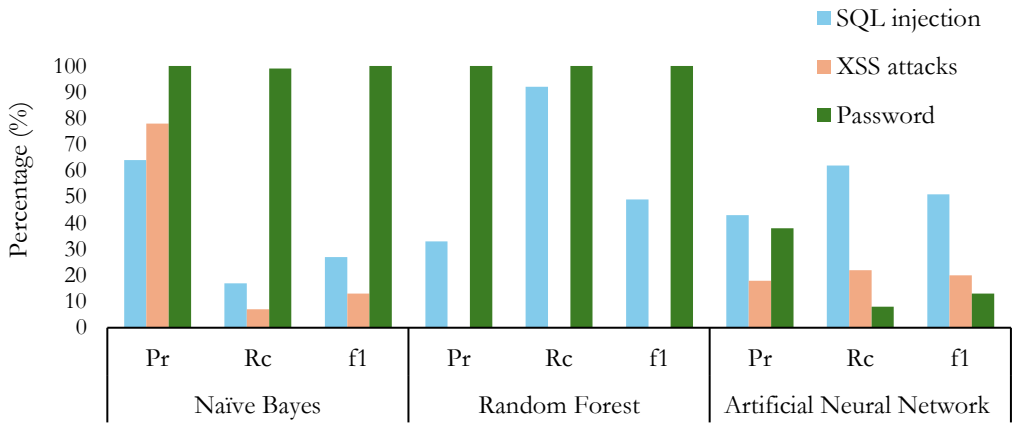


Figure 4.5. Performance analysis of ML models on 34-common features for multilayer attack identification.

4.2.2.3 Comparison Between 13-feature and 9-feature sets in Identifying IoT Multilayer Attacks

As shown in the following Tables 4.5 and 4.6, which represent the analysis of ML models' performance using various evaluation metrics on 13-feature and 9-feature sets for multilayer attack identification demonstrates good performance. The models perform well with both feature sets for the attack types of DDoS (TCP, UDP, HTTP, and ICMP). This indicates that the critical features for detecting these attacks are included in both the 9-feature and 13-feature sets. However, the models are not performing well for password, MITM, XSS, and SQL injection-related attacks, as well as for normal traffic detection.

For the 13-feature set (Table 4.5), the NB model displays notably low precision, recall, and F1-score values —2%, 4%, and 3% respectively—in detecting XSS attacks, indicating a high rate of false positives and false negatives. The testing accuracy is 72%, and the AUC shows a reasonable score in training but experiences a drop in testing from 76% to 68%. This decline could suggest potential overfitting.

In contrast, as demonstrated in Table 4.6, the 9-feature set exhibits significantly poorer performance across several models, such as NB, RF, and ANN, in distinguishing normal traffic, XSS, SQL injection, and password attacks compared to the 13-feature set. While all the metric values for NB in detecting XSS remain very low, they are consistent with those of the 13-feature set. Additionally, for NB in detecting SQL injection, despite high AUC scores of 97% for both training and testing and an accuracy of 90%, the precision, recall, and f-measure drop to 0. This indicates that the model failed to identify any true positives for SQL injection attacks. For RF, despite high AUC scores for both training and testing and high accuracy in distinguishing normal traffic, XSS, and SQL injection attacks, the model achieved 0% in precision, recall, and f-measure. This suggests that the model failed to identify any true positives for these attacks. The ANN model also exhibits low precision, recall, and f-measure values at 35%, 13%, and 19% respectively, indicating poor performance in identifying XSS attacks, and 26%, 11%, and 16% respectively, in identifying password attacks.

Although the 9-feature set is generally sufficient for detecting DDoS (TCP, UDP, HTTP, and ICMP) and MITM-related attacks, it is less effective in detecting normal traffic, password, SQL injection, and XSS attacks compared to the 13-feature set. This highlights the importance of including specific features that are critical for identifying normal traffic and more sophisticated attacks such as password, SQL injection, and XSS. The AUC scores for 13 features are quite high for both training and testing across all models, suggesting good model performance. However, the drop in AUC from training to testing for NB observed in both feature sets may indicate overfitting, particularly for XSS attacks.

In summary, the application of the SAIDS demonstrates that the 13-feature set is more adept at detecting and identifying multilayer attacks compared to the 9-feature set, 34 common features, and all 62 features.

Table 4.5. Analysis of ML models performance on 13-feature sets for multilayer attack identification. *Yellow highlights indicate worst predictions.

Alg	Metric	Normal	DDoS_TCP	DDoS_UDP	DDoS_HTTP	DDoS_ICMP	SQL injectio	XSS attacks	MITM	Password
NB	Precision	0.99	1.00	1.00	0.36	0.96	0.84	*0.02*	1.00	1.00
	Rc	0.22	0.63	1.00	0.99	1.00	0.42	*0.04*	1.00	1.00
	f1	0.36	0.77	1.00	0.53	0.98	0.56	*0.03*	1.00	1.00
	Acc.	0.81	0.96	1.00	0.82	0.99	0.93	*0.72*	1.00	0.99
	AUC Training	0.91	0.99	1.00	0.96	0.99	0.94	*0.76*	1.00	0.99
	AUC Testing	0.90	0.99	1.00	0.96	0.99	0.94	*0.68*	1.00	0.99
RF	Pr	0.79	0.84	1.00	1.00	0.99	0.60	0.59	0.72	1.00
	Rc	0.73	0.60	1.00	1.00	1.00	0.98	0.46	1.00	1.00
	f1	0.76	0.70	1.00	1.00	0.99	0.74	0.52	0.83	1.00
	Acc.	0.89	0.95	1.00	0.99	0.99	0.93	0.91	0.99	0.99
	AUC Training	0.94	0.95	1.00	0.99	1.00	0.93	0.90	0.99	1.00
	AUC Testing	0.94	0.96	1.00	0.99	1.00	0.93	0.90	0.99	0.99
DT	Pr	0.96	0.99	1.00	0.85	1.00	0.30	0.66	0.71	1.00
	Rc	0.60	0.65	1.00	0.38	0.99	1.00	0.34	1.00	0.97
	f1	0.74	0.78	1.00	0.52	0.99	0.47	0.45	0.83	0.98
	Acc.	0.90	0.96	0.99	0.93	0.99	0.77	0.92	0.99	0.99
	AUC Training	0.92	0.99	1.00	0.86	1.00	0.90	0.88	1.00	0.82
	AUC Testing	0.91	0.98	1.00	0.87	1.00	0.93	0.85	1.00	0.83
ANN	Pr	0.93	0.85	1.00	0.45	0.99	0.58	0.36	0.97	0.38
	Rc	0.44	0.88	1.00	0.30	0.99	0.89	0.82	1.00	0.30
	f1	0.60	0.86	1.00	0.36	0.99	0.70	0.50	0.98	0.33
	Acc.	0.84	0.96	0.98	0.86	0.99	0.87	0.83	0.99	0.87
	AUC Training	0.93	0.99	1.00	0.87	0.99	0.95	0.91	1.00	0.87
	AUC Testing	0.93	0.98	1.00	0.85	0.99	0.95	0.87	1.00	0.85
KNN	Pr	1.00	1.00	1.00	0.70	1.00	0.70	0.85	1.00	0.79
	Rc	1.00	1.00	1.00	0.72	1.00	0.80	0.95	1.00	0.54
	f1	1.00	1.00	1.00	0.71	1.00	0.75	0.89	1.00	0.64
	Acc.	0.99	0.99	1.00	0.94	0.99	0.94	0.97	1.00	0.94
	AUC Training	0.99	1.00	1.00	0.98	1.00	0.98	0.99	1.00	0.97
	AUC Testing	0.99	0.99	1.00	0.88	0.99	0.95	0.97	1.00	0.82

Table 4.6. Analysis of ML models performance on 9-feature sets for multilayer attack identification. *Yellow highlights indicate worst predictions.

Alg	Metric	Normal	DDoS_TCP	DDoS_UDP	DDoS_HTTP	DDoS_ICMP	SQL injectio	XSS attacks	MITM	Password
NB	Precision	0.99	1.00	1.00	0.32	0.96	*0.00*	*0.02*	1.00	1.00
	Rc	0.17	0.65	1.00	0.99	1.00	*0.00*	*0.04*	1.00	1.00
	f1	0.29	0.79	1.00	0.48	0.98	*0.00*	*0.03*	1.00	1.00
	Acc.	0.80	0.96	1.00	0.78	0.99	*0.90*	*0.72*	1.00	0.99
	AUC Training	0.92	0.99	1.00	0.99	0.99	*0.97*	*0.76*	1.00	0.99
	AUC Testing	0.92	0.99	1.00	0.99	0.99	*0.97*	*0.68*	1.00	0.99
RF	Pr	*0.00*	1.00	1.00	0.19	0.99	*0.00*	*0.00*	0.21	1.00
	Rc	*0.00*	0.60	1.00	1.00	1.00	*0.00*	*0.00*	1.00	1.00
	f1	*0.00*	0.75	1.00	0.32	1.00	*0.00*	*0.00*	0.34	1.00
	Acc.	*0.76*	0.96	1.00	0.58	0.99	*0.90*	*0.90*	0.95	0.99
	AUC Training	*0.78*	0.89	1.00	0.79	1.00	*0.79*	*0.77*	0.99	0.99
	AUC Testing	*0.77*	0.88	1.00	0.76	1.00	*0.76*	*0.73*	0.99	0.99
DT	Pr	0.98	1.00	0.99	0.92	1.00	0.67	0.64	1.00	0.93
	Rc	0.92	1.00	1.00	0.58	1.00	1.00	0.95	1.00	0.51
	f1	0.95	1.00	1.00	0.72	1.00	0.80	0.76	1.00	0.66
	Acc.	0.97	1.00	0.99	0.95	1.00	0.95	0.94	1.00	0.95
	AUC Training	0.99	1.00	0.99	0.97	1.00	0.96	0.97	1.00	0.96
	AUC Testing	0.99	1.00	1.00	0.97	1.00	0.97	0.97	1.00	0.97
ANN	Pr	0.61	0.40	1.00	0.23	0.95	0.53	*0.35*	0.88	*0.26*
	Rc	0.43	0.69	0.99	0.27	1.00	1.00	*0.13*	1.00	*0.11*
	f1	0.51	0.51	1.00	0.25	0.97	0.70	*0.19*	0.94	*0.16*
	Acc.	0.80	0.87	0.99	0.83	0.99	0.91	*0.89*	0.99	*0.88*
	AUC Training	0.86	0.88	1.00	0.78	0.99	0.95	*0.90*	1.00	*0.83*
	AUC Testing	0.86	0.87	1.00	0.76	0.99	0.90	*0.87*	1.00	*0.80*
KNN	Pr	1.00	1.00	1.00	0.83	0.99	0.87	0.91	1.00	1.00
	Rc	1.00	0.99	1.00	0.81	1.00	0.87	0.97	1.00	0.98
	f1	1.00	0.99	1.00	0.82	1.00	0.87	0.94	1.00	0.99
	Acc.	0.99	0.99	1.00	0.96	0.99	0.97	0.98	1.00	0.99
	AUC Training	1.00	1.00	1.00	0.99	1.00	0.99	0.99	1.00	1.00
	AUC Testing	0.99	0.99	1.00	0.96	0.99	0.98	0.99	1.00	0.99

4.3 PERFORMANCE EVALUATION OF ML MODELS USING 13-FEATURE SET

The 13-feature set is selected based on previous findings in section 4.2.2 to optimise the balance between model performance and computational efficiency. This set includes the most critical features necessary for effectively detecting and identifying a wide range of IoT multilayer attacks. By reducing the feature count to 13, the model's processing speed is improved, and computational resource requirements are minimised, making the models more suitable for real-time applications.

Despite the reduced feature count, the 13-feature set maintains high detection accuracy by retaining essential information. It avoids the significant performance drop seen with the 9-feature set, particularly in distinguishing Normal traffic, SQL injection, XSS, and password attacks, ensuring a robust detection capability.

4.3.1 Evaluation of ML Models for IoT Multilayer Attacks Detection Using 13-Feature Set

The performance metrics (precision, recall, F1-score, accuracy, and AUC for both training and testing datasets) of different machine learning models for distinguishing IoT multilayer attacks from normal traffic using the 13-feature set are analysed, as shown in Table 4.7.

When comparing the models, KNN stands out as the top performer, achieving perfect scores across all metrics for both normal and multilayer attack detection. Specifically, KNN achieved precision, recall, F1-score, accuracy, and AUC values ranging from 99% to 100% for both categories, indicating exceptional reliability and performance.

RF, DT, and ANN models also exhibit strong performance, with high values across all metrics ranging from 79% to 100%, making them reliable choices for IoT multilayer attack detection. However, ANN shows slightly lower precision and accuracy for normal traffic at 68% and 83%, respectively.

NB, while stable in terms of AUC, shows the lowest performance overall, indicating that it may not be the best choice for this application.

Table 4.7. Comparative analysis of IoT multilayer attacks detection using different ML models.

ML Model	Metric	Normal	Multilayer
NB	Precision	0.34	0.97
	Recall	0.51	0.44
	f1-score	0.54	0.61
	Accuracy	0.56	0.56
	AUC Training	0.92	0.92
	AUC Testing	0.92	0.92
RF	Precision	0.83	0.98
	Recall	0.95	0.94
	f1-score	0.88	0.96
	Accuracy	0.94	0.94
	AUC Training	0.98	0.98
	AUC Testing	0.98	0.98
DT	Precision	0.79	1.00
	Recall	1.00	0.92
	f1-score	0.88	0.96
	Accuracy	0.93	0.93
	AUC Training	0.98	0.98
	AUC Testing	0.97	0.97
ANN	Precision	0.68	0.99
	Recall	0.97	0.86
	f1-score	0.80	0.92
	Accuracy	0.83	0.99
	AUC Training	0.99	0.99
	AUC Testing	0.99	0.99
KNN	Precision	1.00	1.00
	Recall	1.00	1.00
	f1-score	1.00	1.00
	Accuracy	0.99	0.99
	AUC Training	0.99	0.99
	AUC Testing	0.99	0.99

Figure 4.6 visualises the IoT multilayer evaluation metrics (precision, recall, F1-score, accuracy, and AUC) from Table 4.7, providing a clear comparison of the classification algorithms. Highlighting the exceptional performance of KNN and the strong performances of RF, DT, and ANN models.

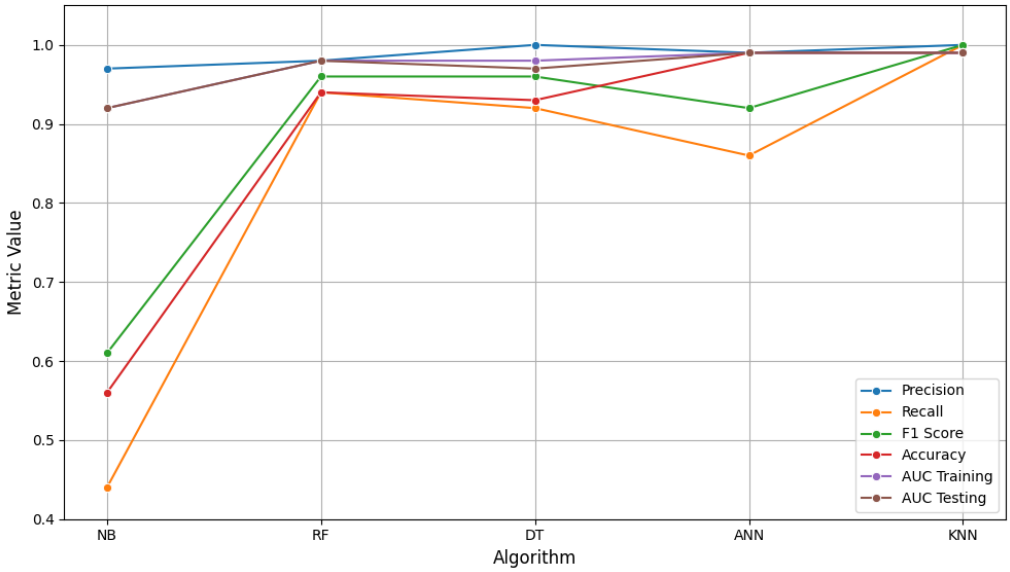


Figure 4.6. Comparison of classification algorithms in IoT multilayer attacks detection.

4.3.2 Evaluation of ML Models for IoT Multilayer Attacks Identification Using 13-Feature Set

In this section, the performance of various machine learning models for identifying IoT multilayer attacks is comprehensively evaluated using the 13-feature set. The evaluation metrics include precision, recall, F1-score, testing accuracy, AUC for training, and AUC for testing, as detailed in Table 4.5 (in Section 4.2.2).

To provide a clear comparison and facilitate understanding, these metrics are visualised in Figure 4.7, which highlights the key performance indicators for each model across different attack types.

In this figure, KNN and RF generally show the highest precision across most attack types, with KNN achieving nearly perfect precision for all attack types (ranging from 70% to 100%). NB shows lower precision of 2% for XSS attacks, suggesting that this area requires significant

improvement. Such failings have a significant impact on the practical deployment of the model, resulting in numerous false alerts or missed attacks.

KNN and RF again lead in recall, indicating their effectiveness in identifying true positive instances of attacks. ANN shows a high recall for most attack types, though it dips for normal traffic, DDoS_HTTP, and password attacks, achieving 44%, 30% and 30% respectively. NB shows lower recall of 4% for XSS attacks. KNN consistently achieves high F1-scores across all attack types, followed by RF, DT, and ANN. NB shows lower F1-scores for XSS attacks, indicating a balance between precision, recall and F1-scores that needs improvement.

