Journal of Information Systems Engineering and Management
2025, 10(45s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Framework for Detection and Comprehensive Review of IoT Botnet Techniques

Vikrant[1*], Dr. Gesu Thakur[2]
[1*]Research Scholar, College of Smart Computing, COER University, Roorkee, Uttarakhand, India,
Orcid ID: 0009-0007-0215-1469
[2]Professor, College of Smart Computing, COER University, Roorkee, Uttarakhand, India
Email: [1]vikrantrana.in@gmail.com, [2]drgesuthakur@gmail.com

ARTICLE INFO

Received: 18 Dec 2024

Revised: 14 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

IoT devices have basic security flaws that make them susceptible to a variety of security threats and attacks, including botnet attacks. As a result, botnet developers keep using the security holes in IoT devices to obtain control of multiple host devices on networks and launch cyberattacks against the systems they plan to target. Finding IoT bot vulnerabilities is challenging since methods to get around detection and security measures are constantly being developed. The conceptual frameworks of IoT botnet attacks and different machine-learning-based botnet detection techniques will be looked at in this study. In this article, various botnet detection techniques are reviewed and compared on realistic IoT dataset that covers cutting-edge IoT botnet attack scenarios. The experiments and evaluations unequivocally demonstrate the effectiveness of our approach in detecting botnet activity while minimizing false positives. This research makes a significant contribution to improving IoT security by presenting a robust and scalable solution for detecting botnet attacks, with far-reaching implications for safeguarding critical infrastructure and upholding user privacy. Moving forward, our focus will be on addressing any remaining challenges and validating the practical utility and effectiveness of our methodology in real-world IoT deployments.

Keyword: IoT Botnet, Detection Technique, Botnet Architecture.

I. INTRODUCTION

The Internet of Things (IoT) is very popular in the recent decade with the smart IoT applications in industries and academia. There are various application areas of IoT such as smart homes, healthcare, transportation, and smart city are just a few of the innovations that will be made possible by the IoT, which is becoming a vital in the daily life. Because of the exponential growth of IoT devices and technological improvements, IoT is being implemented in many applications to improve services. IoT devices may collect, connect, interact, and share data for specific software, sensors, actuators, and connections between them. IoT devices are becoming more and more popular due to their affordability [1]. With the Internet of Things' enormous potential comes enormous concern, particularly in terms of cybersecurity. The connected devices are clearly distinct from computers, which contributes to the completely distinct security landscape of the Internet of Things. These gadgets often have a limited range of specific capabilities and are quite simple. Among the most difficult tasks in academic and industrial research is accurately identifying and detecting botnets, especially unidentified botnets from the initial infection, given the security concerns caused by their constant evolution. First, the C&C mechanism of botnets displays a variety of cognitive traits.

Smart household appliances and industrial sensors are among the many connected gadgets that have sprouted from the growing number of IoT devices. IoT devices are, nevertheless, appealing targets for cybercriminals to launch botnet assaults because of their vulnerabilities. Attackers can utilize these botnets, or networks of compromised devices, to launch DDoS (Distributed Denial of Service) assaults, exfiltrate data, and distribute malware, among other harmful activities. Botnets are a highly dangerous network of fictitious accounts that are linked together to form a web to lure an unwary victim into a series of attacks that will be covered in section III. In the meantime, let us examine and comprehend the current botnet situation. Like worms, botnets spread quickly through several computers; however, apart from their malevolent intent, bots can work together and target targets with effectiveness. [2] As a botnet proliferates, regular computers become vulnerable to compromise due to the 'bot' application, which acts as an order taker for the attack host and can even interact with other bots in the same network or wide region. Because they are directly managed, as opposed to most viruses, trojans, and malicious software, they only can steal information rather than “hack” into anything.

Botnets propagate by making use of zero-day weaknesses, peer-to-peer connections, phishing, bitcoin networks, and lightning networks. Additionally, botnets propagate very quickly, have distinct attacks and vulnerabilities, destructive nature, and substantial network flaws in the IoT network. Finally, as botnets are typically in a passive mode, they frequently lack traditional attack features and merely sustain the connection status via C&C channels [4]. Authors will describe the operation of the IoT botnet and the key detection techniques that have been applied to identify IoT botnet attacks. And we will emphasize the difficulties for the upcoming projects. There are distinct tools for IoT botnet:

"Bot: A 'zombified' or infected device that awaits orders from the master bot.

A "horde" of zombie computers, a network of bots or web of bots controlled by a master bot and used for either individual or collective goals.

The botmaster communicates with the bots under their control via the C&C channel.

Internet Relay Chat (IRC) is a widely used instant messaging service that offers one-to-one or one-to-many chat on several networks covering a variety of topics through distinct channels.

The proliferation of computer networks, the growth of IoT, and a variety of applications have all raised demand for cybersecurity. Botnets are being created because of an increase in malicious software attacks, which can interfere with day-to-day operations. A recent report from 2021 states that the AV-TEST institute registers over 350,000 new instances of malware and potentially harmful software per day [5]. Because malware is widely available, it may be used to build botnets, which give hackers access to hundreds or even millions of infected devices' combined bandwidth. This allows hackers to disrupt government and commercial operations daily. Botnets are networks of infected computers managed by botmasters, used for intrusion attacks like DDoS, click fraud, flooding, or spamming. The worldwide scope of attacks aided by botnets makes botnet detection a key concern. Machine Learning (ML) techniques, which necessitate feature extraction prior to ML model learning or training, are among the methodologies that have been developed.

Botnets can be categorized into three structures: central, distributed, and mixed. Central structures use client-server (C/S) mode, while distributed structures use non-central P2P mode. Mixed structures combine central and distributed structures, enhancing communication efficiency and robustness. These structures are essential for controlling botnets [6]. The article offers a thorough framework for anomaly detection methods utilizing SVM and random forest algorithms for botnet attack analysis and detection. By utilizing communication patterns, packet flow, and traffic volume, this technique enhances network security [7].

Supervised and unsupervised approaches detect anomalies using network traffic analysis. This is a new approach using supervised Machine Learning Algorithms, integrating decision tree and multilayer classifier. The Anomalies Detection System (ADS) is tested and evaluated for accuracy and computing time [8]. Due to the increase in network attacks, which will highlight the vulnerability of all the attacked device, to combat this problem an exploring machine learning (ML) as a potential solution has been developed. We will suggest an enhanced machine learning framework to identify botnet assaults on Internet of Things devices by fusing the decision tree classification models with the Bayesian optimization Gaussian process [9].

