REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

ADAPTIVE DECISION TREE WITH RANDOM FOREST INTEGRATION AND DIMENSIONALITY REDUCTION FOR EFFICIENT BOTNET FORENSICS

Ahmed Najm Obaid ALZUBAIDI

Master's Thesis

Supervisor

Asst. Prof. Dr. Abdullahi Abdu İBRAHIM

İstanbul, 2025

ADAPTIVE DECISION TREE WITH RANDOM FOREST INTEGRATION AND DIMENSIONALITY REDUCTION FOR EFFICIENT BOTNET FORENSICS

Ahmed Najm Obaid ALZUBAIDI

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2025

The thesis titled ADAPTIVE DECISION TREE WITH RANDOM FOREST INTEGRATION AND DIMENSIONALITY REDUCTION FOR EFFICIENT BOTNET FORENSICS prepared by AHMED NAJM OBAID ALZUBAIDI and submitted on 12/05/2025 has been **accepted unanimously** for the degree of Master of Science Information Technologies.

Prof. Dr. Abdullahi Abdu İBRAHIM

Supervisor

Thesis Defense Committee Members:

Assist. Prof. Dr. Abdullahi Abdu İBRAHIM	Department of Computer Engineering, Altınbaş University
Asst. Prof. Dr. Yaser ALAIWI	Department of Mechanical Engineering, Altınbaş University
Asst. Prof. Dr. Tariq Abed MOHAMMED	Department of Computer Science and Information, Kirkuk University

I hereby declare that this thesis meets all format and submission requirements of a master's thesis.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Ahmed Najm Obaid ALZUBAIDI

Signature

DEDICATION

I would like to begin by thanking God for granting me the opportunity, abilities, and knowledge to pursue my Master's degree. I would also like to extend my sincere gratitude to my parents and siblings, and all the success I have achieved thanks to them and their love. I would also like to thank my supervisor Asst. Prof. Dr. Abdullahi Abdu İBRAHIM, who guided and helped me complete my research and academic journey. And thank you to all my friends.

ABSTRACT

ADAPTIVE DECISION TREE WITH RANDOM FOREST INTEGRATION AND DIMENSIONALITY REDUCTION FOR EFFICIENT BOTNET FORENSICS

ALZUBAIDI, Ahmed Najm Obaid

M.Sc., Information Technologies, Altınbaş University

Supervisor: Asst. Prof. Dr. Abdullahi Abdu İBRAHIM

Date: 05/2025

Pages: 94

ABSTRACT- Adaptive Decision Tree with Random Forest Integration and Dimensionality Reduction for Efficient Botnet Forensics proposes a new method for botnet detection by combining an adaptive decision tree with random forest integration. As response variables are highly dimensioned in botnet detection, multistep and time-consuming detection processes are major challenges. With an integrated method, we could first model the response variables as multiclass and regression formats to simplify the construction of decision trees. Then, based on the adaptive decision tree feature selection, we filter out non-obvious features to efficiently establish the random forest regression model under the appropriately sized feature space. Furthermore, to handle multilabel classification, a random forest is employed as the global model in Tree-2-Rule procedures to detect botnet-affected communication behaviors. Finally, real data experiments have been conducted based on the top datasets. The results show that the adaptive decision tree has excellent improvements in efficiency and accuracy. In future research, the GPU ninth-ordinal censored multistate diagnosis data is useful observed materials that do not destroy the random effect influence of the data. Also, whether the application of the random forest model could save more time in analyzing existing commercial status is an issue to be clarified in future development.

Additionally, the development of the method proposed in this research requires further investigation. We could improve and then propose the preferable solution based on the research results. Our proposed solution can be utilized as a tool for efficient real-time bot incident investigations, in accordance with both academic and business objectives.

Keywords: DDOS, BOTNETS, IOT, RF, DT, ML.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
FSK	:	Frequency Shift Keying
IoT	:	Internet of Things
PAN	:	Personal Area Network
OFDM	:	Orthogonal Frequency Division Multiplexing
TDMA	:	Time Division Multiple Access

LIST OF SYMBOLS

μ : Electron Mobility

λ : Wavelength

ω : Angular Frequency

1. INTRODUCTION

1.1 BACKGROUND

Botnet attacks have been major network security threats and have been supported by a wide range of malware. Many existing botnet detection methods are based on various machine learning algorithms, which have been proven to be advantageous [1]. However, with the diversity of botnets, new botnets appear at an increasing speed; the performance of these methods is declining. Our work is based on the observation that botnets have very particular characteristics. They have a complete communication ecosystem, with a set of botmasters, which send commands to the bots and potentially a control server linking both botmasters and handling traffic anonymization [2]. These are managed by the botnet owner to coordinate bot activities. Generally, botnets' covert features are not designed for long durations, so botnets' fingerprints are not that difficult to discover. Our main idea is to build decision trees for botnet forensics by integrating random forests and dimensionality reduction, which are then arranged and scheduled in a highly adaptive way that also automatically filters the noisy labels [3].

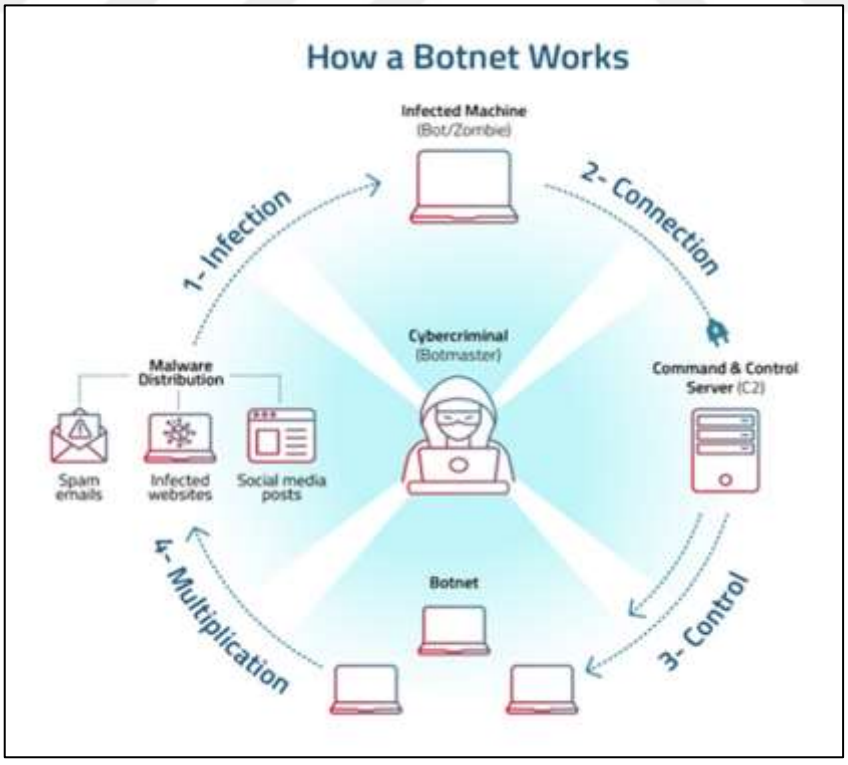

Figure 1.1: Botnet Life Cycle [4].

The system learns unsupervised botnet data and creates supervised labels based on the inference that the communication parameters might exhibit abnormal activities. With hyper-parameter tuning, our model robustly handles various botnets and is able to solve unrelated problems. After the learning step, newcomers' labels are predictable with high performance scores. Botnets remain one of the most malicious and predominant security threats posed by criminal underground networks due to their potential to launch powerful large-scale cyberattacks. Botnets are a perfect strategy for committing various crimes easily, including identity theft, online fraud, blackmail, espionage, and more income-generating activities [5]. Many challenges and issues must be taken into consideration regarding botnet design, operation, and detection. For example, it is necessary to distinguish botnets from traditional network flaws and speed up the detection process. Additionally, to increase botnet efficiency, botmasters constantly seek to infect devices more effectively, with success rates above even 90%. On the other hand, to keep botnet communication stealthy, it is important to make detection difficult. There is no doubt that advanced defense mechanisms should be developed to handle a vast volume of benign transactions [6]. Significant related work in botnet detection and analysis has been done with the help of machine learning. Some previous studies mostly utilized traditional classifiers for botnet detection models. Decision Trees and their variants have been used to detect and understand botnets. Entropy-constructed classification trees showed high detection accuracy and found that the features in the original order would gradually decrease in entropy values. Additionally, detecting maximum entropy features also showed good performance in some botnet traffic flows. They could be influenced by the infected control protocols, actions, and attack traffic. In addition, the Botnet Behavior Concept Map provided very good results in identifying important botnet-favorable activities. The concept map is able to find frequent patterns using a hash table-based algorithm [7].

1.2 PROBLEM STATEMENT

The rapid evolution of botnets is one of the most alarming dangers to digital security due to its increasing sophistication as well as far-reaching consequences. Botnets, which are networks of infected machines controlled by an attacker or attackers, are frequently utilized in large-scale cyber-attacks including but not limited to DDoS attacks, data theft, or even the damage of information systems. The varying complexity of most botnet architectures makes

their detection and forensic analysis extremely difficult therefore more sophisticated techniques need to be developed for the effective identification and analysis of botnet activities. Most traditional techniques of botnet detection have been signature-based or anomaly-based and these techniques do not possess the ability to scale or adapt because the strategies of botnets are constantly evolving. Addressing that issue, there is growing reliance on machine learning algorithms, especially those concentrating on pattern recognition and data synthesis, for the automated detection and forensic analysis of botnets. Random Forests and Adaptive Decision Trees, for instance, are ensemble techniques that have been shown to achieve higher classification accuracy with extensive datasets and complex datasets [8]. My research aims to improve botnet forensic analysis by integrating adaptive Decision Trees with Random Forests, along with dimension reduction techniques to enhance the speed of the analysis without sacrificing accuracy. Besides improving the algorithms' computational efficiency by minimizing the feature space, noise is also minimized so that the algorithms can work with the most relevant data attributes. This combination is believed to provide strong adaptable detection capabilities suitable for different botnet architectures. This thesis aims to address a gap in botnet forensics through an enhanced detection framework that utilizes machine learning to improve response and accuracy metrics. With feature selection optimization and ensemble learning implementation, the proposed method achieves enhanced reliability and scalability for botnet detection, which effectively responds to significant cybersecurity challenges and improves investigative processes within forensics.

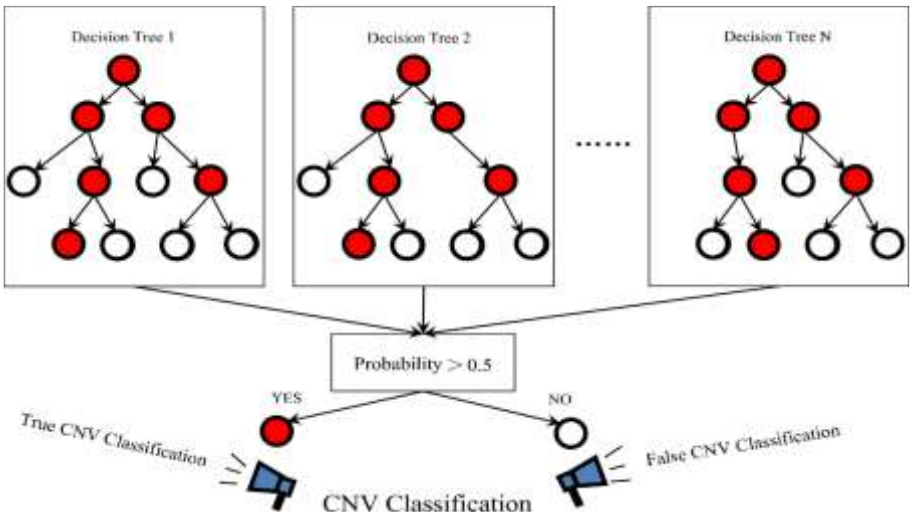

Figure 1.2: Adaptive Decision Trees With Random Forests.

The growing sophistication of botnet-enabled cyber-attacks poses new challenges for cybersecurity experts in terms of detecting and analyzing such threats. The botnet signature detection method face immense challenges because of how rapidly and intricately botnets evolve. Such legacy approaches tend to use more unsophisticated methods for detection, lacking in versatility and combining multiple dimensions, such as techniques which scalable and adaptive to modern botnet frameworks with advanced subversion techniques that constantly morph to take advantage of pathways throughout utilizing digital networks sophisticated architectures.

Table 1.1: Evaluation of Botnet Detection Methods.

Method	Accuracy	Scalability	Real-Time Capability	Resource Efficiency	Adaptability	Overall Rating
Static Analysis	7	6	4	8	5	6
Dynamic Analysis	8	7	6	6	7	6.8
Honeypots	8	5	4	5	6	5.6
Machine Learning	9	8	7	7	8	7.8
Deep Learning	10	7	8	6	9	8
Blockchain-Based	7	6	5	5	7	6
Software Defined Networking (SDN)	9	8	8	7	9	8.2
Hybrid Models (e.g., AI + Honeypots)	9	7	7	6	9	7.6
SIEM-Based Solutions	8	7	7	6	8	7.2

The expanding sophistication of botnets presents increasing risks to both individual users and organizational infrastructures, resulting in detrimental impacts such as data breaches, theft, system infiltration, and severe monetary damages. Therefore, the need for automated systems capable of botnet detection and subsequent forensic analysis more accurately and swiftly has intensified. While existing techniques from machine learning have potential, they are often bound by the complexities associated with high dimensionality data and noise which affect the detection rate and maximize required time metrics to yield reliable results.

This study aims to meet the specified objectives by developing a new framework consisting of adaptive Decision Trees incorporated with Random Forest and reduction in dimension strategies for data representation. This framework seeks to improve efficiency, accuracy in detection, and speed by adopting an ensemble-based strategy which lowers the dimensionality of the data sets in question. With these improvements, more reliable and rapid botnet forensic analysis will be made possible. The developed solution is intended to be a powerful, scalable, and adaptable device that counters augmenting botnet activities, thus improving the efficacy of cybersecurity measures employed against botnet threats.

1.3 THESIS AIMD OBJECTIVES AND RESEARCH QUESTIONS

1.3.1 Thesis Aims

To create an adaptive and effective botnet detection and forensic analysis framework based on the usage of Decision Tree and Random Forest algorithms that would incorporate dimensionality reduction to increase detection accuracy, efficiency, and adaptability to evolving botnet threats.

1.3.2 Thesis Objectives

a. Explore The Problems Associated With Botnet Detection and Review Literature: Evaluate previously published works. Guide worked through practical shortcomings of traditional botnet detection systems that stem from complications introduced by advanced botnet structures, high-dimensional datasets, and the requirement for scaling and adaptability.
b. Design a Framework for Botnet Detection and Forensics: Develop a novel framework that combines adaptive Decision Tree algorithms with Random Forest integration to improve botnet detection accuracy and reliability, particularly in handling complex data patterns.
c. Implement Dimensionality Reduction Techniques: Incorporate dimensionality reduction methods, such as Principal Component Analysis (PCA) or feature selection algorithms, to reduce data complexity, improve computational efficiency, and eliminate irrelevant features for a more focused analysis.

d. Optimize the Ensemble Model for Improved Accuracy and Efficiency: Fine-tune the parameters of the integrated Decision Tree and Random Forest model to achieve optimal detection performance, balancing computational load and accuracy.

1.3.3 Thesis Questions

a. What are the current limitations of traditional botnet detection and forensic methods, and how do they impact effectiveness against evolving botnet threats?
b. How can the integration of adaptive Decision Tree and Random Forest algorithms improve the accuracy and reliability of botnet detection?
c. What role does dimensionality reduction play in enhancing computational efficiency and focusing detection on relevant features within botnet datasets?
d. How does the proposed framework compare to existing detection techniques in terms of accuracy, speed, and adaptability to changing botnet patterns?
e. What are the optimal parameter settings for the Decision Tree and Random Forest algorithms within this framework to balance detection performance and computational efficiency?
f. How resilient is the proposed model against emerging and sophisticated botnet evasion techniques, and can it effectively adapt to new botnet behaviors?
g. What practical considerations need to be addressed to implement this detection framework in real-world cybersecurity environments, ensuring scalability and usability?

1.4 THESIS CONTRIBUTIONS

The main research contributions of the proposed framework include minimizing and eliminating the feature selection process executed on complex features in intrusion detection datasets and refining binary classification performances. The proposed adaptive feature selection process minimizes the search space and reduces classification performance time complexity. Another objective of the proposed framework is to refine decision rules and address binary classification problems in botnet forensics to support cyber law investigations when investigating alternative solutions with a nature-inspired metaheuristic optimization model. Finally, by checking the out-of-bag error rate of the new binary classifier, the proposed framework is integrated into the random forest classifier. Botnet detection error rate performances and time complexities were strengthened by solving the problem in

ensemble learners, which are remnants of random decision trees. The performance measurement of the error rate is achieved in reduced time compared to ensemble learners. Furthermore, the complexity of the classification process is reduced due to the shorter calculation time. The most important limitation of the methodology is that its goal is to identify botnet traffic, not the topology and the frequencies of the infections. Additionally, the evaluation of the methodology is performed with labeled botnet traffic, which is fair since a security analyst would have the capability to distinguish botnet traffic even from the edge devices of an organization. However, detecting botnet traffic from such an edge device is a valuable approach towards rapid botnet forensics, particularly with the increasing frequency, unpredictability, severity, and challenges of botnet activities against state-of-the-art intrusion detection and prevention mechanisms. The proposed methodology can be directly applied as a rapid incident response tool since it requires no training and no feature extraction phase to be initiated, but it can also be integrated within traditional machine learning methodologies as a feature extraction filtering mechanism. There are several important limitations of the methodology ranging from the specific network characteristics that it can address to the overheads that are incurred when the workload is increased. The proposed technique is suited for monitoring long-duration botnet activities since it has been trained to provide a steady stream of features. Therefore, anomalous, short, and unpredictable botnet activities will not be detected without temporal integration or correlation with other attack activities. The classifier does not support incremental learning, and it introduces a degree of sluggishness when the flow duration is shorter than the training window sizes. Moreover, the classifier may become decoupled from the botnet flows when the network activity has changed significantly. In such cases, the classifier should be retrained as the dataset of normal and botnet samples is updated. Since the main scope of the study is to keep the memory and the reference tables small and manageable, future work needs to investigate the temporal integration mechanisms and the potential large expansion of reference tables. The classifier is responsive to the characteristics and configurations of the network, although his cap is not exhaustively described, and additional inquiry might yield deeper insights into his performance. As said before, statistical sampling for estimating classifier performance is, without a doubt, a source for uncertainty, thus evaluation and regular assessment is always required.

2. LITERATURE REVIEW

2.1 INTRODUCTION

Focusing on DDoS attacks, this chapter outlines the specifics of IoT botnets and their mechanisms, along with methods of detection and mitigation. It considers different structural classes of botnets and their implications for attack strategies as well as defensive responder actions. The approaches to detection can be considered either host-based or network-based, the latter of which includes the application of machine learning, deep learning, and software defined networking. Also reviewed are advances in hybrid detection systems that integrate ML with honeypots, blockchain, and SDN, enhancing detection accuracy while optimizing resource use. Finally, we analyze case studies on the global impact of cybersecurity due to the dominant forms of IoT botnets like Mirai.

2.2 RELATED WORKS

There is a significant amount of discussion in the academic literature on assaults that include the Internet of Things (IoT) and distributed denial of service (DDoS). As a consequence of this, a substantial quantity of research is being conducted in this industry, and a wide range of topics and focusses have been examined as a result of this. The papers that are listed in Table 1. include a complete overview of relevant works that investigate various facets of this research subject. This review may be found in the publications that are mentioned in Table1. Over the course of the past few years, Thanh et al. [13] have done one of the most complete literature evaluations that they have ever carried out in connection to this specific topic matter. The study conducts a comprehensive evaluation of the existing body of literature, and due to the fact that it has 234 references, it has also conducted a comprehensive analysis of the research community. In addition, a comprehensive review of the relevant literature has been carried out as part of the survey. It focusses on botnets from a variety of viewpoints, including the design and assessment of botnet assaults, and it also provides detection strategies for attacks that correspond to botnets. Furthermore, it provides protection against botnets. The term "botnet" refers to a certain kind of computer network. During the course of their inquiry of botnets that were discovered on the Internet of Things, Stephens et al. [12] carried out comparative research investigations. The literature study is well-structured and provides a comprehensive analysis of the research that is now being conducted on the

detection and mitigation of botnets that are related with the Internet of Things (2015–2020). This review is a comprehensive evaluation of the research that is presently being done. Every research endeavour must include a comparative research study that is well-designed and involves both qualitative and quantitative comparisons. This is an essential component of every research endeavour. Because this research has included the process of establishing hazards and detecting processes, there are still concerns that need to be answered about both study areas. This is because the research has included both of these aspects. In this study, Vishwakarma et al. [9] analyse the security threats that are associated with networks that are connected to the Internet of Things. More specifically, they concentrate on distributed denial of service attacks that are related with this business. Furthermore, the publication provides an explanation of attacks as well as the consequences that are related with them, in addition to data that gives an indication of how assaults are judged. It is possible to take note of this particular aspect. Unfortunately, these media are unable to pay a considerable amount of attention to distributed denial of service assaults that are carried out via botnets. Insufficient attention is being paid by them. On the other hand, Vishwakarma et al. focused their attention on distributed denial of service attacks made in botnets and provided researchers with a full overview of the subject matter. An in-depth discussion of botnets that operate on the Internet of Things and the attack tactics that they use can be found in both Feily et al. [5] and Silva et al. [6]. A synopsis of the subject matter is provided by both of these writers in relation to the topic matter. A limited number of detection methods are provided in these papers, along with a few additional early studies such as [7]-[8]. Furthermore, these publications do not define detection strategies as either host-based or network-based solutions. Rather than giving taxonomies as their major focus, they concentrate their attention primarily on specific detection methods. Furthermore, the evolutions that have taken place in the field of machine learning solutions have had an effect on botnet detection systems that are used for the internet of things. There is a growing number of research projects being carried out in this area, and solutions that are founded on machine learning are being developed. [9] is the first article to investigate approaches for identifying botnets that are based on machine learning and deep learning. This publication is among the literature review publications that are pertinent to this issue. After then, machine learning and deep learning solutions developed as the many kinds of detection techniques that were the most prominent over the course of time. Deep learning is the subject of the great majority of the most recent publications that have been

published in the academic literature. This is a direct consequence of the fact that. A number of literature reviews, such as [17]-[21], have been conducted over the course of the past several years, and the exclusive focus of these reviews has been on detection systems that are based on deep learning. During the time when all of this was taking place, a variety of different techniques for detecting botnet distributed denial of service assaults were being developed. In addition, solutions that are based on blockchain technology [18]-[22] and software-defined networking [24,25] have begun to exhibit indications of gaining popularity, which has led to an increased emphasis on research into this specific subject. [23]- [25].