All the models demonstrate good testing accuracy across all attack types ranging from 72% to 100%. KNN outperforms the other models, with an accuracy of 100% in detecting DDoS_UDP and MITM, 99% in detecting normal traffic, DDoS_TCP, and DDoS_ICMP, 97% in detecting XSS, and 94% in detecting DDoS_HTTP, SQL injection, and password attacks.

The AUC values for both training and testing are high for KNN, RF, DT, and ANN across all attack types, suggesting strong model performance. NB shows stable AUC but lower values for XSS attacks.

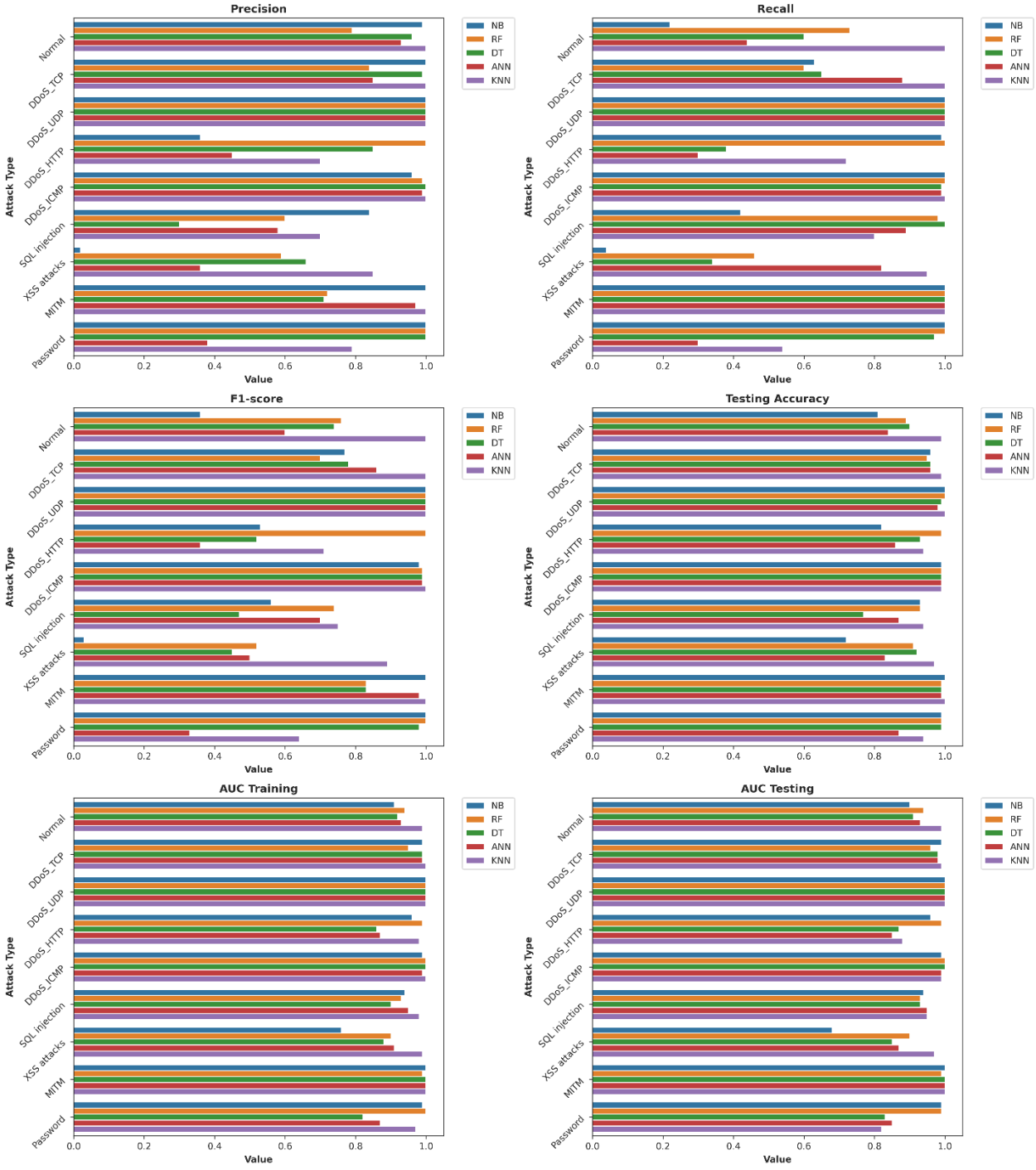


Figure 4.7. Precision, recall, f1-score, testing accuracy, and AUC for 13-feature set.

4.3.3 Detailed Evaluation of KNN Model Using 13-Feature Set

This section delves deeper into the performance of the KNN model, which has shown superior performance in detecting and identifying IoT multilayer attacks using the 13-feature set. The detailed evaluation includes an analysis of the confusion matrix and the Receiver Operating Characteristic (ROC) curve, providing further insights into the model's effectiveness.

Confusion Matrix Analysis

Figure 4.8 displays the confusion matrix using the KNN model for binary classification of normal traffic and multilayer attacks. It indicates that a total of 7,364 instances are predicted as "Normal" and 24,179 as "Multilayer Attacks". Only 18 instances are incorrectly labeled as "Multilayer Attacks", and 26 instances are incorrectly classified as "Normal". This means that the model has a high true positive rate, suggesting it is effective at detecting "Multilayer Attacks". The relatively low rates of false positives and false negatives indicate that the model is also capable of correctly identifying "Normal" traffic.

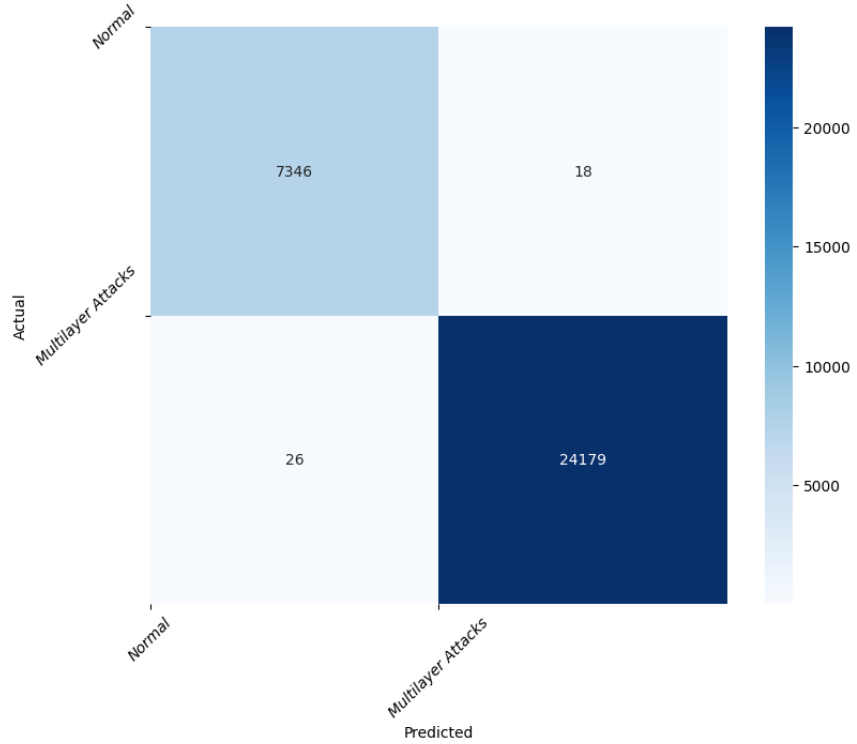


Figure 4.8. Confusion matrix for attack detection using KNN model with 13-feature set.

Figure 4.9 presents the confusion matrix for multiclass classification, illustrating the KNN model's effectiveness in identifying various types of multilayer attacks. The model demonstrates high accuracy, correctly predicting 7,333 samples of Normal traffic and showing strong performance in identifying DDoS_TCP (3,059 samples) and DDoS_UDP (4,416 samples) with minimal misclassifications. The KNN model also performed well in classifying DDoS_HTTP attacks, accurately predicting 2,270 samples, though some were misclassified as DoS_ICMP and SQL Injection.

For DoS_ICMP, 4,187 samples were correctly identified, with few errors. The model effectively classified 2,472 SQL Injection samples, despite minor misclassifications into DDoS_HTTP and Password categories. Similarly, the model accurately identified 2,835 XSS samples and 358 MITM samples, both with minimal errors. Lastly, for Password attacks, the model correctly classified 1,606 samples, though some were misclassified as SQL Injection and DDoS_HTTP.

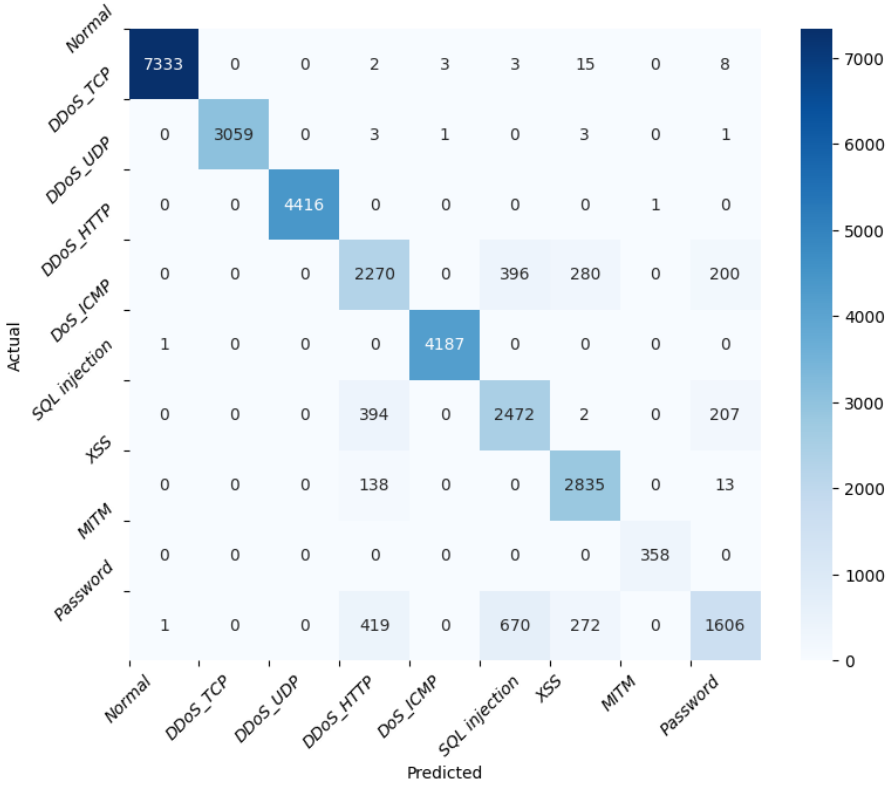


Figure 4.9. Confusion matrix for attack identification using KNN model with 13-feature set.

ROC Curve Analysis

Figure 4.10 presents the ROC curve for the KNN model during binary classification testing, illustrating the model's capability to distinguish between normal and Abnormal (multilayer attacks). The curve's proximity to the top left corner indicates a high true positive rate and a low false positive rate, demonstrating the model's robustness. The ROC curve demonstrates an AUC value of 100% for both normal traffic and multilayer attacks, indicating perfect classification ability.

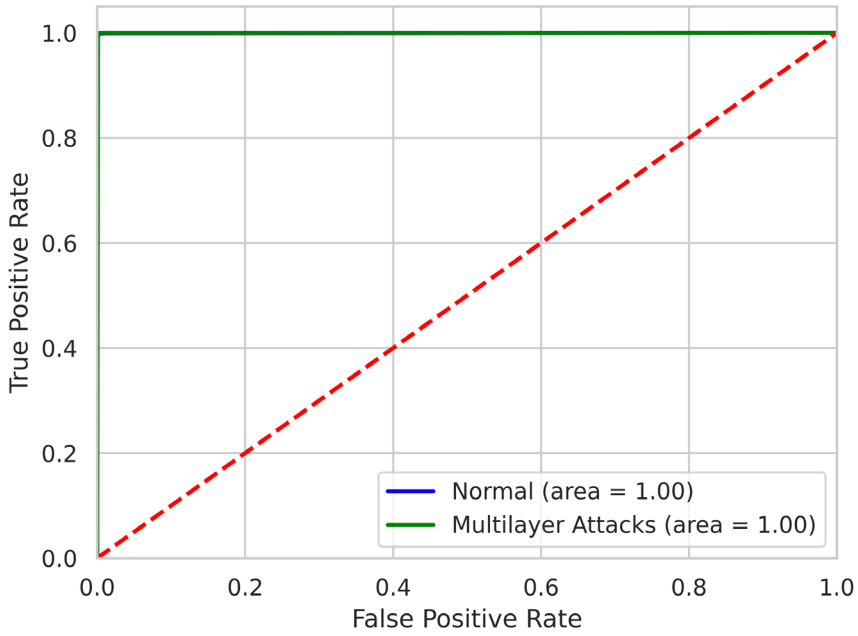


Figure 4.10. ROC curve for binary classification using KNN model.

Figure 4.11 presents the ROC curve for the KNN model during testing. The curve illustrates the model's capability to distinguish between normal and attack classes. The curve shows high AUC values across multiple attack types, reflecting excellent distinguishing capabilities. Specifically, the AUC values are as follows: Normal traffic, DDoS_TCP, DDoS_UDP, DoS_ICMP, and MITM all achieve an AUC of 100%, showcasing the model's perfect performance in these categories. XSS attacks and SQL Injection also have high AUC values of 98% and 95%, respectively, indicating

strong performance. However, DDoS_HTTP and Password attacks have slightly lower AUC values of 88% and 82%, respectively, suggesting some improvement in these specific areas.

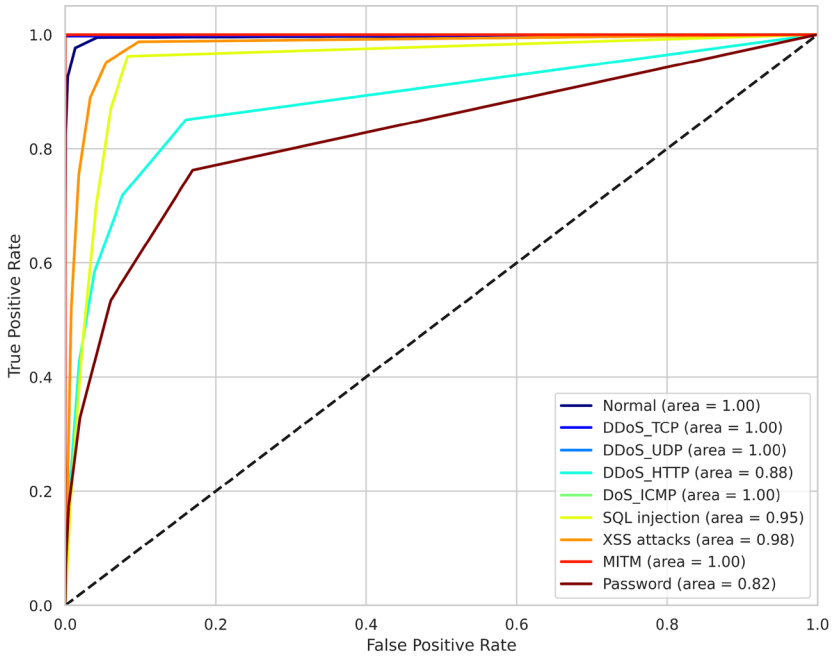


Figure 4.11. ROC curve for multiclass classification using KNN model.

4.4 CHAPTER SUMMARY

The chapter evaluates the hyperparameter tuning and performance of five machine learning models (DT, RF, KNN, ANN, and NB) for detecting and identifying multilayer attacks in IoT networks using the Edge-IIoTset dataset. The study uses random search for hyperparameter tuning and assesses the models using various feature sets, including 62 features, 34 common features, 13 features, and 9 features.

Feature selection methods in the binary classification, MI and IG significantly improved model accuracy, especially for ANN model. MI with permutation tests is particularly effective as the classification models achieve the highest accuracy. In contrast, the least effective method for feature selection is DTE.

Model performance varies with different feature sets in multiclass classification. Using 62 features, KNN and RF demonstrate the highest overall performance. However, NB, RF, DT, and ANN struggle to detect SQL injection and XSS attacks. With 34 features, KNN and RF continue to show evidence of strong performance, but NB, RF, and ANN struggle with XSS, SQL injection, and password attacks detection. The 9-feature set is less effective than the 13-feature set, particularly for detecting normal traffic, password, SQL injection, and XSS attacks.

Using the 13-feature set for binary classification, the KNN emerges as the top performer, achieving near-perfect scores ranging from 99% to 100% across all evaluation metrics. In multiclass classification, the KNN continues to be the top performer, achieving an accuracy above 94%, followed by RF, DT, and ANN. In contrast, the NB is the weakest performer, particularly in detecting XSS, with scores of 2%, 4%, and 3% in precision, recall, and F1-score respectively. This indicates that the NB in the 13-feature set needs improvement to reduce the high rates of false positives and false negatives.

The proposed SAIDS incorporates multilayer IoT attacks detection, distinguishing it from recent studies that utilised the Edge-IIoTset dataset. Notably, it employs only 13 features, significantly fewer than the 20 to 63 features used in other research, thereby enhancing both its efficiency and effectiveness in detecting IoT multilayer attacks.

5 SAIDS EVALUATION USING ADDITIONAL DATASETS

In the previous chapters, specifically Chapters 3 and 4, the SAIDS framework was assessed using the Edge-IIoTset dataset. Building upon this foundation, this chapter introduces additional datasets from various attack scenarios generated from both simulated and real-world experiments to further evaluate the robustness and effectiveness of the SAIDS framework, as shown in Figure 5.1.