Botnets are a serious security risk because they use sophisticated control methods and are constantly evolving on a worldwide scale. Hacking groups have developed bots that can remotely take over compromised networks, spread malware, send unsolicited emails, and steal confidential data. Approximately 80% of all Internet traffic is associated with botnet activity. Bots in botnets access server IP addresses using Domain Name Service (DNS). The efficiency of a bonnet detection approach utilizing DNS query data is evaluated in this study using machine learning algorithms. Monitoring DNS query data can uncover harmful activity [10].

Botnets are complicated malware networks that use command-and-control servers to function under the supervision of botmasters, resulting in substantial financial losses. Because of their versatility, detection measures like masking and community-based procedures are useless [11]. ML classification techniques are effective in network security and intrusion detection, making them ideal for detecting botnets. However, optimizing these models is crucial to ensure their maximum capacity, as default versions may not be sufficient for optimal performance [12]. Internet security is crucial as cyberattacks can lead to data loss and unauthorized access to computer systems, networks, and devices. Botnets, which are computer networks infected with malware, are becoming a significant risk against cybersecurity. Botmasters, who control these networks, use C&C channels to create and manage botnets, which are an army of bots [13]. Botnets are infected devices controlled by a hacker, acting as a foot soldier for their botmaster. They pose a significant threat to information security, with botmasters constantly improving their skills.

Current detection methods struggle with evolving botnets [14]. The paper utilized distinct ML algorithm for the detection of IoT botnet attacks, which involve the connected devices infected with malware. This method improves accuracy in detecting and distinguishing botnets, a crucial aspect of computer security as more devices become potential botnets [15]. The IoT's growing number of devices makes them vulnerable to cyberattacks. ML can detect these threats, but high dimensionality data can lead to performance issues. This paper evaluates the impact of wrapper and hybrid feature selection techniques on ML models for IoT botnet identification [16]. The rising susceptibility of IoT devices to assaults renders standard intrusion detection systems inadequate. Although a machine learning-based architecture for botnet attack detection has been developed, its implementation is problematic due to its complexity and difficulties in gathering typical, everyday data [17].

The study aims to find a solution to the serious issue of protection against botnets, which may spread like Internet worms and engage in DDoS assaults, using an anomaly-based botnet detection system that uses IP headers to distinguish between botnet DNS requests [18]. It is essential to maintain confidentiality, integrity, and availability as internet services become more and more dependent on them. Botnet detection techniques grounded in machine learning aim to mitigate the impact of malevolent behavior [19].

IoT Botnet Life Cycle:

A newly registered DDNS connection is sent out to the first bot in the network, which is set up by the botmaster, also known as the botherder, in accordance with the Botnet Lifecycle (Figure 1). They spread by doing this, attacking various targets with denial-of-service attacks [20]. The botnets gather data by using a technique known as "traffic sniffing." Is Traffic Sniffing Explained? Bluetooth connections, both functional and unworkable, are typically blamed for it. It observes while cornering a target into being "captured" and examined. It listens in on conversations like a herding tactic. 21] Due to the fact that they are not all owned by the same person and can have distinct, hostile goals, a botnet may occasionally lose its edge to another horde of botnets [22]. The botnet leaves its location and unregisters from the DDNS after it has been routed or has lost its usefulness. The Single Bot Lifestyle, displayed on the right from the left image, provides a closer-up view of the bots and what they do while left idle, if they have been used effectively. As explained below, there are at least three major stages in which IoT botnet's function [23]:

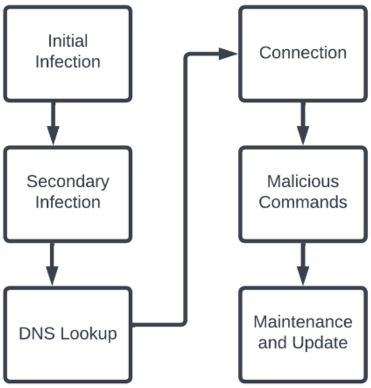

Fig. 1. Botnet lifecycle

Phase 1: Phase of scanning: To locate a vulnerable device, a bot does reconnaissance and scanning. The botmaster looks for vulnerable Internet of Things devices. When it does, it starts attacking it with force or by finding a weakness. Once compromised, the compromised device turns into a bot and initiates communication with the botmaster. Sending fingerprint packets to look for pseudorandom IPv4 addresses is one way the Mirai virus locates IoT devices that can be accessed through Telnet service on port 23 or port 2323 [24]. By exploiting well-known IoT device vulnerabilities or abusing weak credentials through brute force assaults, the bot finds new victims.

Phase 2: Propagation: Depending on the vulnerable device's architecture, an appropriate version of the bot is installed and launched. Typically, the bot locks ports to itself and ends the process associated with the relevant service to eliminate any prior infection and gain complete control of the device. The malicious code spreads and attracts additional bots to quickly expand the IoT botnet [25]. The bots are still awaiting orders from the botmaster as of right now.

Phase 3: Attack Phase: The attack phase, during which malicious operations including DDoS attacks, bitcoin mining, and spamming are carried out. Through the command-and-control server, the attacker sends commands to each dispersed bot, starting the attack. The same instructions are thus sent to the bots, who then launch the attack.

The botnet network operated by Mirai, which is used in DDoS attacks and has reached a record 1.1 Tbps [26], consists of more than 100,000 IoT devices intended for homeowners. Researchers have been creating novel approaches and strategies to identify malicious code on IoT devices in response to these risks [27]. The two primary categories of these research are the static analysis [28] and the dynamic analysis [29].

To find malicious code, the static analysis method uses analytical techniques such as byte code, system calls API, or Printable-Strings-Information (PSI). These techniques allow for precise control of the control flow (CFG) and data flow (DFG) [30]. This technique provides the ability to activate harmful code and permits in-depth file examination. Malicious code, however, only stores in the RAM of the device, making it challenging to apply the static analysis method to it or to collect samples using complex techniques (obfuscation). The static analysis method ought to be applied in conjunction with the dynamic analysis method, as per Andreas Moser [31].

A technique for tracking, gathering, and evaluating system behavior to identify malicious code is called dynamic analysis [32]. To establish whether the program deliberately breaks these predefined criteria, algorithms applied with pre-defined rules of dynamic analysis. While there are various approaches to employing dynamic analysis techniques, ML technology is currently the most widely adopted option. To support the training process and produce an appropriate learning model with high forecast accuracy, ML technologies require labels of data. Moreover, constructing a simulation by using sandbox with all the required features of the tested devices and having the ability to watch how dangerous code behaves as it runs are crucial prerequisites for dynamic analysis. This helps to prevent any infections of real field tissue (Figure 2). The accuracy of identifying malicious code is significantly influenced by all the data gathered from the Sandbox. Collecting benign files is not as simple as gathering virus samples, though. Many malware sample databases are available, including Detux and IoTPot, among others. Consequently, when machine learning models are used in sandbox environments, the data set used for training them has bias. As a result, the authors use the one-class categorization approach to address the issue. One-class classification is a technique that uses distinct samples for the identification of patterns that exists in the class [33].

