RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF
MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt24321

Botnet Attack Detection in IoT Devices using Ensemble Classifiers with Reduced Feature Space

N. Dharini [a, *], Jeevaa Katiravan [b], S.P. Shakthi [c]

[a] Department of Computer Science Engineering (Cyber Security), R.M.K College of Engineering and Technology Chennai, Tamil Nadu India
[b] Department of Information Technology, Velammal Engineering College, Chennai, Tamil Nadu, India
[c] Tata Consultancy Services, Chennai, Tamil Nadu, India
* Corresponding Author Email: dharini1990@gmail.com

DOI: https://doi.org/10.54392/irjmt24321
Received: 09-04-2024; Revised: 10-05-2024; Accepted: 15-05-2024; Published: 22-05-2024

Abstract: The Internet of Things (IoT) is an advancing important technology offers multiple perks, such as webcams, baby monitors, room temperature controllers, smart security cameras and intelligent home automations resulting in the creation of intelligent settings that greatly simplify daily living. However, there are cybersecurity dangers associated with IoT devices due to their lack of protection. For example, Internet of Things botnets have become a major risk. IoT has been a boon for attackers to perform malicious attacks like information theft, DDoS, sending junk data to disrupt networks. IoT devices face serious security issues, from having default weak and common passwords, and a lack of security, rarely and poorly monitored, to having open access to management systems, always connected to the internet. In this paper, we used the N-BaIoT dataset which includes datasets of 9 IoT devices infected with 2 Bot viruses Mirai and Bashlite, where each botnet has 5 sub-attacks and the benign datasets of 9 devices. An analysis with the N-BaIoT dataset which initially had 115 features were reduced to 35 features by using manual reduction and further reduced to single feature in 5-time instances equivalent to 5 features using heat map. We then classified the sub-attacks of 2 botnets and benign of 9 IoT devices by using 7 Machine Learning based classifiers in the Weka tool and Python and compared our results with the manually reduced 35 Features and Heat map based 5 features. Performance metrics like correctly classified, incorrectly classified instances and time taken to build the model were evaluated to verify the proposed work. We found out that over 3 ensemble machine learning classifiers performed extremely well with 99 % accuracies for all devices. In order to verify the logic of our work we tried implementing our proposed model in a different dataset with 3 ensemble classifiers and were able to achieve high detection rates.

Keywords: Internet of Things, BotNet Attacks, Ensemble Machine Learning. Heat Map, Reduced Feature Space

1. Introduction

IoT has evolved and has become an essential part of modern society in a variety of applications and considered as the most preferred technology [1]. The proliferation of electronic services and applications has spurred remarkable progress in telecommunications networks, ushering in the era of the Internet of Things (IoT) [2, 3]. In this evolving communications landscape, devices function as interconnected entities capable of sensing their surroundings, establishing connections, and sharing data online [4, 5]. Projections indicate that by 2022, the Internet will host a staggering one trillion IP addresses or interconnected objects via IoT networks [6]. The common IoT devices we use in our homes like security cameras, baby monitors, and doorbells when bought we often look at whether they are cost-efficient or not but when considering security of these devices we often neglect them and these devices will be installed and not touched for years this results in making the attacker to easily invade into these devices. As a result, an effective detection system must be developed and implemented. Despite the existence of numerous previous detection systems, they are insufficient to detect all types of attacks effectively because of the proliferation of new variants of malicious software. Cyber-attacks on Internet connected devices have increased since 2016.The IoT devices we used in our study [2] are small common IoT devices used in our day-to-day life like in home, schools, and offices environments the Danmini_doorbell and Ennio_doorbell are modern smart doorbells which will be always connected to Wi-Fi and can be controlled with their app in multiple mobiles when someone presses the doorbell we have to slide to answer and look for the visitor, the visitor will be able to hear our voice and respond to us and we will be able to see them with a motion detector and alerts us if any motion is detected .The next device is the ecobee_thermostat this device can adjust the temperature automatically subjected with the climatic conditions of the environment which can be controlled by mobile, then the next device is Philips B120N10_Baby monitor is monitoring device designed for babies, when parents are away from the children doing some chores this device will help them to monitor them remotely from anywhere using a mobile app from phone, this device enables us to talk and sing to the baby. The next devices are 4 models of different security cameras they are Provision PT737E Security Camera ,Provision838 Security Camera, Samsung SNH1011 N Webcam, Simple Home XCS71002 WHT Security Camera Simple Home XCS71003 WHT Security Camera used to monitor the important areas over the cloud transmission using the internet, they are compatible in both android and IOS compatibility having a common default password as 888888, we can speak and hear with help of built-in microphone and speaker, we can also view the recorded videos, Samsung SNH1011N Web camera in order to connect home routers and personal computers must be connected, router can be connected wirelessly or with an Ethernet cable, the LED (Light Emitting Diode) will indicate that the device is on, WIFI(Wireless Fidelity) LED in this device must stop blinking to ensure that the device is being connected. The account creation must be done in website with the serial number in the packaging so that the device can be connected and accessible ,the device has an anytime and anywhere view ,motion detection is enabled, schedule can be set up for meetings ,can view footage from this WIFI camera using a smartphone or a tablet. All the above devices are used by more than half of the people in this era, these devices are very useful as well as they are vulnerable to various security threats, because these devices though may have many advantages lacks security, these devices, in order to work, must always be connected to the internet, and must always be turned ON, common passwords which are easily cracked, rarely monitored and poorly monitored, low cost, these devices are installed and not touched for 10-20 years, non-patchable devices connected to the internet, sort of source to attack somebody else and used for flooding of packets and DDOS attacks. So an IoT device can be attacked very easily. Some of the well-known botnets are Bashlite and Mirai. A botnet runs a bot on multiple IoT devices to form a botnet that is controlled by Command and Control centre (C&C). Numerous issues are brought about by the botnet, such as service interruption and information theft.

Currently available IDS have several limitations such as lack of flexibility and scalability [1]. Before machine learning-based intrusion detection, traditional approaches relied on rule-based, signature-based, and anomaly-based methods, deployed either network-based or host-based. These methods often suffered from limitations such as high false positives, inability to adapt to new threats, and manual intervention required. Machine learning-based intrusion detection outperforms traditional methods by leveraging data-driven models to automatically detect and classify intrusions more effectively, overcoming limitations like adaptability to evolving threats and scalability. Machine learning-based classification is a possible detection mechanism that will help us to differentiate normal from the attack nodes. While a great deal of prior research has been done with machine learning techniques, most of the studies have been done with old datasets such as KDDCUP99 with outdated data with less information for prediction. We tried different ML algorithms, to create a botnet detection model suitable for all 9 devices. Hence the N-BaIoT modern dataset with 115 features in each dataset, was created by infecting them with Bashlite and Mirai botnets into 9 IoT devices. N-BaIoT had separate datasets for ten attack classes and one benign class for all 9 IoT devices by 2 botnets such as Mirai and Bashlite (Gafgyt).

So with help of ML classifiers, we classified the 10 attack classes and benign of the 9 IoT devices. So our comparative study has been performed with 9 IoT devices.

Further, single classifier-based models are inefficient to detect all types of intrusions efficiently under all scenarios [7-11]. An ensemble classifier involves the application of multiple classifiers and aggregates their outcomes to receive more correct results [9, 12].

Ensemble learning is a machine learning technique where multiple models are combined to improve the performance of the overall system. The idea is to leverage the strengths of various individual models to compensate for their weaknesses, ultimately leading to better predictive performance. Ensemble methods are powerful for several reasons:

Reduction of overfitting: Ensemble methods help reduce overfitting by combining multiple models. When different models are trained on the same dataset but with different algorithms or subsets of data, they are likely to make different errors. By combining them, the errors tend to cancel out, leading to a more robust and generalized model.

Improved predictive accuracy: Ensemble methods often outperform individual models by leveraging the diversity of different models. Each model may capture different aspects of the data, and combining them can lead to better overall predictions.

Increased stability: Ensemble methods are more stable because they are less sensitive to small variations in the training data. This makes them particularly useful when working with noisy data or datasets with outliers. Versatility: Ensemble methods can be applied to a wide range of machine learning tasks, including classification, regression, and clustering.

Our previous works [13-15] exemplifies the usage of significant network parameters such as energy, packet count etc as the most distinguishing factors to detect attacks in wireless network [16, 17]. Thus in this proposed work, we utilize ensemble classifiers to enhance efficiency. Our paper aims to devise an efficient detection system with minimal features, utilizing the training and testing of seven top-performing machine-learning classifiers identified in our prior research, both on the full dataset and a dataset reduced through principal component analysis. Through extensive experimentation, we identified seven algorithms exhibiting strong performance. Subsequently, we applied these algorithms to a reduced feature space. Our primary research objective is to establish a highly efficient detection mechanism through a comparative analysis, achieving both minimal feature usage and high accuracies, thereby bolstering the security of IoT devices.

The main contributions of this study can be summarized as follows:

1) For our study, we used the N-BAIoT dataset and selected 9 IoT devices infected with the two botnets, which have ten attack classes and one benign class.
2) We have manually reduced the extensive 115 features in our dataset to 35 features and then applied a heat map and further reduced it to a single feature in five-time instances (5 features) with minimal constructing time and higher accuracies.
3) A comparative study has been performed to distinguish the 35 and 5 features trained with 7 machine learning algorithms for 9 IoT devices for both Mirai and Bashlite. We have compared the correctly and incorrectly classified instances along with the time taken to build the models.
4) The results show that the 3 ensemble classifiers random tree, random forest, and random committee had overall out-performed in all 9 IoT Devices for both Bashlite and Mirai.
5) To verify the robustness of our model and to check the logic of our work we tried implementing our proposed model in a different dataset and were able to classify the good accuracies.