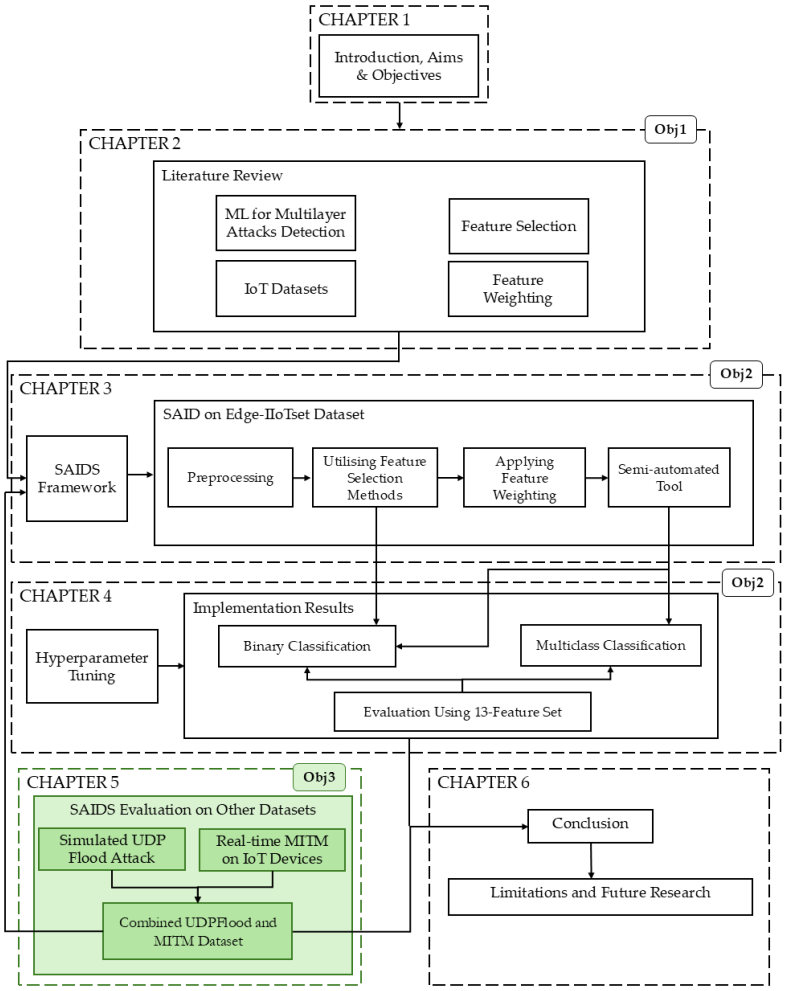


Figure 5.1. Thesis structure showing Chapter 5's placement within the overall project.

5.1 Simulated UDP Flood Attack and Real-Time MITM Attacks on IoT Devices

The evaluation was performed in a simulated environment with a dataset from real-world IoT devices and simulated one, as illustrated in Figure 5.2. The first part utilised the Cooja simulation platform on the Contiki operating system to simulate a UDP flood attack. Cooja is an IoT simulator tool that provides a virtual environment where researchers can simulate and monitor the behaviour of IoT networks under various conditions, including network attacks, without the need for physical hardware. Although it is a Java-based simulator, Cooja allows the nodes to be programmed in the C language (Farea and Küçük, 2022; Contiki-NG, 2022).

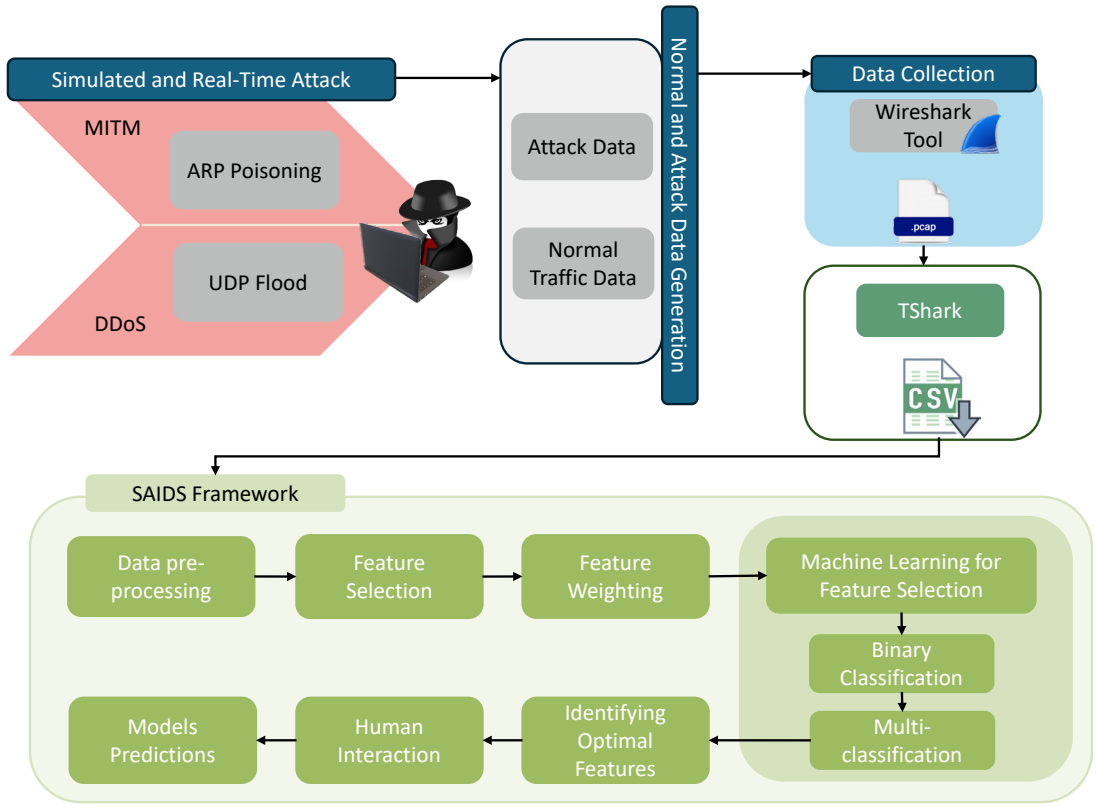


Figure 5.2. Framework for generating and evaluating simulated and real-time IoT attacks

The second part involved connecting real IoT smart home devices and launching a MITM attack, specifically an ARP poisoning attack, on a Xiaomi Redmi Note 9S device. The data collected using Wireshark from these experiments were then converted into CSV format using TShark. The generated datasets were pre-processed and merged into one dataset before being employed in the

SAIDS framework to assess its adaptability and accuracy in detecting these two types of IoT multilayer attacks. The devices and software used in creating both datasets are listed in Tables 5.1 and 5.2.

Table 5.1. Devices used in the simulated and real-time attacks.

Device	Description
Xiaomi Redmi Note 9S	Target device for the MITM attack, running Android OS with 128GB storage.
Lenovo Laptop	Used for capturing network traffic, executing ARP poisoning.
	Specifications: 8GB RAM, Windows 10 Enterprise, 236GB SSD.
Amazon Echo Dot (3rd Gen)	Provides normal traffic data with Alexa voice assistant integration.
Apple HomePod mini	Provides normal traffic data with Siri voice assistant integration.

Table 5.2. Software tools used in the simulated and real-time attacks.

Component Name	Description
Wireshark	Open-source network analyser used to capture and analyse network traffic.
Kali Linux	Linux distribution used for hacking the Xiaomi device.
VMware Workstation 17	Virtualisation software used to run Kali Linux on Windows 10 Enterprise.
Contiki 2.7	Operating system used to run the Cooja Simulator.
Cooja Simulator	Simulator used to implement the IoT traffic and the UDP flood attack.
6LoWPAN Analyser Tool	A tool integrated with Cooja to capture the traffic, saving it in .PCAP format.
Arpspoof	Tool used for performing ARP poisoning attacks.
Tshark	Command-line version of Wireshark used for extracting features from the .PCAP files then converting them into CSV files.

5.1.1 Simulated UDP Flood Attack

This subsection details the evaluation using the Cooja simulation platform on the Contiki operating system to simulate a UDP flood attack, providing insights into the framework's performance under simulated high-traffic conditions. The tools used in creating this testbed are listed in Tables 5.1 and 5.2.

A) Simulated Normal and UDP Flood Attack Data Generation and Collection

The experimental setup consisted of a network of 10 IoT devices (nodes) connected to a single server to emulate a typical IoT environment where multiple devices communicate with a central access point.

- **Normal Traffic Capture**

Initially, as shown in Figure 5.3, normal traffic was generated by the 10 nodes, each with a unique IP address (nodes 2-11, represented by yellow circles), interacting with the server (node 1, represented as a green circle). The blue lines between the server (node 1) and the nodes (2, 3, and 11) represent the radio traffic. The mote output window displays the data transmissions between nodes. This traffic was captured without any malicious interference using Wireshark and stored in a .pcap file.

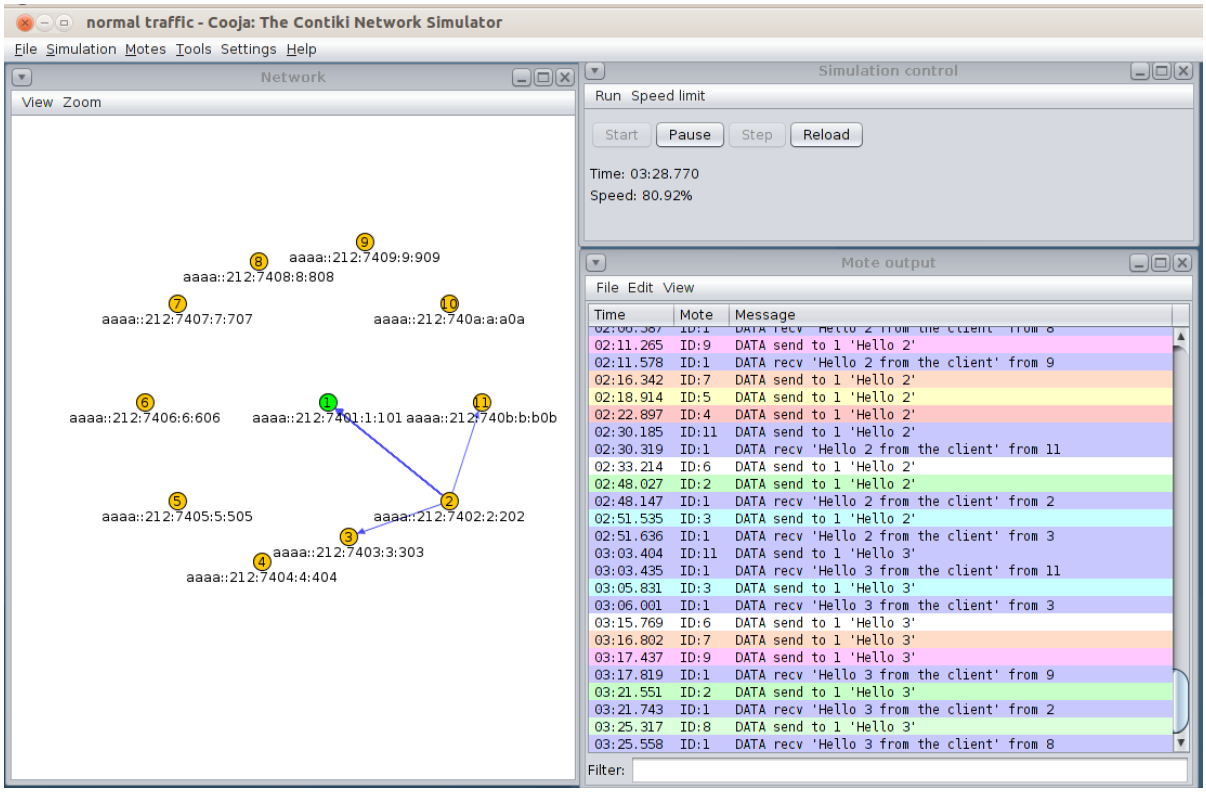


Figure 5.3. Normal traffic scenario and nodes output.

- **Attack Execution**

To simulate a UDP flood attack, as shown in Figure 5.4, node 1 (represented by a green circle) acts as the server, while node 8 (indicated by a red circle) is the malicious node responsible for launching the attack. The other nodes (nodes 2-7 and 9-10), shown as yellow circles, functioned as normal nodes. Node 8 is programmed to flood the network by sending a high volume of UDP packets to the router every second. In contrast, the normal nodes sent UDP packets at regular intervals, such as once every minute. This approach followed the methodology described in the research conducted by (Farea and Küçük, 2022). The network traffic during the UDP flood attack was captured using Wireshark, integrated with the Cooja simulator. The resulting traffic was stored in a separate .pcap file.

Tshark was used to extract the features from the two PCAP files and store them in CSV format, resulting in the **UDPFlood_Attack** dataset with two traffic categories: normal traffic and UDP flood attack traffic.

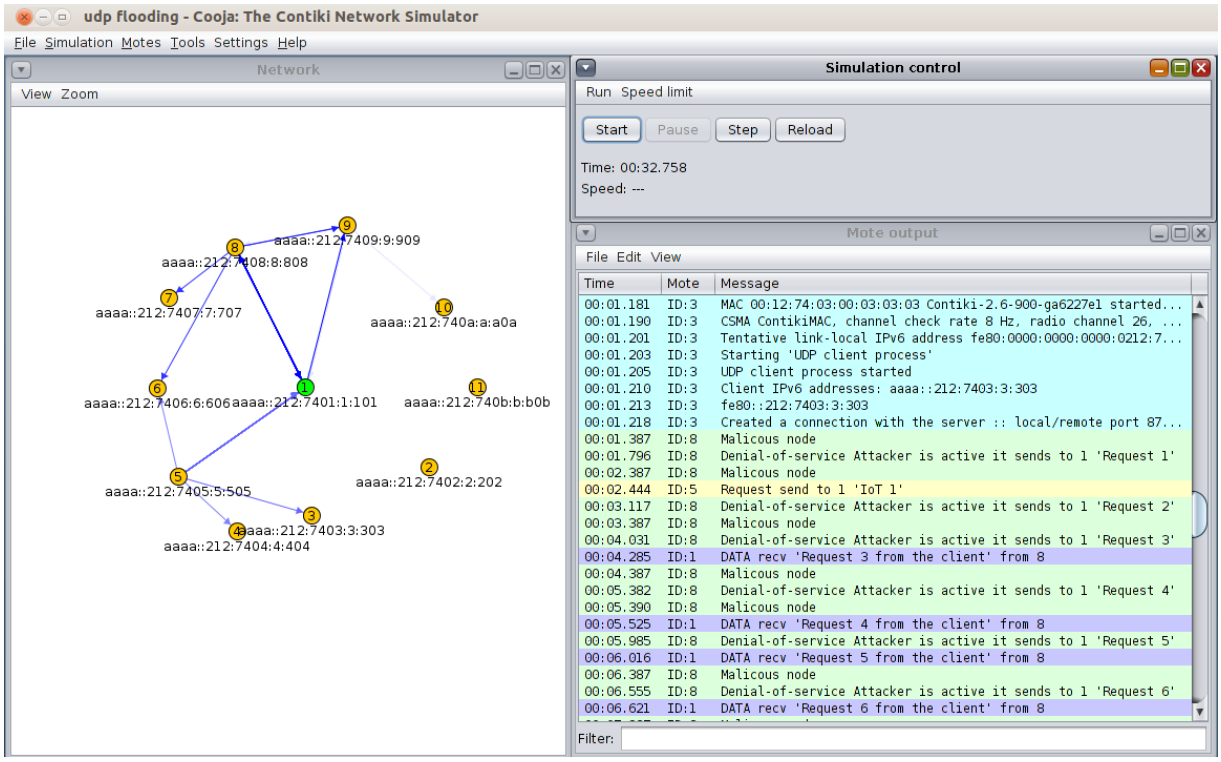


Figure 5.4. UDP flood attack scenario and nodes output.

5.1.2 Real-Time MITM Attacks on IoT Devices

This subsection covers the evaluation of a MITM (ARP poisoning) attack on a Xiaomi Redmi Note 9S device. This evaluation helps in understanding the SAIDS framework's efficacy in detecting real-world security threats.

A) MITM_Attack Testbed: Hardware and Software Components

Several hardware and software tools were used in creating this testbed, as shown in Table 5.1 and Table 5.2. The hardware setup included various devices to create a real-world home network environment, as illustrated in Figure 5.5. Each device served a specific purpose in the experiment. The Xiaomi Redmi Note 9S served as the target device for the MITM attack. The Lenovo laptop was employed to execute the ARP poisoning attack via Kali Linux and capture network traffic using Wireshark from the Xiaomi device, Amazon Echo Dot (3rd Generation), and Apple HomePod mini.

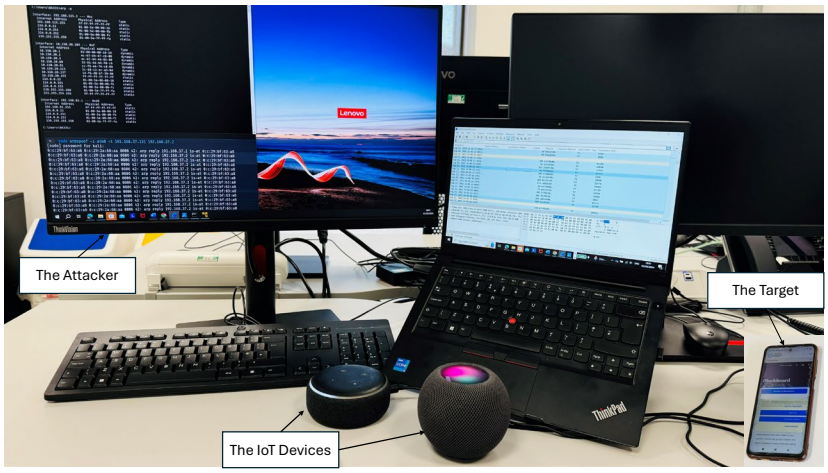


Figure 5.5. Hardware setup for real-world MITM attack experiment.