This method's primary approaches include the following: employing deep learning for the extraction of features that are utilized for data preprocessing and provide the input in more conventional class classification techniques like Support Vector Data Description (SVDD) or One-Class SVM. The contribution of the paper is as follows:

- We suggest a method that detects IoT botnets using the One-class CNN.
- Modify the QEMU-based Sandbox environment to gather more monitoring data.
- Give the One-class classifier deep features as input based on the system call's structure.
- We test the method for IoT botnets with up to 98.01% accuracy and 97.25% F-measure.

The remaining paper can be categorized into the following sections: section II focuses on the latest research which has been conducted for the detection and mitigation of IoT botnet techniques. Section III discussed the distinct IoT botnet architecture such as centralized, decentralized, and hybrid. Section IV focused on the different techniques for the detection of IoT botnet. Section V proposed an enhanced system framework with experimental testbeds. Section VI demonstrated the results with discussion of results. Finally, section VII concludes the article.

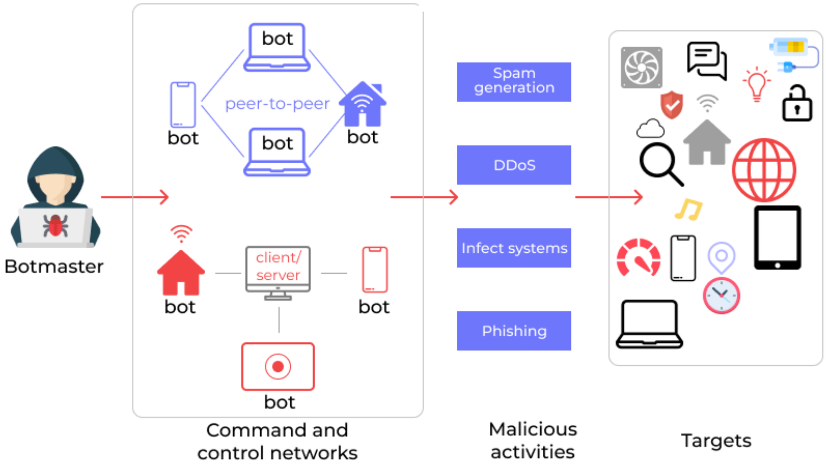

Fig. 2. Botnet Attacks and Infected Server

II. Literature Review

A brief study on Botnet Detection papers published earlier has been done and formatted in a tabular format. The addition of attributes like paper title, author name, algorithm used, and parameters.

The application of IoT is increasing day by day. Therefore, information security concerns are increasing. In the proposed system [34], combining artificial fish swarm algorithm and SVM. In the proposed system [35] usage of behavior-based botnet detection system based on fuzzy pattern recognition technique. It achieves high detection rate up to 95 %. In this document [36], the network flow with the connection logs approaching the dataset. Rule induction algorithm gives higher accuracy up to 98%. Using ISCX dataset for research purpose. System detects botnet using these four machine learning models: Naïve bayes, decision tree, support It [37] uses UNSW-NB15 dataset, Decision trees model gives higher accuracy than other. Various communication protocols are used by botmasters to communicate with the command-and-control server in a botnet. The proposed system [38] detects botnets based on DNS traffic analysis. A novel hybrid rule detection model technique is proposed by the union of the output of two algorithms. This paper [39] proposes to make a model using hybrid approach by combining KNN, Naïve bayes kernel and ID3 classifier which gives us more accurate results. It has used two datasets scenario-6 and scenario-2 This paper [40] gives the botnet detection technique which involves two levels: Host and Network. In host level it detects Bot using bayes classifier and in network level it estimates the probability of the botnets presence in the network using the entire distributed system. this developed classifier shows that the accuracy of this is about 88% Proposed [41] approach use the clonal selection algorithm which mainly focus on improving Bot GRABBER system. It is able to detect the IRC, HTTP, DNS and P2Pbotnets. It has high accuracy of about 95% and very low rate of false positive at about 3-5% vector.

To prevent a single point of failure, P2P botnets deployed numerous main controllers. However, both encryption and misuse detection systems were unsuccessful. The data mining scheme discovers the host of p2p botnet in real internet.

This paper [42] uses the ensemble of classifier algorithm to analyses the botnet traffic. It uses ISCX dataset. The results indicate that an ensemble of classifiers performs better than a single classifier when it comes to discovering bot evidence. Soft Voting of KNN & Decision Tree gives higher accuracy. Proposed [43] approach uses dataset having Mirai and Bashlite. IPR algorithm with XGBoost to identify nine most Important features that distinguish between benign and anomalous traffic for IoT devices. But it prefers the decision tree as it is simple and gives more accuracy. This paper proposes to use machine learning techniques like multilayer perceptron's and decision trees on network traffic analysis to detect botnet traffic.

For IoT networks capable of deliberately detecting IoT botnet attacks, a Deep Neural Network (DNN) that is significantly expandable is designed. Based on the evaluation, our DNN performs better than the current systems with high precision and delicacy. In this paper [44], the researcher suggested a two-layered method for identifying android botnets that combines static analysis with ensemble machine learning at the second layer and signature-based identification at the first layer. With the Logistic Regression classifier, the accuracy achieved is 95.4%. After combining the top three algorithms, this accuracy was significantly increased to 95.8%.

In this research [45], a deep learning model for Android botnet detection based on 1D CNN. Through thorough testing using 1,929 botnet apps and 4,387 clean apps, evaluation of the model. Comparing our CNN-based technique to other well-known machine learning classifiers, the findings demonstrate that it had the highest overall prediction accuracy. The graph-based machine learning approach [46] is used in this paper to detect botnet activity.The efficacy of the suggested graph-based botnet detection method using multiple machine learning algorithms was assessed using two heterogeneous datasets, CTU-13 and IoT-23. With 100% recall, all algorithms were able to identify every bot on both datasets. The optimal accuracy range for Extra Tress is 99%–100%. The Internet of Things is essential to our everyday lives. Cyber threats could target Internet of Things devices and services. Attackers may try to take advantage of holes in application protocols, such as HTTP (Hyper Text Transfer Protocol) and Domain Name System (DNS), among others. Data leaks and security lapses are the outcome. Using the ensemble detection method, this research [47] aims to reduce cyberattacks. AdaBoost ensemble learning is built with naïve bayes, decision trees, and artificial neural networks.