2. Related Works

In order to protect against the vulnerabilities encountered by IoT devices, a dataset known as N-BaIoT was built for the identification and categorization of botnet assaults. The authors of this research study [2] produced this dataset and utilized Deep autoencoder neural networks and mathematical attributes extracted from each device's traffic and benign information to train their model. When identified anomalies are incorporated into fresh data from an IoT device, they can indicate that the gadget is attacked. Device attempts to rebuild the input following compression, and a large reconstruction error is found to determine whether the data is considered anomalous or not. They have succeeded in achieving a maximum TPR(True Positive Rate) by identifying assaults as soon as they happen and a minimum FPR(False Positive Rate) by erroneously classifying benign data as harmful [18]. In this work they have used the Pearson correlation coefficient and selected 9 best features and used random forest, gradient boosting, XG boost and ada boost classifiers and got accuracies such as 99.5% for random forest 85.2% for decision tree, 99.2% for k nearest neighbour and 95.5% for logistic regression. Nomm and Hayretdin, [19] in this work had used isolated forest, SVM and Entropy based feature selection reduced to 5 features with 90% accuracy with 5 features. Almomani, (2023) [20], used ensemble learning system with random forest, support vector networks and ANN (Artificial Neural Networks) and logistic regression. The proposed ensemble method had 96.74% accuracy. Yousra Javed and Navid Rajabi, (2018) [21] Yosura Javed and Navid Rajabi had done the testing phase of IoT botnet categorization. Abu Al-Haija *et al .(2022)* [22] used the following algorithms with their respective accuracies bagging trees-99.5%, adaboost decision trees-95.4%, RUS boosted DT-93.9%, optimized DT-97.3%, optimisable k- nearest neighbour-97.3%, optimizable discriminant 83.6% in 5-fold cross validation with no feature selection [23]. Have used the Fisher score to reduce features into 3, and classified using KNN and decision tree and got 94% and 98% accuracies respectively. Mohanty *et al.(2022)* [24] have made a stacking ensemble model and used sequential feature selection and used 15 features in stacking and utilized random forest, kNN, decision tree and achieved 98.89% for binary classification and 97.89% for multi classification. Hasan *et al. (2022)* [25] had used LSTM which had overcome the limited learning in RNN and in addition to that they have combined DNN & LSTM together which performed with 99.4% detection accuracy. Rust-Nguyen, M. Stamp, (2022) [26] had classified traffic and applications using 4 algorithms SVM -99.3% and 84.8% random forest 99.8% 92.2%,CNN- 99.8% AND 88.8% AC-GAN-98.7% and 75.9% respectively. Hasan *et al.* (2021) [27] made use of CNN&LSTM and achieved an Accuracy of 90.88%. Bharath *et al.* (2021) Used principal component analysis and reduced to 2 features with random forest and decision tree and classified the 10 attacks and got closely to 100% accuracies [28]. Mahi *et al.* (2020) [29] Used fisher score for reducing features and extreme gradient boosting (classification) and had 99.96% accuracy. Prokofiev *et al.* (2018) [30] used Fisher score and GXG boost classification and had 99.96%. Alzahrani and Bamhdi (2022) used logistic regression model and had Efficiency up to 97.3% [31]. Jiyeon *et al.* (2020) [32] Used CNN AND LSTM. Cullen *et al.* [33] did feature engineering for 84 features and for classification they utilized light gradient boosting algorithm with 98.2% for 3 classes and 88.8% for 8 classes and also the authors did a study for synthetic data and got 98.7% accuracy for 3 classes and 90.3% for 8 classes. The authors of [34] created the Iotid20 dataset which had 83 feature and applied several classifiers for which the accuracies of gaussian NB, SVM, logistic regression, LDA, decision tree had and ,random forest was 73%,40%,40%,70% 88% and 84% respectively.

Alissa *et al.* (2022) [35] decision tree, XgBoost, and logistic regression performed with 94% accuracy. The authors of [36] utilized LSTM and CNN to give 94% accuracy and Multi-CNN gave 96% accuracy. In [37] we have made a comparative analysis for full datasets and reduced datasets (using principal component analysis) and found 7 better performed machine learning algorithms and those identified best performing algorithms were used in this work.

2.1 Drawbacks of the existing works

- Lack of Generalization: Some studies achieve high accuracies on specific datasets or scenarios, but their models may struggle to generalize to new, unseen data.
- Limited Evaluation Metrics: Many works focus solely on accuracy without considering other important metrics like precision, recall, or F1-score, which provide a more comprehensive understanding of model performance.
- Overfitting: Certain studies may overfit their models to the training data, resulting in high accuracies on the training set but poor performance on unseen data.
- Lack of Scalability: Some approaches may not scale well to larger datasets or real-world deployment scenarios, limiting their practical utility.

Table 1. Summary of Literature Review Works

State of art works	Techniques used	Accuracy
"N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Auto encoders" [2]	Deep auto encoder	1. Minimizes FPR rate 2. Maximum TPR rate
"Unsupervised Anomaly Based Botnet Detection in IoT Networks" [19]	Entropy based feature SVM isolated forest(IF)	90% accuracy with 5 features and 3 features
"Multi-Layer Perceptron Artificial Neural Network Based IoT Botnet Traffic Classification" [20]	Multi-layer perceptron artificial neural network	100% in testing phase
"Dimensionality Reduction for Machine Learning Based IoT Botnet Detection" [23]	Fisher score, Decision tree, KNN	3 features used Decision tree = 0.98 KNN = 0.94
"Securing Industrial Internet of Things Against Botnet Attacks Using Hybrid Deep Learning Approach" [25]	LSTM, DNN	99.4% detection accuracy.
"Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications" [27]	CNN & LSTM	90.88% 88.61% - Doorbell 88.53-thermostat devices 87.19%, 87.6% , 89.4%,89.23%-Security cameras
"Edge2Guard: Botnet Attacks Detecting Offline Models for Resource-Constrained IoT Devices Pattern Recognition." [28]	PCA Random forest & decision tree	Close to 100% detection rate
"IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection" [29]	Fisher score & GXG boost classify	99.96% accuracy
"A Method to Detect Internet of Things Botnets"[30]	Logistic Regression model	Efficiency up to 97.3%
"Hybrid Deep-Learning Model to Detect Botnet Attacks over the Internet of Things Environments"[31]	CNN AND LSTM	Accuracy=1.00

In conclusion Ensemble classifiers are powerful and widely used for several reasons. Ensemble methods combine multiple base learners to make predictions, often resulting in higher accuracy compared to individual models. Ensemble methods can capture complex relationships in the data by combining different models that may specialize in different aspects of the problem. This flexibility allows them to handle diverse types of data and relationships. Ensemble methods are often more robust to noise and outliers compared to individual models. By combining multiple predictions, they can mitigate the impact of noisy data points or outliers on the final decision. Thus addressing the above drawbacks various ensemble classifier models are utilized in our work on the proposed reduced dataset using minimal features thus reducing the complexity.

3. Background

3.1 IoT attacks

There are numerous IoT attacks in our work, we have mainly focussed on two botnets Mirai and Bashlite which initiate DDOS attacks. Mirai and Bashlite are flooding based attacks, flooding of unwanted packets to disrupt the network activities so that the server will be busy and users might not be able to access the particular server. This is achieved by transmitting a flood of network packets like ACK,SYN,TCP,UDP ,for instance the attacker will be a client who continuously sends a malicious flood of SYN requests and the server in turn will respond with a SYN-ACK, the server will be waiting for an acknowledgement from the client, but the client here will not send a ACK but keeps sending SYN requests to make the server busy, on the other hand server will wait for a while and again send a SYN-ACK packet to random places waiting for an acknowledgement. Multiple malicious users send SYN to the server with different devices making a distributed denial of service attack to the particular server and blocking it, because of this the legitimate user will not be able to access the server. The botnet-based attacks in IoT are one of the popular attacks in smart environments because of the security vulnerabilities, low memory and computing resources making them being suitable for the attackers to attack.

3.2.1 Botnets-Bashlite and Mirai

The Mirai Botnet, known as the celebrity of botnets, emerged in 2016 and focused on a significant percentage of the internet. The threat actor initiates the scanning process to search for vulnerable devices. The bot then finds an IP address and tries to log in with random passwords. Next, the scanning and listening process communicates with the command and control server, followed by loading malicious code onto the IoT device. The attack is then initiated when the attacker commands the center for the initiation of a DDoS attack. Numerous IP addresses will be used by these IoT devices to submit requests. Major Websites like Twitter and Netflix would go down as a result of this Mirai attack, thus disrupting networks. Bashlite, otherwise known as Gayfgt/Gafgyt, was found in 2014. This botnet instructs Internet of Things devices to initiate DDoS assaults by taking advantage of vulnerabilities in Linux platforms. Primarily, this is an IoT botnet comprised of cameras and mostly infects Linux devices. This may develop further and evolve, resulting in the invasion of new devices. This botnet generates a variety of DDoS assaults, such as flooding TCP and UDP with a junk of data.

3.2 IoT Botnet

A botnet consists of internet-connected robots that function as controlled agents for a hacker, executing instructions from a botmaster (attacker). These botnets operate collectively, resembling an army, and are utilized to target specific destinations or engage in information theft. By infecting interconnected devices with malware and subsequently controlling them via a command-and-control server (C&C), malicious actors or hackers establish botnets. Once a device on a network has been compromised by an attacker, all vulnerable devices within that network are at risk of infection. In our research, we focus on two prominent botnets, Bashlite and Mirai, among the various types available.

3.3 IoT Botnet Life Cycle

The IoT botnet lifecycle has three phases. The Figure.1 represents the three phases of IoT botnet life cycle.

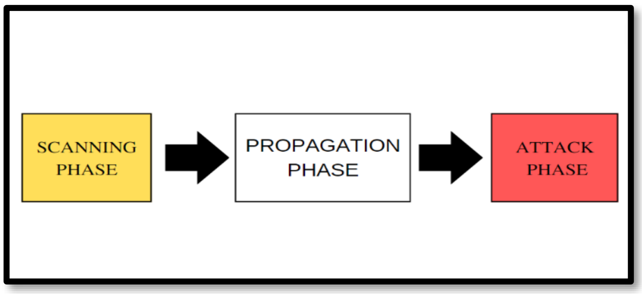

Figure 1. IoT botnet lifecycle phase

1) ***Scanning Phase:*** Botmaster scans vulnerable IoT devices, and after finding a device it starts infecting that device. After the vulnerable device being compromised completely it becomes a bot that communicates with its master.
2) ***Propagation Phase:*** Propagation of appropriate variety of bot is loaded and exhibited. If any existing program is found in the device it kills and loads the new malware program and recruits new bots for expanding the army.

3) ***Attacking Phase:*** It sends a command to the C&C to trigger an attack. The bots will start attacking and perform various attacks like sending spam, stealing data and DDoS attacks etc after receiving the commands from the C&C server.

4. N-BaIoT

4.1 Dataset Description

The dataset N-BaIoT we utilised for our research was proposed by Yair *et al.* (2018) and is intended to identify Mirai and Bashlite botnets. These two recent IoT malicious softwares are present in the dataset, which are members of the Botnet family to infect internet-attached devices and perform attacks like DDoS and cause network disruption. This is a sequential, multivariate dataset contain 89 different files for 9 commercial IoT devices, divided by the benign 9 IoT devices and distinct files for 5 Mirai and Bashlite subattacks. There are 115 features in each file, and each file has a different number of rows or data; overall, there were 7062606 instances throughout all data files. This dataset [1] was generated from real network traffic, by using the 9 IoT devices infected with Mirai and Bashlite and the same 9 devices non-infected (benign) were taken and they used port mirroring to gather raw network traffic data. Switch devices were set up for port mirroring in order to collect and evaluate network traffic. Several access point devices were used to connect these devices to the internet by a WIFI Port mirroring. This has been set up on the switch devices to capture and analyse actual network traffic. The Wireshark tool was used to record the retrieved data. The below table 1 shows the 9 IoT devices that were used, we found the benign of 9 IoT devices and the same IoT devices infected with 5 sub-attacks of Bashlite and 5 sub-attacks of Mirai except (for the 3rd and 7th device) so we found there were a total of 89 distinct data files with varying features.