B) Normal and MITM Attack Data Generation and Collection

In this phase, IoT data was generated from various components connected to the home router, including the Xiaomi Redmi Note 9S, Apple HomePod, and Amazon Echo. Initially, normal traffic data was collected by running Wireshark to capture traffic from these devices. The IP and MAC addresses of these devices are listed in Table 5.3. The normal data generation continued for a few hours. During this period, an ARP poisoning attack was launched using the Arpspoof tool against the Xiaomi device at different times to ensure a comprehensive dataset.

By launching Arpspoof, the Kali Linux machine (attacker) positioned itself in the middle of the connection between the home router and the Xiaomi Redmi Note 9S. The Kali Linux machine acts as a router for the Xiaomi device by having the same MAC address as the router, and it acts as the Xiaomi device for the home router by having the same MAC address as the Xiaomi device. This setup allowed the attacker to effectively intercept the traffic between the two, enabling the theft of information as the Xiaomi device communicated with the network.

Table 5.3. IP and MAC addresses of devices used in MITM attack.

Device	IP Address	MAC Address
Home router	192.168.8.1	94:E9:EE:8D:81:91
Xiaomi Redmi Note 9S	192.168.8.100	18:87:40:EF:19:B1
Kali Linux	192.168.8.157	08:00:27:5C:65:26
Apple HomePod	192.168.8.152	4C:20:B8:DB:B2:8C
Amazon Echo	192.168.8.111	08:84:9D:D5:C7:76

Features were extracted from the PCAP files specific to the MITM attack scenario using the same process described in Section 5.1.1. This resulted in the **MITM_Attack** dataset, which includes both normal traffic and MITM attack traffic.

5.2 DATASETS EVALUATION RESULTS

The **UDPFlood_Attack** and **MITM_Attack** datasets were merged into a single dataset called the Combined UDPFlood and MITM Attacks (CUMA) dataset. This comprehensive dataset includes both multilayer attack scenarios as well as normal traffic. It contains a total of 25,027 instances, with 15,988 instances representing normal traffic, 14,598 instances representing UDP flood attack traffic, and 1,740 instances representing MITM attack. The dataset includes 36 features (excluding the "label" and "Attack_type" features), as shown in Table 5.4.

Table 5.4. List of 36-features of the CUMA dataset.

No.	Feature Name	No.	Feature Name	No.	Feature Name
1	frame.time_WithoutIP	13	http.request.method	25	tcp.checksum
2	frame.time_WithIP	14	http.content.type	26	tcpstream
3	arp.hw.type	15	http.request.full_uri	27	ip.src_host
4	arp.protocol.type	16	http.connection	28	ip.dst_host
5	arp.hw.size	17	http.cookie	29	udp.length
6	arp.protocol.size	18	tcp.srcport	30	udp.payload
7	arp.opcode	19	tcp.dstport	31	icmp.checksum
8	arp.src.MAC	20	tcp.seq	32	udp.src.port
9	arp.src.proto_ipv4	21	tcp.ack	33	icmp.flags
10	arp.dst.MAC	22	tcp.win.size	34	udp.dst.port
11	arp.dst.proto_ipv4	23	tcp.flags	35	udp.stream
12	dns.flag	24	tcp.len	36	udp.time_delta

- **Identifying Common Features**

Pre-processing was performed on the CUMA dataset to prepare it for accurate classification. The next step involves identifying common features between both multilayer attacks. Out of the 36 features, 21 have been identified as common features. These features are listed in Table 5.5.

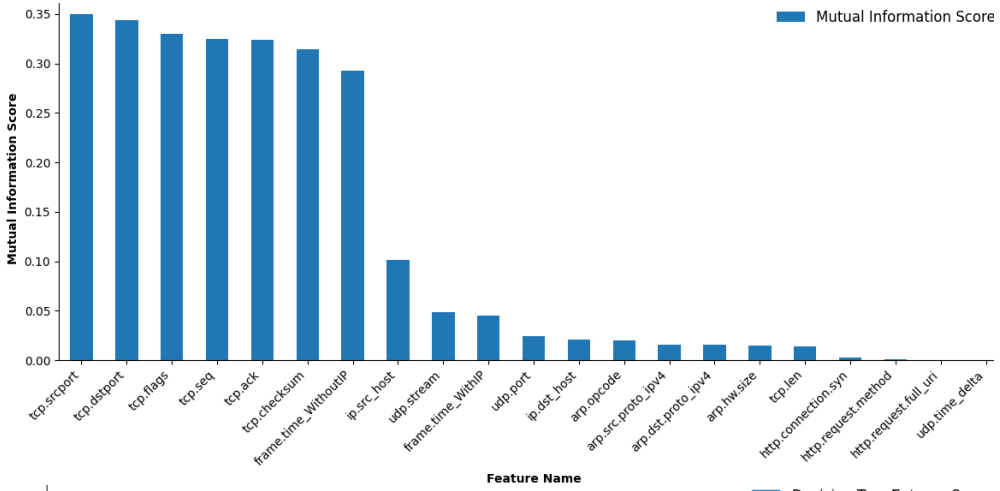
Table 5.5. Twenty-one common features between the multilayer attacks in the CUMA dataset.

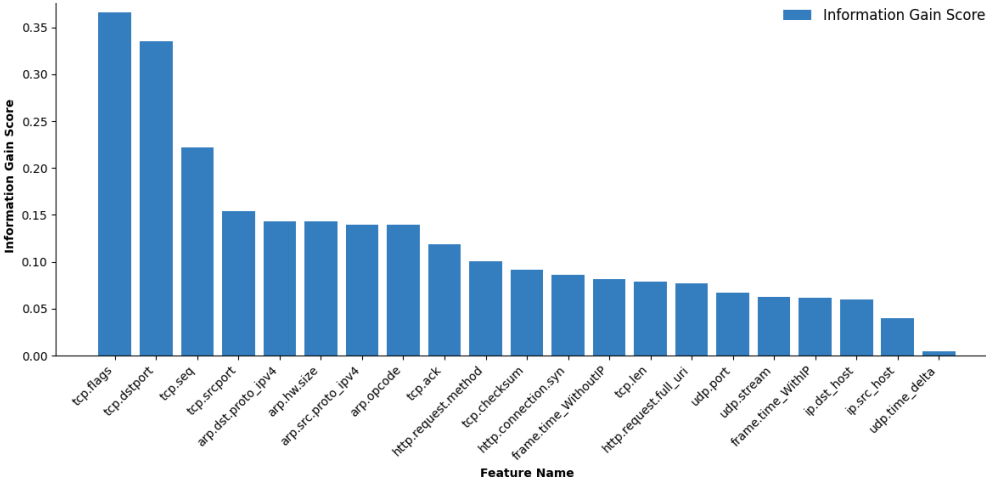
No.	Feature Name		No.	Feature Name
1	frame.time_WithoutIP		12	tcp.seq
2	frame.time_WithIP		13	tcp.ack
3	arp.hw.size		14	tcp.flags
4	arp.opcode		15	tcp.len
5	arp.src.proto_ipv4		16	tcp.checksum
6	arp.dst.proto_ipv4		17	ip.src_host
7	http.request.method		18	ip.dst_host
8	http.request.full_uri		19	udp.port
9	http.connection.syn		20	udp.stream
10	tcp.srcport		21	udp.time_delta
11	tcp.dstport			

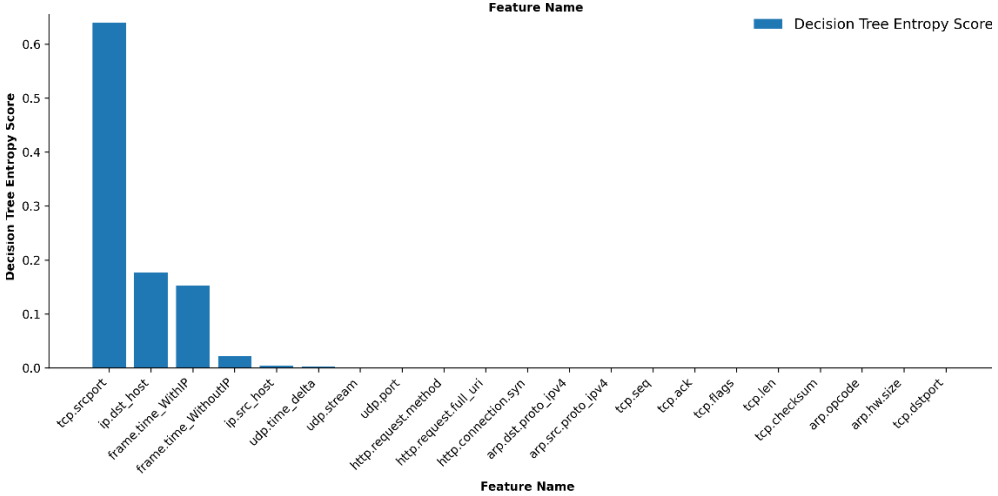
- **Feature Selection Methods**

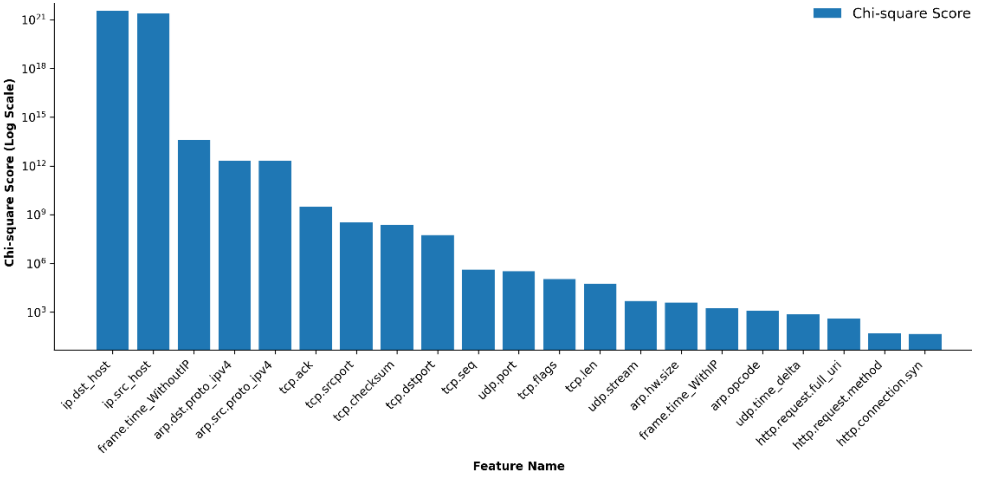
Feature selection techniques, including mutual information (MI), information gain (IG), random forest (RF), decision tree entropy (DTE), principal component analysis (PCA), and chi-square (chi^2), were applied to the 21common features between UDP flood and MITM multilayer attacks. The features scores for each method are presented in Figure 5.6.

The Mutual Information identified 19 features, ranking "tcp.srcport", "tcp.dstport", "tcp.flags", and tcp.seq as the top features. Both IG and Chi^2 identified all 21 features as significant. For Information Gain, the highest-scoring features are "tcp.flags", "tcp.dstport", and "tcp.seq", while the top features in the Chi-square method are "ip.dst_host" and "ip.src_host". DTE identified 6 features, highlighting "tcp.srcport", "ip.dst_host", and "frame.time_WithoutIP" as the most significant. PCA identified 11 features, giving the highest scores to "frame.time_WithoutIP", "frame.time_WithIP", and "arp.hw.size". RF identified 18 features as significant for detecting multilayer attacks, with "ip.dst_host", "tcp.srcport", and "frame.time_WithIP" being the most important.


Mutual Information Score
Mutual Information Score
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
tcp.srcport
tcp.dstport
tcp.flags
tcp.seq
tcp.ack
tcp.checksum
frame.time_WithoutIP
ip.src_host
udp.stream
frame.time_WithIP
udp.port
ip.dst_host
arp.opcode
arp.src.proto_ipv4
arp.dst.proto_ipv4
arp.hw.size
tcp.len
http.connection.syn
http.request.method
http.request.full_uri
udp.time_delta
Feature Name


Information Gain Score
Information Gain Score
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
tcp.flags
tcp.dstport
tcp.seq
tcp.srcport
arp.dst.proto_ipv4
arp.hw.size
arp.src.proto_ipv4
arp.opcode
tcp.ack
http.request.method
tcp.checksum
http.connection.syn
frame.time_WithoutIP
tcp.len
http.request.full_uri
udp.port
udp.stream
frame.time_WithIP
ip.dst_host
ip.src_host
udp.time_delta
Feature Name


Decision Tree Entropy Score
Decision Tree Entropy Score
0.6
0.5
0.4
0.3
0.2
0.1
0.0
tcp.srcport
ip.dst_host
frame.time_WithIP
frame.time_WithoutIP
ip.src_host
udp.time_delta
udp.stream
udp.port
http.request.method
http.request.full_uri
http.connection.syn
arp.dst.proto_ipv4
arp.src.proto_ipv4
tcp.seq
tcp.ack
tcp.flags
tcp.len
tcp.checksum
arp.opcode
arp.hw.size
tcp.dstport
Feature Name


Chi-square Score
Chi-square Score (Log Scale)
10^{21}
10^{18}
10^{15}
10^{12}
10^{9}
10^{6}
10^{3}
ip.dst_host
ip.src_host
frame.time_WithoutIP
arp.dst.proto_ipv4
arp.src.proto_ipv4
tcp.ack
tcp.srcport
tcp.checksum
tcp.dstport
tcp.seq
udp.port
tcp.flags
tcp.len
udp.stream
arp.hw.size
frame.time_WithIP
arp.opcode
udp.time_delta
http.request.full_uri
http.request.method
http.connection.syn
Feature Name

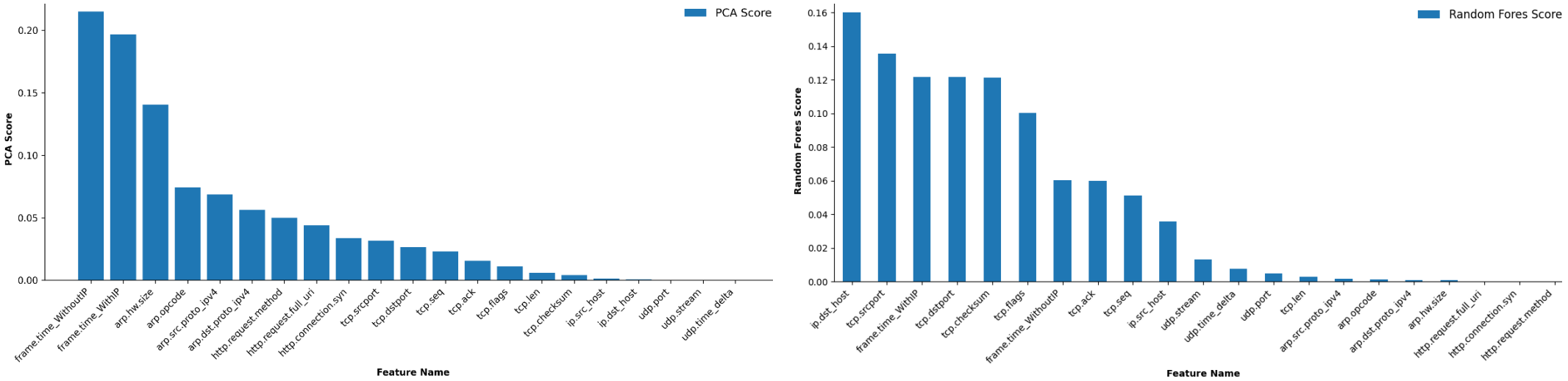


Figure 5.6. Comparative analysis of the results of feature selection methods.

- **Feature Weighting**

The scores obtained from the feature selection methods were normalised and then combined using Equation 6 in Section 3.2.3 to assign a weight to each feature. Figure 5.7 presents the sorted weights of the features based on their importance. It shows that "tcp.srcport" has the highest weight, approximately 0.56, followed by "tcp.dstport" at around 0.46, and "tcp.flags" at 0.43.