P2P botnets are more successfully detected by the suggested [48] botnet detection method. Our model uses a feed-forward artificial neural network to develop a classification model. The results of the experiments demonstrate that the convolutional features provide a higher level of detection accuracy than the standard features. Training of the model using convolutional neural network (CNN). It gives high accuracy with low false positive rate. This work [49] presents a novel approach for botnet identification, namely neuro-fuzzy classification techniques. With 15,000 occurrences and 56 attributes, the system's accuracy was 94.78%. To create the necessary dataset, cloud attacks have been carried out. A botnet detection method called BotEye is proposed in this research [50]. It is based on the network's traffic flow characteristics. Additionally, this method finds encrypted botnets. using CTU-13 dataset with three classifiers namely – Random Forest, ADA boost, decision tree. According to the proposed method, the precaptured pcap files are used to calculate the specified features over constant time intervals.

This study [51] demonstrates how high accuracy unsupervised learning models may be created with smaller feature sets, resulting in a reduction in the number of computational resources needed. Another solution for resource optimization is to train a single common model for all IoT devices rather than a unique model for each device. To recognize botnets and lawful actions at the DNS application layer semantically, this research [52] suggests a two-level deep learning framework-based botnet detection method. incorporating characteristics. Significant gains in F1score, detection speed, and false alarm rate were shown by the experimental results.

The author of this paper [53] suggests developing a system for identifying prospective botnets by examining their Internet traffic flows. This system can be installed on a server or a network. The classification model is then constructed using the behavior patterns that are retrieved from the groups of traffic flows that are classified as being like each other. To train the model, using bidirectional NetFlow files. The objective of the NetFlow protocol is to gather IP traffic data and track network traffic to get a clearer picture of the network traffic flow. Creating a system that uses machine learning to categorize the flow of botnets. The dataset was subjected to different classifiers. An extensive analysis of botnet kinds and life cycles is presented in this work. Examining the characteristics of peer-to-peer botnet detection approaches with a variety of the most recent detection techniques.

III. IoT Botnet Architecture

Centralized botnets: One central server, which serves as both the botmaster's command center and the point of contact for all bots, is used to manage and keep an eye on all the bots to reduce latency. One or more central servers could be accessible to the botmaster in this configuration. Protocols like HTTP and IRC are used by the server. As a single point of potential total failure, the botmaster server may have one disadvantage. Among the most well-known class of centralized Internet of Things botnets is the Mirai family.

Decentralized botnets: They are sometimes referred to as peer-to-peer (P2P) botnets. Every bot is a client and a server and is linked to a minimum of one other bot. Until all the bots are connected to each other, the orders will not go to them. In addition to being more complex and difficult to detect, this architecture's diversified peer communication makes it difficult for bots to coordinate. The IoT botnet in question uses peer-to-peer networking as its communication channel. Decentralized (P2P) IoT botnets, like Hajime, are among the most well-known.

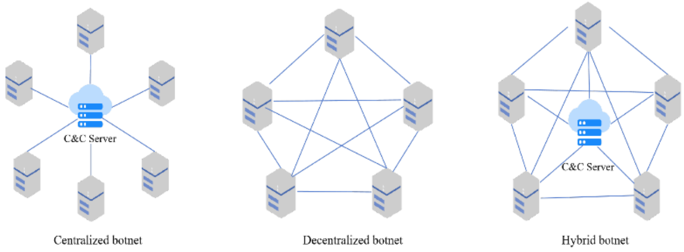

Fig. 3. IoT botnet Architecture

Hybrid botnets: Because it is made up of two separate kinds of bots, some of which can function as both servers and clients as well as clients solely, a hybrid botnet combines the two prior architecture types (centralized and decentralized). There is an excessive message delay.

IV. IoT Botnet Detection Technique

Three machine-learning methods were used by the researchers on the Bot-IoT sub dataset, which included all 46 attributes: Support Vector Machine (SVM), Recurrent Neural Network (RNN), and Long-Short Term Memory Recurrent Neural Network (LSTM-RNN) [54]. According to the experiment's findings, the SVM classifier required the longest training times but also had the highest accuracy and recall rates when all the attributes were present. This classifier yielded the best accuracy and lowest relative fall-out rates when it was limited to the ten highlighted qualities. Stated differently, the SVM produced better outcomes than the other two methods.

A hybrid intrusion detection system (HIDS) was implemented in the study reported in [55] with the goal of increasing the precision with which IoT threats were discovered. This system combines a

behavioral IDS to find zero-day attacks with a signature-based IDS to identify known threats. The boosting approach was used to merge the classification results from the two systems; a C5 decision tree served as the signature-based component and a One-Class SVM as the behavioral component. For binary classification, only 13 of the 46 initial features from the Bot-IoT datasets were employed. The proposed system was compared to several algorithms' detection accuracy, such as NB, SVM, KNN, CART, RF, and Multilayer perceptron. According to their research, the recommended approach had the highest accuracy rate, and the Random Forest algorithm generating 92.67% accuracy.
BLSTM-RNN for botnet detection was introduced by the authors of [56]. A unidirectional LSTMRNN was put up against the suggested solution in comparison. This was done to determine whether the latter approach could match the increased accuracy and loss metrics obtained on the collected dataset. The two models equally achieved excellent levels of accuracy and minimum loss metrics for the various Mirai attack pathways. To improve the detection of IoT botnet malware, researchers in [57] suggested a method for creating a PSIgraph that represents the interactions among PSI. This approach worked quite well for static analysis details. Without the requirement for the pre-selected criteria, the visual convolution neural network classifier—which is based on a convolutional neural network (CNN)—was also used to effectively identify IoT malware. By fusing a CNN classifier with a PSI graph, they demonstrated a novel method for Linux IoT botnet detection in their work. The test results showed a 92% accuracy rate and a 94% F-measure for the PSI graph CNN classifier. ANN were employed in [58] to identify DDoS attacks. Due to the previously mentioned imbalance in the Bot-IoT dataset, more normal samples were added using the Synthetic Minority Oversampling Technique (SMOTE) until the total number of samples matched the size of the DDoS records. The suggested method was tested on 34% of the dataset after being trained on 66% of it for binary classification. Out of the 46 features that were originally included, only 41 were utilized. The testing findings demonstrated that the SMOTE approach enabled 100% detection accuracy for DDoS attacks.

A. Defending Technique for Mirai

It is possible to prevent your computer from being infiltrated by the "horde" of bots in several ways, including by using third-party software that has been approved. Let us first review the set of issues we initially encountered and examined. By probing unwary PCs and files for information, other bots in their botnet network were able to steal hard-coded passwords from open ports, which Mirai used to get access to previously guarded corporations. This resulted in 65k IoT devices going rogue and becoming worthless—another zombie computer or device. How can we counter this system of fast-moving, unique scanning AI? Blockchain is one of the ways to protect the security vulnerability of IoT botnet. After the Mirai-based botnet code was made public and made available to any attacker or host eager to use it for their own purposes, there is a suggestion to use block chains as a form of retaliation. Once banks are filled, the Block-chain is utilized to continuously exchange information, swapping them back and forth to generate fluidity within the autonomous system. It is refreshed often to prevent Bots from taking more information than is traded and just the information that is swapped. Nevertheless, who is to say the bots would reach that level? To guarantee that bots would not be able to access the blockchain and its constantly updating data, this system employs two additional procedures [59].