4.1.1 Scan Attack

The attack described is a common feature of both Mirai and Bashlite. In this type of attack, the botmaster actively seeks out IoT devices with weak configurations that make them vulnerable. Once identified, the attacker proceeds to infiltrate these devices, initiating an attack. Additionally, the attacker scans the network to locate other devices in the vicinity, exploiting any weaknesses found and propagating the malware further. Figure 2 provides a visual representation of this scan attack as observed in both Bashlite and Mirai.

4.1.2 Flooding Attack

This attack is further divided into

- For Bashlite-Junk, Udp, Tcp, Combo
- For Mirai-Ack, Syn, Udp, Plain Udp Flooding attack is simply flooding of packets to disrupt the network flow. This of flooding attacks will mainly be used in Distributed denial of service attacks where flooding of enormous amount of packets to a network from different devices making the users unable to access a particular website which eventually causes disruption in the network flowT he Junk attack entails the transmission of spam data, connecting to an IP and port address and randomly generating strings to different IPs. UDP flooding occurs when a target is overwhelmed with a substantial volume of UDP packets, including control protocols. COMBO attack involves inundating the target with an extensive amount of junk data. PLAIN UDP attack resembles UDP but with a higher transmission rate of user datagram protocol packets per second. TCP attack, observed in Bashlite, floods the target with transmission control protocol requests. ACK and SYN attacks, found in Mirai, operate differently.

Table 2. Nine IoT Devices used

Device ID	Device Name
1	Danmini_Dooebell
2	Ecobee_Thermostat
3	Ennio_Doorbell
4	Philips_B120N10_Baby Monitor
5	Provision_PY_737E_Security_Camera
6	Provision_PY_838_Security_Camera
7	Samsung_SNH_1011_N_Webcam
8	SimpleHome_XCS7_1002_WHT_Security_Camera
9	SimpleHome_XCS7_1003_WHT_Security_Camera

The 10 sub-attacks of Bashlite and Mirai can be divided into two type of attacks Scanning and Flooding Attack

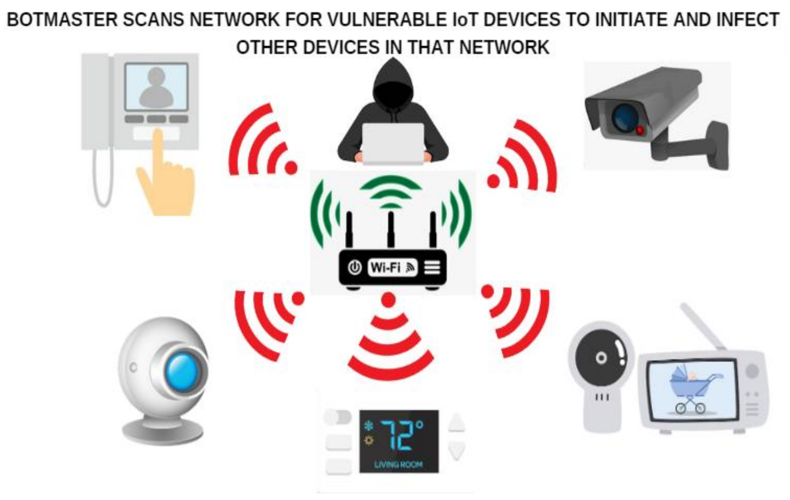
BOTMASTER SCANS NETWORK FOR VULNERABLE IoT DEVICES TO INITIATE AND INFECT OTHER DEVICES IN THAT NETWORK
Wi-Fi
72°
LIVING ROOM

Figure 2. Scan attack in Bashlite and Mirai

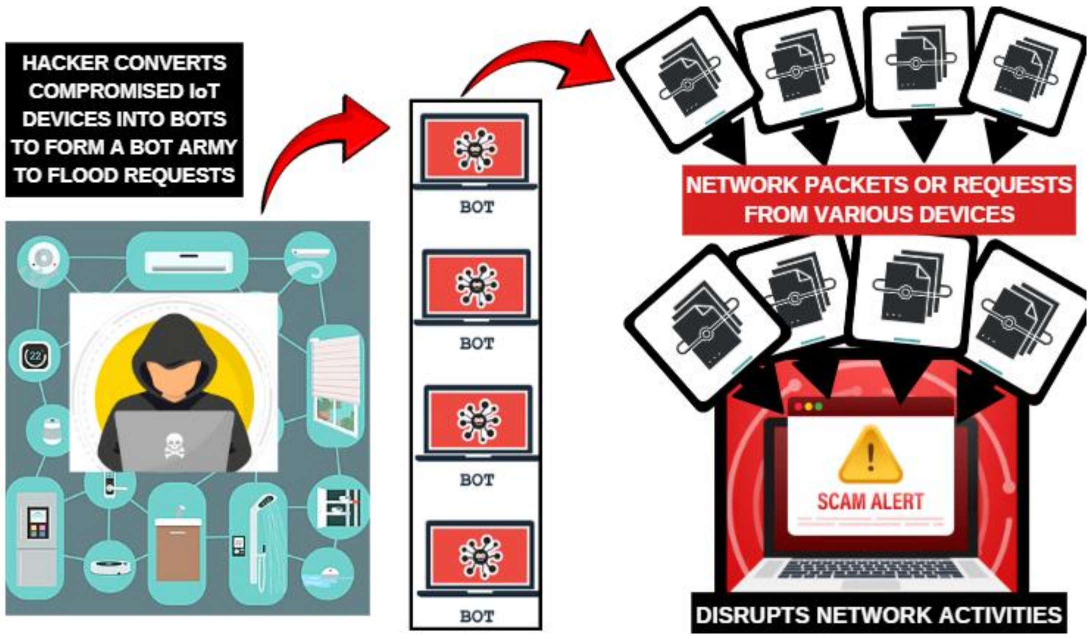
HACKER CONVERTS COMPROMISED IoT DEVICES INTO BOTS TO FORM A BOT ARMY TO FLOOD REQUESTS
BOT
BOT
BOT
BOT
NETWORK PACKETS OR REQUESTS FROM VARIOUS DEVICES
SCAM ALERT
DISRUPTS NETWORK ACTIVITIES

Figure 3. Flooding Attack in Bashlite and Mirai

Table 3. Statistics of the detailed features of the dataset

S.No	To Determine	Statistical Attributes	Aggregated by	Total Number of Attributes
1.	**Packet Size** [Outgoing Packets]	1. Mean 2. Weight	1. Source or Host Internet Protocol 2. Source or Host MAC-Internet Protocol 3. Channel 4. Socket	2 X 4 = 8
2.	**Packet Count**	1. Weight	1. Source or Host Internet Protocol 2. Source or Host MAC-Internet Protocol 3. Channel 4. Socket	1 X 4 = 4
3.	**Packet Jitter** [Time period between the arrival of packets]	1. Mean 2. Variance 3. Weight	1. Network jitter	3 X 1= 3

4.	**Packet Size** [of incoming and outgoing together]	1.Radius 2.Correlation coefficient 3.Magnitude 4.Covariance	1. Channel 2.Socket	4 X 2 = 8
Total Number Of Features= 23 Time instances in which 23 features were collected: 5 Total= 23*5=115				

In an ACK attack, the target's server is flooded with a large number of acknowledgement packets from various devices, causing confusion and busying the server. In a SYN Attack, the attacker inundates the server with malicious SYN packets, prompting the server to reply with a SYN-ACK packet, awaiting an Acknowledgement packet that is never sent by the attacker. This confusion is exacerbated in DDoS attacks, where multiple users send SYN-ACK packets to random destinations, leading to network disruption and the eventual inability of the target server to function. Figure 4 provides a comprehensive depiction of the flooding attack mechanism.

The Table 3 represents the calculation of the 115 attributes, for calculating the outbound packet size, packet count, packet jitter, outbound and inbound packet size they have specific statistical features and aggregated features which are So when calculating we get a total of 23 attributes, the dataset originally was taken in 5-time windows which are 100ms, 500ms, 1.5sec, 10sec, 1minute, represented as L5, L3, L1, L0.1, L0.01in the datasets. For instance HHJIT_L5_Mean represents the mean of the network jitter in 100th milliseconds.

4.2 Feature Description

1 ***Source/Host Ip***-denoted as H. Host internet protocol summarizes recent traffic coming from specific packet's source IP.

2 ***Host/Source - Mac IP***-denoted as MI. This enhances the capacity to distinguish between traffic coming from various gateways and traffic with forged IP addresses.

3 ***Channel-*** denoted as HH, indicates attributes of recent traffic travelling from the source to the destination of this packet between specific sites.

4 ***Network Jitter-*** represented as HH_jit in channel communication this feature is related to the time interval or delay between the transmission and reception of the traffic jitter going from specific packet's source to destination.

5 ***Socket***-denoted as HpHp. An IP address and port number together make up a socket address. The traffic between a certain source and ports that are identified by the source and destination UDP or TCP port numbers. For instance, all traffic sent from 192.142.13.50:80 to 192.157.5.12:14 is the traffic sent from one socket to another.

5. Pre-Processing

We have taken the datasets of 9 IoT devices namely

1. Danmini doorbell
2. Ecobee thermostat room temperature
3. Ennio doorbell
4. Philips b 120N10 baby monitor
5. Provision pt 737E security camera device
6. Provision pt 838 security camera device
7. Samsung snh 1011 n webcam
8. Simple home XCS71002 WHT security camera
9. Simple home XCS71003 WHT security camera

This includes distinct files for 9 devices with independent Bashlite and Mirai subattacks which are 5 for each, along with the benign versions of these devices. We have combined five Bashlite subattacks with benign, five Mirai subattacks with benign, and generated new datasets for nine IoT devices for Bashlite and Mirai. We then added an extra feature which was initially not there in all the datasets called class for classifying the subattacks and benign for Bashlite and Mirai datasets. We have created 9 datasets for bashlite and 7 datasets for Mirai (7th device and 3rd device datasets were not available) so a total of 16 datasets were created for our analysis the detailed data summary of the 16 datasets after merging is given in table 4 and Figure 4. So our new datasets for Bashlite and Mirai consist of

1. Bashlite Dataset-Benign, Combo, Junk, Scan, Tcp, And Udp For 9 Iot Devices
2. Mirai Dataset -Benign, Ack, Scan, Syn, Udp,Udp Plain For 9 Iot Devices

6. Proposed Work

We apply and load our newly preprocessed datasets into a variety of machine learning classifiers for training and testing.

For the nine IoT devices, we were able to differentiatethe sub-attacks of Bashlite and benign (which is Unaffected normal datasets of all 9 IoT devices.) and sub-attacks of Mirai and benign. Figure 5 represents a more complete depiction of our model's process.