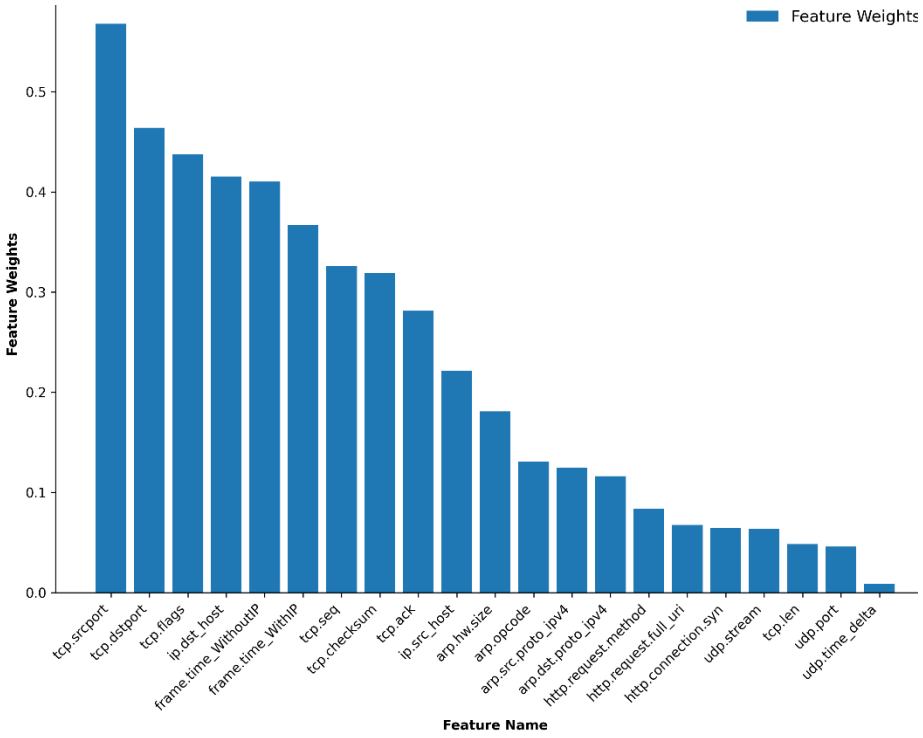


Figure 5.7. Feature weights analysis.

- **Semi-automated tool for Identifying Optimal Features**

The semi-automated tool successfully identified 7 significant features for detecting and identifying UDP flood and MITM attacks using the KNN machine learning model, as shown in Figures 5.8 and 5.9. These 7 features are listed in Table 5.6, with "tcp.srcport" and "tcp.dstport" being the most significant.

Tabel 5.6. The 7-significant features for detecting and identifying UDP flood and MITM multilayer attacks.

No.	Feature Name
1	tcp.srcport
2	tcp.dstport
3	tcp.flags
4	ip.dst_host
5	frame.time_WithoutIP
6	frame.time_WithIP
7	tcp.seq

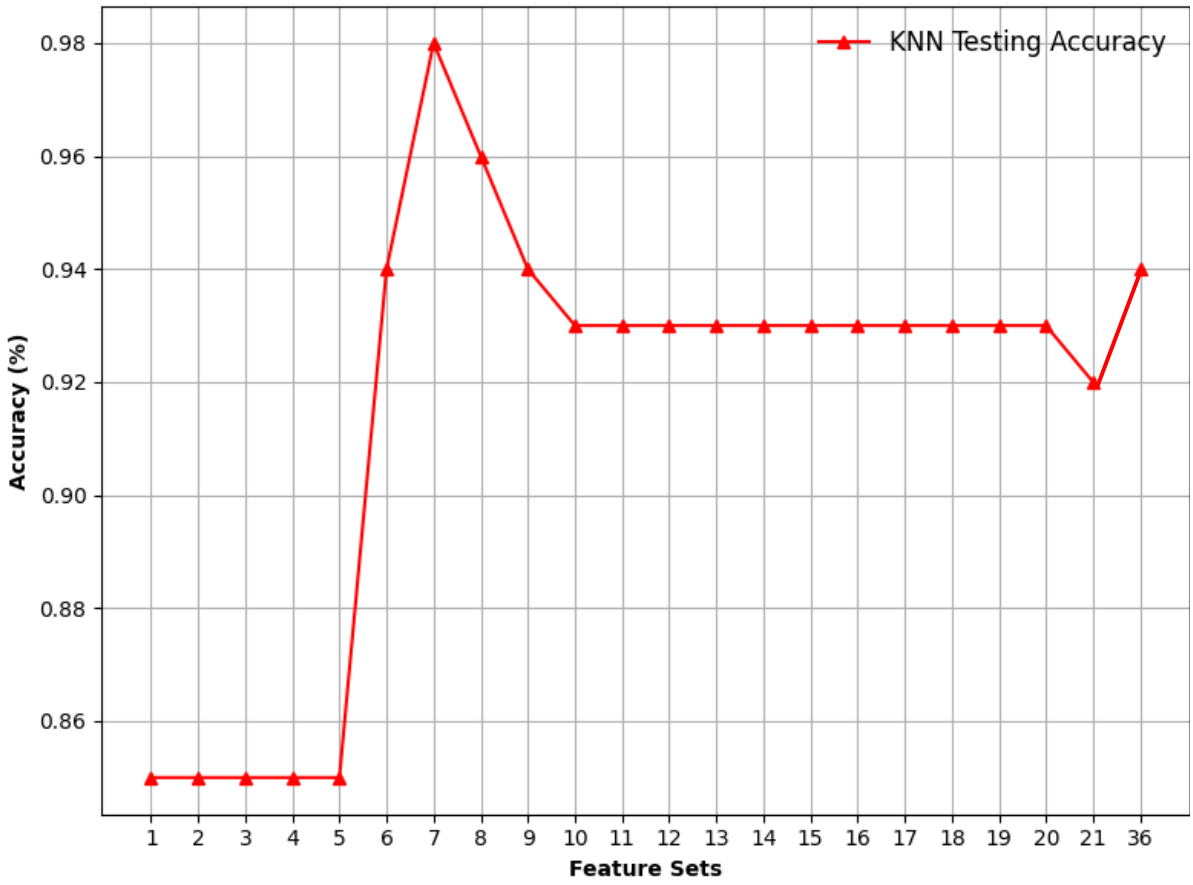


Figure 5.8. Visualising binary classification using KNN model.

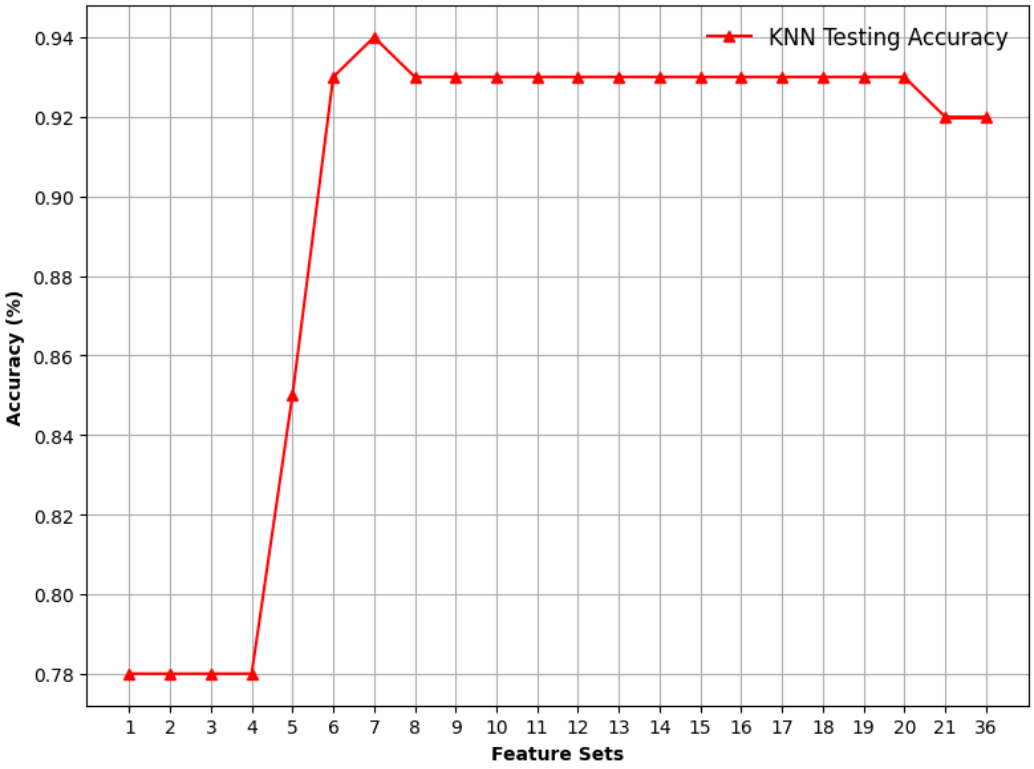


Figure 5.9. Visualising multiclass classification using KNN model.

- **Performance Evaluation of ML Models Using the 7 Significant Features.**

The identified 7 significant features were then used to analyse the performance of machine learning models in distinguishing between normal and multilayer attacks traffic.

Binary Classification Results

Table 5.7 shows the performance of different machine learning algorithms applied to the CUMA dataset. The models were evaluated based on precision, recall, F1-score, accuracy, and AUC for both normal and attack traffic categories.

The KNN and DT algorithms achieved good scores across all metrics, ranging from 86% to 100% for both normal and multilayer traffic. This indicates that these models are highly reliable in distinguishing between normal and malicious multilayer activities. The ANN also performed well,

with metrics ranging from 84% to 100%. Both the RF and NB models demonstrated good performance, with all metrics ranging from 70% to 100%. These models achieved 99% and 100% recall for multilayer traffic, highlighting their capability to correctly identify all instances of attack traffic.

Table 5.7. Binary classification results of SAIDS framework on CUMA.

Alg	Metric	Normal	Multilayer
NB	Precision	100	0.70
	Rc	0.76	0.99
	f1	0.86	0.82
	Acc.	0.85	0.85
	AUC Training	0.81	0.81
	AUC Testing	0.81	0.81
RF	Pr	1.00	0.70
	Rc	0.76	1.00
	f1	0.87	0.82
	Acc.	0.85	0.85
	AUC Training	0.98	0.98
	AUC Testing	0.98	0.98
DT	Pr	1.00	0.86
	Rc	0.91	1.00
	f1	0.95	0.92
	Acc.	0.94	0.94
	AUC Training	0.98	0.98
	AUC Testing	0.98	0.98
ANN	Pr	1.00	0.84
	Rc	0.89	1.00
	f1	0.94	0.91
	Acc.	0.93	0.93
	AUC Training	0.97	0.97
	AUC Testing	0.97	0.97
KNN	Pr	0.86	0.95
	Rc	0.98	0.70
	f1	0.91	0.81
	Acc.	0.88	0.88
	AUC Training	0.98	0.98
	AUC Testing	0.98	0.98

Multiclass Classification Results

The multiclass classification results, presented in Table 5.8, evaluate the same set of models on the CUMA dataset for distinguishing between normal, DDoS_UDP, and MITM traffic types. The evaluation metrics include precision, recall, F1-score, accuracy, and AUC.

The KNN and DT algorithms again performed well, with precision, recall, and F1-scores ranging from 83% to 100%. They showed strong ability in classifying both normal and attack traffic, with particularly high scores for normal and DDoS_UDP traffic but slightly lower precision for MITM traffic at 83%. The ANN model's performance metrics ranged from 81% to 100%. It achieved high recall for MITM traffic but struggled with precision at 81%, particularly for MITM traffic, suggesting a need for optimisation in this area.

Finally, both the RF and NB models continued to demonstrate solid performance, achieving metrics between 65% and 100%. These models achieved perfect recall and F1-scores for DDoS_UDP traffic, underscoring their effectiveness in identifying this type of multilayer attacks. However, for MITM traffic, their precision was lower at 65%, indicating room for improvement.

Table 5.8. Multiclass classification results of SAIDS framework on CUMA.

Alg	Metric	Normal	DDoS_UDP	MITM
NB	Precision	1.00	1.00	0.65
	Rc	0.76	1.00	1.00
	f1	0.87	1.00	0.79
	Acc.	0.85	1.00	0.85
	AUC Training	0.79	1.00	0.90
	AUC Testing	0.79	1.00	0.90
RF	Pr	1.00	0.99	0.65
	Rc	0.76	1.00	1.00
	f1	0.87	1.00	0.79
	Acc.	0.85	1.00	0.85
	AUC Training	0.96	1.00	0.90
	AUC Testing	0.96	1.00	0.90
DT	Pr	1.00	1.00	0.83
	Rc	0.91	1.00	1.00
	f1	0.95	1.00	0.91
	Acc.	0.94	1.00	0.94
	AUC Training	0.98	1.00	0.98
	AUC Testing	0.98	1.00	0.98
ANN	Pr	1.00	1.00	0.81
	Rc	0.89	1.00	1.00
	f1	0.94	1.00	0.89
	Acc.	0.93	1.00	0.93
	AUC Training	0.97	1.00	0.97
	AUC Testing	0.97	1.00	0.97
KNN	Pr	0.98	1.00	0.83
	Rc	0.92	1.00	0.95
	f1	0.94	1.00	0.89
	Acc.	0.93	1.00	0.93
	AUC Training	0.98	1.00	0.98
	AUC Testing	0.98	1.00	0.98

5.3 CHAPTER SUMMARY

This chapter evaluates the SAIDS framework using datasets generated from both a simulated UDP flood attack, conducted using the Cooja simulation platform, and a real-world MITM attack (ARP poisoning) executed on a Xiaomi Redmi Note 9S device. This builds upon the assessments conducted in previous chapters with the Edge-IIoTset dataset. The Combined UDPFlood and MITM Attacks dataset (CUMA) was created by merging the generated UDP flood and MITM attack datasets. This dataset, which comprises 36 features, serves as a comprehensive testbed for assessing the robustness and effectiveness of the SAIDS framework in detecting and classifying multilayer attacks.

By applying the SAIDS framework to the CUMA dataset, 21 common features between the multilayer attacks were identified. These features were further refined using feature selection methods, feature weighting, and a semi-automated tool, which ultimately identified 7 significant features, such as "tcp.srcport" and "tcp.dstport," that are critical for detecting and identifying UDP flood and MITM attacks.

In both binary and multiclass classification tasks, the KNN, DT, and ANN algorithms outperformed other models, achieving good scores between 84% and 100% for binary classification and between 81% and 100% for multiclass classification across all metrics for both normal and multilayer traffic. The RF and NB models demonstrated good performance as well, with metrics ranging from 70% to 100% in binary classification and from 65% to 100% in multiclass classification, though they were slightly less effective, particularly for MITM traffic.

6 CONCLUSION

This chapter summarises the findings and insights gained from the research on developing a Semi-Automated Intrusion Detection System (SAIDS) for multilayer attack detection in IoT networks. The research particularly focuses on the detection and identification of complex IoT multilayer attacks using machine learning techniques. The chapter is organised into three main sections: research contribution, research significance, and limitations with future work.

6.1 RESEARCH CONTRIBUTION

The primary aim of this research was to develop a robust SAIDS capable of detecting and identifying multilayer attacks in IoT networks. This aim was achieved through the smart integration of feature selection and feature weighting within machine learning model, the incorporation of human expertise in the overall process to tackle these attacks using the most significant features, and the evaluation of the system using both simulated and real-world datasets. The research was structured around three key objectives where each was successfully addressed, as outlined below.

The first objective was to investigate the existing machine learning algorithms and identify the current limitations of the standard frameworks for detecting IoT multilayer security attacks, with a particular focus on optimisation of features through feature selection and weighting methods. This objective was implemented in Chapter 2, through a comprehensive literature review that highlighted several key challenges in current Intrusion Detection Systems, particularly in managing the complexity of multilayer attacks in IoT environments. These challenges include the absence of benchmark datasets, difficulties in feature extraction from semi-structured data, and the rigidity of some machine learning models, which struggle to adapt to new types of attacks or attacks targeting new devices. Additionally, the review pointed out the absence of comprehensive computational frameworks that cover all stages of the knowledge discovery process, from data preprocessing to

model evaluation. These insights provided the foundation for developing a more robust and flexible framework, addressing the identified gaps and limitations through the integration of human expertise with machine learning techniques.

The second objective focused on the development, implementation, and optimisation of the parameters of the SAIDS framework. This objective was comprehensively addressed through the work presented in Chapters 3 and 4. In Chapter 3, a detailed methodology was presented, describing the data preprocessing techniques, feature selection methods, feature weighting, the integration of cybersecurity experts' feedback with machine learning algorithms, and two stages of classification (binary and multiclass). Chapter 4 further expanded on the hyperparameter tuning process and the performance evaluation of the SAIDS framework. The system then was evaluated using the Edge-IIoTset. Through this evaluation, 13 significant features were identified for the detection and classification of IoT multilayer attacks. The effectiveness of the SAIDS framework is clearly demonstrated by the KNN model's ability to accurately identify both normal traffic and various multilayer attacks, achieving an accuracy of 99% in binary classification and above 94% in identifying each type of multilayer attacks. These results indicate that the SAIDS framework significantly improved detection accuracy while reducing computational complexity.

The final objective was to evaluate the SAIDS framework using additional datasets derived from both simulated and real-world experiments. This objective was achieved through the work presented in Chapter 5, where the framework was tested on datasets generated from a simulated UDP flood attack and a real-world MITM (ARP poisoning) attack. The results from these evaluations showed that the identified seven significant features successfully detected and classified the IoT multilayer attacks. These results confirmed the robustness and adaptability of the SAIDS framework in detecting multilayer attacks across different IoT environments. The framework's ability to maintain high detection accuracy with a reduced feature set highlighted its potential for real-time application in diverse IoT scenarios.

6.2 RESEARCH SIGNIFICANCE

The development of SAIDS may influence significant implications across IoT security domains. This approach may compensate for the limitations of existing research that mostly focuses on single-layer attacks while offering a comprehensive and innovative approach to IoT multilayer attacks detection method:

1. The proposed approach comprises an ensemble feature analysis technique by combining multiple feature selection and feature weighting methods to optimise the process of extracting the most significant features from IoT datasets utilised for training the machine learning models to develop the intrusion detection systems (IDS) for identifying IoT multilayer attacks.

2. This framework supports the ML training and testing process by minimising the total numbers of features in a dataset, indirectly improving the computational complexity for IDS, reducing the training time while maintaining the required accuracy and enhancing the scalability across various IoT applications.