Second, the recently suggested system depends on hosts. We have two types of hosts: normal hosts and IoT hosts. The system will only use IoT hosts because they have the advantage of being able to be remotely monitored and controlled. Additionally, IoT hosts are less likely to compromise than a standard computer when placed near a botnet. Lastly, the Autonomous System (AS) is utilized in both Block-Chain and Internet of Things technology. Its functions include storing a list of IP addresses and signaling who has, has not, will, or will not be compromised.

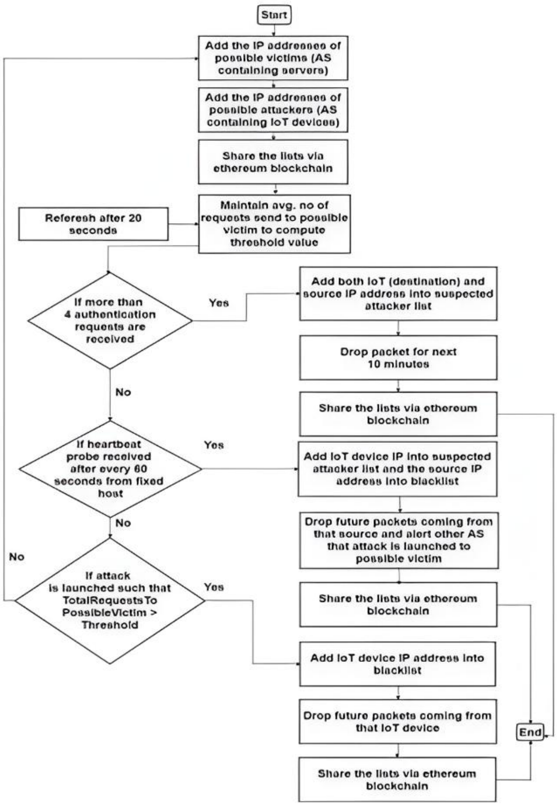
Start
Add the IP addresses of possible victims (AS containing servers)
Add the IP addresses of possible attackers (AS containing IoT devices)
Share the lists via ethereum blockchain
Refresh after 20 seconds
Maintain avg. no of requests send to possible victim to compute threshold value
If more than 4 authentication requests are received
Yes
Add both IoT (destination) and source IP address into suspected attacker list
Drop packet for next 10 minutes
Share the lists via ethereum blockchain
No
If heartbeat probe received after every 60 seconds from fixed host
Yes
Add IoT device IP into suspected attacker list and the source IP address into blacklist
Drop future packets coming from that source and alert other AS that attack is launched to possible victim
No
No
If attack is launched such that TotalRequestsTo PossibleVictim > Threshold
Yes
Share the lists via ethereum blockchain
Add IoT device IP address into blacklist
Drop future packets coming from that IoT device
End
Share the lists via ethereum blockchain

Fig. 4. Defending Mirai Botnet

B. Honeypot Based Technique

One of the few and most effective methods to stop bot-netting and robots from thinking like hordes and adopting their life cycles is to use honeypots. According to principle (5), honeypots function as bots that a bot controller would send forward to infect unwary users and computers with malicious applications. This effectively gives us "inside man" status within the botnet, allowing us to use fire with fire to extract information about the bot herder and bring an end to their operations. Authors [60] illustrates this requirement: the bot controller must confirm the phony bot before allowing it to join the "horde" after noticing that its sensor has lit up and sent him a message informing him that a bot is approaching. Usually, the botmaster has placed up several sensors to stop malfunctions and malicious traffic from coming his way. However, he is unable to determine whether the bots are once more his or someone else's.

C. DNS server

In a botnet, a bot will normally establish a connection with a C&C server to receive commands. DNS detection is based on the DNS information that the bot provides. This indicates that, in addition to promptly verifying the DNS origins of every machine that passes through, it would be advisable to keep an eye on the C&C server going forward to ensure that no malicious material is being trafficked through. Since bots frequently attempt to conceal their identity and location on the network by fluxing, the most effective method for identifying bots is to look at the number of times a query for DDNS rates has been pinged.

V. Proposed System Framework

The systems that are currently in place have assisted us in preventing malware attacks from harming our devices and machines. However, more research and studies have been conducted on the topic to find solutions because the security of our gadgets and their data is a delicate and serious issue that we have been dealing with for years. In the past, several solutions have been put up to improve the security of our data and gadgets. For instance, the Posts and Telecommunications Institute of Technology studied how to distinguish bots from real data using machine-learning classifiers in conjunction with static and dynamic vector analysis. They were 99% accurate in their identification. In a different study, deep learning was used to detect botnet attacks. Compared to other existing systems, the system is more scalable when machine algorithms are used at the back end. In this instance, a deep neural network model is used for the detection of attacks. We will examine a study that delves deeply into the Multilayer framework for Botnet detection. In this case, the attacks are detected using two layers of modules. Using the supplied data, the first layer filters network traffic. This module's goal is to minimize the time interval by minimizing traffic. The second module focuses on identifying botnet assaults using IP addresses. We created a layered architecture for botnet identification in the previous session using machine learning methods. K-means was included in the first module, and multilayer perceptions, KNN, and SVM were included in the second. The details of why the suggested algorithms outperform those that have been employed in the past. The primary elements of the suggested strategy are discussed in this section along with their purposes. Figure 5 depicts this strategy's general layout.