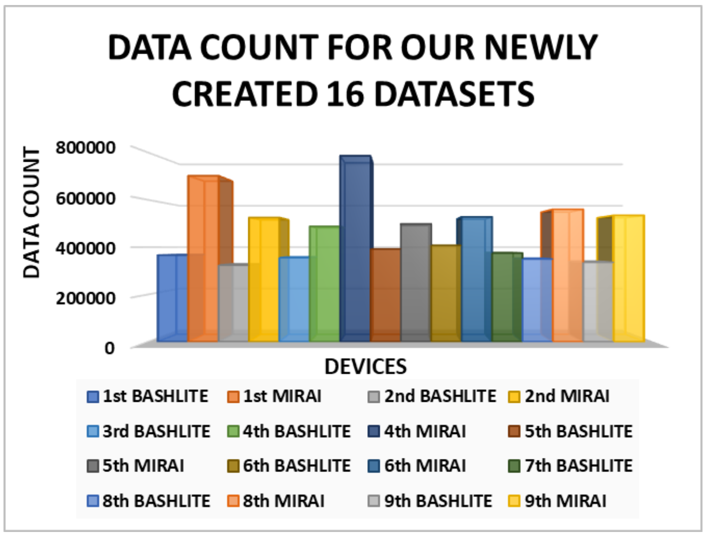

Figure 4. Data Count of newly created Bashlite and Mirai Dataset

Table 4. Data count for newly created Bashlite and Mirai datasets

DEVICE NO	BASHLITE	MIRAI
1	366198	701648
2	323743	525246
3	355500	-
4	487963	785954
5	392250	498164
6	407554	527851
7	375222	-
8	349808	559833
9	335966	534388

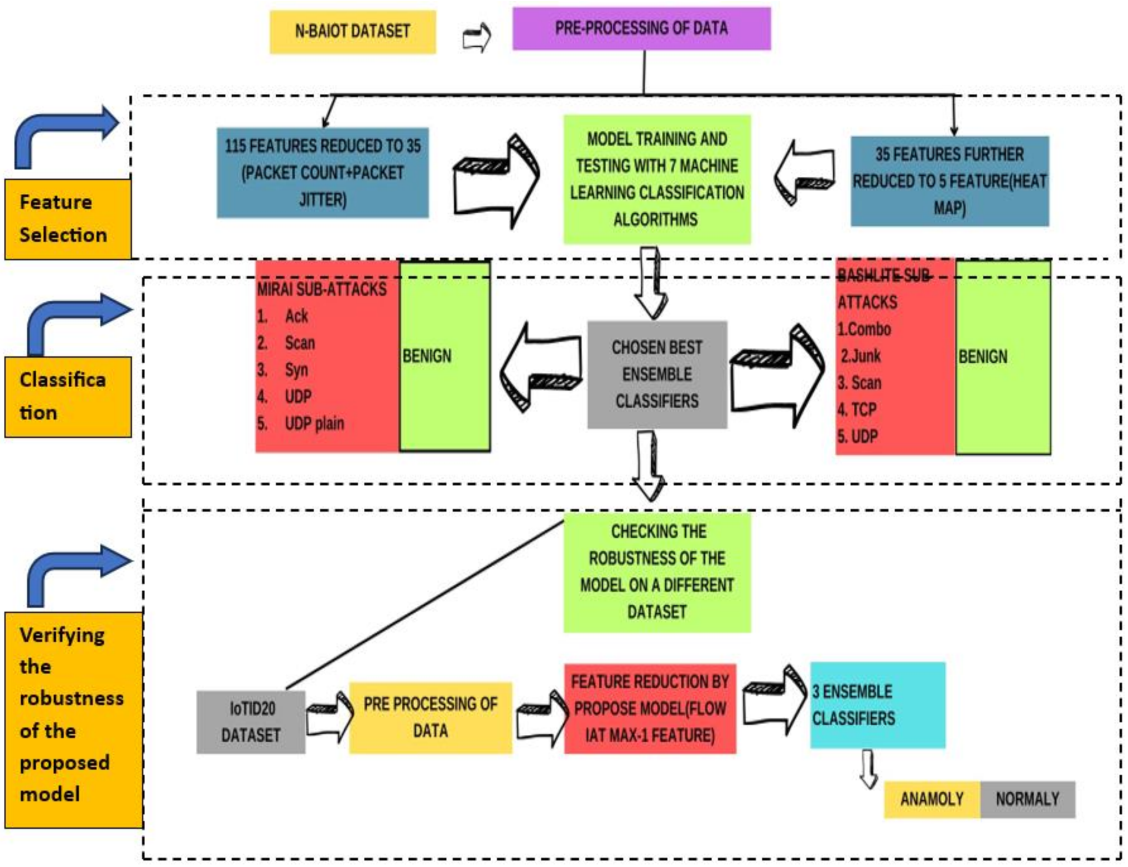
N-BAIOT DATASET
PRE-PROCESSING OF DATA
Feature Selection
115 FEATURES REDUCED TO 35 (PACKET COUNT+PACKET JITTER)
MODEL TRAINING AND TESTING WITH 7 MACHINE LEARNING CLASSIFICATION ALGORITHMS
35 FEATURES FURTHER REDUCED TO 5 FEATURE(HEAT MAP)
Classification
MIRAI SUB-ATTACKS
1. Ack
2. Scan
3. Syn
4. UDP
5. UDP plain
BENIGN
CHOSEN BEST ENSEMBLE CLASSIFIERS
BASHLITE SUB-ATTACKS
1.Combo
2.Junk
3. Scan
4. TCP
5. UDP
BENIGN
Verifying the robustness of the proposed model
CHECKING THE ROBUSTNESS OF THE MODEL ON A DIFFERENT DATASET
IoTID20 DATASET
PRE PROCESSING OF DATA
FEATURE REDUCTION BY PROPOSE MODEL(FLOW IAT MAX-1 FEATURE)
3 ENSEMBLE CLASSIFIERS
ANAMOLY
NORMALY

Figure 5. Workflow of our model

In the proposed workflow, the original N-BaIoT dataset undergoes preprocessing by consolidating it into 16 new data files. Subsequently, training and testing are conducted on these 16 datasets, each containing 115 attributes. Manual feature reduction is then performed by evaluating packet count and packet jitter, resulting in a total of 35 features. Seven machine learning algorithms previously identified using Weka are utilized to classify the 10 sub-attacks (for both botnets) and benign instances across the 16 datasets.

Table 5. Proposed algorithm

```
Algorithm: Feature Reduction and Ensemble Classification for Intrusion Detection

Input: Dataset with 115 features
Output: Reduced feature set and classification results

1. Preprocess the dataset to handle missing values, outliers, and normalization
2. Perform feature reduction:
    a   Combine features based on packet count and packet jitter to reduce from 115 to 35 features.
    b   b. Further reduce the 35 features to a single feature, HH_Jitt_Mean, in 5 time instances, resulting in 5 features.
3. Train and test ensemble classifiers on the reduced feature set:
    a   Select ensemble classifiers (e.g., Random Forest, Random Tree, Random Committee).
    b   Split the dataset into training and testing sets.
    c   Train the ensemble classifiers on the training data.
    d   Test the trained classifiers on the testing data.
    e   Evaluate the performance of the classifiers using accuracy metrics.
4. Analyze results:
    a   Compare the performance of ensemble classifiers across all 16 datasets.
    b   Assess the accuracy of classifications, ranging from 99.1% to 99.9%.
    c   Evaluate the computational efficiency of the reduced feature set compared to the original 115 features.
5. Output the reduced feature set and classification results for further analysis and decision-making.
```

Further dataset reduction is achieved by applying a heat map to the 35 features, revealing that a single feature, HH_jit_mean, across five instances, or five features in total, significantly contributes to classification. Analysis is conducted using the 35 and 5 features with the seven ML classifiers, including random tree, random forest, random committee, j48, rep tree, hoeffding tree, and part. The accuracy of the models is enhanced across the 16 datasets.

Assessment metrics such as correctly categorized instances, erroneously classified instances, and model building time are utilized to analyze and distinguish results. Comparative analysis of the new 16 datasets with 35 and 5 features is performed, demonstrating decreased model development time and reduced complexity as the number of attributes decreases. Tree classifiers exhibit strong performance, particularly the ensemble classifiers random tree, random forest, and random committee, achieving approximately 99% accuracy in both Mirai and Bashlite datasets. To validate the methodology, the model is implemented on a different dataset, the IoTID20 dataset, where 86 features are reduced to a single feature, Flow IAT MAX. Training and testing using the three ensemble classifiers successfully differentiate anomalies from normal instances, with high detection rates.

6.1 Feature Selection

6.1.1 Manual Reduction of Features

We started with the raw N-BaIoT dataset, comprising 115 features. Through manual analysis, we endeavored to condense these attributes to 35, focusing on key metrics such as packet count and packet jitter. Our selection of these metrics was influenced by the prevalence of flooding attacks within the dataset; among the ten identified sub-attacks, eight were classified as flooding attacks. Flooding attacks are characterized by the dissemination of a large volume of unwanted control and data packets across a network. Consequently, malicious flooding attacker nodes typically generate a significantly higher volume of packets compared to legitimate nodes. By evaluating the packet count, we aimed to distinguish between different types of flooding attacks, including junk, TCP, UDP, and combo attacks in the case of Bashlite, and ACK, SYN, UDP, and plain UDP attacks in the case of Mirai. Furthermore, we assessed the packet jitter value, which indicates the time intervals between the arrivals of packets. The continuous influx of unwanted packets from malicious nodes can contribute to an increase in network jitter [23-24]. This elevated jitter value serves as an indicator of flooding attacks. By leveraging packet count and packet jitter values, we aimed to effectively classify all ten identified attacks within the dataset. We did the manual feature reduction by determining the values of packet count and packet jitter by

1. For **PACKET COUNT** (4 X 1 =4)

- ➢ Host IP(H) ,Host Mac IP(MI),Channel(HH),Socket(HpHp)= 4 features
- ➢ Weight=1 feature

So total 4 features will be taken in the packet count

Table 6. 35 features-Manual Reduction

MI_dir_L5 weight	H_L0.1_weight	HH_jit_L3_weight	HH_jit_L0.01_mean	HpHp_L1_weight
MI_dir_L3 weight	H_L0.01_weight	HH_jit_L1_weight	HH_jit_L5_ variance	HpHp_L0.1_weight
MI_dir_L1 weight	HH_L5_weight	HH_jit_L0.1_weight	HH_jit_L3_ variance	HpHp_L0.01_weight
MI_dir_L0.1 weight	HH_L3_weight	HH_jit_L0.01_weight	HH_jit_L1_ variance	
MI_dir_L0.01 weight	HH_L1_weight	HH_jit_L5_ mean	HH_jit_L0.1_ variance	
H_L5_weight	HH_L0.1_weight	HH_jit_L3_ mean	HH_jit_L0.01_ variance	
H_L3_weight	HH_L0.01_weight	HH_jit_L1_ mean	HpHp_L5_weight	
H_L1_weight	HH_jit_L5_weight	HH_jit_L0.1_ mean	HpHp_L3_weight	

2. for **PACKET JITTER** (1 X 3 = 3)

- Network jitter (HH_JIT)=1
- Mean, Variance, Weight=3

So total 3 features will be taken in the packet jitter

3+4=7 X (By 5 Time Windows) = 35+1 (Attack Class)

These 7 features which we found from packet count and jitter are multiplied by 5-time instances that is L5, L3, L1, L0.1, and L0.01 which represent 100ms,500ms,1.5s,10s, and 1min respectively. So a total of 35 features plus an attack label which is shown in table 6 were taken and then we applied 7 ML classifiers to it by determining the packet count and packet jitter and we were able to get good accuracy results (Table 6).