3. The approach includes Human-Machine Teaming by utilising human expertise in extracting feature selection and tuning of data mining parameters to facilitate a more robust approach in identifying attack patterns. This collaboration between human and machine enhances the system's adaptability and accuracy in detecting multilayer intrusions.

4. It shows a unique visualisation tool that graphically represents how individual features influence the detection process that guides researchers and developers in understanding and selecting the minimum significant features for detecting and identifying multilayer IoT attacks.

5. The framework employs a two-stage classification process where the first stage uses a binary classifier to filter traffic into normal and abnormal categories whereas the second stage applies multiclass classification to identify specific types of multilayer attacks, enabling targeted mitigation strategies.

6. The system is designed to easily integrate and add classifiers as needed. For the specific use case of the data, we utilised classifiers such as Decision Tree, K-Nearest Neighbours, Naive Bayes,

Random Forest, and Artificial Neural Network as they are suitable for this prototype. The research further confirmed the adaptability of the SAIDS framework across different datasets and IoT scenarios, including simulated and real-world environments. This adaptability indicates that the framework can be applied to multiple IoT sectors, serving as a valuable tool for enhancing security measures in critical infrastructure.

6.2.1 Comparative Analysis of IoT Multilayer Attacks Detection Frameworks

As shown in Table 6.1, existing frameworks, such as those by Bansal et al. (2011); Mahale et al. (2017); Mythili and Seetha (2021), while employing traditional techniques for multilayer attack detection, lack critical components like feature selection, feature weighting, and hyper-parameter optimisation, limiting their adaptability in complex IoT environments. These methods suffer from high false positive and false negative rates and struggle to meet the computational demands required for real-time IoT security. Although effective in simulation environments, they face challenges with real-world complexities such as interference and evolving attack patterns.

Table 6.1 Comparative analysis of existing frameworks and SAIDS framework for multilayer IoT attacks.

Ref.	Techniques Used	Multilayer Detection	Feature Selection	Feature Weighting	Hyper-parameter Opt.	Classify
Bansal et al. (2011)	Cross-layer	Yes	No	No	No	Binary
Mahale et al. (2017)	Cross-layer	Yes	No	No	No	Binary
(Sodagudi and Rao, 2014)	Behavior-based anomaly detection	Yes	No	No	No	Binary
Mythili and Seetha (2021)	Distributed mobile agents	Yes	No	No	No	Binary
SAIDS	ML	Yes	Yes	Yes	Yes	Binary, Multiclass

6.2.2 Comparative Analysis of IoT Attack Detection Using Edge-IIoTset Dataset

Similarly, more recent ML and DL models, like those by (Keserwani, Aggarwal and Chauhan, (2023); Tareq et al., (2022); Khacha et al., (2022); Al Nuaimi et al., (2023); Samin et al., (2023); Ullah et al., (2023); Ferrag et al., (2022), have improved detection for single-layer attacks. Table 6.2 presents a comparative analysis of the proposed SAIDS with recent related works using the Edge-IIoTset dataset. The proposed SAIDS stands out by incorporating multilayer IoT attack detection, which is not addressed by the other studies. Furthermore, this approach utilises 13 significant features, as illustrated in Figure 6.1, which is fewer than the features used in most other studies, ranging from 20 to 63 features with some studies relying on manual feature selection methods.

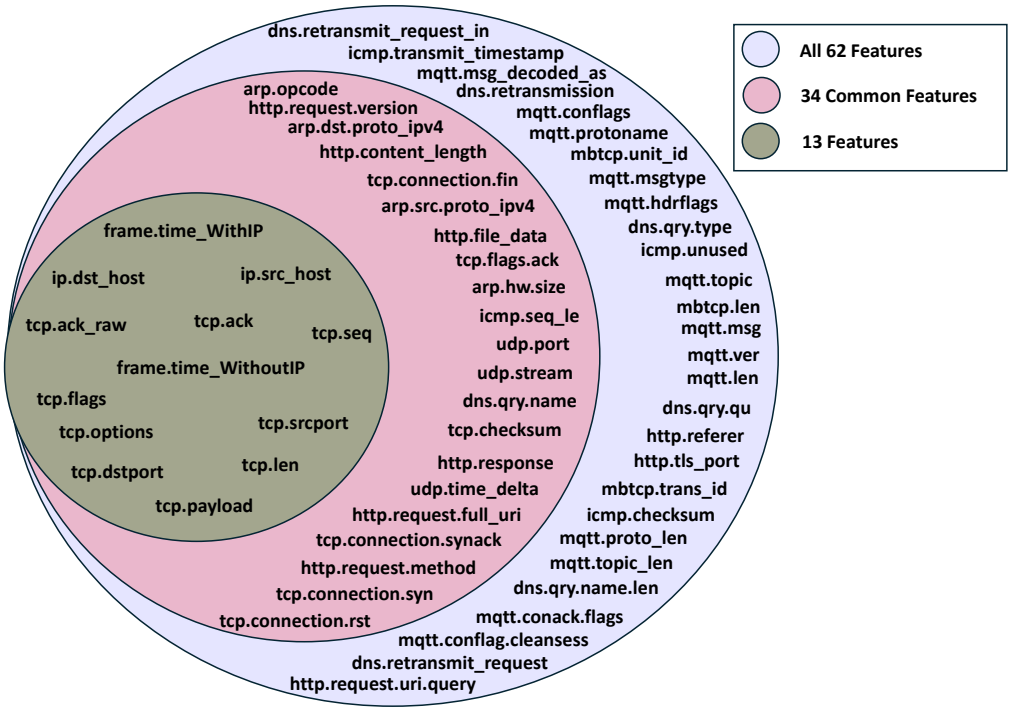


Figure 6.1. Intersection of all features, 34-common features, and 13-features sets.

Table 6.2. Comparison between proposed model and relevant works on EdgeIIoT-set dataset.

Ref.	Models	Multilayer Detection	Features	Feature Selection	Feature Weighting	Hyper-parameter Opt.	Classify
Keserwani, et al. (2023)	CatBoost, XGBoos, RF, DT	No	20	Manual	No	No	Binary, Multiclass
Tareq et al. (2022)	Inception Time, DenseNet	No	All 63	_	No	Yes	Multiclass
Khacha et al. (2022)	CNN-LSTM	No	_	_	No	Yes	Binary, Multiclass
Al Nuaimi et al. (2023)	J48, PART, BayesNets, AdaBoos, LogitBoost, ASC	No	All 63	_	No	Yes	Binary, Multiclass
Samin et al. (2023)	NB, DT	No	46	Manual	No	No	Multiclass
Ullah et al. (2023)	MAGRU	No	31	XGBoost	No	Yes	Multiclass
Ferrag et al. (2022)	RF, DT, SVM, KNN, DNN	No	46	Manual	No	Yes	Binary, Multiclass
The proposed method	NB, DT, KNN, RF, ANN	Yes	13	MI, DTE, IG, RF, Chi^2	Yes	Yes	Binary, Multiclass

6.3 LIMITATIONS AND FUTURE WORK

While this research has successfully developed and validated the SAIDS framework, there are several limitations and opportunities for future work that could enhance its capabilities and applicability. The first limitation of this research is that, although a comprehensive range of multilayer attacks were identified, the research focused on a subset of these attacks when implementing the SAIDS framework. Specifically, the detected attacks include DDoS TCP SYN Flood, DDoS UDP, DDoS HTTP, DoS ICMP, MITM (ARP and DNS Spoofing, ARP poisoning), cryptanalysis (password cracking), SQL injection, and XSS attacks. This means that the remaining multilayer attack types, such as side-channel attacks, replay attacks, and other types of DDoS, DoS, MITM, cryptanalysis, and malicious code injection attacks, were not addressed in the framework.

Future work could expand detection capabilities of SAIDS to study these additional multilayer attacks, further strengthening the framework's ability to comprehensively safeguard IoT networks.

Another limitation of this research is that inference time was not taken into consideration in the SAIDS framework while detecting and identifying attacks. Although the framework demonstrated high accuracy and robustness in multilayer IoT attack detection, the absence of an analysis on inference time leaves a gap in understanding the system's real-time performance. This is particularly critical in IoT environments, where rapid detection and response are essential to prevent damage or further intrusions. Future work could focus on evaluating and optimising the inference time to ensure the SAIDS framework is not only accurate but also efficient for real-time deployment in various IoT scenarios.

Acknowledging the observed limitations results within SAIDS, particularly the underperformance of certain machine learning models like the Naive Bayes model in detecting XSS attacks, it is evident that the framework could benefit from the adoption of advanced machine learning models. A significant extension to this work would be involve integrating Continuous Machine Learning (CML) into SAIDS, following the methods used by (Ariyadasa, Fernando and Fernando, 2024; Seetha et al., 2023). This integration would allow the system to automatically update its understanding of multilayer attack patterns, reducing overfitting risks and improving detection of evolving threats.

Furthermore, another future work direction is to employ existing feature weighting algorithms and develop an algorithm to identify common features without compromising the significant features for each specific attack. By leveraging these two approaches, the framework can refine the detection process more effectively.

Moreover, implementing the SAIDS framework in various environments, such as critical energy infrastructure, smart cities, industrial IoT, and healthcare IoT systems, would provide valuable insights into its scalability and performance under real-world conditions. This could involve collaboration with industry partners to deploy the system in live networks.

REFERENCES

Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S.A. and Khan, M.S. (2021) 'Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set', *EURASIP Journal on Wireless Communications and Networking,* 2021(1), pp. 1–23 Available at: 10.1186/s13638-021-01893-8.

Ahmad, R. and Alsmadi, I. (2021) 'Machine learning approaches to IoT security: A systematic literature review', *Internet of things (Amsterdam. Online),* 14, pp. 100365 Available at: 10.1016/j.iot.2021.100365.

Ahmad, Z., Khan, A.S., Cheah, W.S., Abdullah, J.B. and Ahmad, F. (2021) 'Network intrusion detection system: A systematic study of machine learning and deep learnin.g approaches', *Transactions on Emerging Telecommunications Technologies,* 32(1), pp. e4150–n/a Available at: 10.1002/ett.4150.

Al Nuaimi, T., Al Zaabi, S., Alyilieli, M., AlMaskari, M., Alblooshi, S., Alhabsi, F., Yusof, M.F.B. and Al Badawi, A. (2023) 'A comparative evaluation of intrusion detection systems on the edge-IIoT-2022 dataset', *Intelligent systems with applications,* 20, pp. 200298 Available at: 10.1016/j.iswa.2023.200298.

Al Sukhni, B., Manna, K.S., Dave, M.J. and Zhang, L. (2023) *Exploring Optimal Set of Features in Machine Learning for Improving IoT Multilayer Security.* , Aveiro, Portugal. 2023. IEEE, pp. 1.

Alalhareth, M. and Hong, S. (2023) 'An improved mutual information feature selection technique for intrusion detection systems in the Internet of Medical Things', *Sensors,* 23(10), pp. 4971 Available at: https://doi.org/10.3390/s23104971.

Albulayhi, K., Abu Al-Haija, Q., Alsuhibany, S.A., Jillepalli, A.A., Ashrafuzzaman, M. and Sheldon, F.T. (2022) 'IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method', *Applied sciences,* 12(10), pp. 5015 Available at: 10.3390/app12105015.

Al-Garadi, M.A., Mohamed, A., Al-Ali, A.K., Du, X., Ali, I. and Guizani, M. (2020) 'A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security', *IEEE Communications Surveys & Tutorials,* 22(3), pp. 1646–1685 Available at: 10.1109/COMST.2020.2988293.

AL-Hawawreh, M., Moustafa, N. and Sitnikova, E. (2018) 'Identification of malicious activities in industrial internet of things based on deep learning models', *Journal of information security and applications,* 41, pp. 1–11 Available at: 10.1016/j.jisa.2018.05.002.

Alhowaide, A., Alsmadi, I. and Tang, J. (2021) 'Ensemble Detection Model for IoT IDS', *Internet of things (Amsterdam. Online),* 16, pp. 100435 Available at: 10.1016/j.iot.2021.100435.

Ali, Y.A., Awwad, E., Al-Razgan, M. and Maarouf, A. (2023) 'Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity', *Processes,* 11(2), pp. 349 Available at: 10.3390/pr11020349.

Aljawarneh, S., Aldwairi, M. and Yassein, M.B. (2018) 'Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model', *Journal of computational science,* 25, pp. 152–160 Available at: 10.1016/j.jocs.2017.03.006.

Alotaibi, Y. and Ilyas, M. (2023) 'Ensemble-learning framework for intrusion detection to enhance internet of things' devices security', *Sensors,* 23(12), pp. 20 Available at: https://doi.org/10.3390/s23125568.

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A. and Anwar, A. (2020) 'TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems', *IEEE Access,* 8, pp. 165130–165150 Available at: 10.1109/ACCESS.2020.3022862.

Al-Yaseen, W.L., Othman, Z.A. and Nazri, M.Z.A. (2017) 'Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system', *Expert Systems with Applications,* 67, pp. 296–303 Available at: 10.1016/j.eswa.2016.09.041.

Anthi, E., Williams, L., Slowinska, M., Theodorakopoulos, G. and Burnap, P. (2019) 'A Supervised Intrusion Detection System for Smart Home IoT Devices', *IEEE internet of things journal,* 6(5), pp. 9042–9053 Available at: 10.1109/JIOT.2019.2926365.

Ariyadasa, S., Fernando, S. and Fernando, S. (2024) 'SmartiPhish: a reinforcement learning-based intelligent anti-phishing solution to detect spoofed website attacks', *International journal of information security,* 23(2), pp. 1055–1076 Available at: 10.1007/s10207-023-00778-9.

Atlam, H.F. and Wills, G.B. (2019) 'IoT Security, Privacy, Safety and Ethics', *Digital Twin Technologies and Smart Cities,* , pp. 123–149 Available at: 10.1007/978-3-030-18732-3_8.

Bagaa, M., Taleb, T., Bernabe, J.B. and Skarmeta, A. (2020) 'A Machine Learning Security Framework for Iot Systems', *IEEE access,* 8, pp. 114066–114077 Available at: 10.1109/ACCESS.2020.2996214.

Bansal, D., Sofat, S. and Kumar, P. (2011) 'Distributed cross layer approach for detecting multilayer attacks in wireless multi-hop networks', *2011 IEEE Symposium on Computers & Informatics,* , pp. 692–698 Available at: 10.1109/ISCI.2011.5959000.

Bansal, D. and Sofat, S. (2010) 'Use of cross layer interactions for detecting denial of service attacks in WMN', *2010 14th International Telecommunications Network Strategy and Planning Symposium (NETWORKS),* , pp. 1–6 Available at: 10.1109/NETWKS.2010.5624900.

Beerman, J.&., Berent, D.&., Falter, Z. and Bhunia, S.&. (2023) *A Review of Colonial Pipeline Ransomware Attack.* , Bangalore, India. 2023. IEEE, pp. 8.

Belkacem, S. (2024) *IoT-Botnet Detection Using Deep Learning Techniques.* , Portugal. 2022. Singapore: Springer, pp. 10.

Bolboacă, S.D., Jäntschi, L., Sestraş, A.F., Sestraş, R.E. and Pamfil, D.C. (2011) 'Pearson-Fisher Chi-Square Statistic Revisited', *Information (Basel),* 2(3), pp. 528–545 Available at: 10.3390/info2030528.

Burton, S.D., Tanczer, L.M., Vasudevan, S., Hailes, S. and Carr, M. (2021) *The UK Code of Practice for Consumer IoT Security.* Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/978692/The_UK_code_of_practice_for_consumer_IoT_security_-_PETRAS_UCL_research_report.pdf (Accessed: 8 August 2024).

Butun, I., Osterberg, P. and Song, H. (2020) 'Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures', *IEEE Communications Surveys & Tutorials,* 22(1), pp. 616–644 Available at: 10.1109/COMST.2019.2953364.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C. and Faruki, P. (2019) 'Network intrusion detection for IoT security based on learning techniques', *IEEE Communications Surveys & Tutorials,* 21(3), pp. 2671–2701 Available at: 10.1109/COMST.2019.2896380.

Chen, Y., Sheu, J., Kuo, Y. and Van Cuong, N. (2020) 'Design and Implementation of IoT DDoS Attacks Detection System based on Machine Learning', *2020 European Conference on Networks and Communications (EuCNC),* , pp. 122–127 Available at: 10.1109/EuCNC48522.2020.9200909.

Chkirbene, Z., Eltanbouly, S., Bashendy, M., AlNaimi, N. and Erbad, A. (2020) 'Hybrid Machine Learning for Network Anomaly Intrusion Detection', *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT),* , pp. 163–170 Available at: 10.1109/ICIoT48696.2020.9089575.

Contiki-NG (2022) *Running Contiki-NG in Cooja.* Available at: https://docs.contiki-ng.org/en/develop/doc/tutorials/Running-Contiki-NG-in-Cooja.html (Accessed: 23 August 2024).

Cvitic, I., Perakovic, D., Gupta, B.B. and Choo, K.R. (2022) 'Boosting-Based DDoS Detection in Internet of Things Systems', *IEEE internet of things journal,* 9(3), pp. 2109–2123 Available at: 10.1109/JIOT.2021.3090909.

Deogirikar, J. and Vidhate, A. (2017) 'Security attacks in IoT: A survey', *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC),* , pp. 32–37 Available at: 10.1109/I-SMAC.2017.8058363.

Department for Digital, Culture, Media & Sport (2021) *Cyber Security Breaches Survey 2021.* Available at: https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2021/cyber-security-breaches-survey-2021 (Accessed: 4 July 2024).