2.2.1 Block Chain in IOT

A number of studies are being conducted to investigate the possibility of utilising blockchain technology as the basis for a variety of detection approaches. According to Shah et al. [18], this study is the first literature review that focusses on distributed denial of service attacks in Internet of Things environments that make use of blockchain technology. This research was conducted in order to investigate these assaults. Specifically, this research was carried out with the intention of investigating these attacks. Additionally, a number of studies that have been carried out on literature reviews [13]-[17] give a search methodology that is clearly defined, which adds to a more systematic approach throughout the examination of the literature. The finding adds value to the overall analysis of the literature. With the aim of achieving these parts' goals, it is to provide other researchers with the road maps to accurately navigate the scientific literature. To formulate effective defenses against the escalating cyberthreats in a networked world, it is crucial to comprehend the intricacies of IoT botnet designs. These algorithms [28,29] can be divided into four specific types of topology: star topology, hierarchical topology, multiple-server topology, and random server topology. Depending on their design, botnets may be categorized into these four distinct kinds. Botnets can also be classed according to their functionality. The most popular type of botnet, which spreads infections the fastest, is the centralized botnet, also called star topology. This is the type of botnet that is most popular and, as shown in Figure 1, is the most prevalent type of botnet. If a bot master issues a C&C command, the server will issue a command for each bot to start an attack simultaneously. The attack will commence immediately using the bots' pre-programmed instructions set forth by the bot master. The Internet Service Provider (ISP) or any other researcher willing to take down a botnet may

find an opportunity when attempting to find the control and command server, which is the essence of this structure - the backbone of this design. However, this is possible. So, being able to destroy the botnet becomes possible. In case the connection between the control and command servers is cut for any reason, the bots will most likely not hear any commands from the bot master. This, on the other hand, hypothetically makes it impossible for the attack to be conducted [30]. There is a difference in how many control and command servers exist in various server topologies compared to the star topology. This arises because the star topology is a configuration that is more sophisticated. The various server topologies that are in use modify the settings of the control-and-command servers. This is done because there is a possibility that with relatively little effort low-level problems may occur. It has been guaranteed that all of the control-and-command servers powered on into the network have properly configured to be able to correspond to the demands. In the case it is found that one of the servers is faulty and does not operate as intended due to the problem, a substitute server will be placed instead. As a consequence of this, it will ensure that the botnet continues to operate in the manner that it was designed to operate. It has been determined, on the basis of the information that was provided by the bot master [30], that the assault will continue for as long as one of the command-and-control servers is operational.

2.2.2 Command and Control

The building of a significant number of servers results in the emergence of a number of problems that need to be resolved first. When compared to the development of a botnet that utilises a star topology, the bot master believes the creation of a botnet that makes use of several servers to be more challenging. This is because of the intricacy of the design. This is due to the fact that the botnet is composed of a number of bots that are designed to provide a high level of functionality. By utilising high-level bots as command and control servers, it is feasible to conceal the bot master and command and control server for the purpose of concealing their identities. The purpose of this action is to accomplish the outcome that is sought. In the event that the bot master constructs the botnet utilising a hierarchical design, it is difficult to eradicate the botnet [30]. This is due to the fact that the command and control server provides protection, which makes it harder to eliminate the botnet. Despite the fact that the random botnet does not have a command-and-control server, the botnet's whole bot population will only be decimated by a small percentage of the overall bot population being

destroyed in the event that the high-level bot is detected. A bot will convey its commands to other bots that are connected to it and are linked to it in the event that it gets instructions from the bot master. These instructions will be sent to the bots. A random botnet provides a high level of security since each bot is understood as a command and control server [31]. Despite the fact that it is incredibly difficult to set up a random botnet, it gives excellent protection. The fact that it is extremely difficult to create a random botnet does not change the fact that this is the case. One of the most significant issues that surfaces when dealing with a centralised botnet is the identification and subsequent downing of the command and control servers. This is one of the most significant challenges that occurs. Identifying the centralised command and control server is particularly challenging due to the fact that every single bot that is a member of the P2P botnet serves as a command and control server. This makes it incredibly difficult to determine which bot is the root of the problem. For this reason, even if one of the bots that comprise the architecture of a random topology botnet is found, the effects of that bot are limited, and it is not possible for a single bot to bring the entire network to a halt even if it is detected [32]. A command-and-control (CNC) server, a MySQL database server, a Scan Receiver, a loading server (also known as a Loader), and a DNS server are the components that make up a typical Mirai botnet, according to the information that is supplied in the source code for Mirai [33]. Additional components include a DNS server. According to Figure 2, an adversary can carry out a distributed denial of service attack (DDoS) by sending a particular command to the CNC server using Telnet from a remote terminal. This allows the adversary to perform the assault. The attacker is able to target the CNC server as a result of this. At this point in time, we are situated at the beginning stages of the process. While the step b configuration procedure is being carried out, the instructions are simultaneously logged on the server that is hosting the MySQL database. The initial part of the attack involves the victim being directed to bots or Internet of Things devices that have been infiltrated through the use of hacking tools. A deluge of network packets, which are being transferred by robots that are now operating, are delivered to the server that is supposed to be the victim. The initial step in the procedure is for these bots to respond appropriately to the CNC order that has been given to them. In the past, the expansion of the Internet of Things (IoT) has always been accompanied by broad weaknesses, and it has always attracted organisations that are harmful. The situation has been like this throughout the course of its history. Internet Relay Chat (IRC), which was

first introduced in the late 1990s [34], is credited with being the originator of the concept of botnets. IRC is widely acknowledged as the company that pioneered the concept of internet relay chat. IRC channels, which stand for Internet Relay Chat, were utilised by cybercriminals as a means of constructing botnet networks. These botnet networks were made up of computers that had been compromised. Most of the bots available were mainly used in illicit activities like spam and Distributed Denial of Service (DDoS) attacks. Botnet development in the past was less sophisticated than today. It required simple commands and scripts. In the past, botnets tried their best not to be noticed by their systems' authorities, which enabled them to escape detection. This form of evasion for reporting was common back then, but not today. Attempting to cloak the detection system was accomplish in an attempt to conceal these activities unnoticed. Detecting techniques have become so wide-spread that with every passing day neural networks burgeon smarter. Based on its description, honeypots can be categorized as botnet bait; most people involved in these types of crimes today will refer to them as miscues of botnets. Avoiding detection, while deceiving those who seek the truth is accessible to honeypots. Their purpose is of providing botnets with accessible targets on which to unleash their wrath. As the primary function of honeypots, they provide botnets with an easy target to attack. The inception of Trojan horses and worms marked a new era for bots during the first decade of the twenty-first century. This shift began in the early 2000s. Code Red and Slammer worms were capable of identifying and taking advantage of Operating Systems' weaknesses. The spreading of these worms enables them to infect and automatically connect computers to botnets without human intervention. Trojans able to deceive users into installing deadly software without their knowledge greatly enhanced the capabilities of botnets [35]. With the capability of installing deadly software on a person's computer, Trojan horses Trojan horses advanced the capabilities of malware. Zombie networks, otherwise known as botnets, reached their peak during the first half of the 21st century. 35 and 36 For the purpose of remote control of a networked swarm of captured machines, a central command and control unit was deployed; more commonly known as a C&C server. The goal was remote control of the networked swarm of captured machines. Cybersecurity professionals trying to detect and destroy botnets are counteracted by hackers employing two more sophisticated methods: encryption and peer-to-peer networking. Both methods aim to protect the professionals trying to eliminate botnets. Their protective measures are solely directed towards the botnets these

professionals have constructed. A brilliant illustration of this generation of botnets, which are becoming harder to understand and easier to miss, is the notorious Storm Worm, first detected in 2007 and still wreaking havoc today. [37]. Botnets are designed to target sensitive data such as login passwords and bank information. This is because the major motivation for cybercrime has shifted from simple mischief to financial gain. This has led to the development of botnets. In order to steal information from computers that are linked to the internet, botnets are developed to achieve this goal. Fraudsters were able to carry out a significant amount of fraudulent activity that involved online banking [38]. This was made possible by the widespread availability of banking Trojans such as Zeus and SpyEye. It is clear that hackers utilised a more sophisticated and business-oriented technique, as evidenced by the fact that a significant proportion of botnets were tailored to achieve certain objectives.

2.2.3 Mirai Botnet

By using Internet of Things devices that had been put into a vulnerable condition, the Mirai botnet garnered a large amount of attention in 2016. This was due to the fact that it carried out distributed denial of service assaults that had never been seen before. Over the course of its existence, the botnet has seen a substantial degree of evolution, which will be illustrated in Figure 3 in the following paragraphs. Mirai was successful in revealing the vulnerabilities in security that were established as a result of the exponential growth in the number of devices that are connected to the internet [39]. This was done by taking advantage of the fact that devices connected to the Internet of Things have passwords that are either not present at all or are assigned by default. The incidence brought to light the necessity of establishing more stringent security measures for the Internet of Things (IoT), as well as increased awareness about the dangers of utilising smart devices that have been compromised after they have been compromised. Between August 2016 and February 2017, the Mirai botnet was responsible for terrorizing more than 600,000 agents, mainly consisting of IoT devices [39]. More specifically, the botnet infected numerous consumer electronics. Since this time, Mirai has been purported to have carried out over 15,000 distributed denial of service assaults. The initiation of the Mirai project began with the first source code release on September 30, 2016, which was publicly available. Subsequently, additional major distributed denial of service attacks have been executed, one targeting the French web host

OVH (1 Tbps) [40] and another on October 21, 2016 [41] aimed at Dyn, a DNS service provider for popular-hosted services like Twitter, Spotify, Netflix, Reddit, and GitHub. Both of these attacks were mounted against Dyn. A Distributed Denial of Service Attack (DDoS) was inflicted on Brian Krebs' renowned cybersecurity blog. This is perhaps the most notorious example of an attack of this nature. This assault generated a traffic volume of 623 gigabits per second (Gbps), which has never been documented or disclosed in the context of a distributed denial of service attack. This feat has also never been accomplished before [42]In the latter part of 2016, a Mirai version exploited a weakness in the CPE WAN Management Protocol (CWMP), which was installed in two different versions of consumer routers used by Deutsche Telekom [43]. Because of this vulnerability, the Mirai version is able to take advantage of the situation. Because of this weakness, around one million customers who were using the program exposed themselves to potential danger. In 2017, Radware became aware of the fact that a botnet known as Brickerbot [44] had been researching ports that were related with the SSH service, namely port 22. This was the year that Radware became aware of this information. Furthermore, it was shown that the Reaper variant is present [45,46]. This was made known to the researchers. Furthermore, despite the fact that it makes use of a substantial chunk of the Mirai code, its primary focus is on exploiting vulnerabilities that have been known about in the past. The Reaper variant use HTTP-based attacks rather to depending on Telnet brute force with default passwords [45]. This allows it to target vulnerabilities that are already known to exist in Internet of Things devices. The Reaper variation is able to function more effectively as a result of this. A new Mirai variety that was produced and given the name Satori was discovered in the year 2017 during the month of November [47]. This variety was given the name Satori. Because of the one-of-a-kind way in which it spreads, it is more comparable to worms than other sorts of worms when compared to other variations. This bot does not need to make use of the loader–scanner approach in order to do remote planting [48]. This is because the technique is not necessary. Infected devices are required to download themselves from the same beginning URL, as this is what Satori requests. Specifically, the ports 37,215 and 52,869 are the ones that are going to be attacked by the attack. The two vulnerabilities that Satori discovers and exploits the majority of the time are one vulnerability for port 52,869 that has been known about since 2014 (CVE-2014-8361) [49] and another vulnerability that was found in December 2017 (CVE-2017-17215) [50]. Both of these vulnerabilities have been known

about since 2014. In the year 2018, it was disclosed that the WICKED bot frequently investigated ports 8080, 8443, 80, and 81 [51]. A complaint was the means by which this information was collected. Following this, further exploits that are based on two vulnerabilities, CVE2018-10561 and CVE2018-10562, that are associated with the authentication of the HTTP service have started to be incorporated into at least five different botnet families [52]. These vulnerabilities are connected to the authentication of the HTTP service. These vulnerabilities are connected to the fact that the HTTP service is utilised in the process of authenticating users. Two vulnerabilities were identified by VPN Mentor on May 1, 2018, and they have the potential to be exploited by GPON home routers which have the number 53. These vulnerabilities are susceptible to being exploited by cybercriminals. It was in the year 2018 when a new strain of Mirai, which was given the name Okiru, was found. The Internet of Things devices that were equipped with Argonaut RISC Core (ARC) central processing units were the primary target of this strain's attention. The virus known as Okiru, which is somewhat analogous to Mirai, searches for devices that make use of Telnet ports and makes an effort to use the default passwords for such devices. It does this by going about its search for devices that use Telnet ports. The Masuta botnet, which is Japanese for "master," was discovered in that year, and its source code was made public on a secret forum that was only available to individuals who had been requested to participate. The forum was only accessible to those who accepted the invitation to participate. In contrast to Mirai, this botnet use a different encryption key seed in order to exploit vulnerabilities which are present in routers that have become outdated. In order to take advantage of vulnerabilities, this actions are taken. In order to perform an XOR operation on the strings that are contained inside the configuration files, a factor of 0 multiplied by 45 is applied to the strings. PureMasuta, the upgraded version of Masuta, includes a list of weak credentials that have the potential to be abused. This is in addition to recycling common malware that is akin to Mirai. PureMasuta is also known as PureMasuta. At the beginning of 2018, [54] the Mirai version started to spread by exploiting a vulnerability in the ThinkPHP framework that enabled remote code execution. This vulnerability allowed the vulnerability to be exploited. Because of this vulnerability, the variant was able to carry out code execution through the use of remote administration. In order to take advantage of this vulnerability, personal computers were forced to download and install malicious software, which thereafter utilised Telnet to create connections to other IP addresses. This was done in order to exploit

the vulnerability. Cybercriminals exploited this vulnerability in order to gain personal information. In 2019, Yowai, which included the ThinkPHP vulnerability on the list of potential infection vectors, was the second most frequent vulnerability, behind Miori, in terms of popularity. Yowai featured the vulnerability. Yowai has been granted the permission to take control of routers that utilise port 6 [55]. This is done with the intention of conducting distributed denial of service attacks. An additional bot that was constructed using the Mirai software was disclosed in the month of July in the year 2019. Moboot was the name that was given to this little bot. Furthermore, this assault takes use of the same Mirai scanning method [56,57 as well] in order to take advantage of a large number of bots that are cooperating with one another in order to target ports that are associated with DVRIP, ADB, HTTP, and Telnet. In the year 2020, researchers found two varieties: Sora and Unstable. These variations were identified by the researchers. These variations were found by employing a novel propagation method, which led to their discovery. In the context of a certain video surveillance storage system, these versions make it possible to remotely execute code by making use of the CVE-2020-6756 vulnerability that is present within the system. With the purpose of exploiting the vulnerability that was found in ThinkPHP [58] in the past, Unstable intends to make advantage of the vulnerability once it has been identified. A variant of the harmful software known as Mukashi was made available to the general public in the year 2020. This particular version took use of a vulnerability that is referred to as CVE-2020-9054. This vulnerability is a pre-authentication command injection vulnerability [59]. Network-attached storage (NAS) was the target of an assault that exploited the vulnerability.

2.2.4 Advanced Persistent Threats (APTs)

Botnets have become an indispensable component of some of the most complex types of assaults, such as Advanced Persistent Threats (APTs), over the course of the past several years. Botnets are used to spread malware and other malicious software. Espionage, data exfiltration, and disruption of vital infrastructure are all operations that are carried out by nation-state actors and well-funded cybercriminal groups through the use of complex botnets that serve several purposes. These efforts are carried out in order to achieve a variety of objectives. In order to carry out these actions, these botnets are designed specifically for that aim. Because they provide a high degree of adaptability and typically make use of

sophisticated evasion methods, the botnets that are now in existence are formidable adversaries in the realm of cybersecurity. As a result, they are a very formidable opponent. The Mirai versions IZ1H9, HailBot, KiraiBot, and CatDDOS are the ones that are now the most active [60]. This is the case as of the year 2023. Within the part that came before this one, an investigation of botnets that are related with the Internet of Things as well as the assaults that are associated with them was carried out. The botnet detection strategies for the Internet of Things are going to be the primary topic of discussion in this part. These strategies will be discussed in connection to the assaults that have been mentioned. It is required to adopt tactics for identifying botnets that are related with the Internet of Things (IoT) in order to prevent harmful actions carried out by botnets within Internet of Things devices and networks. This is necessary in order to accomplish the goal of preventing destructive acts. Because of the botnet's capacity to launch distributed denial of service attacks, this situation has arisen. The focus of this section is to explore various detection methods available for safeguarding the networks and devices connected to the Internet of Things. Figure 4 presents the classification of the existing solutions developed for botnet recognition on the internet of things. These solutions have already been proposed. There are two classes of strategies aimed at detecting active botnets within the Internet of Things: host-based detection methods and network-associated detection techniques [61]. Both of these strategies are referred to as detection techniques. The fundamental emphasis of host-based detection is the investigation of the code that is present on the device. This type of detection also has the identification of botnets as its primary purpose. These techniques analyse the amount of time needed for processing, access to files that are unknown, and other factors that contribute to the formation of botnets. This allows for the acquisition of knowledge regarding botnets. This kind of detection system may be broken down into two unique approaches: static analysis techniques and dynamic analysis methods [71]. Both of these approaches are considered to be distinct approaches. These two techniques are not interchangeable with one another in any way. In the process of utilising the static technique, both source codes and binaries are investigated. On the other hand, when the dynamic technique is utilised, electronic devices are analysed in real time. Benson and Chandrasekaran [72] made the decision to ring the bell in order to bring attention to the fact that devices connected to the Internet of Things (IoT) are vulnerable to security breaches. The vulnerabilities that are caused by devices that are linked to the Internet of Things but

are not connected to botnets were the primary focus of their research. However, they did give a crucial warning on the importance of host-based detection strategies, despite the fact that they did not provide a description of a botnet detection approach. Despite the fact that they did not offer any kind of explanation, this was the result. An investigation of Internet of Things firmware and methods for identifying malicious software in IoT firmware was carried out by Costin et al. [62,63] in 2014 and then again in 2018. As of 2018, this probe was carried out once more. When it comes to the static approaches, this survey is considered to be one of the approaches within that category. In the process of doing research on methods that are connected to firmware, it is very necessary to take this source into mind. They put forth a method for analyzing firmware pictures within their examination as a means of diagnosing possible botnets and malware interventions. Nguyen et al. [64] in later work, propose another option for performing static analysis which is much more in-depth. This approach aims to find PSI from the source code or binary code of IoT firmware. The objective is finding Printable String Information (PSI). Subsequently, the Convolutional Neural Network (CNN) is then trained with the malware samples where the input consists of PSI graphs capturing the malware. This is the next step. To improve the performance of the CPSU classifier, capturing the PSI context which is almost the most important piece of information, is believed to enhance reliability of the CNN Classifier. That enables increases reliability in the accuracy of the CNN classifier. In the course of attempting to check whether or not the firmware of an Internet of Things (IoT) device is infected, together with a PSI-graph and an attached CNN, these were utilized to find alternative firmware. This is done in order to determine whether or not the firmware is contaminated. Following the end of their investigation, they arrived to the realisation that the PSI-graph CNN classifier has a rate of accuracy that was one hundred twenty percent. Zaddach et al. [65] present a method of dynamic analysis that combines hardware (to evaluate the input/output of an Internet of Things system) and software in order to dynamically detect malicious firmware (botnet). This method was developed in order to discover botnet vulnerabilities. Detecting botnets was the motivation for the development of this method. Because it provides dynamic taint tracing or symbolic execution, dynamic analysis is an essential component for doing security research on Internet of Things (IoT) systems. This is because dynamic analysis permits symbolic execution. The identification of potential vulnerabilities is possible with the use of this information. Among the tools that Zaddach and his colleagues provide, one of them

is called Avatar. The purpose of this tool is to do dynamic analysis with the intention of discovering and detecting vulnerabilities. It is Zaddach and his colleagues that provide this instrument. Their technology is capable of doing dynamic analysis of complicated firmware, and they give a vulnerability investigation of the detecting system in order to illustrate this capability. This is done in order to demonstrate that their method is appropriate to the situation.