6.1.2 Heat Map

A correlation heatmap, like a regular heatmap, is aided by a color bar to make data more readable and understandable. With the help of correlation we can understand the significance or importance of each feature in our dataset. After reducing the 35 features manually, we applied a heat map to it to further reduce the 35 features. There are two conditions in a heat map Correlation: -

1. Feature have high correlation with the label.
2. Feature have low correlation with other features.

We have taken the 2nd given condition, in which features should have a low correlation with other features, when applying a heat map to 35 features we found that a single feature HH_JITT_MEAN(Mean Jitter) features in 5-time instances have shown a low correlation with other features as you can see in Figure 8 which is the heat map of the 4th device (Bashlite) which contained 35 features, the correlation scale of features are represented from low to high with features with dark colours representing low correlation and the features with light colours representing high correlation. We have executed this in Jupyter notebook with Seaborn which is a Python library for data visualization that is based on matplotlib and found that a single feature called HH_jitt_mean feature in 5 time instanced L5, L3, L1, L0.1, and L0.01 has shown dark violet colour as we can view this clearly in Figure 7, this represents that this particular feature in 5-time instances have a low correlation with other features. By considering these 5 features of HH_jitt_mean we were able to get similar accuracy with comparatively lesser build time and this particular feature has contributed a lot of information to our classification.

6.2 Supervised Learning

When we give a machine a series of inputs marked with their associated outputs, we train the machine with that collection of data. The machine understands the patterns for the inputs given and the relationship between the inputs and outputs. When new input is given, the model tries to predict with the help of the patterns. Each and every data will be tagged with a label. So that the machine will be trained based on the patterns to give the desired output. For botnet detection supervised learning will be very suitable for classifying and distinguishing bashlite, Mirai, and normal. By doing this we can find the network anomalies. We have used 7 machine learning algorithms with 80 % and 20 % split for training and testing. In our previous work we trained and tested several machine learning classifiers for full the dataset with 115 features and dataset reduced using PCA ,We have found that the following seven machine learning algorithms had performed well in our previous work [22] so we used those 7 algorithms in this work.

6.2.1 Random Tree

An ensemble classifier called Random Tree generates a large number of individual learners. They are a blend of two machine learning techniques that are currently in use, combining concepts from Random Forest with single decision trees. In order to create a random collection of data for the purpose of creating a decision tree, it uses the bagging idea. After receiving the input feature vector, the random tree classifier classifies it using each tree in the forest and returns the class label that has the most "votes." When a tree is developing, just a random subset of all characteristics is taken into consideration at each node, and the best split for that subset is determined, as opposed to constantly figuring out the optimal split for every node. In our study, K randomly selected features are taken into consideration at each node when building a tree. Random tree performed no pruning, we set the batch size as 100 and the maximum depth of the tree, as 0 which is unlimited. The size of the tree is 487.

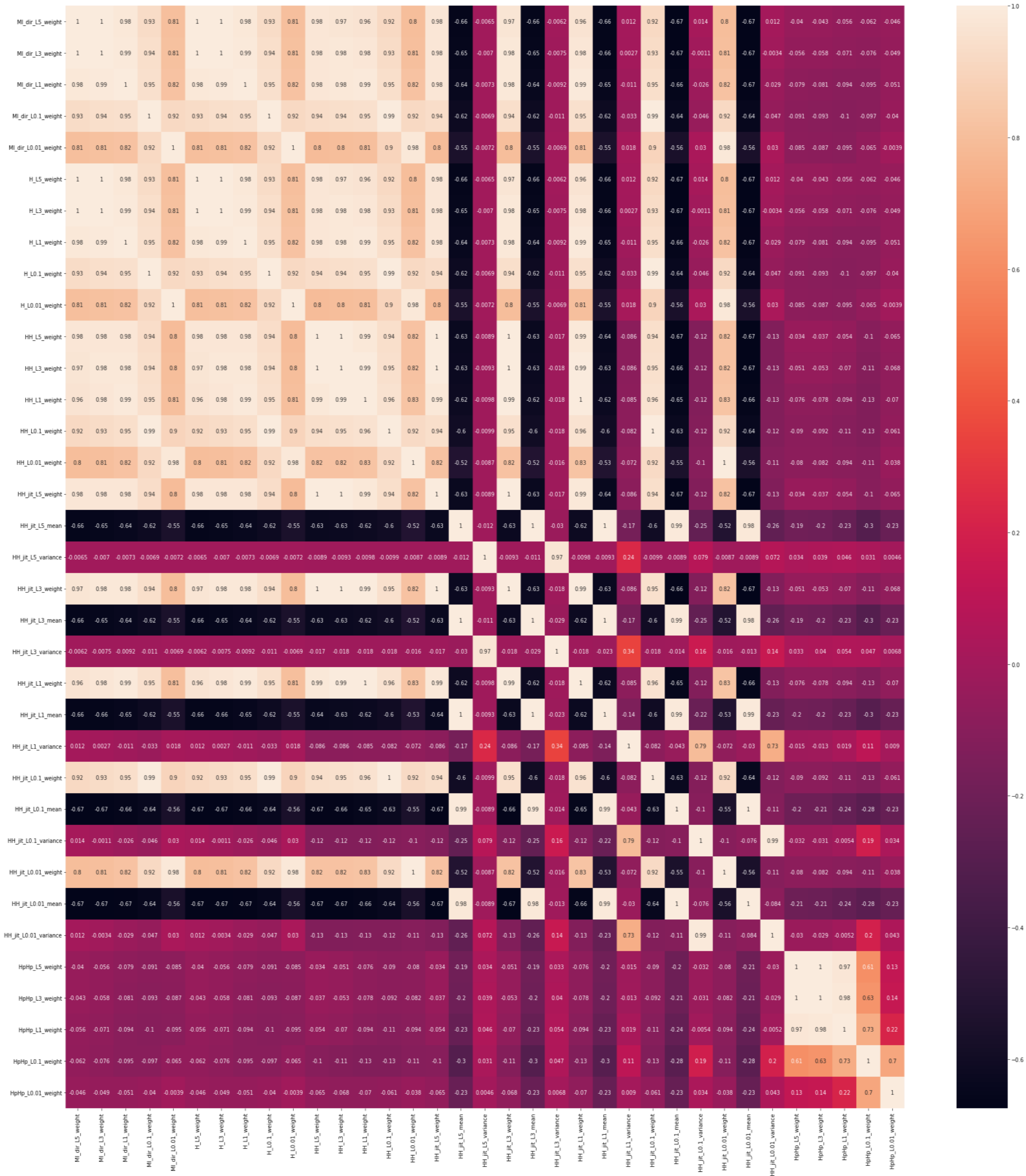

Figure 6. Heat map applied to 35 features

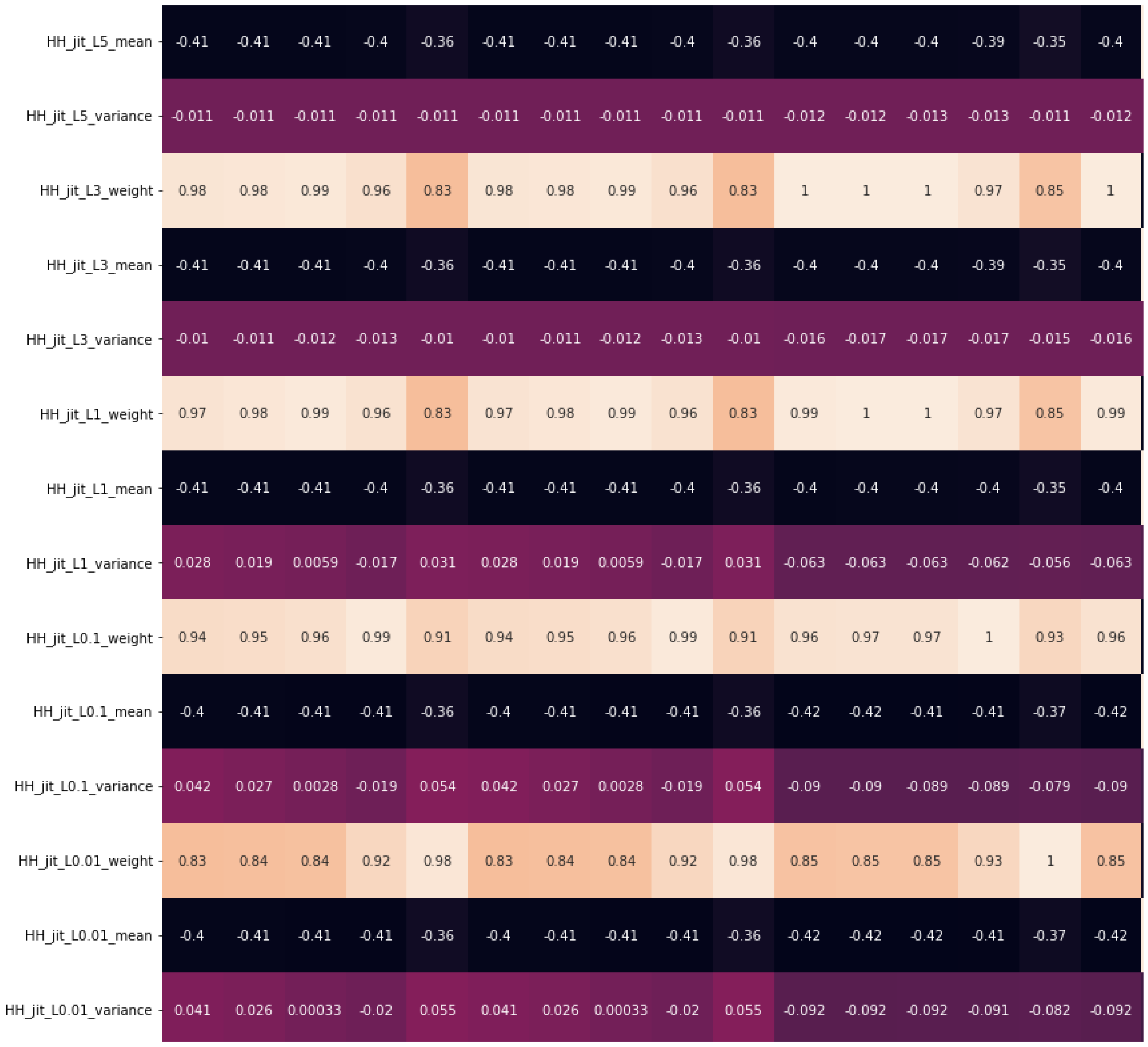

Figure 7. Feature HH_JITT_MEAN in 5-time instances shows that it has a low correlation with other features

6.2.2. Random Forest

Random forest is also an ensemble learning technique classifier model, it randomly selects a subset of features for each tree and uses them for training. It uses (bootstrapping+ aggregating), bootstrapping is used to ensure we are not using the same data and aggregating is used where random feature selection is done for the dataset. Row sampling and feature sampling are done and given to each decision tree and constructs multiple decision trees based on voting, it produces the final outcome. In our work random forest had performed with 100 iterations and base learners, the batch size is set to 100, The maximum depth of the tree is 0 for unlimited. The number of n estimators that is number of trees are 80.