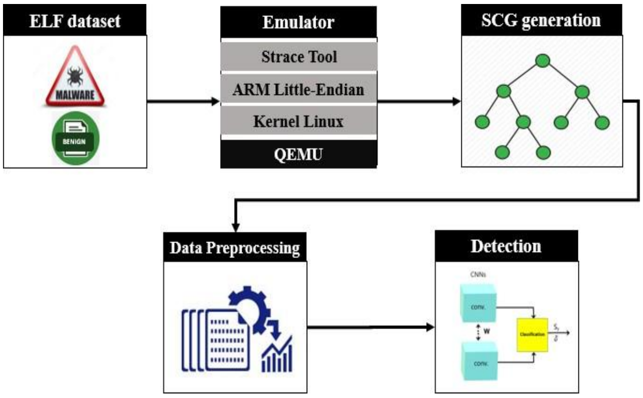

Fig. 5. Proposed System Framework

The four primary components of the suggested method are the following: data preprocessing, detection of botnets, system call graph construction, and emulator. The ARM chip and QEMU are utilized for the execution of ELF file initially in the simulated environment. The Debian VM images can be utilized for ARM and QEMU platforms. It incorporates an agent for gathering system call logs that extract the Strace program source code for the virtualization field. We compared our sandbox environment's system call collection results with those of other sandboxes (Cuckoo, Detux, etc.) throughout the trial. According to the comparison data, gathering system calls is more successful in our modified IoT Botnet sandbox scenario. After that, a straightforward data pre-processor removes the pointless instructions from the system call log. Following that, all the instructions required to generate the system call graph (SCG) are contained in a clear system call log. Next, using this cleared system call log, there is ELF file that generated some system calls. Then convert the graphs into a set of vectors (i.e., a vector for each graph) since the implementation of the classification method on these graphs is exceedingly complex and time-consuming. Given the significance of these graphs, the vectors deviating from them are also excessively huge. These vectors are made smaller by using an algorithm for feature extraction. This learning model is used by a detection module to identify if an input file is dangerous or benign. A portion of these vectors are then utilized as the training set to construct the learning model by one-class classification after feature extraction.

System Logs Collection:

A representative tracing system calls log must display some harmful behavior for the proposed method to identify malicious activity. Malicious activity cannot be identified if there is no distinction between the malware's and the innocuous programs' executions. It is not hard to turn an IoT botnet attacks because most of them attempt to connect to and communicate with a C&C server; they also attempt to affect the other IoT devices by using brute force attacks and scan IP addresses and attack service ports such as FTP, SSH, and Telnet. The traces are gathered by passively observing each program's execution in a confined space that resembles an Internet of Things device that is completely linked. The ARM chip architecture and QEMU platform are utilized to serve as the foundation for this emulated environment.

In order to gather system call logs, the suggested method applies the dynamic analysis method. System call behavior is captured via ELF file templates, which are implemented in an emulated environment and employ the built-in agent from the Strace tool source. When botnet malware records system calls and uses open service ports like SSH, Telnet, and FTP to find additional devices, it can be seen attempting to connect to a command and control (C&C) server. A fragment of the ELF file's system call log is seen in Figure 6. While the Figure 7 shows the convolutional neural network with target and reference dataset.

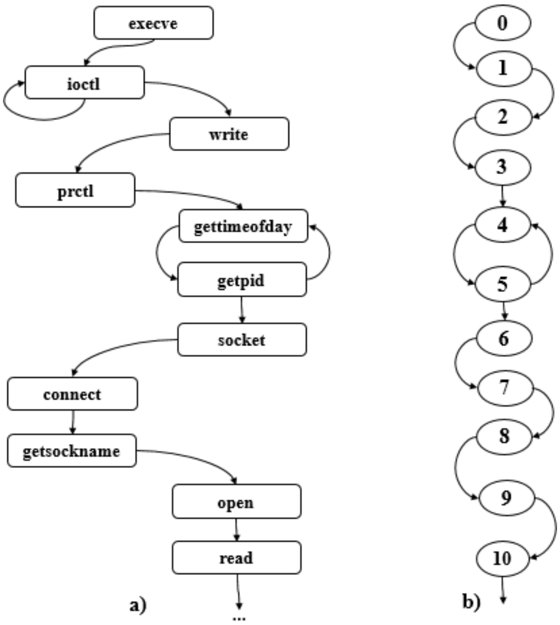

Fig. 6. One-class Classifier

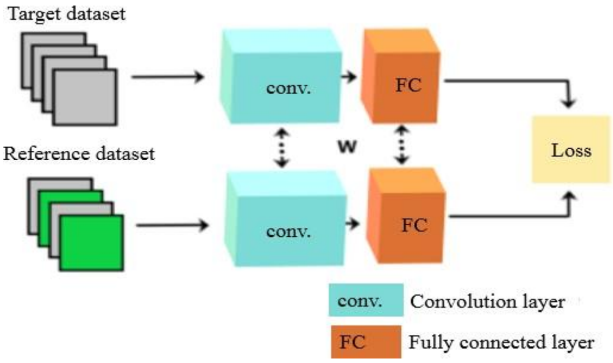

Fig. 7. Target and Reference CNN

V. Results and Discussion

The setup of the setting and the analysis of the test findings are covered in this part. The Intel Core i5-8500, 3.00 GHz CPU, 16GB RAM, and 8GB Nvidia GPU are used by the writers in this experiment. There are 600 ELF samples in the collection, 503 of which are malicious files and 97 of which are benign. The sources of the botnet samples were IoTPOT and VirusShare. Benign samples were gathered from the firmware of Internet of Things devices, including IP cameras and routers. The loss function employed in the experiment was l = lD + λlC, where λ = 0.1. To achieve the greatest outcomes, the deep neural network's weights were adjusted based on this loss function. Figure 5 and Table 1 exhibit the experimental results along with the evaluation indicators' respective values. The results show the comparison of the generated results with other techniques in Table 2.

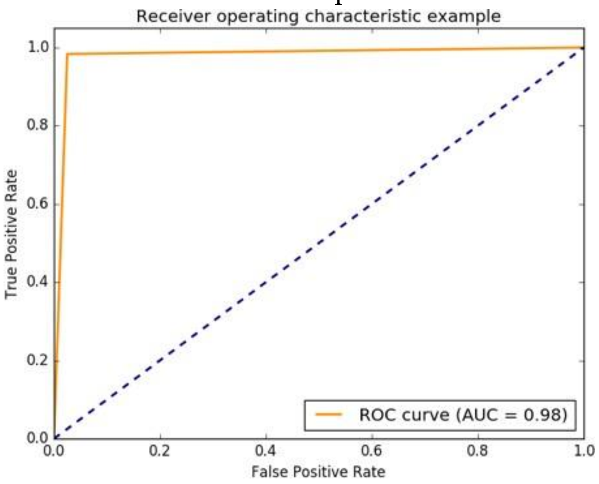

Fig. 8. Results with ROC

Table 1. Performance parameters

Parameters	Values
False Negative Rate	0.026
Precision	98.87%
Accuracy	98.01%
False Positive Rate	0.029
Recall	98.23%
F-measure	97.25%

Table 2. Results Comparisons

References	Accuracy (%)
Catillo et al. [34]	90.68
Thota et al. [40]	93.69
Randhawa et al. [42]	92.58
Mahadik et al. [43]	93.67
Proposed Method	98.01

VI. Conclusion

IoT devices have basic security flaws that make them susceptible to a variety of security threats and attacks, including botnet attacks. The authors of the current research offer a method for identifying Botnet activity on Internet of Things devices. This technique uses a characteristic system call graph and neural network classification of a class convolution as its foundation. The experimental findings show an accuracy of 98.01%, and the authors have attained an F-measurement of 97.25%, indicating the usefulness of the suggested method. The early findings, meantime, solely pertain to IoT botnet files;

additional malicious lines for IoT devices have not yet been included. To increase accuracy and application in ensuring data security and safety on Internet of Things devices, the authors want to apply the proposed approach to data sets that contain more dangerous codes in the future.