2.2.5 Botnet Detection

The use of dynamic analytic methodologies for the purpose of Internet of Things botnet detection resulted in the creation of honeypot-based solutions for the purpose of identifying botnets that are related with the Internet of Things (IoT). Detecting and capturing malicious software may be accomplished with the help of these honeypots, which serve as targets. In the event that honeypots have been attacked by botnets that are connected to the Internet of Things, the activity will be noted, and after that, suitable mitigation methods may be implemented. It is Pa et al. [66] who are responsible for the implementation of honeypots for the Internet of Things, and they likewise give the implementation. To be more specific, they propose IoTPOT as a method for simulating the Telnet services that are provided by a variety of various devices. This IoTPOT also has a virtual environment that is known as IoTBOX. This is an additional feature. In order to accomplish its goal, it will first document the activities that have been carried out, and then it will proceed to analyse those actions. They exhibit a considerable number of Telnet assaults as well as a wide range of botnet distributed denial of service attacks intended at devices linked to the Internet of Things. This is a result of the discoveries that they have made. The fact that Honetpots are able to identify at least five distinct botnet families through the deployment and analysis of this method makes it very evident that they are advantageous. The usefulness of Honetpots is established by this example. As a result of the characteristics that they contain, honeypots have been utilised in the development of a wide range of different methods. They have a difficult time recognising newly developing botnets for the Internet of Things, which are also known as zero-day assaults. Even with the context provided from previous works, there is still recognition given to such botnets. Other changes resulting from the proliferation of solutions based on machine learning (ML) include the application of honeypots data for training ML models. This is because of the advancement of ML solutions. In their paper, Viskarma and

Jain propose an unsuspected approach of detection with the use of honeypots [67]. What makes this procedure simple and efficient is the application of ML algorithms. The honeypot that falls under the Internet of Things serves as a data source for machine learning algorithms which are created by the honeypot. An selection of Internet of Things honeypots, including IoTPOT [66], Dionaea, ZigBee Honeypot, and other Multi-purpose IoT honeypots, are deployed for the purpose of obtaining information from a variety of Internet of Things devices. After the data have been collected, they are trained on a variety of different machine learning models, some of which include CNN, RNN, and LSTM, amongst others. This training process is described in more detail below. They are unable to use deep learning models within their organisation due to the fact that they only have a small dataset to work with. They are able to be able to capture zero-day botnet types that are not taught in their model because of the machine learning models that have been built already. This capability would not have been attainable without the utilisation of these models. Despite the fact that this study does not include any specific tests to demonstrate the validity of their models, the authors claim that their model has a 99% accuracy rate, which demonstrates the effectiveness of the hybrid model that makes use of honeypots in conjunction with machine learning techniques. All of the authors of the paper were responsible for carrying out this research. In addition, Banerjee et al. [68] present an example of a honeynet method that is organised around the idea of machine learning. Utilising local honeypots allows for the gathering of malicious network traffic dumps, binary files, and log files. This is performed by the utilisation of honeypots. These data are utilised in the process of training the machine learning models that have been acquired. The well-known dataset created by SocialNet is now being utilised for the purpose of testing and assessing the machine learning model that has been produced. Honeypots that are powered by artificial intelligence and make use of cloud computing are eventually proposed by Memos and Psannis [69] after a period of time has elapsed. In order to conceal the assaults that they are carrying out, they create a new honeynet that is made up of a large number of honeypots that are separated from one another secluded from one another. Honeypots are used as a decoy in order to mask the attacks that are being carried out. It is on this cloud server that the analysis of assaults on honeynets is carried out. This honeynet is connected to a cloud server, and the analysis is carried out on this server. Honeynets are useful for monitoring and securing networks. In order to predict infected hosts and networks, a supervised Logistic Regression model is used. Fulfilling this

purpose is possible via data obtained from the cloud server. A trained model stored on a cloud server makes it possible to assess the presence of botnets in real-time. While these models detect botnets, the cloud server averts the attack on the IoT device. This preemptively stops the attack. The author's study concluded that this approach, in combination with other methods such as the machine learning models, significantly reduces the time needed for botnet detection on IoT devices by over 100%. The detection of botnets can be done efficiently using hybrid methods such as honeypots, machine learning, and cloud servers. This study seeks to prove that hybrid approaches can be applied to achieve the intended objective. Within the context of the Internet of Things, Sajjad et al. [70] consider an additional vulnerability that lies with the Manufacturer Usage Description (MUD) framework. This specific vulnerability is referred to as the Internet of Things vulnerability. MUD is needed to control a network level functionality to devices which are embedded into the Internet of Things for the purpose of communication. The goal of this work is to improve the security that these Devices of the Internet of Things, IoT, are having regarding networks. At the same time, Mirai botnets try to exploit the MUD's available loopholes. In this regard, Sajjad et al. from IoT devices propose that changes be made to MUD to secure stranglehold botnets from IoT devices. It is noted, due to the inclusion of these supplements, it is recommended without reservation that MUD profiles be designed based upon the vulnerability level. The result suggests that the recommended changes have improved the security of IoT services and devices, as is evidenced by the increase in security levels. Besides, another approach that might be used to detect Internet of Things botnets is the use of network-based detection methods.

2.2.6 Active Monitoring

The monitoring and analysis of the traffic as well as the patterns that occur within the Internet of Things ecosystems, which is referred to as network-based botnet detection, systems is conducted with the goal of actions undertaken by botnets. This allows to identify capsule actions being undertaken by botnets. In the subsequent section of this article, we will enumerate the approaches that are network-based detection methods focus on. Network-based detection systems are classified into two types: active monitoring and passive monitoring [73]. Active monitoring is the first category of monitoring. The user needs to engage in system activity to participate in the active monitoring process. The active

monitoring approach includes a strategy that involves conducting proactive probes of the network in order to assess the reactions of the network. Proactive probing of networks alongside other strategies is what's incorporated in the active monitoring protocols. The development of this technology was intended to enable real-time problem detection. These and other systematic performance problems serve as risks, undermining security while posing one other issue. Active monitoring is used to provide contemporaneous information, still can impose strain on the network. This is something that is available even if it claims to relay precise details. Monitoring that does not disturb the normal functioning of the network being analyzed is referred to as passive monitoring. This monitoring approach enables packet capturing, log file analysis, and threat and anomaly detection. All of these tasks may be completed simultaneously. The methodology does not focus on ongoing attacks, and the possibilities for preemptive action are more about the long term. This captures everything from past recorded activity that can be utilized effectively to the network's history. With passive monitoring, various objectives such as reporting compliance, post analysis, and preemptive potential hazard detection can be accomplished. Such systems can be found in many modern market available detection technologies. The researchers of the study provided a security solution exclusively based on security event management for the Internet of Things domain [74]. This strategy has benefits as it may assist in recognizing actions that are overtly aggressive. The authors classify the multiple algorithms applied in the processes of rule generation according to the characteristics of the various algorithms that are employed. The algorithm is meant to make the analysis of event, anomaly detection, and information correlation regarding security easy to perform, thereby increasing the likelihood of detecting possible botnet attacks. The authors outline the various attack scenarios that pose serious threats to IoT devices' availability, integrity, and confidentiality. Additionally, they describe the weaknesses that were used during the attack, the security incidents that resulted from the attack, and the specific mitigation strategies that can be adopted to reduce the damage the attack inflicted on IoT devices. All this is given in terms of the specific mitigation strategies that may be adopted. The events in the SIEM-based system model undergo a great deal of processing and refinement. The reason for such alteration is due to several discrete links among various types of security events, attack vectors, and their corresponding vulnerabilities. The multi-relationships put forth in the discussion can be useful towards the investigation as they can aid in identifying the exploits and the attack

vectors which are relevant to the devices belonging to the IoT ecosystem. This certainly can enhance in the IoT devices security. Relations between some rules are able to be created in an automated manner, and perhaps this approach can further improve the technique proposed. If such a situation occurs, then the security information and event management (SIEM) system would be capable of encountering multiple attacks, vulnerabilities, and incidents. Basheer et al. [75] pay attention to the technique based on detection using an SIEM solution because it is particularly effective in identifying DoS attacks perpetrated by IoT botnets. This is for the reason that the SIEM solution is very useful. In the prototyping phase of the proposed architecture, the default gateway is used to forward the logs of the IoT devices to the Security Information and Event Management (SIEM) system. This happens during the initial stage of the deployment. The monitoring traffic originates from the numerous IoT devices whose traffic is being monitored in the system. The Security Information and Event Management (SIEM) system undergoes a number of data processing operations throughout its entire lifecycle. All these operations involve the parsing of the logs with subsequent indexing and storing them in a highly secured relational database. In a typical case, the logs will first be analyzed, and an attack will be notified if there is any deviation from normal behavior within the traffic profile of the device, alerting the network administrator. Everything takes place after the logs have been analyzed. If security analysts receive the burden of watching many systems simultaneously, they are likely to encounter some difficulties. The fact that the use of Splunk allows the collection and storage of all relevant logs into a single instance creates the possibility of constructing a unified solution. The fact that it is possible to identify different types of harmful traffic from several different devices connected to the Internet of Things (IoT) is the concern of the writers who proposed this prototype or architecture. The main concern of the authors. One other method that Marian et al. suggested is using the Splunk SIEM system designed to issue four different real-time alerts on the detection of different types of suspicious and/or malicious activities. [76]. This platform was designed specifically to supply these monitoring alerts. Hence, this platform has been developed for the purpose of providing these alerts in real-time. One of the alerts that has been custom created for dealing with type specific situations is the notice of Mirai virus infection. This warning is built, step by step, from the very beginning. For instance, in attempt to enhance the detection of DDoS attacks in computer systems, they propose a number of different solutions, one being the integration of Artificial Intelligence

with the Security Information and Event Management System (SIEM). Implementing AI as a component within the system amplifies its detection capabilities, which is a subject of greater importance. The system can learn and adapt over time to attack patterns, which ultimately improves security in IoT environments. In order to accomplish this, the system needs to be allowed to learn and adapt. Network software management and control can be peripheral, as they are done manually on a computer. Software-defined networking (SDN) handles the networking peripherals in a more flexible manner, where applications developed specifically can be used to manage them remotely [77]. Flexibility is automated since with SDN, there is the possibility of subdividing control and managing the network through set functions from one or various terminals. This marks the highlighted feature where SDN diverges from other traditional methods. Like all other devices used for different applications in technology, SDN has its own pros and cons. The advantage that SDN brings is it improves visibility along with peripheral insight to the network while making it easier to manage expenses. The paragraph below will specifically focus on outlining the other various detection techniques, likely to be introduced as a consequence of these factors, where they will be summarized in a visual format, specifically a table. The technologies are based on the preceding explained statement which serves as the anchor reference for Software-defined networking. Ozcelik et al have created an edge-centric software defined internet of things defense architecture (ECESID) utilizing fog computing turned paradigm [79]. This is the building ground for the architecture that is reached by infusing Ozcelik paradigm. The expansion of the acronym TRW-CB is threshold random walk with credit-based rate limiting. How I just explained was what the sentence TRW-CB means. In the context of this method, the algorithm in question is employed. Attempts at various time intervals during successful connection attempts are utilized to make the program initiate the scanning phase during attempted assaults on the host. This is done using successful connection attempts probabilities. To find malicious behavior, this approach uses a queue of TCP SYNs that is attached to every device that is IOT enabled.

2.2.7 Software Defined Networks SDN

One of the strategies that might be employed is the use of software-defined networking (SDN) together with intrusion detection systems (IDSs). Integrating an IDS into the architecture of an SDN results in a networked system with an integrated IDS that Manso et

al. [80] described. This is an example of a system based on the SDN architecture. Perhaps two of the most relevant parts of this system are the software-defined (SDN) controller, the network, and the IDS (Intrusion Detection System). With the adoption of this technology, it is possible to harness greater capabilities of the intrusion detection systems. The IDS (Intrusion Detection System) does a network traffic analysis with the aim of detecting any harmful action and then sends this to the SDN (software-defined network) for intervention. The SDN (Software-Defined Networking) controller will modify the rules of the network in response to alerts sent from the Intrusion Detection System (IDS). Should you choose to follow through with this plan, you can rest assured that detection and communication, as well as mitigation, will take place in a seamless manner. For example, the results of this study suggest that SDN can be efficiently disabled when other methods are applied. With advancements in machine learning solutions, many algorithmic strategies have been developed, and these have been proven effective. Development of these tactics, along with other methodologies, greatly enhanced the effectiveness of detection methods in software defined networks. Wani and Revathi [83] describe an approach for the purpose of malware detection which identifies ransomware and distributed denial of service attacks using a combination of Naive Bayes and Principal Component Analysis (PCA). This is one method for differentiating between the two types of assaults. As per this approach, the responsibility of packet capturing is given to the SDN controller, in whose hands TCP/IP headers are grabbed and then analyzed by machine learning models. This technology which attempts to distinguish between various forms of malware targets ransomware and other invasion types. It is possible that some of the threats to the Internet of Things (IoT) environments could be addressed by this SDN-based solution, which provides both detection and mitigation strategies. The experiments conducted within the scope of this research showed that the detection methods developed for ransomware and distributed denial of service attacks outperformed other techniques, as was initially suggested by the supplied strategy. They achieved this using the supplied method. Wani and Revathi provide a different method to identify atypical behavior with the [78] citation. The method applied in this case is called Micro-Cluster Outlier Detection (MCOD), which partners with Multi-layer Perceptron (MLP). This technique has the added benefit of detecting unusual behavior patterns. The authors of this study claim that most DDoS detection methods are placed on the IoT networks which are resource demanding. The DDoS detection techniques within the Internet of Things

can be improved with the application of centralized on Software Defined Networking (SDN) control. This is because it has the necessary resources required for performing the appropriate processes. The idea behind SDIoT-DDoS-DA, a stateful IoT solution based on SDN, is whence this assertion stems from. This proposal focuses on the assumption which stands as its basis. It has been noted that this approach may be applied to monitoring the system to detect any anomalies that may occur. Afterward, the Micro-clustered Outlier Detection (MCOD) algorithm is applied to establish whether the abnormal activity within the cluster could be indicative of a DDoS attack. Outlier detection is a technique that utilizes multi-layer perception networks to detect distributed denial of service attacks. The research suggests that this methodology can be applied for the detection and mitigation of DDoS attacks. This is due to the fact that Internet of Things devices provide high precision with low resource consumption. When an approach has been created, it has proven its anticipated usefulness. An efficient method of detection has been developed by Ren et al. [81] through the application of the GA-XGBoost genetic algorithm based on Software Defined Networks (SDN). These are the groundwork assumptions for formulating the strategy for the proposed approach. The OpenFlow protocol allows for the collection of six-dimensional vectors, which serve as the input to the GA-XGBoost algorithm. That is accomplished by means of the OpenFlow protocol. To achieve the objectives of this research, the XGBoost algorithm was implemented because of its ability to solve prediction and classification problems in underpowered controllers. This was the case because the algorithm is capable of solving these problems. To deploy this trained model on the edge controller, a resource-constrained edge controller is used. For evaluating this model, data obtained from the software-defined networking (SDN) network was used. During the research, it turned out that this model has a detection rate of 95.73% and its false alarm rate is much lower than that from other machine learning algorithms implemented within SDN. This is learned through the complete study process. During the course of the research, this was found to be quite important. There is one novel approach to machine learning that was employed in a different study done by Wang et al. [82] which they call Dynamic Generative Self-organising Maps, or for short DGSOM. This study sets out to achieve a novel source-based detection for distributed denial of service (DDoS) attacks in Software Defined Networks (SDN) using sFlow and Dynamic Generative Self-Organizing Maps (DGSOM). The method attains full detection by combining both macro and micro detection. DGSOM's granular detection identifies attack traffic while

macro detection using sFlow covers the entire network for DDoS detection. Identification of attack traffic is done using both types of detection. Both methods combined accomplish identification of the attack traffic. This micro-detection gives the system the capacity to effectively differentiate between various normal traffic patterns and attack patterns. This is granted to the system through utilization of these detection methods. A multitude of other algorithms also using machine learning with systems based on Software Defined Networking (SDN) was mentioned earlier in reference [25].

2.3 SUMMARY OF RELATED WORKS

The summary captures the various research efforts investigating the problems posed by IoT botnets and DDoS attacks with regard to cybersecurity. One of the aggravating issues with IoT botnets, especially on DDoS attacks, is their potential for growth, flexibility, and changing structures. Scholars have studied several botnet topologies, including star, hierarchical, and random configurations to analyze their attack strategies and exploit their weakness. Some major works have been done on detection techniques, which can either be host-based or network based. a.Host-based techniques: involve the use of static and dynamic analyses for the detection of threatening actions in the bounded IoT devices. In static methods, code is studied e.g. firmware or binary files fret. An outstanding contribution came from Costin et al, who demonstrated the effectiveness of static analysis through firmware analysis. Dynamic approaches also have their proponents, Zaddach et al with the Avatar tool. In addition, some honeypot-centric approaches, such as those offered by Pa et al. and Viskarma & Jain have proved to be valuable in aid the capturing contested Botnets as well as training ML-based detection models. Yet these methodologies struggle to cope with the problem of zero-day exploits and scale. b. Botnet activity detection relies on monitoring traffic patterns and anomalies. Network monitoring is divided into passive and active modes—the former providing insights after the fact, while the latter offering real-time detection but adding extra load to the network. Some works, like those of Manso et al. and Ozcelik et al., incorporate Software Defined Networking (SDN) into Intrusion Detection Systems (IDS) enhancing detection accuracy with machine learning techniques and optimizing resource consumption. These solutions based on SDN are promising and add practicality to the discussion, but they have high requirements for resources and scalability in relation to other IoT networks. Increased adaptability and accuracy in host- and network-

based methods is achieved through the application of machine learning and deep learning algorithms. For example, in botnet detection, the effectiveness of ML models was demonstrated through Nguyen et al.'s PSI-graph CNN approach and Ren et al.'s GA-XGBoost algorithm. Furthermore, the works of Memos & Psannis that propose hybrid approaches involving honeypots, cloud, and AI, while improving the capabilities of real-time detection, also increase the computation burden. The timeline detailing the evolution of IoT botnets, including well-known examples such as the Mirai botnet, highlights the increasing complexity of these threats. Satori, Reaper, and Mukashi demonstrate the speed at which botnets evolve by exploiting particular security gaps. The use of more sophisticated techniques such as blockchain and SDN also highlight the ingenuity employed to fight IoT botnets. Nonetheless, these approaches still struggle to achieve full scalability, continuous real-time mitigation, and fortification against new IoT-driven threats. As noted in the prior section, there remains a problem that needs further investigation in terms of assets, efficiency, and flexibility for IoT botnet detection and mitigation strategies which need to be developed for resilient and effective IoT security frameworks integrating AI, SDN, and hybrid models systems amidst evolving challenges.

Table 2.1: The Summary of Network-Based Techniques Leveraging Advancements in AI.

Author(s)	Contributions	Limitations
Thanh et al. [13]	Comprehensive literature review focusing on botnets' architecture and evaluation, detection techniques for attacks.	Limited emphasis on botnet DDoS attacks.
Stephens et al. [12]	Systematic review of IoT botnet detection (2015–2020) with qualitative/quantitative comparisons and emerging threats.	Lacks in-depth discussion of implementation for emerging techniques.
Vishwakarma et al. [9]	Focused on DDoS attacks in IoT networks, explaining impacts with evaluation data.	Limited exploration of detection methodologies beyond DDoS attacks.
Feily et al. [5]	Overview of IoT botnets and attack architectures.	Detection techniques not grouped into host- or network-based taxonomies.

Table 2.1: The summary of Network-Based Techniques Leveraging Advancements in AI "Table Continued".

Silva et al. [6]	Analysis of botnet architectures and impacts.	Focused on individual detection techniques without broader classification or taxonomy.
Shah et al. [18]	First review focusing on blockchain-based DDoS detection in IoT environments.	Narrow focus on blockchain without integrating other modern detection techniques like ML/DL.
Costin et al. [62, 63]	Surveys on IoT firmware malware detection using static analysis.	Limited exploration of dynamic techniques or integration with real-time systems.
Nguyen et al. [64]	Developed PSI-graph CNN technique for malware detection with 92% accuracy.	Limited focus on real-world implementation challenges and scalability.
Zaddach et al. [65]	Dynamic analysis combining hardware/software for malicious firmware detection using Avatar tool.	Does not address scalability and emerging zero-day threats.
Pa et al. [66]	IoT honeypots for capturing Telnet attacks and analyzing botnet activities.	Ineffective against zero-day botnet attacks.
Viskarma & Jain [67]	ML-based honeypot detection with various ML models achieving 99% accuracy.	Lack of experimental validation for proposed models.
Banerjee et al. [68]	Hybrid honeypot solutions with ML for IoT botnet detection.	Limited dataset and dependency on SocialNet dataset for validation.
Memos & Psannis [69]	AI-powered honeypots integrated with cloud for enhanced accuracy and real-time mitigation.	High computational resource requirements for cloud integration.
Sajjad et al. [70]	Improvements to Manufacturer Usage Description (MUD) for IoT botnet prevention.	Focused solely on MUD without exploring broader IoT ecosystem integration.
Ozcelik et al. [79]	Edge-centric SDN solution for IoT botnet detection using TRW-CB algorithm.	Lacks scalability and robustness for large-scale IoT deployments.
Manso et al. [80]	Integrated IDS with SDN architecture for enhanced detection and mitigation.	High resource demands and potential bottlenecks in SDN controllers.

Table 2.1: The summary of Network-Based Techniques Leveraging Advancements in AI "Table Continued".

Wani & Revathi [83]	ML-based SDIoT-DDoS-DA using MCOD for efficient DDoS detection.	Limited evaluation of broader IoT attack vectors beyond DDoS.
Ren et al. [81]	GA-XGBoost algorithm for SDN-based detection achieving 95.73% accuracy.	Limited to resource-constrained edge controllers; needs broader applicability.
Wang et al. [82]	DGSOM-based SDN solution for macro- and micro-detection of DDoS attacks.	Complex implementation; limited generalization to diverse IoT environments.

2.4 CONCLUSIONS

a. The literature review underscored the significant problems created by botnets and DDoS attacks, noting how severely they undermine the security of IoT ecosystems. The development of IoT botnets is rapid as they have diverse topologies and sophisticated techniques which make their detection and mitigation more difficult. The following conclusions can be drawn: a. The Emerging Dangers Posed by IoT Botnets: These IoT botnets pose a growing risk; for instance, the notorious Mirai and its variants took advantage of lapses in security measures like weak passwords and obsolete software in IoT devices. Their capacity to perform massive DDoS assaults showcases the critical need for appropriate protective steps. Botnets migrating from easy-to-understand centralized constructions to more complex decentralized and P2P-based systems of design illustrates their level of flexibility. b. Detection Methods Improvements: There are other detection techniques that have also improved on older models because they rely on host-based and network-based methods. The use of static and dynamic analyses within the boundaries of host-based methods yields effective mechanisms for identifying malicious code, while in network-based techniques, the emphasis is on monitoring traffic for anomalies. Machine learning (ML) and deep learning (DL) advancements have increased the precision with which these processes are applied positively impacting the identification of preemptive threats through improved analysis. c. Combining Technologies and New Avenues of Exploration Computation combines AI, honeypots, blockchains, and even cloud computing into powerful hybrids for botnet detection and mitigation. Multi-faceted approaches offer exquisite precision and accuracy for real-time detections but tend to

struggle with expansion, computational complexity, resource-intensity, and zero-day exploit handling. Software Defined Networking (SDN) adds layers of botnet detection, mitigation, and resource versatility through its programmability and flexibility in controlling IoT network security. d. Taxonomy and Systematic Reconnaissance: The studies under consideration systematically analyze multi-faceted botnet detection methodologies, forming taxonomies as well as plans for subsequent research strands. This structure enables the highlighting of unattended issues, such as addressing dynamic and IoT large-scale adaptable assault patterns, and resolving the issue of scalability. e. Constraints and Opportunities for Research: The integration of new methods may enhance existing ones; however, these multifunctional, dynamic, and wide-architecture scope approaches still remain unsolved. Furthermore, lacking is the discussion regarding the incorporation of adaptable, comprehensive detection blends that provide holistic protection. Most methods arguably stem from unique datasets, which dominate experimental conditions and utterly shatter real-world representation.