6.2.3. Hoeffding Tree

A Hoeffding tree is a decision tree which has the

Capacity to learn from vast amounts of data. It makes use of the fact that a modest sample size will be adequate to determine which splitting property is most successful. Hoeffding bound, a mathematical idea used that quantifies the quantity of data needed to forecast certain statistics within a given attribute's accuracy, provides evidence for this. They have a theoretically appealing feature that other cumulative decision tree learners do not have. The leaf prediction strategy used is naïve Bayes adaptive, the batch size is set to 100, and naïve Bayes Prediction. Threshold is the quantity of occurrences a leaf has to see before enabling naive Bayes to generate predictions, which is set to 0. The splitting criterion is information gain.

6.2.4. Rep Tree

Tree learner who makes quick decision by creating a decision tree based on information gain and to avoid the overfitting problem the decision tree faces, this tree prunes it with reduced-error pruning.The size of the tree is 217.Batch size is set to 100 and The maximum tree depth is -1 for no restriction.

6.2.5. Part

This will generate a PART decision lists, which are organized set of rules. Separate-and-conquer strategy is used. One by one, the new data is compared to every rule in the list, and each value is given the class of the first matching rule. Every iteration creates a partial C4.5 decision tree and turns the leaf that is considered "best" into a rule. Number of rules are 89. Batch size is 100. The seed used for randomizing the data when reduced-error pruning is used is set to 1.

6.2.6. Random Committee

The random committee method is an ensemble of randomizable base classifiers that builds multiple Base classifiers using distinct random number seed values based on the same data. The average of the predictions made by each base classifier is used to determine the final classification outcome. The Size of the tree is 507, batch size is 100 and the random number seed to be used is 1.The base classifier used is random tree.

6.2.7. J-48

J48 is an algorithm to produce decision tree that is generated by C4.5 algorithm. It is a statistical classifier where each node of the decision tree is build using the concept of entropy, chooses attributes that are in the most successful splits subsets. The best attribute to split on to achieve the highest classification accuracy is the attribute with the most information. The j-48 algorithm selects the feature of the data that splits its sample set into smaller groups at each node most efficiently. The feature with the largest normalised information gain is selected in order to determine the decision. The batch size was taken to be 100.The trees are pruned with seed value as 1. Number of Leaves are 129 and Size of the tree is 257.

6.2.8 Ensemble Classifiers

Ensemble learning obtains greater prediction accuracy by combining multiple models together. This will be more effective than using a single algorithm, which enables improvising machine learning outcomes. This method outperforms a single model in terms of detection accuracy. This methodology is used for generating multiple base classifiers from which a new classifier is derived that outperforms a single classifier. The hyper parameters of these learning models may differ. The three base classifiers in our work are Random tree which is an ensemble method that results in the combination of two ML models that is random forest and decision tree, Random forest uses a bagging technique where Bootstrapping and Aggregation takes place. and random committee is a voting-based ensemble classifier. These 3 classifiers had overall outperformed by obtaining detection rates in the range of 99.1% to 99.9% for all 9 IoT devices (both Bashlite and Mirai). The combination of multiple models proved in providing better predictive accuracy rates.

6.3 Robustness of the Proposed Model

To validate the robustness of our approach, we applied our proposed model to a different dataset, the IoTID20 dataset. This dataset was generated using the Wireshark tool, capturing network traffic from two IoT devices, namely Ezviz and Skt Ngu wifi cameras, along with other devices connected to a home router, such as mobile phones and laptops. The dataset includes attack categories such as DoS, Mirai, MITM (Man-in-the-Middle) attack, and Scan, comprising 83 flow-based features. The infected devices were connected to an access point, recorded in pcap format by Wireshark. The dataset features binary classification, distinguishing normal traffic from anomalies, with attack classes including Mirai and Scan. Our focus was on binary classification to distinguish normal nodes from attacker nodes. We applied our feature reduction approach, initially involving 80 network features and three labels, which were then reduced to a single feature named FLOW IAT MaX. This feature represents the maximum inter-arrival time between packets in a flow, indicating the maximum delay between packet transmissions. Our rationale for selecting this feature relates to identifying flooding attackers in nodes, where a high volume of malicious packets increases jitter values.

We leveraged similar features associated with maximum delay to capture the worst-case delay scenario in each node, consequently increasing jitter values. Subsequently, we employed our top three ensemble classifiers for classification, achieving favourable detection rates with reduced model building time.

7. Experimental Results and Discussion

In order to determine which classification algorithms work best, we have conducted a comparison study using the 35 and 5 features on pre-processed N-BaIoT datasets for botnet detection. Our objective is to decrease the extensive 115 attributes manually to 35 and then further reducing it to 5 by applying a heat map to it, We have then applied 7 ML classifiers in the new 16 pre-processed N-BaIoT datasets of 9 IoT devices to these two cases by doing a comparative analysis of manually reduced 35 features and further reduced 5 features, by doing this we can examine how well the 7 machine learning algorithms had performed on the reduced 35 and 5 features

7.1 Performance Metrics

The experiment was done on a laptop using a 1.20GHz Intel (R) Core (TM) i3-1005G1 processor and 8.00 GB RAM. We have used jupyter notebook Seaborn which is a Python library for data visualization that is based on matplotlib for getting a heat map. For implementing the 7 classification algorithms we have used python and Weka, which is a known application that offers a graphical user interface for convenience and a variety of visualisation tools and algorithms for data analysis. Here, we've made advantage of the explorer option, which offers a data exploration for machine learning classification. Evaluation parameters that are included in our research include cases that are correctly classified instances (true positives) and incorrectly classified instances (false positives) along with the time used to develop the model. We selected the building time as one of the important determining parameter, as there are 115 of features in N-BAIOT resulting in difficulties in time computation. Lesser processing time results in lesser computational complexity. For assessment, the confusion matrix is also considered.

7.1.1 Perforomance Metrics for 9 Iot Devices for both Bashlite and Mirai Botnets

From our last analysis, we have found 7 algorithms Random Tree, Random Forest, Random Committee, J48, Rep Tree, and Part had overall performed extremely well in all 9 IoT devices for both bashlite and mirai.

Table 7. 88 features in the IoTID20 dataset

Protocol	Flow IAT Mean	Bwd Header Length	Fwd Packets/S	Fwd Packet Length Mean	Fwd IAT Min	SYN Flag Count	Fwd Act Data Phts
FlowID	Flow IAT Std	Bwd Packets/S	Bwd Packet/Bulk Avg	Fwd Packet Length Std	Bwd IAT Total	RST Flag Count	Fvd Seg Size Min
Tota lFwd Packet	Flow IAT Max	Packet Length Min	Bwd Bulk Rate Avg	Bwd Packet Length Max	Bwd IAT Mean	PSH Flag Count	Idle Mean
Total Bwd Packets	Flow IAT Min	Packet Length Max	Subflow Fwd Packets	Bwd Packet Length Min	Bwd IAT Std	ACK Flag Count	Idle Std
Total Length Of Fwd Packet	Fwd IAT Total	Packet Length Mean	Sub flow Fwd Bytes	Bwd Packet Length Mean	Bwd IAT Max	Dewn/Up Ratio	Idle Max
Total Length Of Bwd Packet	Fwd IAT Mean	Packet Length Std	Sub flow Fwd Bytes	Fwd Packet Length Std	Bwd IAT Min	Average Packet Size	Idle Min
Fwd Packet Length Max	Fwd IAT Std	Packet Length Variance	FWD Init Win Bytes	Fow Bytes/S	Fwd PSH fags	Fwd Segment Size Avg	Category
Fwd Packet Length Min	Fwd IAT Max	FIN Flag Count	Bwd Init Win Bytes	Flow Packets/S	Fwd Header Length	Bwd Segment Size Avg	Sub-Category
Bwd PSH Flags	Fwd URG Flags	Bwd URG Flags	URG Flag Count	CWR Flag Count	Ece Flag Count	Fwd Bytes/Bulk Avg	Label
SRC PORT	DST IP	DST PORT	PROTOCOL	TIME STAMP |	FLOW DURATION	Fwd Packets/Bulk Avg	Bwd Bytes/Bulk Avg
Bwd Packet/Bulk Avg	Fwd Ack Data Packets	Active Min	Active Mean |	Active Max	Active Std	Src IP	Fwd Bulk Rate Avg

In terms of correctly classified and incorrectly classified instances we have found 3 ensemble classifiers random tree, random forest, and the random committee had given 99% accuracies overall in all devices for both the botnets and in terms of time taken, we have inferred that all algorithms had performed comparatively in lesser time in 5 features when compared to 35 features which thus has even simplified the computational complexity the random tree had overall performed exceptionally well in all correctly and incorrectly identified instances along with the time required for building the model it has taken lesser time to build for both the viruses applied in 35 as well as the 5 which features.

For Bashlite botnet for 9 IoT devices when compared with 35 and 5 features given in table 3 and 4, random tree,random forest, Rep tree, Part, Random committee and J48 performed well with 99% accuracies. Random forest had an highest accuracy rate of 99.986% and Hoeffding tree secured 89.7% accuracy as the lowest accuracy. In terms of time taken to build model from table 5 the time taken to build random tree was the lowest when compared to all the algorithms. In our evaluation of Mirai botnet performance across 9 IoT devices, we compared the effectiveness of utilizing 35 and 5 features, as presented in figures (8-13) and figures (14-19).