References

[1] Singh, M.; Singh, M.; Kaur, S. Issues and challenges in DNS based botnet detection: A survey. Comput. Secur. 2019, 86, 28–52.

[2] M. Al-Kasassbeh, M. Almseidin, K. Alrfou and S. Kovacs, "Detection of IoT-botnet attacks using fuzzy rule interpolation," Journal of Intelligent & Fuzzy Systems, pp. 1-11.

[3] A. Karim, R. B. Salleh, M. Shiraz et al., ―Botnet detection techniques: review, future trends, and issues, Journal of Zhejiang University Science, vol. 15, no. 11, pp. 943–983, 2014.

[4] Beltrán-García, P.; Aguirre-Anaya, E.; Escamilla-Ambrosio, P.J.; Acosta-Bermejo, R. IoT botnets. In Communications in Computer and Information Science; Springer Science and Business Media LLC.: Berlin, Germany, 2019; pp. 247–257.

[5] Alzahrani, H.; Abulkhair, M.; Alkayal, E. A multi-class neural network model for rapid detection of IoT botnet attacks. Int. J. Adv. Comput. Sci. Appl. 2020, 11.

[6] Manos, A.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z. 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.

[7] Dange, S.; Chatterjee, M. IoT Botnet: The Largest Threat to the IoT Network. In Advances in Intelligent Systems and Computing; Springer: Singapore, 2019; pp. 137–157.

[8] Beltrán-García, P.; Aguirre-Anaya, E.; Escamilla-Ambrosio, P.J.; Acosta-Bermejo, R. IoT botnets. In Communications in Computer and Information Science; Springer Science and Business Media LLC.: Berlin, Germany, 2019; pp. 247–257.

[9] Bertino, E.; Islam, N. Botnets and internet of things security. Computer 2017, 50, 76–79.

[10] Edwards, S.; Profetis, I. Hajime: Analysis of a decentralized internet worm for IoT devices. Rapidity Netw. 2016, 16, 1–18.

[11] K. Ansam, G. Iqbal, V. Peter, K. Joarder and A. Ammar, "A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks," Electronics, vol. 8, no. 11, p. 1210, 2019.

[12] McDermott, C.D.; Majdani, F.; Petrovski, A.V. Botnet detection in the internet of things using deep learning approaches. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.

[13] 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.

[14] Tzagkarakis, C.; Petroulakis, N.; Ioannidis, S. Botnet attack detection at the IoT edge based on sparse representation. In Proceedings of the 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 17–21 June 2019; pp. 1–6.

[15] 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.

[16] Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Comput. 2018, 17, 12–22

[17] D. A. D. B. o. S. A. w. S. f. I. Environment, "Soe, Yan Naung; Santosa, Paulus Insap; Hartanto, Rudy;," 2019 Fourth International Conference on Informatics and Computing (ICIC), pp. 1-5, 2019.

[18] N. Koroniotis, N. Moustafa and E. Sitnikova, "A New Network Forensic Framework based on Deep Learning for Internet of Things Networks: A Particle Deep Framework," Future Generation Computer Systems, 2020.

[19] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 military communications and information systems conference (MilCIS), pp. 1-6, 2015.

[20] Q. Miao, J. Liu, Y. Cao, and J. Song, “Malware detection using bilayer behavior abstraction and improved one-class support vector machines,” International Journal of Information Security, vol. 15, no. 4, pp. 361–379, 2016.

[21] E. Burnaev and D. Smolyakov, “One-class SVM with privileged information and its application to malware detection,” presented at the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016, pp. 273–280.

[22] P. Perera and V. M. Patel, “Learning deep features for one-class classification,” arXiv preprint arXiv:1801.05365, 2018.

[23] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A tutorial on the cross-entropy method,” Annals of operations research, vol. 134, no. 1, pp. 19–67, 2005.

[24] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, “Iotpot: A novel honeypot for revealing current iot threats,” Journal of Information Processing, vol. 24, no. 3, pp. 522–533, 2016.

[25] K. Angrishi, “Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets,” arXiv preprint arXiv:1702.03681, 2017.

[26] E. Bertino and N. Islam, “Botnets and Internet of Things Security,” Computer, vol. 50, no. 2, pp. 76–79, Feb. 2017.

[27] C. Kruegel and Y. Shoshitaishvili, “Using Static Binary Analysis to Find Vulnerabilities and Backdoors In Firmware,” Black Hat USA, 2015.

[28] P. Celeda, R. Krejci, J. Vykopal, and M. Drasar, “Embedded malware-an analysis of the chuck norris botnet,” presented at the Computer Network Defense (EC2ND), 2010 European Conference on, 2010, pp. 3–10.

[29] A. Diro, H. Reda, N. Chilamkurti, A. Mahmood, N. Zaman, and Y. Nam, “Lightweight Authenticated-Encryption Scheme for Internet of Things Based on Publish-Subscribe Communication,” IEEE Access, vol. 8, pp. 60539–60551, 2020, doi: 10.1109/ACCESS.2020.2983117.

[30] K. Sahlmann, V. Clemens, M. Nowak, and B. Schnor, “Mup: Simplifying secure over-the-air update with mqtt for constrained iot devices,” Sensors (Switzerland), vol. 21, no. 1, pp. 1–21, Jan. 2021, doi: 10.3390/s21010010.

[31] M. Panda, A. A. A. Mousa, and A. E. Hassanien, “Developing an Efficient Feature Engineering and Machine Learning Model for Detecting IoT-Botnet Cyber Attacks,” IEEE Access, vol. 9, pp. 91038–91052, 2021, doi: 10.1109/ACCESS.2021.3092054.

[32] S. Hosseini, A. E. Nezhad, and H. Seilani, “Botnet detection using negative selection algorithm, convolution neural network and classification methods,” Evolving Systems, vol. 13, no. 1, pp. 101–115, Feb. 2022, doi: 10.1007/s12530-020-09362-1.

[33] S. Bagui and K. Li, “Resampling imbalanced data for network intrusion detection datasets,” J Big Data, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-020-00390-x.

[34] M. Catillo, A. Pecchia, and U. Villano, “A Deep Learning Method for Lightweight and Cross-Device IoT Botnet Detection †,” Applied Sciences (Switzerland), vol. 13, no. 2, Jan. 2023, doi: 10.3390/app13020837.

[35] K. Alissa, T. Alyas, K. Zafar, Q. Abbas, N. Tabassum, and S. Sakib, “Botnet Attack Detection in IoT Using Machine Learning,” Comput Intell Neurosci, vol. 2022, 2022, doi: 10.1155/2022/4515642.