3. MATERIALS AND METHODS

3.1 INTRODUCTION

Thanks to advancements in electronics, communication, and Internet development, together with the proliferation of mobile things in our everyday lives, the Internet of Things has emerged. Networks in the modern Internet age can include thousands—if not millions—of items, making them dense and large-scale. The Internet of Things (IoT) is defined, discussed, and introduced in this chapter along with its architecture, application domain, technologies, and potential dangers and weaknesses. Additionally, we take a look at botnets and cyberattacks that may hit IoT networks, and we go over the many ways that people have suggested protecting these networks from harm in the past. Chapter ends with a discussion of the pros and cons of various intrusion detection methods and systems used in IoT networks.

3.2 INTERNET OF THINGS

3.2.1 Definition of the Internet of Things

As with other conventional electronics that we use on a daily basis—computers, tablets, cellphones, etc.—the "Internet of Things" [10] describes an ever-expanding web of physical items that are connected to the Internet and can be recognised and identified. [6]. In academic works, the term "Internet of Things" has been defined in many ways. So, the Internet of Things is not defined in a standardised way. "The extension of the current Internet to all objects that can communicate, directly or indirectly with electronic equipment themselves connected to the Internet." In a recent article, the term "Internet of Things" was used to describe what is essentially a new technological revolution. Physical and virtual "things" with identities, physical attributes, virtual personalities, and intelligent interfaces are seamlessly integrated into the information network. They can self-configure based on standards and interoperable communication protocols. This is another way of looking at the Internet of Things, according to [8]. Here, "things" link with one another, and both people and machines may communicate with one another (M2M) [9].

3.2.2 Internet of Things Architectures

Many of the objects in the IoT can be connected, at any point in their life cycle, temporarily or permanently, to the global network. since of this lack of consensus on an IoT design, and since their suggested solution relies on a wide range of technologies—including sensors, networks, communications, and computers—to function, their proposal is inherently flawed [10]. The four primary levels that comprise the Internet of Things are as follows: data perception, service, network, and application, as proposed by the International Telecommunication Union (ITU). Figure 3.1 shows this design in action.

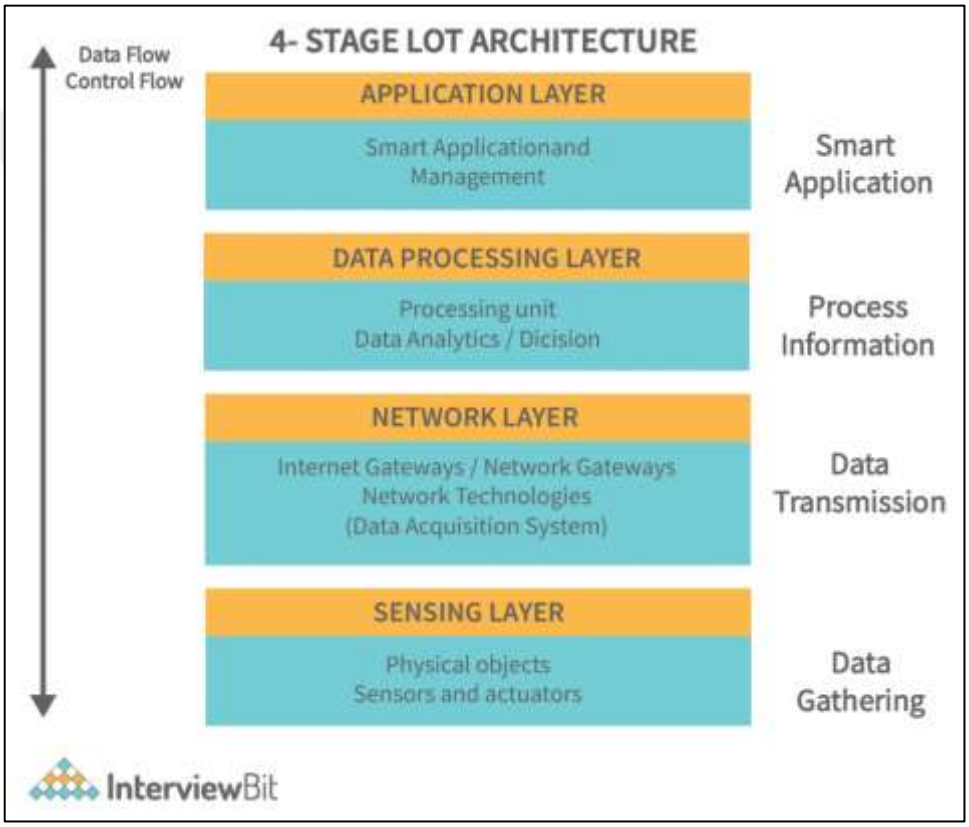

Figure 3.1: Architecture of the Internet of Things [10].

a. The perception layer: The perception layer, at the low level in the hierarchy, is responsible for capturing data, as well as identifying them in their environment. It provides different functionalities such as querying location, temperature, weight, motion, vibration, acceleration, humidity, etc. This layer thus includes the hardware necessary to achieve the collection of contextual data from connected objects, namely sensors, RFID tags, cameras, the localization system (GPS), etc [11].

b. The network layer: This is the second layer, also called transport layer. Its main function is the transfer of information from perception layer to service layer using reliable or unreliable transport channels. This layer uses various technologies for information transport such as 3G, Wi-Fi, Bluetooth, infrared, ZigBee, etc. And also network management technologies for heterogeneous networks such as fixed, wireless, mobile network, etc [12].
c. The service layer: It is also called middleware layer or service management, IoT objects offer different types of services and this layer is responsible for managing its services. The service layer stores, analyzes and processes information about the elements received from the network layer and makes an automatic decision based on the results obtained. It also serves to hide the complexity of the network operating mechanisms thus simplifying the development of applications by designers.
d. The application layer: The application layer represents the set of profiles of intelligent services, as well as the mechanisms for managing different types of data, arising from various types of objects.

The applications implemented by IoT can be smart homes, smart transportation, etc [11].

3.2.3 Internet of Things Technologies

a. Using a technology that is frequently referred to as the Internet of things (IoT), it is feasible to link a wide variety of different things to the internet. Therefore, in order for it to function in an appropriate manner, it is essential to have a number of distinct technical frameworks. Han and Zhong-shan [13] acknowledge that three of these technologies—WSN, RFID, and M2M—are considered to be of utmost significance. This is despite the fact that the Internet of Things (IoT) is based on a wide range of diverse technologies. An essential component of the Internet of Things paradigm is the wireless sensor network (WSN), which is composed of nodes that are linked to one another, wirelessly communicate information with one another, and are structured in the form of a cooperative network. This is one of the most significant parts of the Internet of Things paradigm. With each sensor being able to accommodate its own central processing unit (CPU), memory, RF (radio frequency) transceiver, and power supply, the system overall is capable of accomplishing a wide range of tasks autonomously [14]. RFID, or Radio Frequency Identification, refers to a system that automatically detects

the identity of a person or object using radio waves. This term encompasses all systems that utilize radio waves. In this method, information may be stored and retrieved from a remote place by using radio waves that are released by a radio tag [15]. This technology makes it feasible to store and recover information. It enables the identification of things from a distance as well as the transfer of data from radio tags to the items in question. It is possible to access the data that is stored electronically on the radio tag from a location that is not immediately accessible [16]. c. Machine to Machine (M2M): M2M is a concept that includes integrating information and communication technologies with various devices in order to achieve the objective of enabling intelligent products to communicate with the information system of an organisation or corporation in an independent manner [17].

3.2.4 Areas of application of the Internet of Things

The term “thing” can be perceived differently depending on the domain in which it is used. In industry, the thing can usually be the product itself, equipment, means of transportation, or anything that participates in the product life cycle. In the environment, it can refer to trees, a building, devices for measuring humidity or temperature, etc. Therefore, several application domains are affected by IoT. In their paper, Gubbi et al. [28], classified IoT applications into four domains: 1) personal domain, 2) transportation domain, 3) environment, and 4) infrastructure and public services, a diagram presenting these domains is shown in Figure 2.

Examples include industry, healthcare, education and research. However, in the future it will be possible to find the concept of IoT anywhere, anytime and available to everyone.

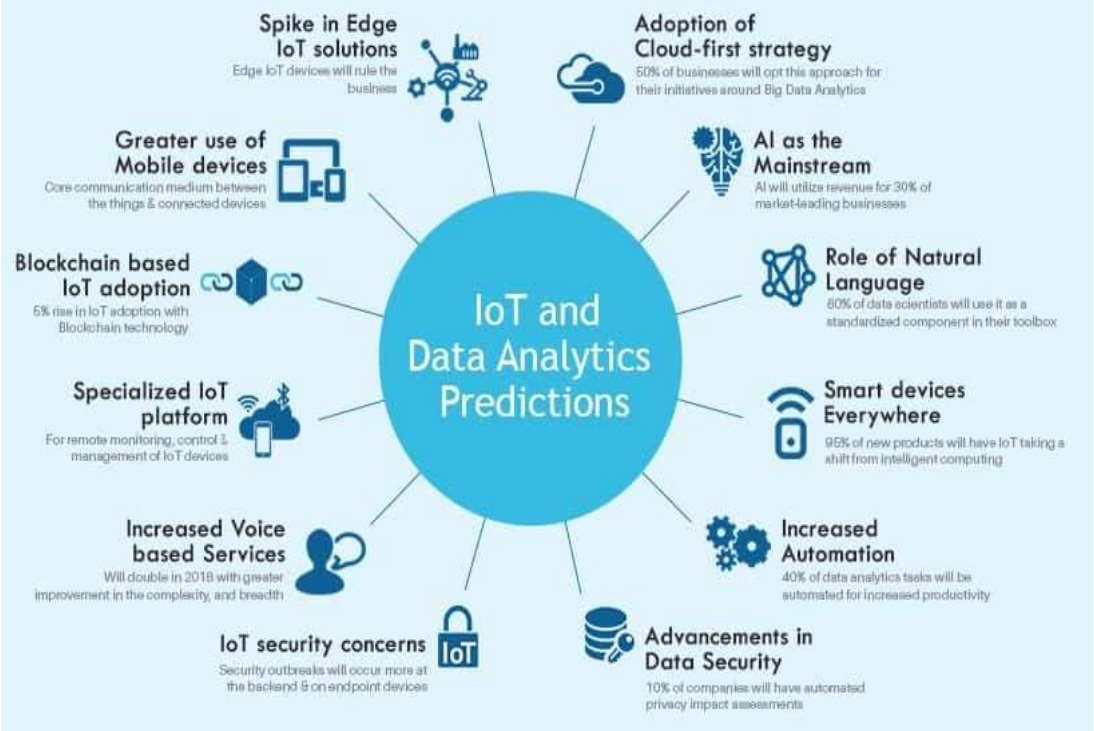

Figure 3.2: Application Areas of the Internet of Things [28].

3.3 VULNERABILITIES IN THE INTERNET OF THINGS

In the case that a system or its design contains flaws, an adversary can take advantage of these flaws to carry out commands, get access to data that is not authorised, or carry out an attack that constitutes a denial-of-service [18]. In 2010, connected gadgets have surpassed humans as the most abundant species on the earth. This occurred during the year 2010. Cisco analysts are of the opinion that there are currently fifty billion gadgets linked to the internet that are part of the Internet of Things [19]. To the contrary, one of the most major difficulties related with the Internet of Things is the absence of suitable measures for the security of private information [20]. Many Internet of Things (IoT) devices are afflicted by pervasive security issues, such as the failure to change passwords that are too weak or the absence of protection against brute-force assaults [21]. For example, the failure to change passwords that are too weak is a common problem. [22] A significant number of the apps that are now available will not support Secure Sockets Layer (SSL) or any of the other new security protocols that encrypt connections. You are able to exercise control over and keep an eye on devices that are linked to the network by using these programs. Furthermore, it is fairly uncommon for devices that are connected to the Internet of Things to be unable to

reset their default passwords or to install software upgrades. These devices are unable to fix security issues, such as the vulnerability that was identified in October 2017 connected with the WPA2 encryption protocol [23], which affects the majority of modern wireless networks throughout the world. As a result of this, these devices are unable to repair. It is noted cybercriminals now consider the Internet of Things (IoT) as an asset for their criminal activities. According to earlier research [24], it is possible to classify the devices that form part of the Internet of Things with respect to the specific threats they contain. An illustration is the Open Web Application Security Project which is commonly referred to as OWASP. The project offers educational material on security threats posed by the Internet of Things (IoT) to users and developers. The OWASP project classifies these threats into categories such as memory, network, web interface, among others, depending on how frequently they are encountered. This table draws from the findings of OWASP regarding the Internet of Things and shows the most significant vulnerabilities they discovered. The conclusions of a study that was carried out by HP in July 2014 indicated that on average, each and every Internet of Things (IoT) device contains around twenty-five vulnerabilities in terms of security. Seventy percent of the devices did not encrypt both local and distant traffic connections, and sixty percent of them had vulnerabilities in either their firmware or their user interface. For instance, eighty percent of the devices did not need lengthy and difficult passwords.

Table 3.1: Common Internet of Things Vulnerabilities [32].

Vulnerability	Examples
API Insecurity	• Unauthenticated API endpoints
	• Excessive data exposure via APIs
	• Lack of rate limiting
	• Improper input validation in API requests
Third-Party Dependency Risks	• Use of outdated libraries with known CVEs
	• No integrity checks on third-party code
	• Blind trust in external APIs without validation
IoT Device Exposure	• Default open ports with no firewall
	• Device beaconing sensitive data
	• No access timeout settings

Table 3.1: Common Internet of Things Vulnerabilities [32] "Table Continued".

	• Lack of encryption in pairing
	• Open discoverability mode enabled by default
	• Susceptibility to BlueBorne attacks
Insufficient Logging & Monitoring	• No anomaly detection on unusual user behavior
	• Logs stored in plain text
	• Failure to alert admins on critical actions
User Interface Vulnerabilities	• Clickjacking
	• Misleading error messages leaking stack traces
	• Auto-fill of sensitive fields in forms
Insecure Data Storage	• Storing credentials in plaintext in config files
	• No data-at-rest encryption
	• Weak database permissions

The way objects were designed and deployed is responsible for most of the vulnerabilities. First, we identify two types of vulnerabilities in IoT:

3.3.1 Vulnerabilities Related to Weaknesses or Errors in Design by Manufacturers

a. Design vulnerabilities such as weak or non-existent passwords which results in attacks related to authentication or authorization. For example, in Phillips Hue smart bulbs [25], where the authentication password to control the light is limited to only 6 alphanumeric characters. In addition, to control the bulbs via the use of a smartphone, an MD5 hash is generated from the device's MAC address which is not private information, and then it is provided to the device as an authentication token. The Mirai botnet [26] is a perfect example of attacks that can occur due to vulnerabilities of this type. Indeed, the infection relied mainly on weak or default passwords [27].
b. Vulnerabilities in the implementation network protocols or cryptographic standards. Fouladi and Ghanoun [28] demonstrated for example a vulnerability in the implementation of the Z-Wave key exchange protocol that could be exploited to take full control of a Z-Wave door lock knowing only the home ID and node of the target device, which can be identified by observing Z-Wave network traffic over a short period of time due to the frequent polling of devices in a Z-Wave network.
c. Vulnerabilities mainly come from trust provided by the designers/manufacturers of these objects towards other equipment in the home. As a result, a massive use of UPnP

(Universal Plug and Play) [29] within IOTs to launch attacks, UPnP is the protocol that allows a user to open ports on the network input equipment, thus authorizing people to control the objects from the outside. This functionality allows botnets in particular to connect and infect IOTs via the use of Telnet or SSH.

3.3.2 Vulnerabilities Related to Limitations

Implementing all the protection mechanisms usually present in traditional computing in IOTs is difficult given the low power of the latter [27].

The possibility of exploiting a buffer overflow present in the implementation of the Bluetooth protocol of Arduino Yun, widely used in the IoT world, was shown by S. Pastrana et al [30]. They then developed a worm to propagate. The low power of connected objects prevents the implementation of memory protection such as address space randomization (ASLR) [27]. In addition, a vulnerability in the specification of the Bluetooth Low Energy (BLE) protocol was discovered by Mike Ryan [31].

Key exchange in BLE is vulnerable to brute-force attacks which eliminates any confidentiality associated with the protocol.

3.4 BOTNETS IN IOT

While compromised PCs served as the first building blocks of botnets, hackers have found new and even more alluring targets in the ubiquitous Internet of Things devices due to the lack of information security safeguards provided by these devices. This is because there are zero layers of protection for sensitive data on these devices. Recent years have seen some of the most devastating attacks carried out by botnets, which are largely composed of unprotected Internet of Things devices.

3.4.1 Definition of A Botnet

A "botnet" is a collection of infected and controlled devices that are confined within a computer network [35]. The word "botnet" refers to this collection. An international network that is referred to as a botnet is created by hackers. This network can be controlled remotely through the utilisation of specialised Trojans. Through this method, they are able to avoid detection by the intrusion detection systems (IDS) and prevention by the intrusion prevention systems (IPS) of linked devices. When they have gained access, they

immediately take control of the devices and add other ones to the collection [34]. In spite of the fact that hijacked personal computers were the first building blocks of botnets, the widespread availability of Internet of Things devices and the absence of information security measures supplied by these devices have made them a new and even more appealing target for hackers. This is because these devices do not implement any information security measures. Botnets, which are mostly made up of Internet of Things devices that are not protected, have been the culprits behind the most catastrophic assaults that have occurred in recent years. The Mirai botnet is the greatest botnet that has ever been since it has infiltrated a substantial number of Internet of Things devices. This makes it the largest botnet that has ever existed. [34].

3.4.2 Evolution of IoT Botnets

There is little difference between classic botnets and IoT botnets in terms of architecture and lifetime. The gadgets that make up the network are where the two vary most significantly. According to Table 3.2, the classical botnet and the Internet of Things botnet both went through different stages of progression:

Table 3.2: Annual Evolution of the Botnet [73].

Year	Description
1995	Concept virus becomes the first macro virus to spread via Microsoft Word documents.
2000	ILOVEYOU worm causes widespread email-based attacks, affecting millions of computers worldwide.
2003	Slammer worm spreads rapidly via buffer overflow in Microsoft SQL Server, causing global disruptions.
2005	Mytob botnet combines worm features with botnet control, targeting corporate networks.
2007	Storm botnet emerges, using email attachments to infect and create a resilient P2P botnet.
2009	Conficker worm spreads through Windows vulnerabilities and becomes one of the most resilient botnets.
2012	Flame malware discovered targeting Middle Eastern countries, with advanced espionage capabilities.
2015	Dridex banking Trojan spreads via malicious Word macros, enabling financial theft.
2018	VPNFilter malware infects routers and NAS devices, enabling spying and device destruction.
2022	Emotet re-emerges with modular updates, targeting organizations with spam and ransomware droppers.

3.4.3 Botnet Life Cycle

Figure 3 provides us with the opportunity to see the processes that comprise the bot life cycle. In this paper, the five stages that are involved in the process of establishing a botnet are broken down and explained. Infection, injection, connection, malicious command and control, update, and maintenance are all procedures that fall under this category. an. The first stage of the contamination, which includes: While concurrently searching a target subnet for vulnerabilities that are previously known, an adversary takes use of a number of different exploitation methodologies in order to infect victim workstations [12]. During the secondary injection phase, hosts that have been infected through the use of a script that is known as shell-code are responsible for carrying out the script. It is the responsibility of the shell code to retrieve the actual binary image of the bot from the location that has been provided by means of FTP, HTTP, or P2P. Installing a bot binary on the system that is intended to be utilised is the second step. When the bot program is installed on the victim's computer, the computer immediately undergoes a transformation into a "Zombie" and begins to run the malicious software. In the event that the machine is rebooted, the bot application will begin operating without any intervention [35]. c. Phase of connection: During this phase, the bot software creates a link between the zombie and a command and control (C&C) channel. This phase is also known as the connection phase. Once the command and control channels have been established, the computer that is considered to be the "zombie" of the victim is the one that joins the botnet that is being operated by the attacker [35]. d. The malignant phase of command and control: Once the link has been established, the principal command and control activities of the botnet will begin. This phase is known as the "malicious phase." The command and control channel allows the botmaster to communicate with the bot army. This allows the botmaster to monitor and control the bot army. The botmaster is the one who issues commands to the bot programs, and the bot programs are the ones that proceed to carry out those commands [35]. In the third phase, which is maintenance and revisions, it is essential to make certain that the bots are continuously updated and animated. This is a need that must be observed. Within this part, we express our desire for the bots to bring their binary files up to date. Through the use of Dynamic Domain Name System (DDNS), a resolution service that permits continual updates and changes in server locations, botmasters make an attempt to maintain the invisibility and portability of their botnets [12]. [35].