Graphical Representation of Performance of 3 Ensemble Classifier in Our Work for Bashlite

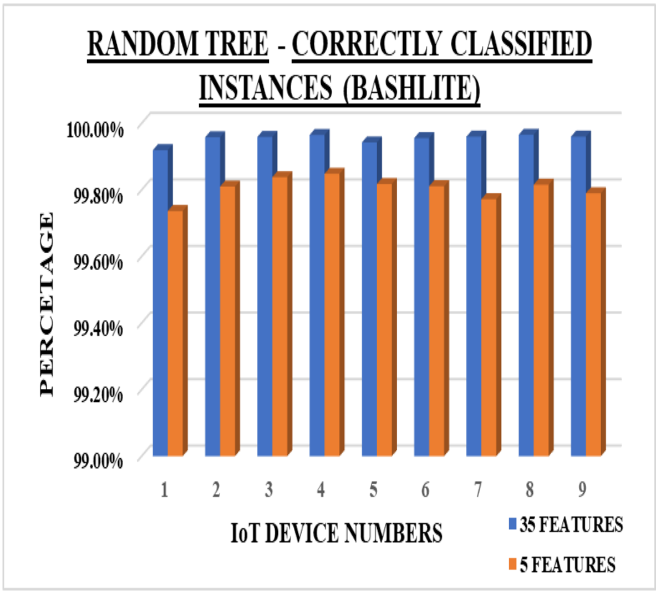

Figure 8. Correctly Classified Instances for Random Tree (Bashlite)

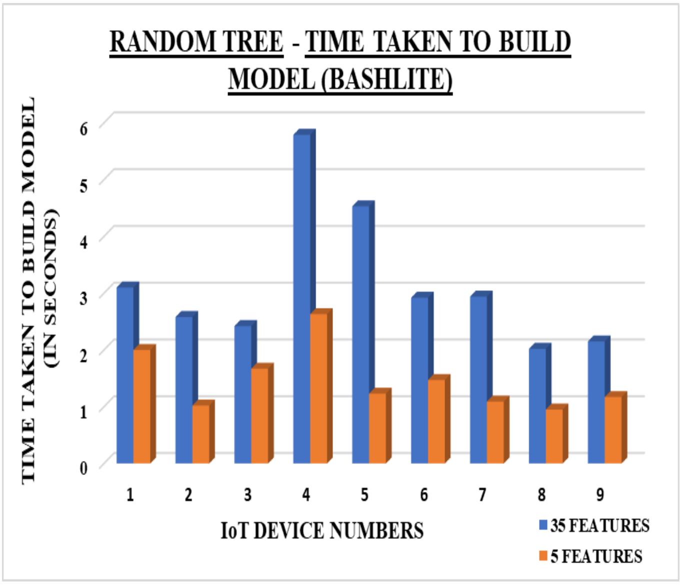

Figure 9. Time Taken to Build Model for Random Tree (Bashlite)

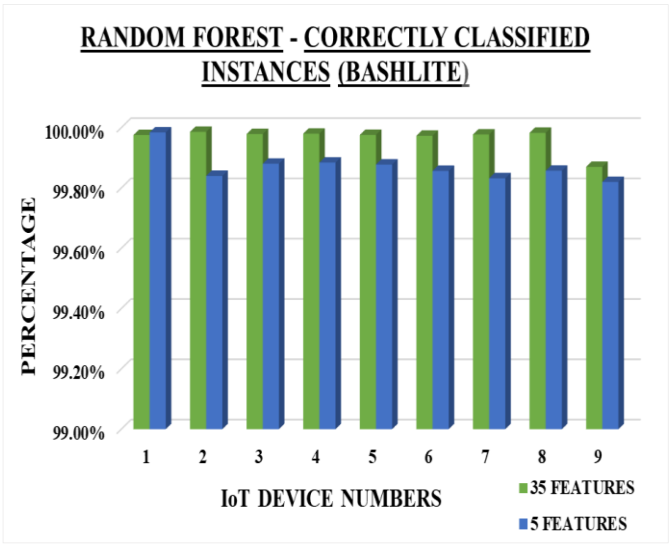

Figure 10. Correctly Classified Instances for Random Forest (Bashlite)

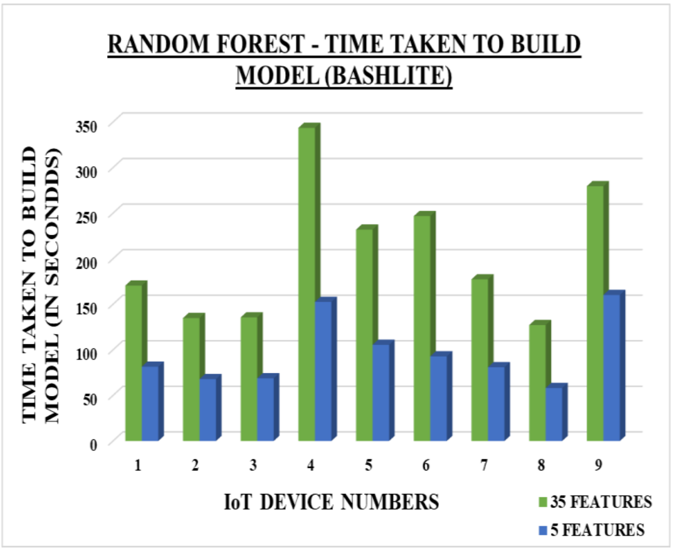

Figure 11. Time Taken to Build Model for Random Forest (Bashlite)

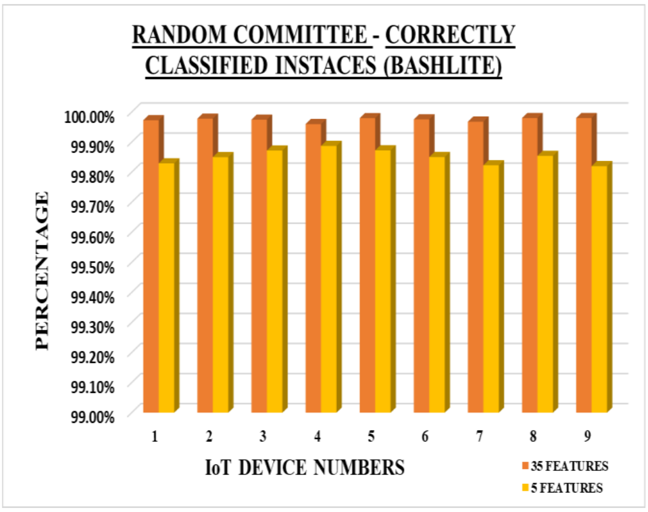

Figure 12. Correctly Classified Instances for Random Committee (Bashlite)

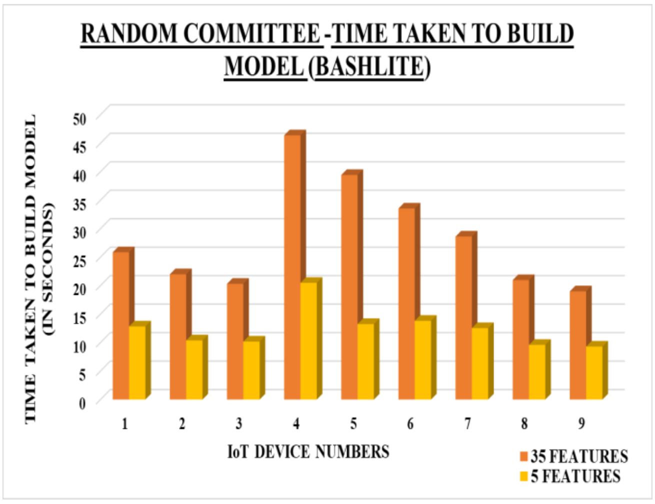

Figure 13. Time Taken to Build Model for Random Committee (Bashlite)

Graphical Representation of Performance of 3 Ensemble Classifier in Our Work for Mirai

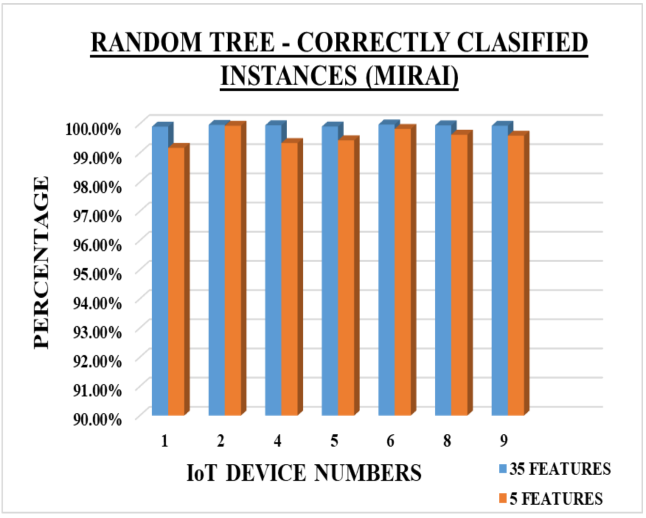

Figure 14. Correctly Classified Instances for Random Tree (Mirai)

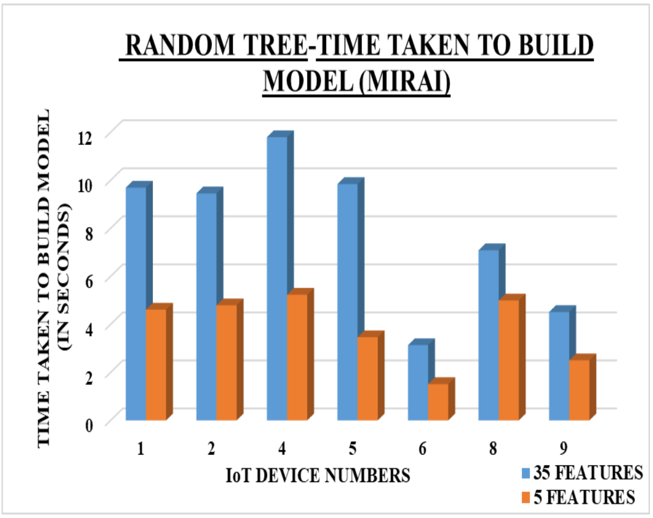

Figure 15. Time Taken to Build Model for Random Tree (Mirai)

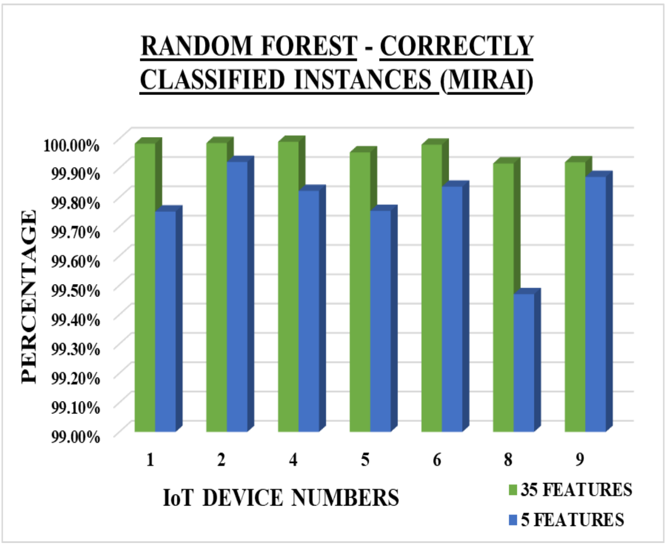

Figure 16. Correctly Classified Instances for Random Forest (Mirai)