[36] T. N. Nguyen, Q. D. Ngo, H. T. Nguyen, and G. L. Nguyen, “An Advanced Computing Approach for IoT-Botnet Detection in Industrial Internet of Things,” IEEE Trans Industr Inform, vol. 18, no. 11, pp. 8298–8306, Nov. 2022, doi: 10.1109/TII.2022.3152814.

[37] S. M. Sajjad et al., “Detection and Blockchain-Based Collaborative Mitigation of Internet of Things Botnets,” Wirel Commun Mob Comput, vol. 2022, 2022, doi: 10.1155/2022/1194899.

[38] T. Hasan et al., “Securing Industrial Internet of Things Against Botnet Attacks Using Hybrid Deep Learning Approach,” IEEE Trans Netw Sci Eng, vol. 10, no. 5, pp. 2952–2963, Sep. 2023, doi: 10.1109/TNSE.2022.3168533.

[39] U. Garg, S. Kumar, M. Kumar, U. Garg, S. Kumar, and M. Kumar, "INFRDET: IoT network flow regulariser-based detection and classification of IoT botnet," 2008.

[40] S. Thota and D. Menaka, "Botnet detection in the internet-of-things networks using convolutional neural network with pelican optimization algorithm," Automatika, vol. 65, no. 1, pp. 250–260, Jan. 2024, doi: 10.1080/00051144.2023.2288486.

[41] J. Azimjonov and T. Kim, "Designing accurate lightweight intrusion detection systems for IoT networks using fine-tuned linear SVM and feature selectors," Comput Secur, vol. 137, Feb. 2024, doi: 10.1016/j.cose.2023.103598.

[42] R. H. Randhawa, N. Aslam, M. Alauthman, M. Khalid, and H. Rafiq, "Deep reinforcement learning based Evasion Generative Adversarial Network for botnet detection," Future Generation Computer Systems, vol. 150, pp. 294–302, Jan. 2024, doi: 10.1016/j.future.2023.09.011.

[43] S. Mahadik, P. M. Pawar, and R. Muthalagu, "Efficient Intelligent Intrusion Detection System for Heterogeneous Internet of Things (HetIoT)," Journal of Network and Systems Management, vol. 31, no. 1, Mar. 2023, doi: 10.1007/s10922-022-09697-x.

[44] R. Sharma, S. Mohi ud din, N. Sharma, and A. Kumar, "Enhancing IoT Botnet Detection through Machine Learning-based Feature Selection and Ensemble Models," ICST Transactions on Scalable Information Systems, Sep. 2023, doi: 10.4108/eetsis.3971.

[45] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 1646–1685, Jul. 2020, doi: 10.1109/COMST.2020.2988293.

[46] H. G. C. Ferreira and R. T. de Sousa Junior, "Security analysis of a proposed internet of things middleware," Cluster Comput, vol. 20, no. 1, pp. 651–660, Mar. 2017, doi: 10.1007/s10586-017-0729-3.

[47] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," IEEE Communications Surveys and Tutorials, vol. 22, no. 3, pp. 1646–1685, Jul. 2020, doi: 10.1109/COMST.2020.2988293.

[48] E. Gelenbe and M. Nakip, "Traffic Based Sequential Learning During Botnet Attacks to Identify Compromised IoT Devices," IEEE Access, vol. 10, pp. 126536–126549, 2022, doi: 10.1109/ACCESS.2022.3226700.

[49] C. D. Mcdermott, A. V Petrovski, and F. Majdani, "Towards Situational Awareness of Botnet Activity in the Internet of Things."

[50] Y. Fu, Z. Yan, J. Cao, O. Koné, and X. Cao, "An Automata Based Intrusion Detection Method for Internet of Things," Mobile Information Systems, vol. 2017, 2017, doi: 10.1155/2017/1750637.

[51] J. C. S. Sicato, S. K. Singh, S. Rathore, and J. H. Park, "A comprehensive analyses of intrusion detection system for IoT environment," Journal of Information Processing Systems, vol. 16, no. 4, pp. 975–990, Aug. 2020, doi: 10.3745/JIPS.03.0144.

[52] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," Journal of Network and Computer Applications, vol. 84. Academic Press, pp. 25–37, Apr. 15, 2017. doi: 10.1016/j.jnca.2017.02.009.

[53] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. V. Pham, S. K. Padannayil, and K. Simran, "A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities," IEEE Trans Ind Appl, vol. 56, no. 4, pp. 4436–4456, Jul. 2020, doi: 10.1109/TIA.2020.2971952.

[54] Y. Jung and R. Agulto, "Virtual ip-based secure gatekeeper system for internet of things," Sensors (Switzerland), vol. 21, no. 1, pp. 1–26, Jan. 2021, doi: 10.3390/s21010038.

[55] C. Singh and A. K. Jain, "A comprehensive survey on DDoS attacks detection &amp; mitigation in SDN-IoT network," e-Prime - Advances in Electrical Engineering, Electronics and Energy, vol. 8. Elsevier BV, p. 100543, Jun. 2024. doi: 10.1016/j.prime.2024.100543.

[56] M. Sanlı, "Detection and Mitigation of Denial of Service Attacks in Internet of Things Networks," Arabian Journal for Science and Engineering. Springer Science and Business Media LLC, Feb. 22, 2024. doi: 10.1007/s13369-023-08669-w.

[57] U. Garg, S. Kumar, and A. Mahanti, “IMTIBOT: An Intelligent Mitigation Technique for IoT Botnets,” Future Internet, vol. 16, no. 6. MDPI AG, p. 212, Jun. 17, 2024. doi: 10.3390/fi16060212.

[58] B. Bojarajulu and S. Tanwar, “Customized convolutional neural network model for IoT botnet attack detection,” Signal, Image and Video Processing, vol. 18, no. 6–7. Springer Science and Business Media LLC, pp. 5477–5489, Jun. 17, 2024. doi: 10.1007/s11760-024-03248-4.

[59] K. Kaur, A. Kaur, Y. Gulzar, and V. Gandhi, “Unveiling the core of IoT: comprehensive review on data security challenges and mitigation strategies,” Frontiers in Computer Science, vol. 6. Frontiers Media SA, Jun. 26, 2024. doi: 10.3389/fcomp.2024.1420680.

[60] M. Gelgi, Y. Guan, S. Arunachala, M. Samba Siva Rao, and N. Dragoni, “Systematic Literature Review of IoT Botnet DDOS Attacks and Evaluation of Detection Techniques,” Sensors, vol. 24, no. 11. MDPI AG, p. 3571, Jun. 01, 2024. doi: 10.3390/s24113571.