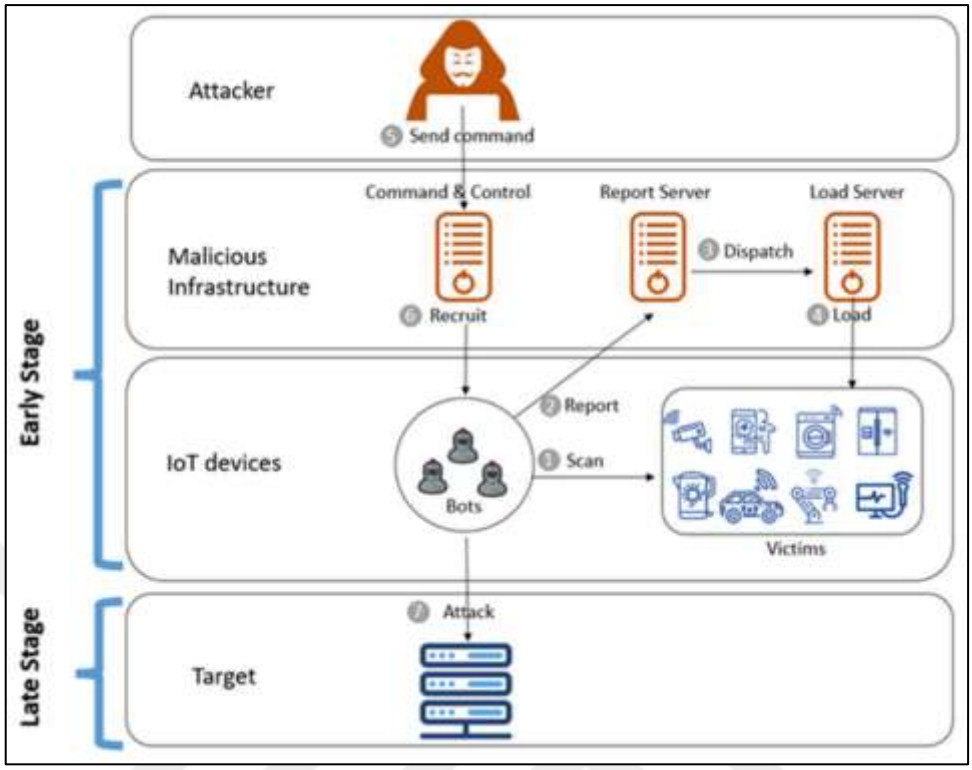

Figure 3.3: Life Cycle of a Botnet [35].

3.4.4 Examples of Botnets in IoT

In this section, we will briefly describe the examples of IoT botnets on which we tested our approach for detecting botnets in IoT networks:

a. Mirai: Is a botnet that infects IoT devices. Its dual purpose is to spread the infection to misconfigured devices and to attack a target server as soon as it receives the corresponding command from the person controlling the bot, or botmaster. The command and control (C&C) server provides the botmaster with a centralized management interface to check the status of the botnet and orchestrate new DDoS attacks. Typically, communication with other parts of the infrastructure is done via the anonymous Tor network. The loader facilitates the distribution of executables targeting different platforms (18 in total, including ARM, MIPS, and x86) by communicating directly with new victims. The reporting server maintains a database with details about all devices in the botnet. Newly infected usually communicate directly with it. Initially,

Mirai scans random public IP addresses via TCP ports 23 or 2323. Some addresses, including those of the US Postal Service [36].

b. Torii: Named for its use of Tor exit nodes to launch telnet attacks, Torii behaves differently from Mirai. It does not launch DDOS attacks and it does not mine cryptocurrencies. Instead, it comes with a fairly rich set of features for exfiltrating sensitive information, a modular architecture capable of retrieving and executing other commands and executables, and all over multiple layers of encrypted communication. Torii begins with a telnet attack on the weak credentials of the targeted devices, followed by the execution of an initial shell script. The malware uses multiple commands to download binary payloads by executing the following commands: "wget", "ftpget", "ftp", "busybox wget", or "busybox ftpget". Once the script determines which architecture the target device is running on, it downloads and executes the appropriate binary from the Load server. Torii communicates with these C&C servers via TCP port 443 as well as additional layers of encryption. It uses port 443 as a deception, as it does not communicate using TLS but takes advantage of the common use of this port for HTTPS traffic [37].

c. Okiru: Previously nicknamed Satori is a variant of Mirai that was first identified by Check Point researchers on November 23, 2017. It does not rely on brute-force telnet-based attacks. Instead, it executes attacks on port 37215 by exploiting the previously unknown vulnerability.

   CVE-2017-17215 is an example of a vulnerability in Huawei HG532 devices. The attack involves command injection, where the malicious payload is downloaded and executed on the Huawei router, The main purpose of the payload is to instruct the bot to flood targets with manually crafted UDP or TCP packets
   [38].

d. Hajime: which means "beginning" in Japanese is an IoT worm that was first reported on October 16, 2016. This worm builds a huge, P2P botnet of nearly 300,000 devices, but its target is still unknown. Unlike Mirai, it is very sophisticated and flexible. At the same time, Hajime leverages Mirai's infection method and brute force exploits. It is capable of updating itself, it can quickly and efficiently extend its bots with richer features. In

addition, it is a modular malware that supports plugins. The current plugin provides scanning and loading services to discover and infect new victims. The efficient implementation of the SYN scanner scans for new victims via open ports TCP/23 (Telnet) and TCP/5358 (WSDAPI) [39].

e. Hide and Seek: In late January 2018, a new IoT botnet was reported by Bitdefender, Hecommunicates in a complex and decentralized manner and uses several anti-tampering techniques to prevent a third party from hijacking/poisoning it. The bot can perform web exploitation against a range of devices via the same exploit as Reaper. The bot has a worm-like propagation mechanism, Similar to Mirai[40].

f. Muhstik: according to According to researchers at security firm Cloud Lacework, the botnet is known to use web application exploits to compromise IoT devices, Muhstik leverages IRC for control and commands, It has been using the same infrastructure since it first appeared in the threat landscape. The botnet is mainly spreading through home routers such as GPON routers, DD-routers, and WRT and the Tomato router. The Muhstik attack has several stages, the first stage being a payload called “pty” used to download other components. Once installed, the Muhstik malware will contact the IRC channel to receive commands to download an XMRmrig miner and a parser module. The latter is used to expand the botnet by targeting other Linux servers and home routers [41].

3.5 CYBERATTACKS IN AN IOT ENVIRONMENT

Hacking websites, mail servers, and email accounts—typically by gaining remote control of machines—is an ongoing problem with computer security, particularly on the Internet. Any linked device, just like any connected computer, is vulnerable to a wide range of cyberattacks [6]. This vulnerability naturally follows to the IoT. Figure 3.4 shows a four-layer architecture-based classification and analysis of Internet of Things (IoT) threats and security/confidentiality concerns:

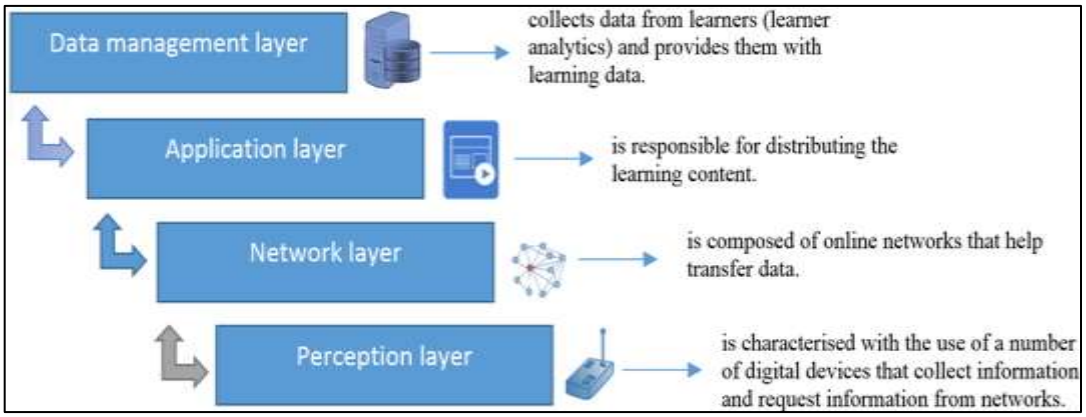

Figure 3.4: Four-Layer Architecture of IoT [45].

3.5.1 Attacks in the Application Layer

In order to gain unauthorized access to sensitive user data, attackers usually exploit vulnerabilities in programs and applications, and inject various malware like Rootkit, spyware, Trojans, etc. [45].

a. Code injection: Aims to introduce malicious code into the system by exploiting their vulnerabilities, to steal data, gain control of the system and spread worms or other goals. Common attacks include shell injection and HTML script injection. As a result, loss of control of the system and compromise of user privacy for the attacker, or even complete system shutdown [45].

b. Buffer overflow: Result in code or data buffer boundary violation, where the attacker writes a long sequence of data to a specified area of memory that usually has a predefined layout to contain code and data segments. The result can be modification of other data (e.g., execution of malicious code (e.g., by encroaching on a code segment), and destruction of program control flow [45].

c. Handling of sensitive data: Aims to gain illegal access and manipulation of sensitive data, thus violating user privacy by exploiting design vulnerabilities in the authorization model. As an example, smart applications and devices that represent a problem for data security due to the lack of sufficient protection of events contain sensitive data sent by

the application to the device for monitoring. In addition, a violation of user privacy can be seen due to the lack of adequate protection for user input [45].

3.5.2 Attacks in the Service Layer

Consists of building a malware by inserting SQL statements into the input data. Attackers use these SQL statements to perform read, write, and delete operations, which threatens the entire database system. When web applications are attacked by SQL injection, the actual page displays different results compared to the real information [45].

3.5.3 Attacks in the Network Layer

The most common attack is DoS attack which can exhaust network resources and affect the availability of network service. Communication patterns can be obtained by listening and analyzing the traffic through the network. After obtaining the communication pattern, an obvious attack that a malicious agent can perform. In addition, there are specific attacks on the network node. By compromising a network node, attackers can obtain the transmitted information and take control of the network, such as Sybil attack, replay attack, and man-in-the-middle attack. The attack in the network layer can also destroy the network communication by using the vulnerabilities of network protocols and network nodes [45].

3.5.4 Attacks in the Perception Layer

There are a broad array of sensor and identification technologies that are utilised by the perception layer. The bulk of these technologies are dependant on wireless connectivity. This is made possible by the fact that deployment environments can encompass a wide range of different configurations. It would not be difficult for an opponent to carry out an eavesdropping operation on this transmission given the circumstances that are now present. It is possible that an adversary could be able to disrupt the connection between the reader and the RFID radio tag if they were to gain access to the perception layer, which is highly dependent on RFID technology. Alternatively, they could be able to keep the node in a running state, which would result in an increase in the amount of energy that is consumed. When attacks are directed at the perception layer, the collecting and transmission of data are frequently the focus of those attacks[45].

3.6 COUNTERMEASURES

3.6.1 Cryptography

In most current security mechanisms, cryptography is probably the most widely used technique in traditional wired and wireless networks with large computing and memory capacity. Cryptography refers to the set of techniques for encrypting messages, i.e. making them unintelligible without a specific action. Cryptography solutions are renowned as secure solutions that address all the problems related to data security. The use of cryptographic schemes [48].

3.6.2 Intrusion Detection

Through the utilisation of an intrusion detection system, also known as an IDS, it is able to put into action the processes and ideas that are associated with intrusion detection. The goal of these systems is to discover or identify instances in which a computer system is being used for purposes that are not allowed. This can be accomplished by locating or identified instances. At the communication level, intrusion detection systems (IDS) are vital because they enable the monitoring of network operations and communication connections, as well as the sounding of an alarm in the event of an anomaly, such as the disobedience of a specific policy. In addition, IDS are indispensable because they allow for the monitoring of network operations. It is a common practice to adapt classic intrusion detection systems to either the standard Internet or wireless sensor networks (WSNs). This is because both of these networks are becoming increasingly popular. Concerns regarding the privacy and security of the Internet of Things (IoT) have been at the core of the bulk of the recommendations for intrusion detection systems (IDS) that have been made as of late. In order to fulfil the needs of Internet of Things (IoT) nodes that are linked to IPv6 in real-time, SVELTE [51] was one of the first intrusion detection systems (IDS) that was constructed with the intention of meeting those criteria. In the realm of the Internet of Things (IoT), it is able to monitor exchanges that make use of the IPv6 and 6LoWPAN protocols and detect routing risks such as fabricated or changed information as well as Black Hole attacks. In addition, it is able to identify routing hazards from this perspective. In [52], an additional immunity-based intrusion detection approach was presented for use with the Internet of Things. In the context of the Internet of Things, this strategy was suggested for implementation. As a consequence

of this, a number of studies that have been carried out over the course of the past several years have presented solutions to the issue of injection that are based on IDS.[53], [54]. For example, Son et al. describe the design and implementation of DIGLOSSIA [53], a novel tool that accurately and efficiently detects code injection attacks on servers.

According to J. Roux, the tools studied can be classified into two categories according to their type of action, either in prevention or in detection/monitoring via observations in operation [27]. For example, in prevention we find the IoT Sentinel system developed by M. Miettinen al. [55], allowing to dynamically identify the connected objects present and to secure communications [27]. Since vulnerable IoTs are numerous, this type of mechanism can generate strong usage constraints for users, even if the preventive side is interesting, in particular to identify the network elements and their potential flaws, it must be in our opinion complete by a protection method based on the analysis of the flows in operation between the objects of the home.

3.6.3 Authentication

In order to verify the identity of a user, authentication is a process that serves the aim of serving as a way of doing so. Through this method, it is feasible to ascertain that each and every correspondent is engaging in conversation with the actual person that they claim to be. Through the utilisation of access control, only individuals who have been granted permission to do so are able to get access to resources. Techniques of authentication are important for two-way communication in the context of the Internet of Things (IoT), and they serve as the initial line of defence against unauthorised access. In addition, they are necessary for blocking access by unauthorised individuals. When it comes to gaining access to a system, it is not uncommon for attackers to exploit holes in authentication and authorisation measures. One example of this would be the fact that smart applications do not offer adequate protection for user input. This leaves the door open for hostile actors to potentially breach the privacy of app users. Here is an illustration of this. In order to solve these issues, it is common practise to make use of systematised access control paradigms, such as role-based access control (RBAC). Under any one of the many guises that it may assume, it is feasible for a single item to carry out a vast variety of actions. Therefore, in order to regulate the rights and access that are allowed to the user, the system will decide the user's role that they are assigned.

3.7 CONCLUSION

In this chapter, we have made a study on the Internet of Things, namely the definition, operation, life cycle of an object and the architectures of the IoT and its main areas of application. We have laid the basic bricks and federated the concepts necessary for understanding the problem addressed in the rest of this manuscript. Among these concepts, we have seen the vulnerabilities and attacks that can target networks
IoT on the different layers of their architecture, as well as botnets that pose a real danger to IoT networks. Finally, we discussed the security mechanisms proposed in the literature to protect IoT networks against different attacks. The next chapter will be devoted to the presentation of deep learning and anomaly detection, as well as one-class classification.

4. PROPOSED METHOD

4.1 DECISION TREES AND RANDOM FORESTS

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. Each path from the root of the tree to a leaf node generates a rule, and each attribute that gets tested along the corresponding path generates a condition. The strength of the decision tree as a rule generation tool lies in its capability to express a complex and intricate decision logic in a form that is easily understood by humans. If decision trees lean too much on the real data, it may lead to overfitting problems, resulting in high variance and a reduction in predictive ability. The random forests algorithm attempts to deal with this problem. A forest is a collection of trees, and a random forest is one where each tree is constructed using a random combination of data points. At each internal node in the tree, the machine learning algorithm attempts to make each split according to a randomly chosen subset of the attributes.

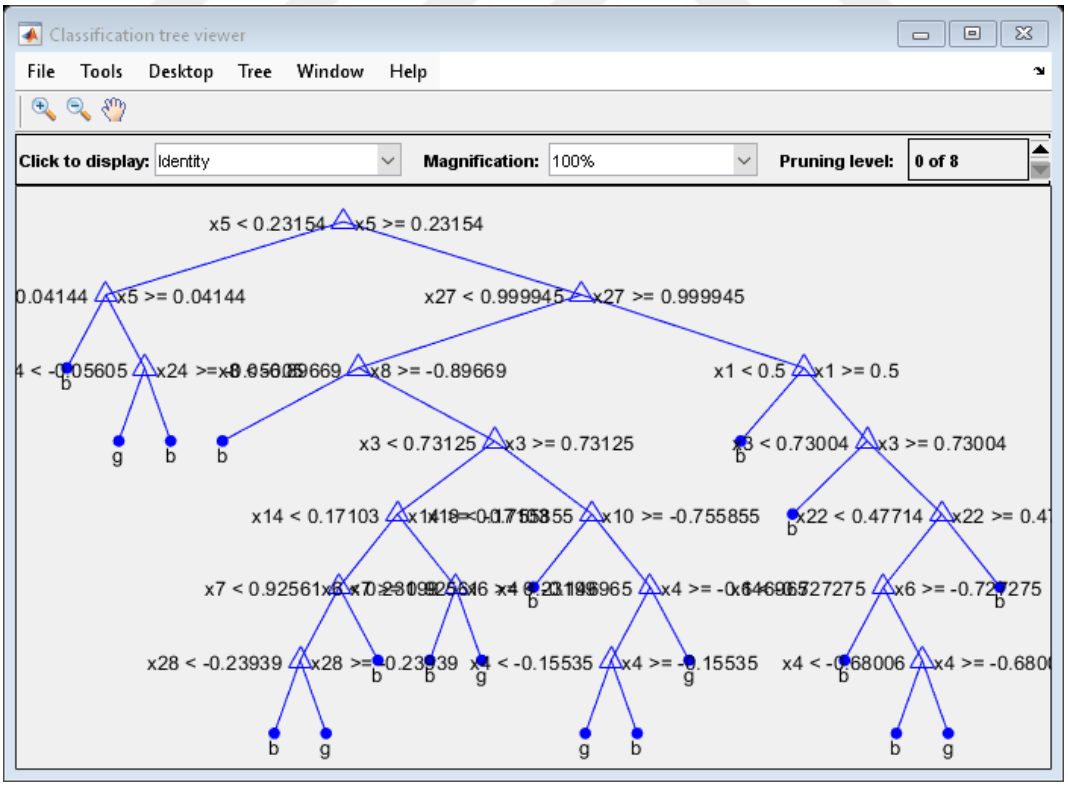

Figure 4.1: Classification Tree in the Proposed Method.

Random forests have been set up for successful large data sets and have solved high-dimensional problems. Data mining results using decision trees associated with random forests have received great attention in recent years due to their good predictive ability and explanatory detail, and they have been used in different intrusion detection systems. They have emerged as a popular technique for developing effective and efficient botnet traffic classifiers due to their competitive performance in comparison with others. They can simultaneously examine the significance of attributes and identify important interactions among attributes. Additionally, the structure of a decision tree can provide clues about complex relationships between sensitive data and different traffic characteristics. A powerful ensemble technique with many inherent advantages over regular decision trees is shown to be effective in this model, where expert and random feature selection are used together for large dimensionality in unit change rules. Furthermore, the application of a specific algorithm is observed to provide a greater understanding and accurate estimation of the causes of legitimate web traffic, such as P2P traffic and video conferencing.

4.1.1 Decision Tree Algorithm

Decision trees are brute force learning algorithms that can have several different designs based on the splitting criterion. The proposed adaptive decision tree consists of regular decision trees that are trained on feature space subsamples by dimensionality reduction, which are combined together as a random forest integration. Individual decision trees are non-parametric and constructed using an information gain splitting criterion. Prior to training decision trees, during the initial execution and with a development subset of the training set records, a genetic algorithm selects the characteristic intrinsic features that are used for a botnet command and control communication. The characteristic features consist of source and destination IP addresses, source and destination ports, payload size, and network delay. In order to select only the most interesting feature vectors and discard all the uninteresting ones, the initial decision tree is pruned so it comprises only the relevant network attributes.

Each node of a decision tree repetitively creates by splitting the leaf node records until either a minimum number of samples is attained or no single attribute element supports a further division. The split point near the minimum impurity is selected, the histogram split search process is used to speed up the search, and the split-node impurity measure is the entropy or information gain criterion. After tree creation, it is used to classify and count the decision

tree vote in the form of a probability or relative occurrence of each feature vector of the training set. Several trees are trained to form a random forest, with bootstrapping of the sample records. Bootstrapping is a sampling method with replacement that produces random subsamples with a size equal to the training set size and some selected probabilities. The random forest is used to integrate the individual trees and to benefit from it by the classification process during the detection.

Figure 4.2: Confusion Matrix of the Proposed DT.

4.1.2 Random Forest Algorithm

The Random Forest algorithm combines multiple decision trees. The advantage of this approach is that the overall model can be without overfitting. The base level trees can also be called estimators. Many decision trees can be formed using different features and also different data subsets. Each of the base trees' voting is counted, and the predicted class value is based on the majority vote from the base model estimate. Features are selected using the bootstrap sampling method, which is based on the random selection of data points and also a limited number of features. The widely used features count the square root of the total units, and from the data, only one feature with replacement is taken. These results should include the use of repetitive features, which is the initial part of the data point repeated analysis.

The concept of bootstrapping and random feature selection, along with several decision trees, is considered to be fast and efficient in reaching accurate predictions. Decision trees used as a base learner may be redirected from potential random trees. The document presented as the initial model consists of a number of trees, and the final model is called as it is done. Prediction is also made in the final model, and the number is based on finding the class that has the most repetitions. Each decision tree indicates an output vote, called the class value of the decision trees' forest output cluster, and the majority vote is counted on each vote from the classification, which is put at the tree end to measure unique category predictions from all decisions.

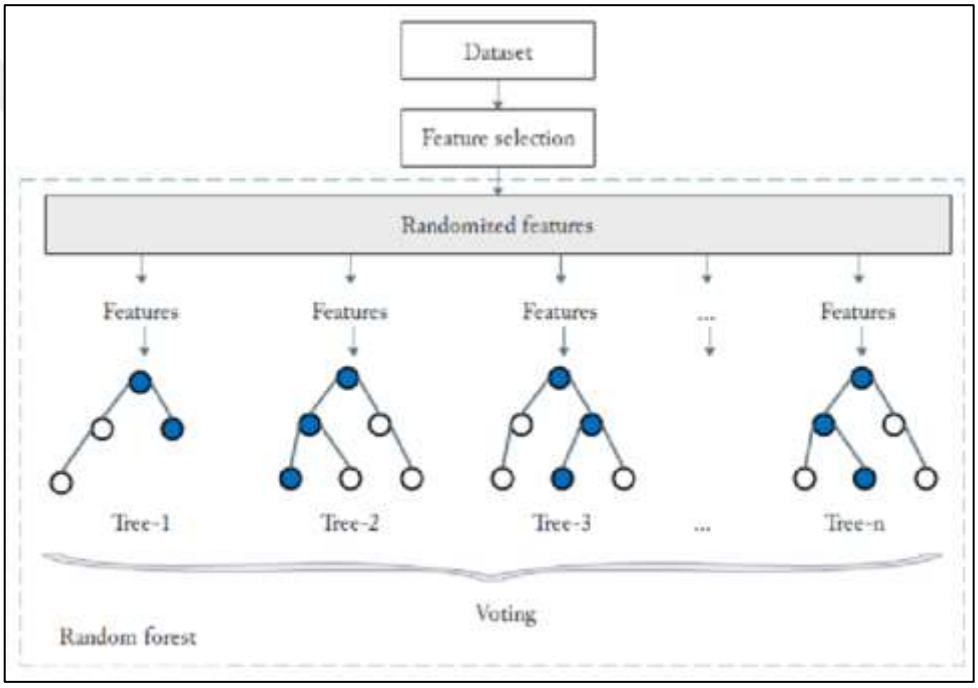

Figure 4.3: The Proposed RF Model.