Department for Science Innovation and Technology (2024) *New laws to protect consumers from cyber criminals come into force in the UK.* Available at: https://www.gov.uk/government/news/new-laws-to-protect-consumers-from-cyber-criminals-come-into-force-in-the-uk (Accessed: 6 July 2024).

Department of Justice (2021) *Department of Justice Seizes $2.3 Million in Cryptocurrency Paid to the Ransomware Extortionists Darkside.* Available at: https://www.justice.gov/opa/pr/department-justice-seizes-23-million-cryptocurrency-paid-ransomware-extortionists-darkside (Accessed: 6 May 2023).

Dini, P., Elhanashi, A., Begni, A., Saponara, S., Zheng, Q. and Gasmi, K. (2023) 'Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity', *Applied Sciences,* 13(13), pp. 7507 Available at: https://doi.org/10.3390/app13137507.

Doshi, R., Apthorpe, N. and Feamster, N. (2018) *Machine Learning DDoS Detection for Consumer Internet of Things Devices.* , San Francisco, CA, USA. 2028. IEEE, pp. 29.

Egea, S., Rego Mañez, A., Carro, B., Sánchez-Esguevillas, A. and Lloret, J. (2018) 'Intelligent IoT Traffic Classification Using Novel Search Strategy for Fast-Based-Correlation Feature Selection in Industrial Environments', *IEEE internet of things journal,* 5(3), pp. 1616–1624 Available at: 10.1109/JIOT.2017.2787959.

Farea, A.H. and Küçük, K. (2022) 'Detections of IoT Attacks via Machine Learning-Based Approaches with Cooja', *EAI Endorsed Transactions on Internet of Things,* 7(28), pp. 12 Available at: 10.4108/eetiot.v7i28.324.

Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L. and Janicke, H. (2022) 'Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning', *IEEE Access,* 10, pp. 40281–40306 Available at: 10.1109/ACCESS.2022.3165809.

Ferrag, M.A., Maglaras, L., Ahmim, A., Derdour, M. and Janicke, H. (2020) 'RDTIDS: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks', *Future internet,* 12(3), pp. 44 Available at: 10.3390/fi12030044.

François, D., Wertz, V. and Verleysen, M. (2006) *The permutation test for feature selection by mutual information.*

Frazão, I., Abreu, P.H., Cruz, T., Araújo, H. and Simões, P. (2019) 'Denial of service attacks: Detecting the frailties of machine learning algorithms in the classification process', in Luiijf, E.Hämmerli, B. (eds.) *Critical Information Infrastructures Security* Cham: Springer, pp. 230–235.

Gad, A.R., Nashat, A.A. and Barkat, T.M. (2021) 'Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset', *IEEE access,* 9, pp. 142206–142217 Available at: 10.1109/ACCESS.2021.3120626.

Göcs, L. and Johanyák, Z.C. (2023) 'Feature Selection with Weighted Ensemble Ranking for Improved Classification Performance on the CSE-CIC-IDS2018 Dataset', *Computers (Basel),* 12(8), pp. 147 Available at: 10.3390/computers12080147.

Hady, A.A., Ghubaish, A., Salman, T., Unal, D. and Jain, R. (2020) 'Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study', *IEEE access,* 8, pp. 106576–106584 Available at: 10.1109/ACCESS.2020.3000421.

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P. and Sikdar, B. (2019) 'A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures', *IEEE Access,* 7, pp. 82721–82743 Available at: 10.1109/ACCESS.2019.2924045.

Hojlo, J. (2021) *Future of Industry Ecosystems: Shared Data and Insights*. Available at: https://blogs.idc.com/2021/01/06/future-of-industry-ecosystems-shared-data-and-insights/ (Accessed: .

Hurtik, P., Molek, V. and Perfilieva, I. (2020) 'Novel dimensionality reduction approach for unsupervised learning on small datasets', *Pattern recognition,* 103, pp. 107291 Available at: 10.1016/j.patcog.2020.107291.

Hussein, A.Y., Falcarin, P. and Sadiq, A.T. (2022) 'IoT Intrusion Detection Using Modified Random Forest Based on Double Feature Selection Methods', in Liatsis, P., and Hussain, A., Mostafa, S.A., Al-Jumeily, D (eds.) *Emerging Technology Trends in Internet of Things and Computing* Cham: Springer, pp. 61–78.

IBM Security (2022) *X-Force Threat Intelligence Index 2022*. Available at: https://www.ibm.com/downloads/cas/ADLMYLAZ (Accessed: 3 August 2022).

Idrissi, I., Azizi, M. and Moussaoui, O. (2020) *IoT security with Deep Learning-based Intrusion Detection Systems: A systematic literature review*. IEEE, pp. 1.

Illuri, B.a.J., D. (2021) ' Design and implementation of hybrid integration of cognitive learning and chaotic countermeasures for side channel attacks ', *Journal of ambient intelligence and humanized computing,* 12(5), pp. 5427–5441 Available at: 10.1007/s12652-020-02030-x.

Keserwani, K., Aggarwal, A. and Chauhan, A. (2023) *Attack detection in industrial IoT using novel ensemble techniques*. IEEE, pp. 1.

Kethineni, K. and Pradeepini, G. (2024) 'Intrusion detection in internet of things-based smart farming using hybrid deep learning framework', *Cluster Computing,* 27(2), pp. 1719–1732 Available at: https://doi.org/10.1007/s10586-023-04052-4.

Khacha, A., Saadouni, R., Harbi, Y. and Aliouat, Z. (2022) *Hybrid Deep Learning-based Intrusion Detection System for Industrial Internet of Things*. Piscataway: IEEE, pp. 1.

Khan, H.U., Sohail, M. and Nazir, S. (2022) 'Features-based IoT Security Authentication Framework using Statistical Aggregation, Entropy, and MOORA Approaches', *IEEE access,* 10, pp. 1 Available at: 10.1109/ACCESS.2022.3212735.

Khanam, S., Ahmedy, I.B., Idna Idris, M.Y., Jaward, M.H. and Bin Md Sabri, A.Q. (2020) 'A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things', *IEEE access,* 8, pp. 219709–219743 Available at: 10.1109/ACCESS.2020.3037359.

Kumar, R. and Sharma, R. (2022) 'Leveraging blockchain for ensuring trust in IoT: A survey', *Journal of King Saud University. Computer and information sciences,* 34(10), pp. 8599–8622 Available at: 10.1016/j.jksuci.2021.09.004.

Kurniabudi, Stiawan, D., Darmawijoyo, Bin Idris, M.Y., Bamhdi, A.M. and Budiarto, R. (2020) 'CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection', *IEEE access,* 8, pp. 132911–132921 Available at: 10.1109/ACCESS.2020.3009843.

Liang, C., Shanmugam, B., Azam, S., Jonkman, M., Boer, F.D. and Narayansamy, G. (2019) *Intrusion Detection System for Internet of Things based on a Machine Learning approach.* Piscataway: IEEE, pp. 1.

Maghrabi, L. (2024) 'Automated Network Intrusion Detection for Internet of Things: Security Enhancements', *IEEE access,* 12, pp. 30839–30851 Available at: 10.1109/ACCESS.2024.3369237.

Mahale, V.V., Pareek, N.P. and Uttarwar, V.U. (2017) *Alleviation of DDoS attack using advance technique.* IEEE, pp. 172.

Makkar, A., Garg, S., Kumar, N., Hossain, M.S., Ghoneim, A. and Alrashoud, M. (2021) 'An Efficient Spam Detection Technique for IoT Devices Using Machine Learning', *IEEE transactions on industrial informatics,* 17(2), pp. 903–912 Available at: 10.1109/TII.2020.2968927.

Malan, J., Eager, J., Lale-Demoz, E., Cacciaguerra, G., Ranghieri, F. and Brady, M. (2020) *Framing the Nature and Scale of Cyber Security Vulnerabilities within the Current Consumer Internet of Things (IoT) Landscape.* Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/900327/Framing_the_nature_and_scale_of_cyber_security_vulnerabilities_within_the_current_consumer_internet_of_things__IoT__landscape.pdf (Accessed: 16 May 2023).

Malhotra, P., Singh, Y., Anand, P., Bangotra, D.K., Singh, P.K. and Hong, W. (2021) 'Internet of Things: Evolution, Concerns and Security Challenges', *Sensors (Basel, Switzerland),* 21(5), pp. 1809 Available at: 10.3390/s21051809.

Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J. and Aharon, D. (2015) *Unlocking the potential of the Internet of Things.* Available at: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world (Accessed: 16 April 2023).

Mitrokotsa, A., Rieback, M.R. and Tanenbaum, A.S. (2010) 'Classifying RFID attacks and defenses', *Information systems frontiers,* 12(5), pp. 491–505 Available at: 10.1007/s10796-009-9210-z.

Mosenia, A. and Jha, N.K. (2017) 'A Comprehensive Study of Security of Internet-of-Things', *IEEE transactions on emerging topics in computing,* 5(4), pp. 586–602 Available at: 10.1109/TETC.2016.2606384.

Moustafa, N., Turnbull, B. and Choo, K.R. (2019) 'An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things', *IEEE internet of things journal,* 6(3), pp. 4815–4830 Available at: 10.1109/JIOT.2018.2871719.

Mukhtar, N., Fournaris, A.P., Khan, T.M., Dimopoulos, C. and Kong, Y. (2020) 'Improved Hybrid Approach for Side-Channel Analysis using Efficient Convolutional Neural Network and Dimensionality Reduction', *IEEE access,* 8, pp. 1 Available at: 10.1109/ACCESS.2020.3029206.

Mukhtar, N., Mehrabi, M., Kong, Y. and Anjum, A. (2019) 'Machine-Learning-Based Side-Channel Evaluation of Elliptic-Curve Cryptographic FPGA Processor', *Applied sciences,* 9(1), pp. 64 Available at: 10.3390/app9010064.

Mythili, B. and Seetha, R.&. (2021) *Accurate Detection of Multi-layer Packet Dropping Attacks Using Distributed Mobile Agents in MANET.* Honolulu, HI, United States. 2021. IOP Science, pp. 13.

National Cyber Security Centre (2024) *NCSC For Startups: Challenges.* Available at: https://www.ncsc.gov.uk/section/ncsc-for-startups/current-challenges (Accessed: 26 July 2024).

National Cyber Security Centre (2022) *Organisational use of Enterprise Connected Devices.* Available at: https://www.ncsc.gov.uk/report/organisational-use-of-enterprise-connected-devices (Accessed: 6 May 2023).

Nimbalkar, P. and Kshirsagar, D. (2021) 'Feature selection for intrusion detection system in Internet-of-Things (IoT)', *ICT Express,* 7(2), pp. 177–181 Available at: 10.1016/j.icte.2021.04.012.

OWASP (2019) *OWASP IoT Top 10.* Available at: https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT%20Top%2010,%202019 (Accessed: 6 January 2022).

Park, J., Chung, H. and DeFranco, J.F. (2022) 'Multilayered Diagnostics for Smart Cities', *Computer (Long Beach, Calif.),* 55(2), pp. 14–22 Available at: 10.1109/MC.2021.3070325.

Peterson, J.M., Leevy, J.L. and Khoshgoftaar, T.M. (2021) *A Review and Analysis of the Bot-IoT Dataset.* Piscataway: IEEE, pp. 20.

Priya, S.S., Sivaram, M., Yuvaraj, D. and Jayanthiladevi, A. (2020) *Machine Learning based DDOS Detection.* , Pune, India. 2020. IEEE, pp. 234.

Rambabu, K. and Venkatram, N. (2021) 'Ensemble classification using traffic flow metrics to predict distributed denial of service scope in the Internet of Things (IoT) networks', *Computers & electrical engineering,* 96(Part A), pp. 107444 Available at: 10.1016/j.compeleceng.2021.107444.

Rashid, M.M., Khan, S.U., Eusufzai, F., Redwan, M.A., Sabuj, S.R. and Elsharief, M. (2023) 'A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks', *Network (Basel),* 3(1), pp. 158–179 Available at: 10.3390/network3010008.

Ravi, N. and Shalinie, S.M. (2020) 'Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture', *IEEE Internet of Things Journal,* 7(4), pp. 3559–3570 Available at: 10.1109/JIOT.2020.2973176.

Rehman, S.u., Khaliq, M., Imtiaz, S.I., Rasool, A., Shafiq, M., Javed, A.R., Jalil, Z. and Bashir, A.K. (2021) 'DIDDOS: An approach for detection and identification of Distributed Denial of Service (DDoS) cyberattacks using Gated Recurrent Units (GRU)', *Future generation computer systems,* 118, pp. 453–466 Available at: 10.1016/j.future.2021.01.022.

Saharkhizan, M., Azmoodeh, A., Dehghantanha, A., Choo, K.R. and Parizi, R.M. (2020) 'An Ensemble of Deep Recurrent Neural Networks for Detecting IoT Cyber Attacks Using Network Traffic', *IEEE internet of things journal,* 7(9), pp. 8852–8859 Available at: 10.1109/JIOT.2020.2996425.

Salman, O., Elhajj, I.H., Chehab, A. and Kayssi, A. (2022) 'A machine learning based framework for IoT device identification and abnormal traffic detection', *Transactions on Emerging Telecommunications Technologies,* 33(3), pp. e3743 Available at: https://doi.org/10.1002/ett.3743.

Samin, O.B., Algeelani, N.A.A., Bathich, A., Qadus, A. and Amin, A. (2023) 'Malicious Agricultural IoT Traffic Detection and Classification: A Comparative Study of ML Classifiers', *Journal of advances in information technology,* 14(4), pp. 811–820 Available at: 10.12720/jait.14.4.811-820.

Sangodoyin, A.O., Akinsolu, M.O., Pillai, P. and Grout, V. (2021) 'Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning', *IEEE access,* 9, pp. 122495–122508 Available at: 10.1109/ACCESS.2021.3109490.

Sarkar, A., Singh, M.M., Khan, M.Z. and Alhazmi, O.H. (2021) 'Nature-Inspired Gravitational Search-Guided Artificial Neural Key Exchange for IoT Security Enhancement', *IEEE access,* 9, pp. 76780–76795 Available at: 10.1109/ACCESS.2021.3082262.

Sarker, I.H., Kayes, A.S.M., Badsha, S., Alqahtani, H., Watters, P. and Ng, A. (2020) 'Cybersecurity data science: an overview from machine learning perspective', *Journal of Big Data,* 7(1), pp. 1–29 Available at: 10.1186/s40537-020-00318-5.

Seetha, A., Chouhan, S.S., Pilli, E.S. and Raychoudhury, V. (2023) 'D i E vD: Disruptive Event Detection from Dynamic Datastreams using Continual Machine Learning: A Case Study with Twitter', *IEEE transactions on emerging topics in computing,* , pp. 1–12 Available at: 10.1109/TETC.2023.3272973.

Shafiq, M., Tian, Z., Bashir, A.K., Du, X. and Guizani, M. (2020) 'IoT malicious traffic identification using wrapper-based feature selection mechanisms', *Computers & security,* 94, pp. 101863–11 Available at: 10.1016/j.cose.2020.101863.

Sivanathan, A., Gharakheili, H.H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A. and Sivaraman, V. (2019) 'Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics', (8, 18), 1745–1759. 10.1109/TMC.2018.2866249.

Sodagudi, S. and Rao, D.K.R. (2014) 'Behavior based Anomaly detection technique to identify Multilayer attacks', *IJARCSMS, May* .

Su, J., He, S. and Wu, Y. (2022) 'Features selection and prediction for IoT attacks', *High-Confidence Computing,* 2(2), pp. 100047 Available at: 10.1016/j.hcc.2021.100047.

Subramani, S. and Selvi, M. (2023) 'Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks', *Optik (Stuttgart),* 273, pp. 170419 Available at: 10.1016/j.ijleo.2022.170419.

Sujatha, G., Ayyannan, M., Priya, S.G., Arun, V., Arularasan, A.N. and Kumar, M.J. (2023) *Hybrid Optimization Algorithm to Mitigate Phishing URL Attacks In Smart Cities.* pp. 1.

Sulaiman, M.A. and Labadin, J. (2015) *Feature selection based on mutual information.* , Sarawak, Malaysia. 2015. IEEE, pp. 1.

Swathi, G., Shwetha, M., Potluri, P., Murthy Raju, K., Kumar, Y. and Rajchandar, K. (2023) *Smart Cities Hybridized to Prevent Phishing URL Attacks.* IEEE, pp. 817.

Tahsien, S.M., Karimipour, H. and Spachos, P. (2020) 'Machine learning based solutions for security of Internet of Things (IoT): A survey', *Journal of network and computer applications,* 161, pp. 102630 Available at: 10.1016/j.jnca.2020.102630.

Tama, B.A., Comuzzi, M. and Rhee, K. (2019) 'TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System', *IEEE access,* 7, pp. 94497–94507 Available at: 10.1109/ACCESS.2019.2928048.

Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R. and Ghogho, M. (2016) *Deep learning approach for Network Intrusion Detection in Software Defined Networking.* IEEE, pp. 258.

Tareq, I., Elbagoury, B.M., El-Regaily, S. and El-Horbaty, E.M. (2022) 'Analysis of ToN-IoT, UNW-NB15, and Edge-IIoT Datasets Using DL in Cybersecurity for IoT', *Applied sciences,* 12(19), pp. 9572 Available at: 10.3390/app12199572.

Ullah, S., Boulila, W., Koubaa, A. and Ahmad, J. (2023) 'MAGRU-IDS: A Multi-Head Attention-based Gated Recurrent Unit for Intrusion Detection in IIoT Networks', *IEEE access,* 11, pp. 1 Available at: 10.1109/ACCESS.2023.3324657.