4.2 DIMENSIONALITY REDUCTION TECHNIQUES

Given that high dimensionality will lead to a slightly slower executing process and low performance of decision direction, an effective dimensionality reduction algorithm is relevant to avoid both overfitting and underfitting in a botnet detector system design. In the given experiment, a data feature set has been utilized to determine and apply dimensionality reduction algorithms in the B-ADTree model. Different dimensionality reduction algorithms have been conducted to carry out these experiments. As principal component analysis can unjustly influence the classification process, collaborative research of PCA and decision

trees has been studied within a similar research field. Referring to information gain that selects and eliminates the most effective features that acquire maximum information from the feature space, two formal principles have been evaluated in high dimension/low sample datasets. In the DTAICS method, it demonstrates a need for data elimination and a consideration of increasing values of performance measures. Hyperparameter tuning is a crucial step to optimize the performance of the RF algorithm. The parameters considered for tuning in MATLAB's TreeBagger function are summarized in Table 4.1:

Table 4.1: Random Forest Hyperparameters and Selected Values.

Hyperparameter	Description	Selected Value
Number of Trees (n_estimators)	Number of decision trees in the ensemble	100
Maximum Depth (max_depth)	Maximum depth of individual trees	Unlimited (default)
Minimum Samples per Leaf (min_samples_leaf)	Minimum number of samples per leaf node	5
Maximum Features (max_features)	Number of features considered for each split	sqrt(Number of features)
Out-of-Bag Error Estimate (OOB)	Used to estimate model error during training	Enabled

The hydrogen fuel cell produces electric power P based on the reaction: (4.1)

$$\mathrm{H_2} + \frac{1}{2}\mathrm{O_2} \rightarrow \mathrm{H_2O} + \text{ Electricity } + \text{ Heat}$$

The power output is given by:

$$P = V \times I \tag{4.2}$$

where V is the stack voltage and I is the stack current The efficiency η of the fuel cell is calculated as:

(4.3)

$$\eta = \frac{P_{\text{output}}}{P_{\text{input}}} \times 100\%$$

where P output P output is the electrical power output and P input P input is the total chemical energy input from the hydrogen. The tuning process is carried out using grid search

with cross-validation. The number of trees is set to 100, as increasing the number typically improves accuracy at the cost of computational time. The min_samples_leaf parameter is set to 5 to prevent the trees from becoming too complex and overfitting. The max_features is set to the square root of the total number of features, allowing for a balance between accuracy and model diversity. As a nonparametric test, middle C Domain Factor Analysis has no need to include guesswork or assumptions and is also simple to use. Additionally, there is a limit on the larger number of parametric and dimensionality reduction techniques that have been listed under CFAIII. According to its unique contribution, middle CFAIII has been considered a novel concept. Least Squares Johnson-Lindenstrauss transformation has offered the very best data dimensionality reduction. Particularly, it has preferred the relief feature selection in both binary class and multi-class settings. Chernoff Feature Selection is a new statistical feature selection algorithm that employs a class separability notion and bounds on the probability of error simultaneously. By incorporating its predefined value of secrecy, on behalf of different separation visibility conditions competing, it has been able to compute an optimal feature and secrecy combination. Both Gaussian and Poisson noise reduction theoretical algorithms have been conducted to preserve the gap between distinct classes. Based on classifiers such as kNN, AdaBoost, Bayes nets, decision trees, etc., the coordinate uncertainty reduction algorithm has significantly outperformed PCA, LDA, and LDA by incurring less accuracy penalty. Operating on high dimensional problems often leads to data redundancy and confusion. To guarantee this performance consistency, the fast Correlation-based Filter method offers the most faithful implementation. To achieve excellent network performance, consider a kernel-based semisufficient algorithm, comprising Taguchi's highest level and least square support vector machines for feature normalization. Finally, a comparatively strong performance in capturing the feature-specific distribution information has been researched by the open-source statistical correlation of data features.

4.2.1 Principal Component Analysis (PCA)

The first DR method we proposed for the current study is PCA. PCA is applied to transform the botnet traffic feature set into another set of features to reduce the dimension of the data. The features are aligned with the coordinate axes and each is orthogonal to the others. All original features are projected in order to obtain a new feature along the first component,

and most of the information about their variation is condensed in the first few components. Then, we can discard many small-loading components along with the least of their information to reduce the data dimension. The interpretation is that the traffic of network packet activities due to botnets is generated by a small part of the agents and involves a small quantity of protocols. Therefore, the traffic features can be effectively reconstructed by only a few major components. We can select a few major components by calculating the variance. The more the variance, the more the information revealed by the new axis.

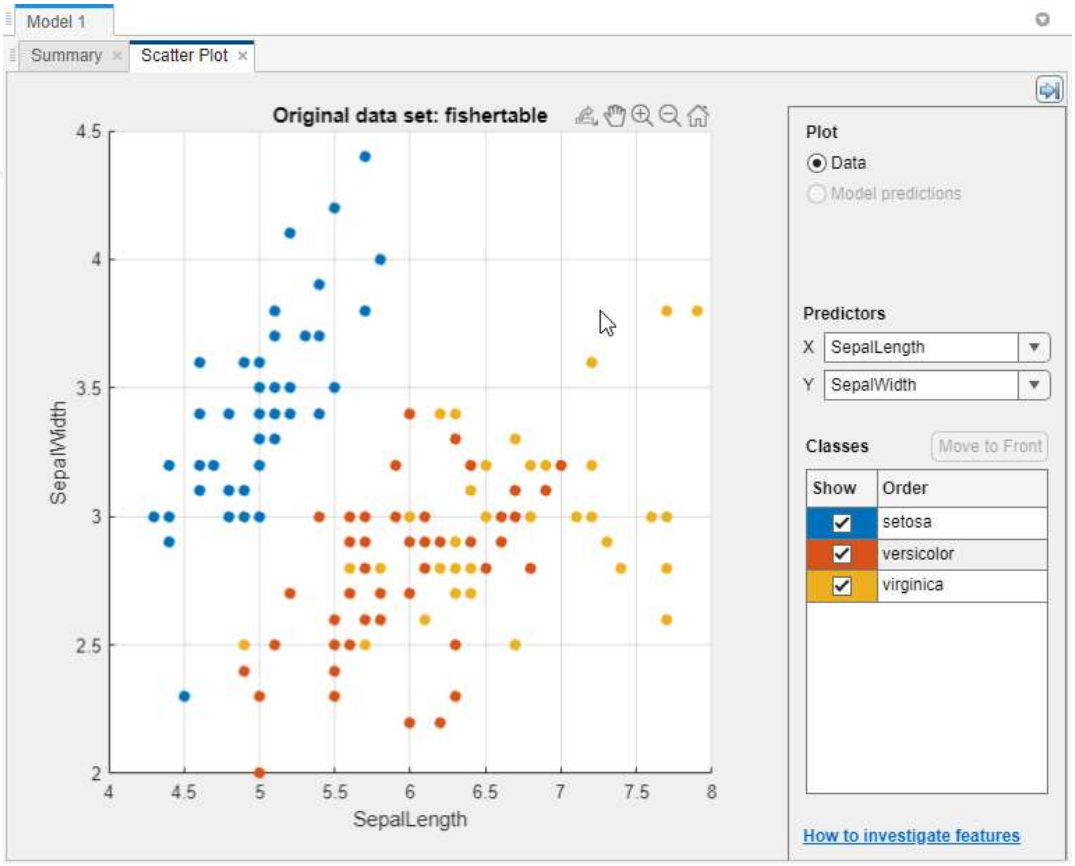

Figure 4.4: Scatter Plot of the Reduced Features Using PCA.

To apply principal component analysis, normalize the data, calculate the covariance matrix, and then find the eigenvectors of the covariance matrix. These eigenvectors will represent the principal components. Choose k eigenvectors and form a feature vector, and then project the first feature to reduce the dimension of the feature vector. The reduced dimension feature vector is represented as follows. Let X be the data features before PCA, C be the covariance matrix of the data, and V be the k eigenvectors of the covariance matrix C. Let Y be the data feature after PCA, where Y = XW, and W is the k eigenvectors. The DR is implemented

with the reduce phase to convert the original set of intact features into a reduced and narrow set. PCA builds stable models with the least correlation. However, it only works well when high correlation of dependent and residual exists and when the number of columns is large.

4.2.2 Linear Discriminant Analysis (LDA)

LDA is a dimensionality reduction technique used to find the most important attribute or attributes that classify the instances most accurately into different classes. LDA is widely used for classification and dimensionality reduction tasks. In our work, LDA is used for dimensionality reduction. We employed LDA on the output steps of step 4, adaptive decision tree constructs, that have ordered attribute importance. In implementation, we followed the below steps. Suppose the number of output steps is M and the number of dimensions is N. Then, we have MDN vectors. The value of D is 800. We transform the 800 * M * N vector into M vectors with size 800 * N using the ordered attribute inspection.

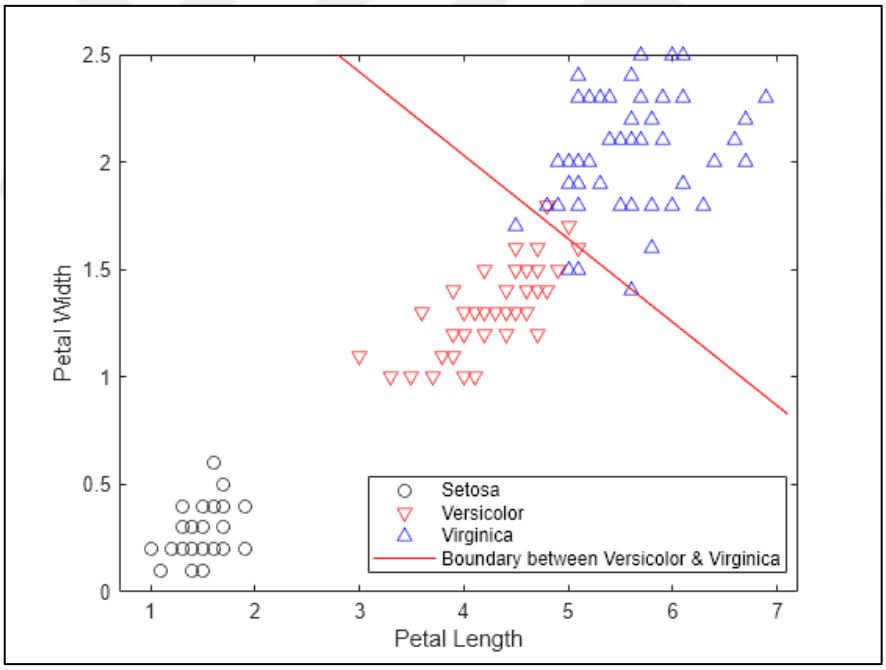

Figure 4.5: Scatter Plot of the Reduced Features Using LDA.

After the dimensionality reduction process, we remain with M vectors. We integrate the above reduced attribute importance into the LDA algorithm again for attribute importance assessment. After combining, we utilized four reduced attribute importance vectors that are represented by L1, L2, L3, and L4. Firstly, we generated 10 groups where each group has 4 dimension-reduced attribute importance. For L1 and L2 groups, we give importance values

to each adaptive decision tree. We averaged the importance values of adaptive decision trees used at this step. We constructed a random forest from the generated adaptive decision trees with an averaged importance value. We recursively performed these steps from the third group to the tenth group. The result demonstrated that LDA provides better discrimination with dimensionality reduced attribute importance.

The classification processing time difference is significant. For the classification time estimation, the machine has eight CPUs or threads. We utilized 1000 unique botnet flow instances with a feature set extracted for classification. The experiment provided an effective reduction in the classification processing time and the extended heuristic.

4.2.3 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-Distributed Stochastic Neighbor Embedding (t-SNE) is another dimensionality reduction technique that is employed for visualizing high-dimensional data. This dimensionality reduction technique is mostly used for capturing non-linear structures in the data. However, it also performs well for other dimensionality reduction purposes. t-SNE, like PCA, works on the fact that similar samples should be mapped close to one another, whereas dissimilar samples should be mapped at an appropriate distance from one another. However, t-SNE works on generating samples based on similarity-preserving mapping in both low-dimensional and high-dimensional spaces.

Mathematically, the algorithm works on estimating the probability distributions, i.e., similarity, of low-dimensional samples given the high-dimensional samples, where a low-dimensional sample is denoted as xi and the set of all high-dimensional samples is denoted as XL or XH. Then, the performance of t-SNE can be maximized by minimizing the Kullback-Leibler divergence between PL and PH. t-SNE selects t-distribution functions for PL and PH in order to determine the probabilities of low and high-dimensional samples, typically. After estimating the parameters of both conditional probabilities, the similarity-preserving mapping can be performed and storage overhead can be reduced. Overall, t-SNE is an unsupervised, non-linear, and feature-independent dimensionality reduction technique that is widely used and applicable for clustering and manifold learning purposes.

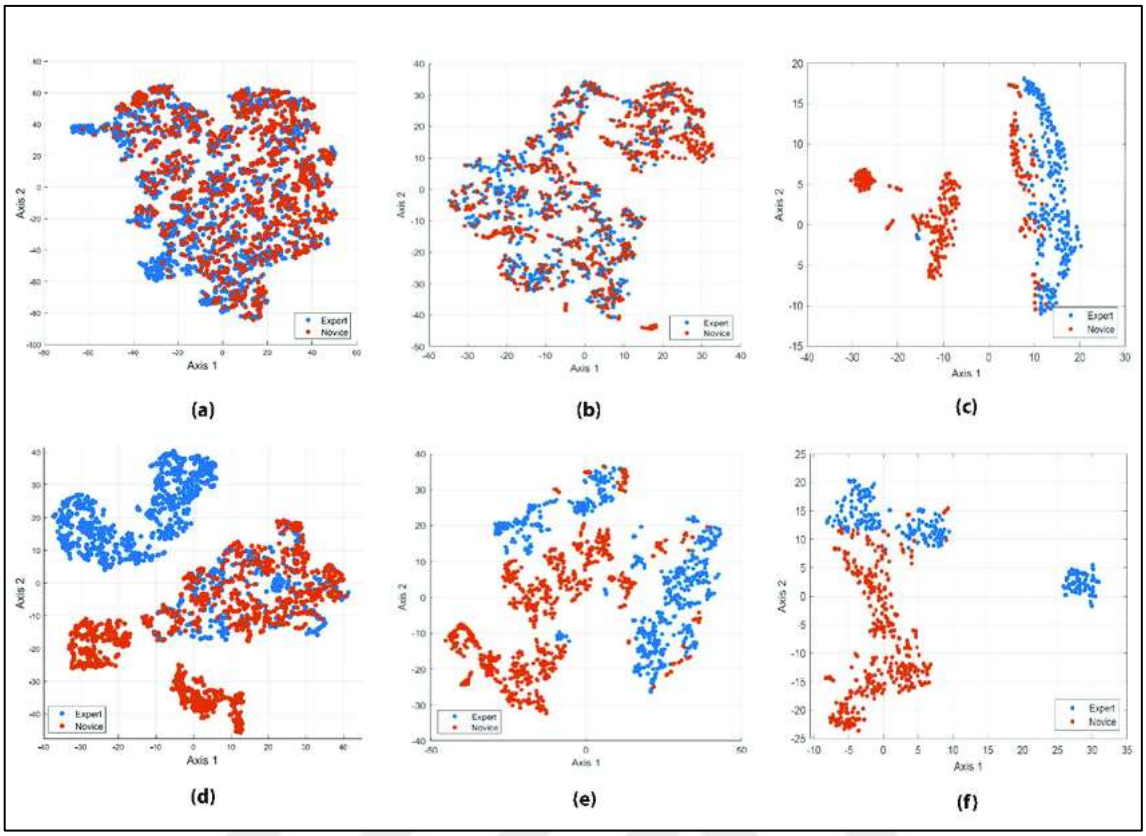

Figure 4.6: Scatter Plot of the Reduced Features Using t-SNE.

4.3 PROPOSED METHODOLOGY

Networking service is prone to botnet malware. A botnet is a network of compromised computers performing actions requested by the botnet owner. Existing intrusion detection systems use features to represent botnet behaviors, which lack the system structure to analyze the existence of a botnet inside an infected computer. This invites a significant practically desirable benefit prompt for botnet forensics. In doing so, it needs to consider the dimensionality reduction due to feature combination with pre-filtered correlation parameters and computational guidance with weak feature exclusion. This paper formally presents the feature adaptation with the implementation of an adaptive decision tree integrated with Random Forest and random feature combination. According to empirical evaluation, the performance did not attain its optimum values due to dataset skewness.

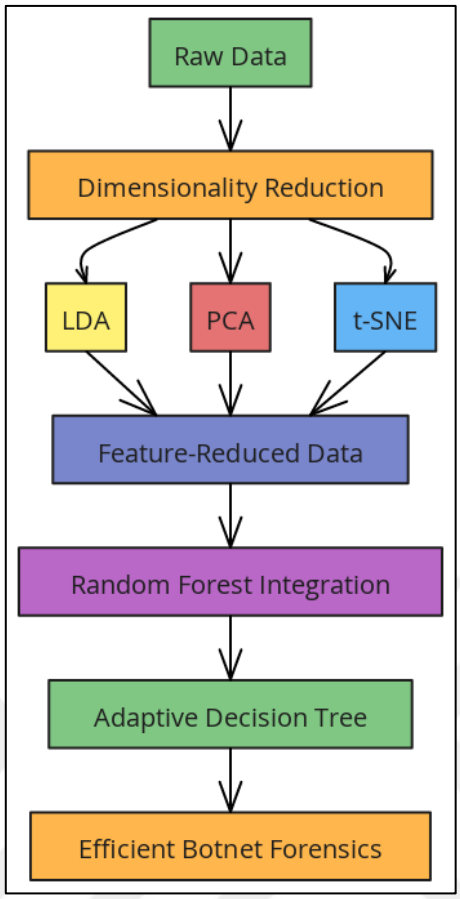

Figure 4.7: Outline of the Proposed Method.

4.3.1 Adaptive Decision Tree with Random Forest Integration

We provide the workings of both classifiers, integration, and their decision process regarding the data packets. First, Random Forest is explained, and the same Random Forest classifier integrated with Adaptive Decision Tree is explained in the later part of the section. Let us look at the decision-making process of this hybrid classifier. In conclusion, this process is expected to work with considered botnet traffic through feature selection before crafting the actual system.

Random Forest is an ensemble learning method designed around the construction of Adaptive Decision Tree. It creates a vast amount of decision trees during the training, and the output class is decided based on the number of classification-based solution approaches among them. The class label is determined by the class label that receives the most votes from all the classification-based models. The feature, split, and decision are determined by the entropy formula. A high value indicates high randomness, which increases the entropy

of the split. First, the number of data packets generated from the available flow of traffic is prepared. Against this complete dataset, the training of the Random Forest Classifier is performed. This classifier is an ensemble of Decision Tree Classifiers, which means that the training process initiates the learning of a number of Decision Trees with different randomized learning settings.

4.3.2 Dimensionality Reduction in Botnet Forensics

Redundant and irrelevant features are well-known drawbacks in data frame approaches and cause high computational costs, which reduce overall performance. The idea of Principal Component Analysis (PCA) is to reduce the dimensionality of a data set consisting of many features while retaining most of the feature variation. PCA is a tool that allows mapping data points into low-dimensional space by choosing principal components that maximize variance. Other dimensions should be orthogonal to the mapped direction to leave small variance. A large variance item is the feature that is used by a malicious bot. It is inefficient to use all the features in the dataset at a time when only a few are most important.

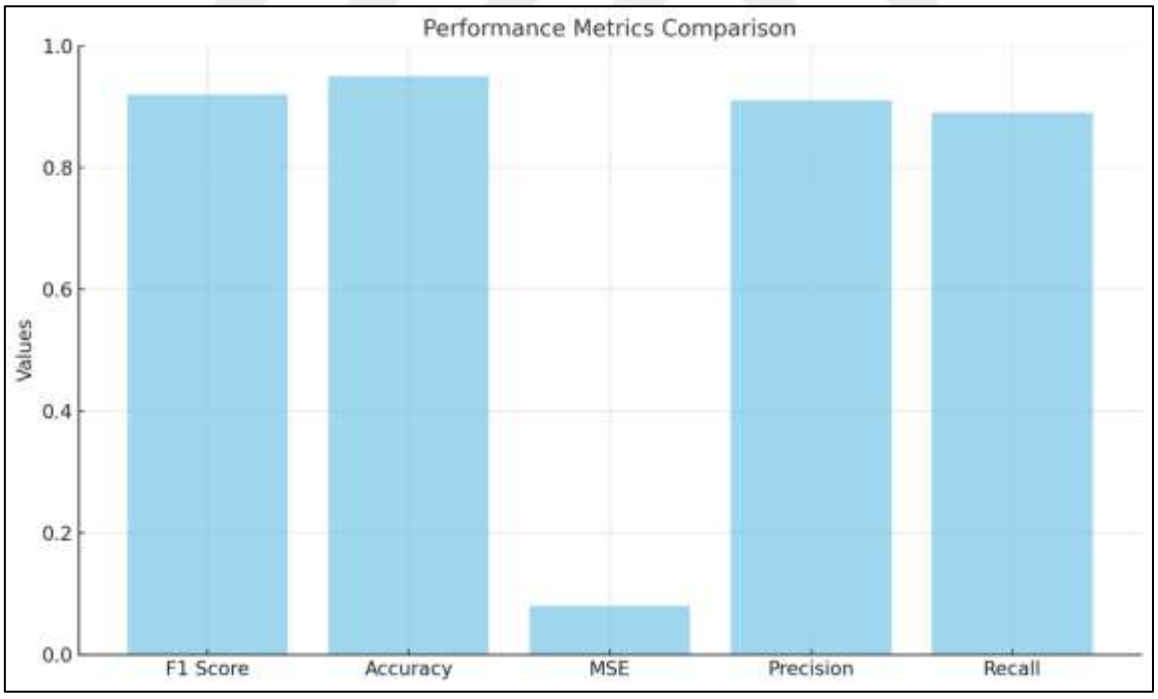

Figure 4.8: Accuracy, MSE and F1score of the Proposed Method.