Vibhute, A.D., Patil, C.H., Mane, A.V. and Kale, K.V. (2024) 'Towards Detection of Network Anomalies using Machine Learning Algorithms on the NSL-KDD Benchmark Datasets', *Procedia Computer Science,* 233, pp. 960–969 Available at: 10.1016/j.procs.2024.03.285.

Wan, Y.&., Xu, K.&., Xue, G.&. and Wang, F.&. (2020) *IoTArgos: A Multi-Layer Security Monitoring System for Internet-of-Things in Smart Homes.* , Toronto, ON, Canada. 2020. IEEE, pp. 874.

Yang, Y., Wu, L., Yin, G., Li, L. and Zhao, H. (2017) 'A Survey on Security and Privacy Issues in Internet-of-Things', *IEEE Internet of Things Journal,* 4(5), pp. 1250–1258 Available at: 10.1109/JIOT.2017.2694844.

Zainudin, A., Ahakonye, L.A.C., Akter, R., Kim, D. and Lee, J. (2023) 'An Efficient Hybrid-DNN for DDoS Detection and Classification in Software-Defined IIoT Networks', *IEEE internet of things journal,* 10(10), pp. 8491–8504 Available at: 10.1109/JIOT.2022.3196942.

Zainudin, A., Akter, R., Kim, D. and Lee, J. (2023) 'Federated Learning Inspired Low-Complexity Intrusion Detection and Classification Technique for SDN-Based Industrial CPS', *IEEE Transactions on network and service management,* 20(3), pp. 1 Available at: 10.1109/TNSM.2023.3299606.

Zolanvari, M., Teixeira, M.A., Gupta, L., Khan, K.M. and Jain, R. (2019) 'Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things', *IEEE Internet of Things Journal,* 6(4), pp. 6822–6834 Available at: 10.1109/JIOT.2019.2912022.

APPENDIX 1

Summary of literature on detecting multilayer attacks.

Author	Year	Focusing	Dataset	ML Algorithm	Number of features	Accuracy
(Doshi, Apthorpe and Feamster, 2018)	2018	Their model focused on one layer the network layer to identify one attack (DDoS). The five features they've focused on are: packet size, regular time intervals between packets, protocols used, the bandwidth, and limited number of endpoints	Created their own dataset	KNN, SVM, DT, RF, and DNN	5	ACC 99% PRE 99% REC 99% F1 99%
(Bagaa *et al.,* 2020)	2020	Their model focused on one layer the network layer to identify DDoS, Probe, U2R and R2L attack.	built their model based on the NSL-KDD dataset	One-class SVM	41	ACC 99.71%.
(Moustafa, Turnbull and Choo, 2019)	2019	Their model focused on one layer the application layer to identify one attack (botnet attacks)	UNSW-NB15 and NIMS botnet datasets	NB, DT and ANN	36	ACC Between 98.29% and 99.54%
(Cvitic *et al.,* 2022)	2022	Their model focused on one layer the network layer to identify one attack (DDoS).	Created their own dataset and used a dataset from the University of New South Wales	Logistic Model Trees (LMT)	76	Between 99.92%– 99.99%
(Sangodoyin *et al.,* 2021)	2021	Their model focused on one layer the network layer to identify one attack DDoS flooding attacks	Created their own dataset	quadratic discriminant analysis (QDA), NB, KNN, and classification and regression tree (CART)	_	98.00%

(Priya *et al.,* 2020)	2020	Their model focused on one layer the network layer and used two features delta time and packet size to identify one attack (DDoS)	Created their own dataset	NB, KNN, RF	2	98.50%
(Ravi and Shalinie, 2020)	2020	Their model focused on one layer the network layer to identify one attack (DDoS)	UNB-ISCX dataset and created their own dataset	semi-supervised machine learning	155	96.28%
(Rambabu and Venkatram, 2021)	2021	Their model focused on one layer the network layer to identify one attack (DDoS).	DEFCON, ADFA, LBNL, KYOTO, and CICIDS2017	KNN	_	95%
(Shafiq *et al.,* 2020)	2020	Their model focused on several DoS attacks on one layer.	Bot-IoT	NB, RF, DT, Bayes Network, and random tree	44	99.79%
(Nimbalkar and Kshirsagar, 2021)	2021	Their model focused on one layer the network layer to identify DDoS attacks. They presented a feature selection for intrusion detection systems utilizing 50% top-ranking features of the Gain Ratio (GR) and Information Gain (IG)	KDD Cup 1999 and BoT-IoT	JRip	16 and 19	99.99%
(Rehman *et al.,* 2021)	2021	They focused on DDoS attacks. And MinMax scaling is used for feature normalization	CICDDoS2019 dataset	GRU, NB and Sequential Minimal Optimization (SMO)	_	99.69%
(Ferrag *et al.,* 2020)	2020	Their model focused on several attacks (DDoS, DoS slowloris, DoS Slowhttptest, DoS GoldenEye, Heartbleed, Port Scan, Infiltration, and web attacks) on one layer. The first two classifiers of the proposed model work in parallel and feed the third classifier.	CICIDS2017 and BoT-IoT datasets	REP Tree, JRip algorithm and Forest PA	_	96%

(Aljawarneh, Aldwairi and Yassein, 2018)	2018	Four Attacks (DDoS, Probe, R2L and U2R attacks) on one layer (network layer)	NSL-KDD	J48, FR, and Naïve Bayes		For the binary class and multiclass NSL-KDD datasets, their hybrid model achieves accuracy of 99.81% and 98.56%, respectively
(Al-Yaseen, Othman and Nazri, 2017)	2017	Three Attacks (DoS, U2R, and R2L) on one layer (network layer)	KDD CUP 1999	SVM, K-means clustering algorithm, and Extreme learning machine (ELM)		95.75%
(Salman *et al.,* 2022)	2019	Several attacks (including TCP-SYN flooding, UDP flooding, and HTTP flooding), network layer	Created their own dataset and used CICIDS2017 dataset	decision tree, random forest, and deep learning models	39	the Random Forest classifier achieved 94.5% device-type identification accuracy, 93.5% traffic-type classification accuracy, and 97% abnormal traffic detection accuracy
(Chen *et al.,* 2020)	2020	One Attack DDoS on Multilayer	Created their own dataset	DT	_	99.98%
(Anthi *et al.,* 2019)	2019	Focused on five attacks (DoS, MITM, spoofing, reconnaissance, and replay) on one layer (network layer).	Created their own dataset	NB, BN, J48, ZeroR, OneR, SL, SVM, MLP, and RF	_	F-measure between 90% and 98%.
(Saharkhizan *et al.,* 2020)	2020	Focused on four attacks (MITM, Ping DDoS, TCP SYN DoS attacks, and Modbus query flood attacks) on one layer (network layer).	Simoe's dataset [46]	Decision tree with a long short-term module (LSTM)	83	99%

(Gad, Nashat and Barkat, 2021)	2021	Their model focused on one layer the network layer and foced on 9 attacks (such as MITM, DoS, DDoS, ransomware, password attack, scanning, data injection, backdoor, and XSS)	ToN-IoT	Logistic Regression, NB, DT, SVM, KNN, RF, AdaBoost and XGBoost	20	98.60%
(Hady *et al.,* 2020)	2020	One Attack MITM, and one layer (network layer) healthcare system	Created their own dataset	RF, DNN, SVM, and ANN	34	ACCU 92.13%, 92%, 92.44%, and 92.04%
(Mukhtar *et al.,* 2020)	2020	One attack (side channel attack)	created their own datasets	CNN	800	Accuracy 67%
(Illuri, 2021)	2021	One attack (side channel attack)	created their own datasets and the sensitivity analysis was evaluated with medical Image datasets such as MRI, Mammogram, and Diabetic Retinopathy images.	extreme learning machine (ELM)	_	95% accuracy
(Mukhtar *et al.,* 2019)	2018	One attack (side channel attack)	created their own datasets	Naive Bayes, Support Vector Machines, Random Forest, and Multilayer Perceptron (MLP)	_	accuracy of 90%.
(Makkar *et al.,* 2021)	2021	several attacks (spam, DoS, DDoS, eavesdropping, tag modification, and malware) used a variety of input features such asinformation.gain, gain.ratio, and symmetrical.uncertainty	REFIT smart home dataset	Bayesian Generalized Linear Model, Boosted Linear Model, xgboost, Generalized Linear Model, and bagged model	15	accuracy ranged from 79.8% to 91.8%.

(Sarkar *et al.,* 2021)	2021	Several attacks (session hijacking, impersonation, replay, brute force, DDoS, Geometric, MITM, and social engineering)	created their own datasets	ANN	–	–
(Anthi *et al.,* 2019)	2019	Several attacks (DoS, MITM, replay, spoofing and reconnaissance) on one layer (network layer)	created their own datasets	NB, J48, Zero R, Bayesian Network, One R, Logistic Regression (LR), SVM, Multi-layer Perception, and RF	10	F-measure, for j48 ranging from 90% and 98%.
(Zolanvari *et al.,* 2019)	2019	Several attacks (backdoor, command injection, and SQL injection attacks)	created their own datasets	SVM, KNN, NB, RF, DT, LR, and ANN	23	F-measure value for RF classifier is 96.81%.
(Liang *et al.,* 2019)	2019	Their model focused on one layer the network layer to identify DDoS, Probe, U2R and R2L attack.	NSL-KDD	DNN	41	accuracy 98%
(Chkirbene *et al.,* 2020)	2020	Reconnaissance, DOS, wormhole and backdoor.	UNSW-NB15 dataset	DT, classification and regression tree (CART) algorithms	13	accuracy 95.37%
(Tang *et al.,* 2016)	2016	DDoS on one layer (network layer), the authors only employed 6 features from the dataset's 14 features	NSL-KDD dataset	DNN	6	accuracy of 75.75%
(AL-Hawawreh, Moustafa and Sitnikova, 2018)	2018	Serveral attacks (Backdoors, DoS, Reconnaissance, Worms, DDoS, Probe, R2L and U2R attacks) on one layer (network layer)	NSL-KDD and UNSW-NB15 datasets	Deep Auto-Encoder (DAE) and a deep neural network algorithm	6	99%t detection rate and a 1.8% false positive rate.

(Tama, Comuzzi and Rhee, 2019)	2019	Serveral attacks (Backdoors, DoS, Reconnaissance, Worms, DDoS, Probe, R2L and U2R attacks) on one layer (network layer). The authors used 19 features from UNSW-NB15 and 37 features from NSL-KDD	NSL-KDD and UNSW-NB15 datasets	Rotation Forest and bagging algorithms	19 features from UNSW-NB15 and 37 features from NSL-KDD	85.8% accuracy
(Alhowaide, Alsmadi and Tang, 2021)	2021	DDos and zero-day attacks, on one layer (network layer), They also utilised 5-fold cross-validation, which divides the datasets into 80% for training and 20% for testing. And used a total of 23 features.	Bot-IoT, NSL-KDD, UNSW-NB15, and BoTNetIoT	15 different classifiers, including ensemble and traditional classifier	23	accuracy ranged from 93% to 100%.

APPENDIX 2

• HARDWARE TOOLS

The experiments are conducted on a Lenovo laptop with the specifications listed in Table 8.2. These specifications provided sufficient computational power for the tasks involved in this research and are suitable for the dataset scale and the complexity of the machine learning models used.

Lenovo Laptop Specifications.

Processor	: Intel(R) Core (TM) i5-1135G7 @ 2.40GHz, 2.42 GHz
RAM	: 8.00 GB (7.71 GB usable)
Operating System	: Windows 10 Enterprise
Storage	: 236 GB SSD

• SOFTWARE TOOLS

- Python

Python 3 on Google Colab, a cloud-based Jupyter Notebook environment, is chosen as the primary programming platform for this research due to its extensive libraries, ease of use, and strong community support. Google Colab provides a flexible and powerful environment for implementing machine learning algorithms and processing large datasets, with the added advantage of cloud-based GPU acceleration. Several libraries are employed to handle different aspects of the research, such as Pandas, Matplotlib, Scikit-learn, NumPy, Imbalanced-learn, TensorFlow, and Keras. These libraries are used for handling structured data, performing complex mathematical operations, feature selection, model training, evaluation, hyperparameter tuning, addressing class imbalance in the datasets, feature scaling, and data visualisation.

- Weka

A graphical user interface tool for comprehensive machine learning algorithms and data analysis written in Java. It is utilised for implementing information gain (IG) as a feature selection technique. This tool provides a wide range of visualisation tools and algorithms for data analysis and predictive modelling. It enables the preprocessing of datasets, the direct application of machine learning algorithms, and visualising the results without requiring any coding (Witten et al., 2017). Its IG implementation facilitated the efficient evaluation and ranking of features based on their relevance to the target variables.

- Feature Selection Tools

For feature selection, Mutual Information (MI), Decision Tree Entropy (DTE), Principal Component Analysis (PCA), Chi-Square (Chi^2), and Random Forest (RF) methods are implemented, in addition to IG using Weka. These techniques are essential in identifying the most significant features contributing to the detection and identification of multilayer IoT attacks. Each method is evaluated for its effectiveness in identifying the most relevant features, revealing unique strengths in different aspects of feature importance.

- Hyperparameter Tuning

Hyperparameter tuning is a critical aspect of optimizing the performance of the machine learning models, and it is used in this research. Initially, Grid Search is employed, however, due to its time-consuming nature, the approach is shifted to Random Search. Random Search provides a faster alternative by sampling a fixed number of parameter settings from the specified distributions (Ali et al., 2023). This method significantly reduces the computational time while still effectively identifying optimal hyperparameters for the models.

- Excel

Excel is used for data analysis and preprocessing, particularly for converting IP addresses into integers using the Excel Add-in (ip2location-ip-conversion).

APPENDIX 3

Testing accuracy analysis using semi-automated for binary classification.

Top Features	NB	DT	RF	KNN	ANN
1	59.03	91.27	88.1	91.05	60.22
2	72.36	99.86	99.84	99.86	80.11
3	38.45	99.91	99.81	99.96	99.81
4	38.45	94.2	99.86	99.93	93
5	80.56	99.57	99.81	99.94	83.98
6	81.23	99.87	99.83	99.94	91.07
7	43.98	95.2	99.83	99.94	99.32
8	49.61	99.86	93.22	99.94	86.78
9	49.61	95.25	99.83	99.92	86.38
10	49.61	99.85	95.59	99.89	99.5
11	49.64	84.09	94.08	99.9	99.54
12	55.96	98.52	99.76	99.89	88.92
13	56.37	93.85	94.22	99.86	98.45
14	56.37	99.86	95.59	98.12	89.12
15	65.54	99.86	98.51	98.12	99.44
16	71.68	98.82	99.81	98.12	99.51
17	70.28	99.86	94.39	98.14	97.52
18	70.44	93.58	87.24	98.15	99.51
19	49.8	99.86	79.37	98.15	99.55
20	49.82	99.89	93.8	98.14	97.63
21	61.09	99.86	95.63	98.14	97.51
22	58.84	87.27	99.81	98.14	90.79
23	60.89	97.32	83.34	98.07	94.41
24	60.97	84.55	96.04	98.07	99.47
25	60.98	97.28	94.98	98.08	99.24
26	59.31	94.58	94.92	98.08	99.51
27	59.05	91.39	93.65	98.08	92.6
28	59.05	99.89	98.44	98.08	98.51
29	61.25	67.68	99.84	98.08	94.12
30	61.29	97.13	86.01	98.08	99.5
31	61.28	99.55	99.83	97.96	89.52
32	61.28	91.82	95.55	97.89	88.81
33	61.24	98.55	99.84	97.89	89.02
34	61.18	99.87	95.93	97.89	98.07
62	66.77	94.3	94.58	98.4	89.44

Testing accuracy analysis using semi-automated for multiclass classification.

Top Features	NB	DT	RF	KNN	ANN
1	28.68	53.35	41.66	52.1	36.49
2	53.47	67.78	49.6	77.12	41.86
3	64.1	77.9	74.39	92.22	41.44
4	55.47	81.16	69.53	94.95	70.78
5	59.55	78.64	60	95.31	57.6
6	60.85	68.25	64.55	95.49	56.3
7	62.19	71.49	61.88	95.5	45.07
8	63.82	95.43	82.62	95.98	62.77
9	58.29	88.85	53.54	96.12	63.41
10	60.64	87.9	64.29	89	61.06
11	59.94	94.5	61.34	89.63	63.61
12	61.14	75.99	64.84	89.66	59.92
13	63.47	74.31	84.44	90.39	69.28
14	63.53	79.51	67.36	85.36	67.55
15	63.9	70.67	64	85.36	63.82
16	63.91	81.83	75.75	85.38	51.72
17	62.31	84.33	68.52	85.38	63.39
18	62.31	76.21	54.71	85.37	42.12
19	62.31	66.15	83.02	85.37	64.3
20	62.49	69.57	66.05	85.37	45.58
21	62.49	79.7	86.15	85.36	65.51
22	62.31	66.15	83.02	85.37	59.46
23	64.11	61.53	59.2	86.8	70.17
24	63.54	66.51	53.76	86.79	64.47
25	63.53	74.48	70.93	86.81	39.5
26	63.53	64.71	56.68	87.2	66.7
27	63.53	83.36	59.06	87.2	63.52
28	63.53	80.76	62.72	87.24	63.6
29	63.53	80.13	69.23	87.24	69.62
30	63.53	73.41	72.64	87.27	68.64
31	63.53	82.49	70.87	86.88	58.96
32	63.53	79.17	70.75	86.84	58.89
33	64.25	88.14	69.16	87.32	65.99
34	64.64	76.49	67.66	87.31	62.35
62	67.85	70.23	76.13	87.58	72.65