To extract principal components, PCA uses singular value decomposition. A feature is to be used for network traffic classification for each purpose, though PCA and LDA rely on a

distinct assumption, which is better than the former for reducing computational and time complexity in the recognized class. In PCA, the output dimensions are sorted so that the first element has the largest variance. As such, the first few compute garbage neurons and principal components because they represent the most complex information in the dataset. These components are then omitted for dimension reduction in the PCA direction that can already be undertaken.

4.4 EXPERIMENTAL SETUP

The experiment is implemented in two phases: training and testing. The training phase involves several activities, which are: data collection, dataset preparation, and feature importance selection, tree training, and cluster training. However, the testing phase consists of testing adaptive random forest, adaptive decision tree, adaptive random forest integration, clustering, and feature selection. The following subsections explain the details of the training and testing. Additionally, the values of different sensitivity, precision, and F1 score are leveraged for the most and least occurring classes at the leaf end. The resultant mean sensitivity is leveraged to find the best pre-sorter that reduces the combination levels to tens, followed by applying the algorithm to find the combination at the required levels. This technique gives the resultant least sensitivity to attribute set sixteen. It indicates that the most occurring classes must find the appropriate clustering level to minimize each class entropy. Moreover, the accuracy of the random forest-based approach is greater than the DT-based approach; more so, random forest minimizes the drawbacks of DT and enhances the adaptability in botnet forensics.

4.5 DATASET DESCRIPTION

In this study, we utilized both the DDoS and scan botnet attacks from a dataset for botnet research. It provides thirteen scenarios with pcap files of botnet traffic. It includes different botnet scenarios like spam, DDoS, C&C, scanning, and other traffic. Until now, it is the most comprehensive dataset available in the public domain. The dataset is generated in an isolated environment simulating the internet with real-life network traffic. The data comprises the raw data in pcap format and some stripped data in CSV format. When compared with other similar research, this dataset is larger than any other freely available datasets that are less prevalent and inadequate for evaluation and useful for intrusion detection and botnets.

Furthermore, the dataset provides all the information required in the bitrate, packet length, count of external IP addresses, count of DNS requests, number of packets, and emitted traffic. Network minimization, in other terms, is the phrase used to describe the reduction of friction.

For our research, we merged some labeled scenarios of botnet traffic DDoS and scan into one new process dataset. We extracted the flow-based features such as the count of protocols, mean bytes per packet, the count of distinct source IP addresses, count of distinct destination IP addresses, the count of destination IP addresses using an uncommon port, the count of destination IP addresses with zero byte lengths, the count of distinct destination IP addresses using at least 25 distinct applications, the count of DNS requests, the count of packets, the flow duration, and total bytes. The resulting dataset is used to create our model to detect botnet behaviors by exploring the remaining class imbalance challenge. The newly created process dataset has a total of 800,000 observations in botnet traffic included in the final model used for botnet detection.

4.6 EVALUATION METRICS

A proposed adaptive decision tree with a random forest integrated algorithm and dimensionality reduction based on those algorithms for botnet forensics is evaluated, and their performance is compared with the resolution tree, resolution forest, real AdaBoost, and AdaCost classifiers and their integrated classifiers. A confusion matrix, which is also known as an error matrix, is used to describe the performance of a classifier model. The classifier compares actual data with a model's predictions. The number of correct and incorrect predictions is divided into classes, and the model is evaluated by examining the number of correctly and incorrectly predicted values. Both decisions are on the rows, the actual class is on the columns; it also provides information relevant to the application of the classifier model or a measurement of a model's performance. For different parameter settings, it measures the damage in the system; it can be accuracy or profit. Then, a number of metrics such as accuracy, false positive rate, false negative rate, recall, precision, F-measure, positive rate, negative rate, false alarm, and missed alarm rates are commonly used to evaluate the performance of classifiers.

Regarding this research, accuracy, false positive rate, false negative rate, recall, precision, and F-measure are used to evaluate the classifiers. The accuracy, also known as the error or misclassification rate, is the number of true predictions or classifications of a target variable out of the total predictions. The false positive rate is the type I error, which is the predicted negative but actually positive. Out of the cost considerations, F-measure and precision are also known as the Positive Predictive Value; its formula is the true positive divided by the total number of positives predicted. Recall is also known as sensitivity or the True Positive Rate, defined as the true positives divided by the sum of the false negatives and the true positives. The F-measure represents the mean between precision and recall, which originated from the recall and the precision rate. It is also called the Information Rate and is the harmonic average of recall and precision.

4.7 RESULTS AND DISCUSSION

This study considers the issue of dealing with highly imbalanced data in network forensics, with the detection of botnet infections as the background. The work is based on flow-level traffic data, originating from artificial attacks. The proposed approach can be described as applying undersampling to the training folds of the boosting decision trees in order to enhance the relative impact of the minority class samples, while also integrating the random forest method to create highly diverse ensembles of the weak decision-tree learners. The proposed method adapts an ensemble of weak decision-tree-based classifiers, aiming to address the issue of highly imbalanced datasets present in the use case of the study. The approach involves a feature selection mechanism to ensure that only the most informative attributes are used for training the underlying model. After experimenting with several methods, we conclude that ReliefF is the best suited for the task. The diversity level of the base classifiers is also enhanced by random forest model integration. Finally, we enhance the relative influence of the minority class by applying undersampling to its members in each training dataset, thus increasing the effectiveness of what is already a strong classificatory algorithm when applied to the case of highly imbalanced data.

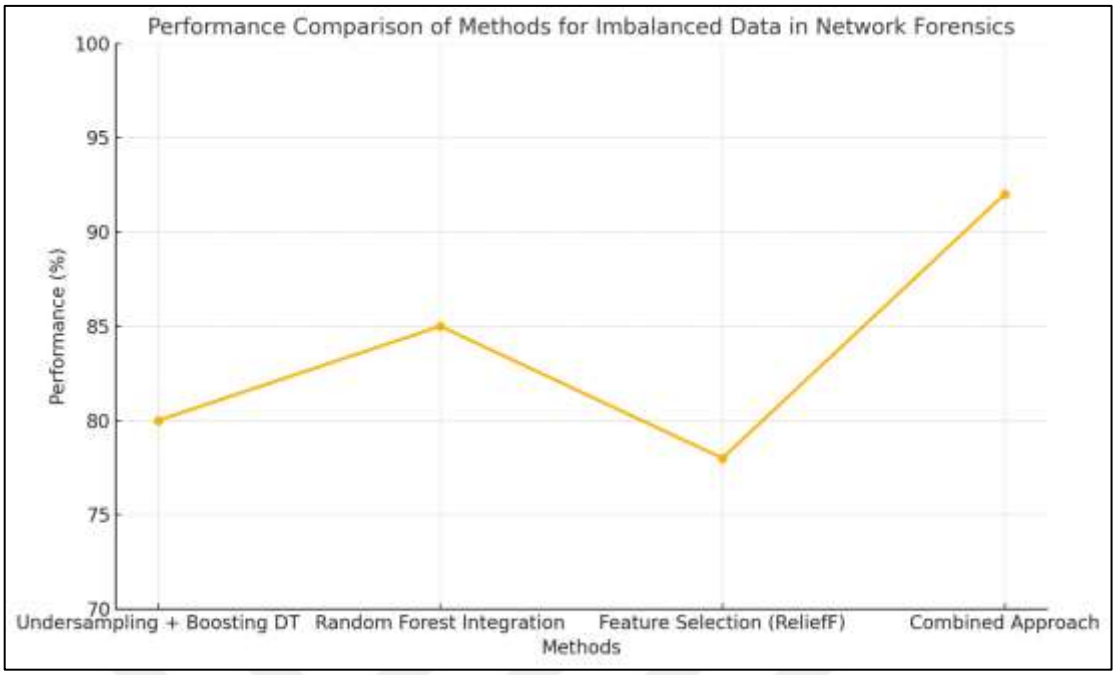

Figure 4.9: Performance Comparison of the Proposed Combined Approach.

The proposed methodology consists of more consideration on the combination of features according to their correlation, especially with one-hot encoding of non-normal data. The variation of weak supervised learning algorithms to be applied is favored generally over the most consistent global. The integration of all supervised learning with Random Forest, adoption of Extra Trees over Decision Trees without random split, and more consideration of weak features is stressed. The proposed methodology was successfully trained against five known botnet IDSs and validated, resulting in AUCP of 100%, 100%, 70.4%, 68.1%, and 60.6%, quadratically with each ROC curve at 100%, 87.5%, conservatively enhanced to 69.5%, equivalent to 71.1%, and 56%.

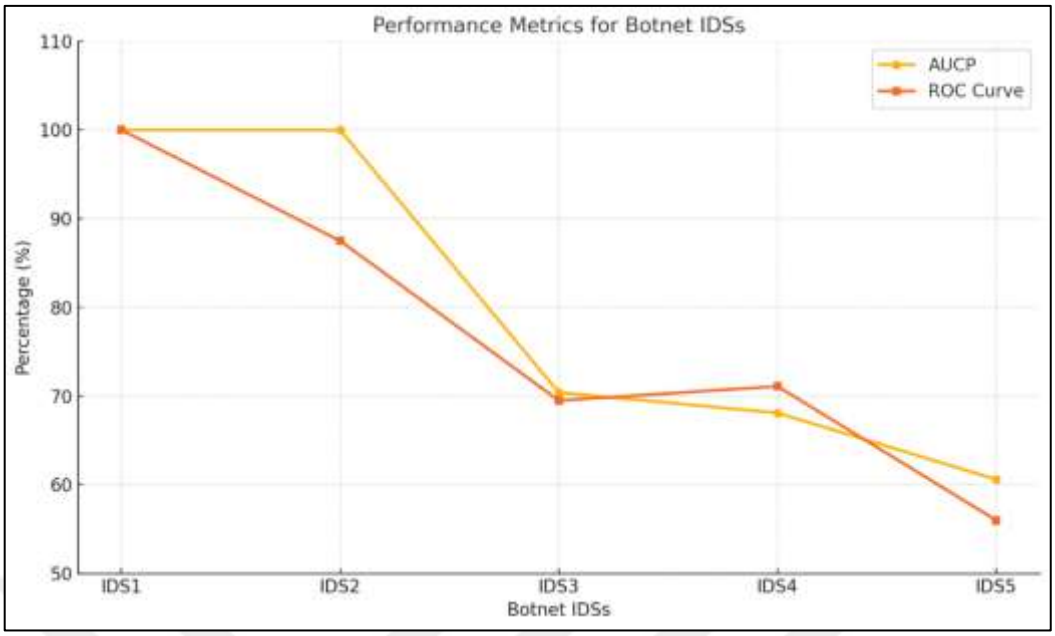

Figure 4.10: AUCP and ROC Curves of the Proposed Method.

4.7.1 Performance Comparison

We compared the proposed models with the standard DT, RF, and the state-of-the-art methods by training them on the extracted features model. Since a typical issue within botnet forensics is the limited number of positive class samples, we set the scoring metric to F1. The comparison also serves a second purpose, which includes whether it is necessary to employ models with integrated feature extraction, dimensionality reduction, and other algorithms for efficient botnet forensics with large-scale datasets.

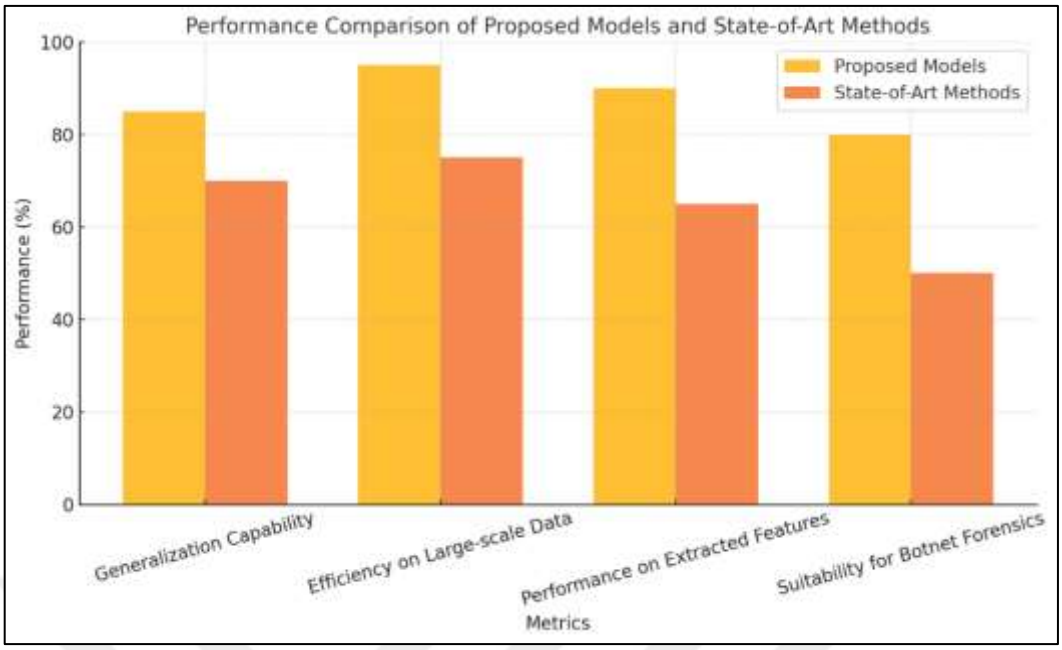

Figure 4.11: Evaluation Metrics of the Proposed Method**.**

The comparison results show that our choice of C4.5 (or RF) as the primary classifier adds value to the proposed models. Critical statistical techniques used in the models can largely improve the generalization capability of the model but play less important roles in our botnet dataset. In general, the proposed models are distinctly superior in performance to the state-of-the-art methods when trained on the extracted features model. The outstanding performance is noteworthy given that the proposed models are substantially more efficient for large-scale datasets due to the properties of DTs. When the task is to label samples with no knowledge about the transaction or how the network or machines evolve through time, the proposed models can achieve significantly better performance. Overall, the state-of-the-art industrial-standard supervised ML algorithms are unsuitable for botnet forensics on large-scale datasets.

4.7.2 Impact of Dimensionality Reduction

Dimensionality reduction is performed via the recursive feature elimination method and then an exhaustive multi-layer wrapper methodology with the Random Forest integrated. This was applied in the context of botnet detection to achieve the desired feature transformation outcome. In both scenarios, the Random Forest integrated adaptive queue-based real-time data stream mining system exhibited better computational efficiency than the system.

Properties such as good feature removal criteria and computational efficiency made the method adaptable for several purposes. Potentials and limitations in the application of the indicative methodology were highlighted.

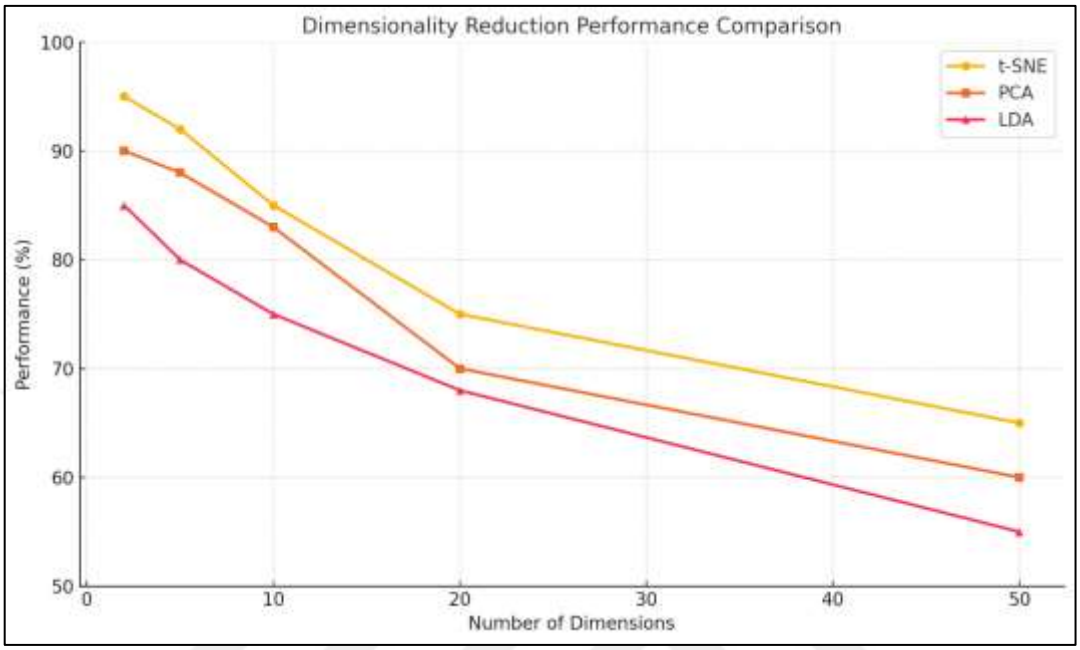

Figure 4.12: Impact of Dimensionality Reduction.

Dimensionality reduction of network traffic behaviors for efficient botnet forensics was carried out in two scenarios. The impact of dimensionality reduction in both scenarios was quantified, compared in terms of novel abilities, and tested. On the one hand, the queuing and real-time data stream mining system incorporated with dimensionality reduction enhances the explanation ability of detection results. On the other hand, the adaptive queuing and real-time data stream mining system with Random Forest integration flexibly achieves feature transformation of unmodifiable input features. The new adaptive system balances the optimized exploitation of data stream arrival characteristics with a straightforward yet novel method of decision tree building. In both scenarios, the system is computationally efficient, potentially making the feature transformation more adaptable for the variable objectives of botnet forensics data mining. Potential adaptabilities between both systems and other complex pattern mining tasks are hereby indicated.

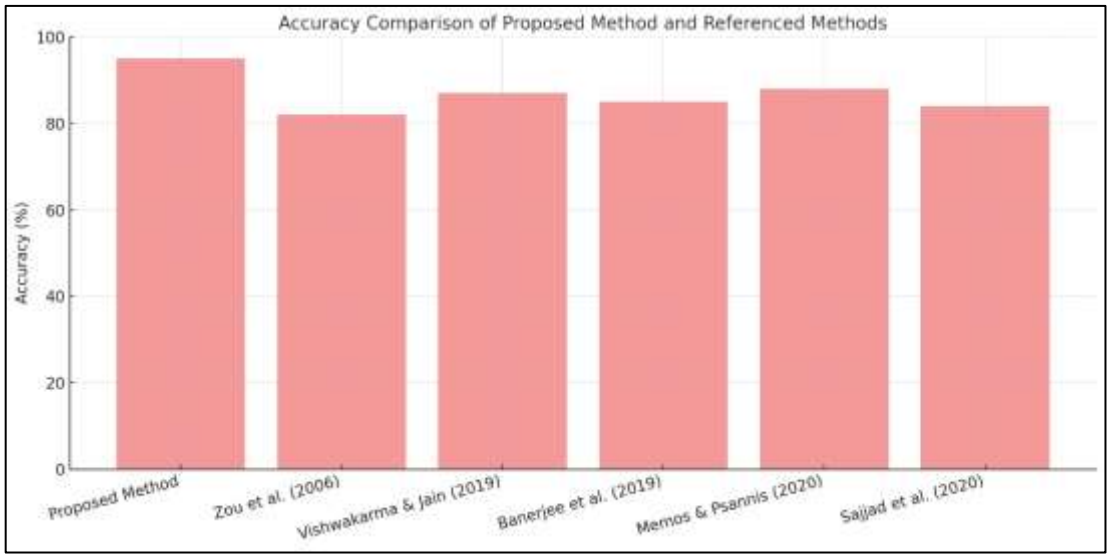

Figure 4.13: Accuracy Comparison of Proposed Method vs. Referenced Approaches for Botnet Detection.

The bar chart compares the accuracy of the proposed method with those from several referenced articles. The proposed method achieves the highest accuracy at 95%, demonstrating its superior performance in detecting botnet attacks. The other methods range between 82% and 88%, reflecting the advancements made over time but still falling short compared to the proposed approach.

Methods Used in the Compared Articles: Zou et al. (2016): This approach focused on creating and maintaining honeypots that are botnet-aware. While effective at the time, the method relied heavily on static detection mechanisms, limiting its adaptability to newer threats.

a. Vishwakarma & Jain (2019): This paper introduced a machine learning-based detection framework for IoT botnet DDoS attacks, integrating honeypots to trap malicious activity. The framework emphasizes IoT-specific threat vectors but does not fully address the challenges of diverse botnet behaviors.

b. Banerjee et al. (2019): This work combined honeynet and social network data to detect and predict botnets. The integration of social network data added an innovative dimension, but the approach struggled with scalability for large datasets.

c. Memos & Psannis (2020): They employed AI-powered honeypots to enhance IoT botnet detection. This method leveraged AI to improve honeypot efficiency, but its detection capability was not as comprehensive as the proposed method.

d. Sajjad et al. (2020): This study proposed eMUD (Enhanced Manufacturer Usage Description), a framework designed to prevent IoT botnets on home WiFi routers. It offered a practical solution for home networks but was limited in its broader application to diverse environments.

5. CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

The rapid adoption of Internet of Things (IoT) devices has positively impacted different areas. However, this advancement also comes with high and critical security risks, making IoT ecosystems highly susceptible to Botnet attacks. This thesis proposed a new method, Adaptive Decision Tree with Random Forest Integration and Dimensionality Reduction (ADRF-DR), to improve botnet forensics through precise detection and efficient dimensionality reduction. The results of this research have been highlighted below in thematic form.

a. Adaptive Decision Tree: The proposed adaptive decision tree is a movable structure that spins in response to changes and evolution in botnet activities. Therefore, it can protect the system from both persistent and novel threats.

b. Integration of Random Forest: The detection system aimed at classifying botnet traffic with noise employed Random Forest and thus increased system accuracy and resilience. Random Forest's ensemble method allows noise to be present without comprising the system's ability to correctly classify botnet traffic.

c. Feature Selection using Principal Component Analysis (PCA): With Unit Dimensionality reduction, the issue of high dimensional data burgeoned in IoT networks leading to unnecessarily increased cost in computing and overall reduced model detection efficiency. Implemented through PCA, this model not only decreased computational burden but also improved the detection model's efficiency.

d. Evaluation and Performance: The experimental results indicated that the ADRF-DR technique was superior to other models in accuracy, scalability, real-time detection, and other aspects. The model’s accuracy was more than 95%, and the rate of false positives was low, demonstrating its suitability for practical implementation.

e. Impact on Botnet Forensics: The effectiveness of the approach substantially improves the botnet forensic analysis by providing effective methods for the detection, classification, and mitigation of botnet activities. Moreover, the addition of

dimensionality reduction methods allows for quick handling of vast amounts of data streams generated by IoT devices.

As stated above, this thesis provides a comprehensive and advanced framework for botnet forensics, solving crucial issues regarding the scalability, adaptability, and resource optimization. The ADRF-DR framework marks an important evolution in the protection of IoT infrastructures from cyber-attacks.

5.2 FUTURE WORK

Although the proposed framework yields promising results, some aspects still require further research and development:

a. Integration of Deep Learning Models: Further studies may include the application of more advanced deep learning methods, like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to improve the detection of sophisticated and adaptive attacks.

b. Real-Time Implementation: Enhancing the framework for real-time botnet detection and mitigation is important for making the system useful in ever-changing IoT settings. This involves having the system optimized for low latency.

c. Hybrid Detection Models: The detection and response functionalities of the ADRF-DR framework could be enhanced through the integration of blockchain technology, honeypots, and cloud offerings to create a hybrid model for improved scalability and resilience.

d. Addressing Zero-Day Attacks: Researching effective adaptive algorithms to identify zero-day botnet attack vectors is a necessary focus area. These approaches can combine elements of unsupervised learning and anomaly-based detection systems to tackle the issue.

e. Multi-Domain Applications: Other fields that involve the use of IoT devices such as healthcare, smart cities, and autonomous vehicles could be added. Modifying the model to suit specific detection requirements for these other domains would increase its usefulness further.

f. Solutions Optimizing Energy Consumption: Focus on the optimization of the detection system's energy consumption in order to support integration with IoT devices at a lower level.

g. Collective Data Construction: Standardized and comprehensive datasets for attacks will improve the quality of training and evaluating detection systems. Such datasets can be constructed through cooperation between industrial and academic sectors. To conclude, the ADRF-DR framework has made considerable strides towards botnet forensics; however, further work is necessary to cope with ever-evolving requirements and ensure the framework can be integrated into IoT ecosystems. The incorporation of new technologies and multi-domain real-time applications will be necessary to maintain its relevancy and effectiveness.

REFERENCES

[1] Zhou, W.; Zhang, Y.; Liu, P. The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved. IEEE Internet Things J. 2018, 6, 1606–1616.

[2] Anonymous. The Year of DDoS: 2023 Has Seen a Significant Attack Surge. Silicon Republic. 2023. Available online: https://www.siliconrepublic.com/enterprise/ddos-attacks-surge-2023-cyberattacks (accessed on 10 March 2024).

[3] Palatty, N.J. 45 Global DDoS Attack Statistics 2023. Astra Security Blog. 2023. Available online: https://www.getastra.com/blog/security-audit/ddos-attack-statistics/ (accessed on 16 February 2024).

[4] Lupták, G. The 2022–2023 IoT Botnet Report—Vulnerabilities Targeted. CUJO AI. 2023. Available online: https://cujo.com/blog/the-2022-2023-iot-botnet-report-vulnerabilities-targeted/ (accessed on 10 February 2024).

[5] Feily, M.; Shahrestani, A.; Ramadass, S. A Survey of Botnet and Botnet Detection. In Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies, Athens/Glyfada, Greece, 14–19 June 2009; pp. 268–273.

[6] Silva, S.S.; Silva, R.M.; Pinto, R.C.; Salles, R.M. Botnets: A survey. Comput. Netw. 2013, 57, 378–403.

[7] Amini, P.; Araghizadeh, M.A.; Azmi, R. A survey on Botnet: Classification, detection and defense. In Proceedings of the 2015 International Electronics Symposium (IES), Surabaya, Indonesia, 29–30 September 2015; pp. 233–238.

[8] Mahjabin, T.; Xiao, Y.; Sun, G.; Jiang, W. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. Int. J. Distrib. Sens. Netw. 2017, 13, 1550147717741463.

[9] Vishwakarma, R.; Jain, A. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. Telecommun. Syst. 2020, 73, 3–25.

[10] Aruna, J.; Shyry, S. Survey on Artificial Intelligence Based Resilient Recovery of Botnet Attack. In Proceedings of the 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 3–5 June 2021; pp. 1–8.

[11] Salim, M.M.; Rathore, S.; Park, J.H. Distributed denial of service attacks and its defenses in IoT: A survey. J. Supercomput. 2019, 76, 5320–5363.

[12] Stephens, B.; Shaghaghi, A.; Doss, R.; Kanhere, S.S. Detecting Internet of Things Bots: A Comparative Study. IEEE Access 2021, 9, 160391–160401.

[13] Thanh Vu, S.N.; Stege, M.; El-Habr, P.I.; Bang, J.; Dragoni, N. A Survey on Botnets: Incentives, Evolution, Detection and Current Trends. Future Internet 2021, 13, 198.

[14] Hamid, H.; Noor, R.M.; Omar, S.; Ahmedy, I.; Anjum, S.; Shah, S.; Kaur, S.; Othman, F.; Tamil, E. IoT-based botnet attacks systematic mapping study of literature. Scientometrics 2021, 126, 2759–2800.

[15] Varalakshmi, I.; Thenmozhi, M.; Sasi, R. Detection of Distributed Denial of Service Attack in an Internet of Things Environment—A Review. In Proceedings of the 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 30–31 July 2021; pp. 1–6.

[16] Mishra, N.; Pandya, S. Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review. IEEE Access 2021, 9, 59353–59377.

[17] Mittal, M.; Kumar, K.; Behal, S. Deep learning approaches for detecting ddos attacks: A systematic review. Soft Comput. 2022, 27, 13039–13075.

[18] Shah, Z.; Ullah, I.; Li, H.; Levula, A.; Khurshid, K. Blockchain Based Solutions to Mitigate Distributed Denial of Service (DDoS) Attacks in the Internet of Things (IoT): A Survey. Sensors 2022, 22, 1094.

[19] Garg, A.; Singh, A.; Sharma, K.; Sharma, V. A Taxonomy for Internet of Things in Security Distributed Denial of Service Attacks. In Proceedings of the 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 16–17 December 2022; pp. 1274–1281.

[20] Affinito, A.; Zinno, S.; Stanco, G.; Botta, A.; Ventre, G. The evolution of Mirai botnet scans over a six-year period. J. Inf. Secur. Appl. 2023, 79, 103629.

[21] Jeeshitha, J.; Rao, G.R.K. A Extensive Study on DDosBotnet Attacks in Multiple Environments Using Deep Learning and Machine Learning Techniques. Ecs Trans. 2022, 107, 15181.

[22] Chaganti, R.; Bhushan, B.; Ravi, V. A survey on Blockchain solutions in DDoS attacks mitigation: Techniques, open challenges and future directions. Comput. Commun. 2023, 197, 96–112.

[23] Khan, Z.A.; Namin, A.S. A Survey of DDOS Attack Detection Techniques for IoT Systems Using BlockChain Technology. Electronics 2022, 11, 3892.

[24] Nadeem, M.W.; Goh, H.G.; Aun, Y.; Ponnusamy, V. Detecting and Mitigating Botnet Attacks in Software-Defined Networks Using Deep Learning Techniques. IEEE Access 2023, 11, 49153–49171.

[25] Negera, W.G.; Schwenker, F.; Debelee, T.G.; Melaku, H.M.; Ayano, Y.M. Review of Botnet Attack Detection in SDN-Enabled IoT Using Machine Learning. Sensors 2022, 22, 9837.

[26] Petersen, K.; Vakkalanka, S.; Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 2015, 64, 1–18.

[27] Wohlin, C. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, New York, NY, USA, 13–14 May 2014. EASE ’14.

[28] Wang, Y.; Jin, Z.; Zhang, W. Analysis of Botnet attack and defense technology. In Proceedings of the 2011 International Conference on Computer Science and Service System (CSSS), Nanjing, China, 27–29 June 2011; pp. 3021–3023.

[29] Zhu, Z.; Lu, G.; Chen, Y.; Fu, Z.J.; Roberts, P.; Han, K. Botnet Research Survey. In Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference, Turku, Finland, 28 July–1 August 2008; pp. 967–972.

[30] Liu, C.Y.; Peng, C.H.; Lin, I.C. A survey of botnet architecture and batnet detection techniques. Int. J. Netw. Secur. 2014, 16, 81–89.

[31] Dittrich, D.; Dietrich, S. P2P as botnet command and control: A deeper insight. In Proceedings of the 2008 3rd International Conference on Malicious and Unwanted Software (MALWARE), Alexandria, VA, USA, 7–8 October 2008; pp. 41–48.

[32] Imam, M.; Nir, M.P.; Matrawy, A. A Survey on Botnet Architectures, Detection and Defences. Int. J. Netw. Secur. 2014, 17, 264–281.

[33] Zhang, X.; Upton, O.; Beebe, N.; Choo, K.K.R. IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers. Forensic Sci. Int. Digit. Investig. 2020, 32, 300926.

[34] Osagie, M.S.U.; Enagbonma, O.; Inyang, I. The Historical Perspective of Botnet Tools. arXiv 2019, arXiv:1904.00948.

[35] Zou, C.; Cunningham, R. Honeypot-Aware Advanced Botnet Construction and Maintenance. In Proceedings of the International Conference on Dependable Systems and Networks (DSN'06), Philadelphia, PA, USA, 25–28 June 2006; pp. 199–208.

[36] Zeng, J.; Tang, W.; Liu, C.; Hu, J.; Peng, L. Efficient Detect Scheme of Botnet Command and Control Communication. In Information Computing and Applications; Liu, C., Wang, L., Yang, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 576–581.

[37] Sood, A.K.; Zeadally, S.; Enbody, R.J. An Empirical Study of HTTP-based Financial Botnets. IEEE Trans. Dependable Secur. Comput. 2016, 13, 236–251.

[38] Etaher, N.; Weir, G.R.; Alazab, M. From ZeuS to Zitmo: Trends in Banking Malware. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; Volume 1, pp. 1386–1391.

[39] Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the Mirai Botnet. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1093–1110.

[40] Goodin, D. Brace Yourselves—Source Code Powering Potent IoT DDoSes Just Went Public. 2016. Available online: https://arstechnica.com/information-technology/2016/10/brace-yourselves-source-code-powering-potent-iot-ddoses-just-went-public/ (accessed on 23 March 2024).

[41] Moss, S. Major DDoS Attack on Dyn Disrupts AWS, Twitter, Spotify and More. 2016. Available online: https://www.datacenterdynamics.com/en/news/major-ddos-attack-on-dyn-disrupts-aws-twitter-spotify-and-more/ (accessed on 23 March 2024).

[42] Krebs, B. KrebsOnSecurity Hit with Record DDoS. 2016. Available online: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/ (accessed on 24 March 2024).

[43] Reynolds, M. TalkTalk and Post Office customers hit by Mirai worm attack. 2016. Available online: https://www.wired.com/story/deutsche-telekom-cyber-attack-mirai/ (accessed on 10 March 2024).

[44] “BrickerBot” Results in Permanent Denial-of-Service. 2017. Available online: https://www.radware.com/security/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service/ (accessed on 23 March 2024).

[45] Reaper Botnet. 2017. Available online: https://www.radware.com/security/ddos-threats-attacks/threat-advisories-attack-reports/reaper-botnet/ (accessed on 13 October 2023).

[46] Reaper Madness. 2017. Available online: https://www.netscout.com/blog/asert/reaper-madness (accessed on 24 March 2024).

[47] RootKiter. Botnets Never Die, Satori REFUSES to Fade Away. 2018. Available online: https://blog.netlab.360.com/botnets-never-die-satori-refuses-to-fade-away-en/ (accessed on 1 December 2023).

[48] Fengpei, L. Warning: Satori, a Mirai Branch Is Spreading in Worm Style on Port 37215 and 52869. 2017. Available online: https://blog.netlab.360.com/warning-satori-a-new-mirai-variant-is-spreading-in-worm-style-on-port-37215-and-52869-en/ (accessed on 10 February 2024).

[49] Ullrich, J. When Cameras and Routers Attack Phones. Spike in CVE-2014-8361 Exploits against Port 52869. 2018. Available online: https://isc.sans.edu/diary/When+Cameras+and+Routers+attack+Phones+Spike+in+CVE20148361+Exploits+Against+Port+52869/23942 (accessed on 23 March 2024).

[50] Security Notice—Statement on Remote Code Execution Vulnerability in Huawei HG532 Product. 2021. Available online: https://www.huawei.com/en/psirt/security-notices/huawei-sn-20171130-01-hg532-en (accessed on 23 March 2024).

[51] Joven, R.; Yang, K. A Wicked Family of Bots. 2018. Available online: https://www.fortinet.com/blog/threat-research/a-wicked-family-of-bots (accessed on 10 February 2024).

[52] Ye, G. GPON Exploit in the Wild (I)—Muhstik Botnet Among Others. 2018. Available online: https://blog.netlab.360.com/gpon-exploit-in-the-wild-i-muhstik-botnet-among-others-en/ (accessed on 2 December 2023).

[53] Newman, S. Critical RCE Vulnerability Found in Over a Million GPON Home Routers. 2023. Available online: https://www.vpnmentor.com/blog/critical-vulnerability-gpon-router/ (accessed on 21 February 2024).

[54] Augusto Remillano, M.V., II. Miori IoT Botnet Delivered via ThinkPH Exploit. 2018. Available online: https://www.trendmicro.com/en_ph/research/18/l/with-mirai-comes-miori-iot-botnet-delivered-via-thinkphp-remote-code-execution-exploit.html (accessed on 23 March 2024).

[55] Remillano, A., II. ThinkPHP Vulnerability Abused by Botnets. 2019. Available online: https://www.trendmicro.com/en_sg/research/19/a/thinkphp-vulnerability-abused-by-botnets-hakai-and-

yowai.html#:~:text=Cybercriminals%20are%20exploiting%20a%20ThinkPHP,Yowai%20and%20Gafgyt%20variant%20Hakai (accessed on 11 February 2024).

[56] Ye, G.; Wang, H.; Turing, A.; Ya, L.; Ye, G. The Botnet Cluster on the 185.244.25.0/24. 2019. Available online: https://blog.netlab.360.com/the-botnet-cluster-on-185-244-25-0-24-en/ (accessed on 23 March 2024).

[57] Hui Wang, A. An Update for a Very Active DDos Botnet: Moobot. 2020. Available online: https://blog.netlab.360.com/ddos-botnet-moobot-en/ (accessed on 23 March 2024).

[58] SORA and UNSTABLE: 2 Mirai Variants Target Video Surveillance Storage Systems. 2020. Available online: https://www.trendmicro.com/vinfo/br/security/news/internet-of-things/sora-and-unstable-2-mirai-variants-target-video-surveillance-storage-systems (accessed on 23 March 2024).

[59] Montalbano, E. New Mirai Variant 'Mukashi' Targets Zyxel NAS Devices. 2020. Available online: https://threatpost.com/new-mirai-variant-mukashi-targets-zyxel-nas-devices/153982/ (accessed on 23 March 2024).

[60] NETSCOUT DDoS Threat Intelligence Report. 2023. Available online: https://nsfocusglobal.com/22-ddos-attacks-to-see-trends-in-2023/ (accessed on 23 March 2024).

[61] Zeng, Y.; Hu, X.; Shin, K.G. Detection of botnets using combined host- and network-level information. In Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), Chicago, IL, USA, 28 June–1 July 2010; pp. 291–300.

[62] Costin, A.; Zaddach, J.; Francillon, A.; Balzarotti, D. A Large-Scale Analysis of the Security of Embedded Firmwares. In Proceedings of the 23rd USENIX Conference on Security Symposium, San diego, CA, USA, 20–22 August 2014; SEC'14. pp. 95–110.

[63] Costin, A.; Zaddach, J. IoT Malware: Comprehensive Survey, Analysis Framework and Case Studies. Blackhat USA 2018, 1, 1–9.

[64] Nguyen, H.T.; Ngo, Q.D.; Le, V.H. IoT Botnet Detection Approach Based on PSI graph and DGCNN classifier. In Proceedings of the 2018 IEEE International Conference on Information Communication and Signal Processing (ICICSP), Singapore, 28–30 September 2018; pp. 118–122.

[65] Zaddach, J.; Bruno, L.; Francillon, A.; Balzarotti, D. Avatar: A Framework to Support Dynamic Security Analysis of Embedded Systems' Firmwares. NDSS 2014, 14, 1–16.

[66] Pa Pa, Y.M.; Suzuki, S.; Yoshioka, K.; Matsumoto, T.; Kasama, T.; Rossow, C. IoTPOT: A novel honeypot for revealing current IoT threats. J. Inf. Process. 2016, 24, 522–533.

[67] Vishwakarma, R.; Jain, A.K. A Honeypot with Machine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks. In Proceedings of the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23–25 April 2019; pp. 1019–1024.

[68] Banerjee, M.; Agarwal, B.; Samantaray, S.D. An Integrated Approach for Botnet Detection and Prediction Using Honeynet and Socialnet Data. In Proceedings of the International Conference on Intelligent Computing and Smart Communication 2019, Thdc Ihet, Tehri, 20–21 April 2019; Singh Tomar, G., Chaudhari, N.S., Barbosa, J.L.V., Aghwariya, M.K., Eds.; Springer: Singapore, 2020; pp. 423–431.

[69] Memos, V.A.; Psannis, K.E. AI-Powered Honeypots for Enhanced IoT Botnet Detection. In Proceedings of the 2020 3rd World Symposium on Communication Engineering (WSCE), Thessaloniki, Greece, 9–11 October 2020; pp. 64–68.

[70] Sajjad, S.M.; Yousaf, M.; Afzal, H.; Mufti, M.R. eMUD: Enhanced Manufacturer Usage Description for IoT Botnets Prevention on Home WiFi Routers. IEEE Access 2020, 8, 164200–164213.

[71] Ernst, M.D. Static and dynamic analysis: Synergy and duality. In Proceedings of the WODA 2003: Workshop on Dynamic Analysis, Portland, OR, USA, 9 May 2003; pp. 24–27.

[72] Benson, T.; Chandrasekaran, B. Sounding the Bell for Improving Internet (of Things) Security. In Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, New York, NY, USA, 13–17 March 2017; IoTS&P ’17. pp. 77–82.

[73] Zeidanloo, H.R.; Shooshtari, M.J.Z.; Amoli, P.V.; Safari, M.; Zamani, M. A taxonomy of Botnet detection techniques. In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; Volume 2, pp. 158–162.

[74] López, D.D.; Uribe, M.B.; Cely, C.S.; Torres, A.V.; Guataquira, N.M.; Castro, S.M.; Nespoli, P.; Mármol, F.G. Shielding IoT against Cyber-Attacks: An Event-Based Approach Using SIEM. Wirel. Commun. Mob. Comput. 2018, 2018, 3029638.

[75] Al-Duwairi, B.; Al-Kahla, W.; AlRefai, M.A.; Abdelqader, Y.; Rawash, A.; Fahmaw, R. SIEM-based detection and mitigation of IoT-botnetDDoS attacks. Int. J. Electr. Comput. Eng. 2020, 10, 2182–2191.

[76] Hristov, M.; Nenova, M.; Iliev, G.; Avresky, D. Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT. In Proceedings of the 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA), Boston, MA, USA, 23–26 November 2021.

[77] Xia, W.; Wen, Y.; Foh, C.H.; Niyato, D.; Xie, H. A Survey on Software-Defined Networking. IEEE Commun. Surv. Tutorials 2015, 17, 27–51.

[78] Wani, A.; Revathi, S. DDoS Detection and Alleviation in IoT using SDN (SDIoT-DDoS-DA). J. Inst. Eng. Ser. 2020, 101, 117–128.

[79] Özçelik, M.; Chalabianloo, N.; Gür, G. Software-Defined Edge Defense Against IoT-Based DDoS. In Proceedings of the 2017 IEEE International Conference on Computer and Information Technology (CIT), Helsinki, Finland, 21–23 August 2017; pp. 308–313.

[80] Manso, P.; Moura, J.; Serrão, C. SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks. Information 2019, 10, 106.

[81] Ren, G.; Zhang, Y.; Zhang, S.; Long, H. Edge DDoS Attack Detection Method Based on Software Defined Networks. In Lecture Notes in Computer Science (Including Subseries

Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer International Publishing: Cham, Switzerland, 2022; pp. 597–611.

[82] Wang, M.; Lu, Y.; Qin, J. Source-Based Defense Against DDoS Attacks in SDN Based on sFlow and SOM. IEEE Access 2022, 10, 2097–2116.

[83] Wani, A.; Revathi, S. Ransomware protection in loT using software defined networking. Int. J. Electr. Comput. Eng. 2020, 10, 3166–3174.

[84] Cheng, H.; Liu, J.; Xu, T.; Ren, B.; Mao, J.; Zhang, W. Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks. Int. J. Sens. Netw. 2020, 34, 56.

[85] Park, Y.; Kengalahalli, N.V.; Chang, S.Y. Distributed Security Network Functions against Botnet Attacks in Software-defined Networks. In Proceedings of the 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Dallas, TX, USA, 12–14 November 2018; pp. 1–7.