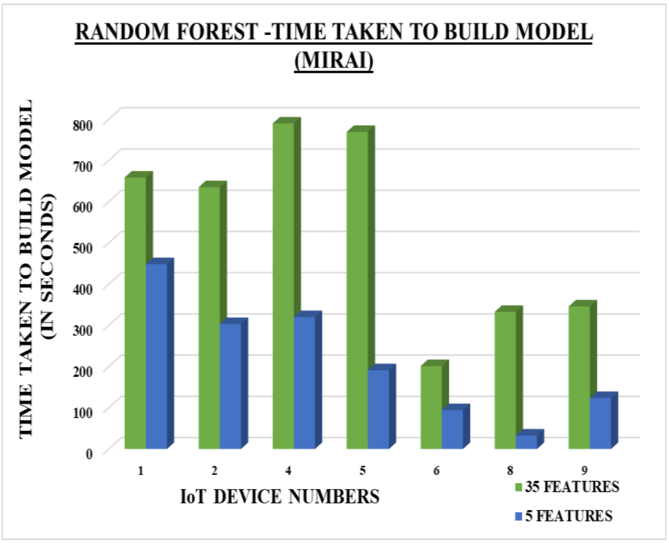

Figure 17. Time Taken to Build Model for Random Forest (Mirai)

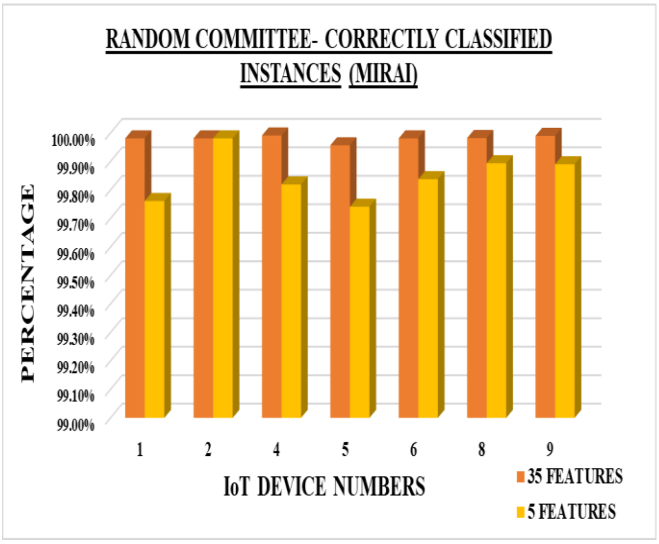

Figure 18. Correctly Classified Instances for Random Committee (Mirai)

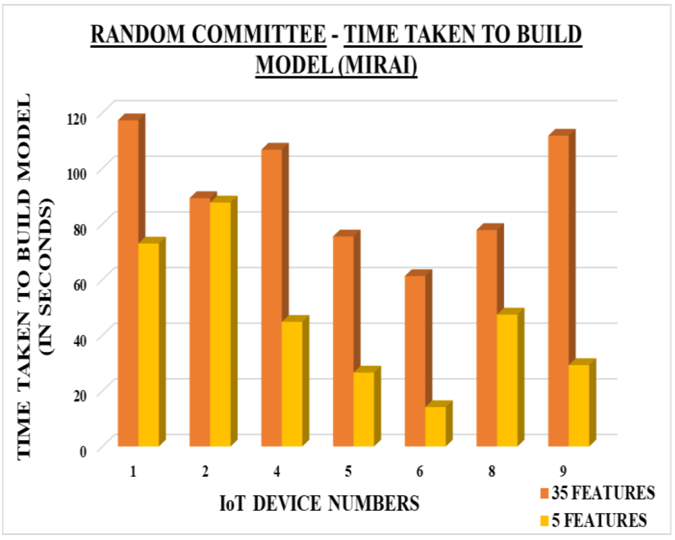

Figure 19. Time Taken to Build Model for Random Committee (Mirai)

Random tree, random forest, and random committee consistently demonstrated strong performance across all 9 devices, with random committee achieving the highest accuracy of 99.99% and Hoeffding tree registering the lowest accuracy of 74.56%. Figure 9 illustrates that random tree exhibited the shortest model building time.

Our comparative analysis across 16 datasets revealed the consistent effectiveness of three ensemble algorithms. Graphical representations of their performance in terms of correctly classified instances and model building time for both Bashlite (Figures 8-13) and Mirai (Figures 14-19) datasets are provided. Random tree, random forest, and random committee consistently outperformed, achieving accuracies ranging from 99.1% to 99.99%. Additionally, random tree demonstrated the shortest model building time across all 9 IoT devices for both botnets.

7.1.2 Algorithm Complexity

Assumptions: n = 80% of the total no of training samples, m =No of features, k' = No of trees, Depth of the tree=0 which is unlimited.

The computational complexities, both time- and space- complexities, for the 3 best-performed machine learning algorithms were analyzed. The complexity of Random Forest and Training Time Complexity is O(n*log(n)*d*k), testing time complexity is O(m*k') and space complexity is O(depth of tree *k)where k'=number of Decision Trees here the no of decision trees is 80 n is the 80% of total number of samples mentioned in table 9.9 and m is the depth of the tree that is 0 ,unlimited. m is the number of features. We have compared with the complexity of the existing work and complexity of the proposed work with 35 and 5 features. Random tree algorithm Training Time Complexity is O(n*log(n)*m) ,testing time complexity is O(m) and space complexity is O(depth of tree) where n is the 80% of total number of samples mentioned in table 10 and m is the depth of the tree that is 0 ,unlimited. m is the number of features we have compared with the complexity of the existing work and complexity of the proposed work with 35 and 5 features. The complexity of Random committee is mentioned in Figures 18 and 19. The Training Time Complexity is O(n*log(n)*d*k) ,testing time complexity is O(m*k') and space complexity is O(depth of tree *k)where k'=number of Decision Trees, where the no of decision trees is 507 ,n is the 80% of total number of samples mentioned in table 9.9 and m is the depth of the tree that is 0 ,unlimited. m is the number of features we have compared with the complexity of the existing work and complexity of the proposed work with 35 and 5 features. Thus comparing the proposed work with the existing works and also with other peer algorithms, random tree algorithm's Training time, Testing time and Space complexity is less when compared with other algorithms used in the literature without the feature selection process. Thus our proposed feature selection process has significantly reduced the complexity and also achieved improved accuracy.

7.1.3 Performance Metrics for IOTID20 Dataset

In order to check the robustness and verify the logic of our study, We took the IOTID20 dataset which initially had 88 features we reduced it to a single feature called flow IAT max and applied the three best ensemble algorithms which worked well in our datasets we then classified the normal data and attack data from table 11, From Figure 20 we can see that random forest had performed with 95.99% accuracy while random committee and random tree had performed with 95.96%.Thus random tree had performed well with lesser time when compared to other.

Table 8. Complexity Analysis of Random Forest

Random Forest Complexity	Existing Work	Proposed work with 35 features	Proposed work with 5 features
Train Time Complexity=o(k'*n*(log(n)*m)	O(80*n*log(n)*115)	O(80*n*log(n)*35)	O(80*n*log(n)*5)
Test Time Complexity= O(m*k')	O(115*80)	O(35*80)	O(80*5)
Space Complexity= O(k'*depth of tree)	O(80*0)	O(80*0)	O(80*0)

Table 9. Complexity Analysis of Random Tree

Random Tree Complexity	Existing Work	Proposed work with 35 features	Proposed work with 5 features
Train Time Complexity=o(k'*n*(log(n)*m)	O(n*log(n)*115)	O(n*log(n)*35)	O(n*log(n)*5)
Test Time Complexity= O(m*k')	O (115)	O(3)	O(5)
Space Complexity= O(k'*depth of tree)	O (80*0)	O(80*0)	O(80*0)

Table 10. Complexity Analysis of Random Committee

Random Committee Complexity	Existing Work	Proposed work with 35 features	Proposed work with 5 features
Train Time Complexity=o(k'*n*(log(n)*m)	O(507*n*log(n)*115)	O(507*n*log(n)*35)	O(507*n*log(n)*5)
Test Time Complexity= O(m*k')	O(115*507)	O(35*507)	O(5*507)
Space Complexity= O(k'*depth of tree)	O(507*0)	O(507*0)	O(507*0)

Table 11. Comparison of various algorithms

Algorithm	**Correctly Classified Instances**	**Time Taken To Build**
Random Forest	95.99%	38.941
Random Committee	95.96%	110.75
Random Tree	95.96%	2.58

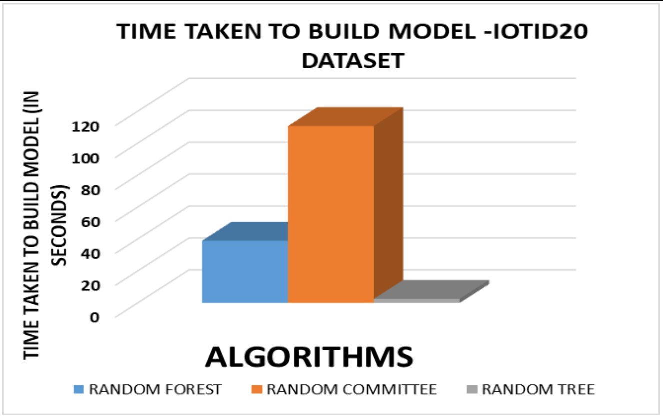

Figure 20. Time Taken to Build Model for IOTID20

8. Conclusion

IoT is rapidly evolving everywhere in this world. These devices talk with each other and communicate using the internet. There are numerous benefits like improved performance, reduced cost, improved data collection, and customer engagement but when considering security they have low security and people don't consider security issues as a major threat when buying these devices, so an effective detective mechanism is needed, to implement this supervised learning with labelled data are given to the machine to learn various patterns of the attack behaviours and are made to classify them as attack or normal. In this analysis, we used 16 datasets of 9 different IoT devices infected with Mirai and Bashlite. We pre-processed and found 7 better performing machine learning algorithms for classification. Initially our dataset had 115 features we tried to reduce to 35 with the values of packet count and packet jitter and we further reduced that 35 to a single feature HH_Jitt_Mean in 5 time instances so 5 features considered in our study out of the original 115 features in the original dataset. Considering all the 115 features were time-consuming and had computational complexity. By reducing to 35 and 5 features, the time and space complexity is reduced and were able to achieve results much quicker with less complexity. We have found that three ensemble classifiers outperformed well in all 16 datasets with accuracies ranging from 99.1%to 99.9%. In terms of time complexity, we found that the random tree algorithm was executed in lesser time. In order to check the robustness of our proposed model we verified the logic of our work in a different dataset called IoTID20 which had 88 features and then we reduced it to a single feature and applied our 3 best ensemble classifiers and achieved 95% accuracy. By implementing the proposed model in a different dataset we are still able to classify with good accuracies, with this we can reduce the danger of botnet attacks on the IoT devices which are in use today.

References

[1] P.L.S. Jayalaxmi, R. Saha, G. Kumar, M. Conti, T.H. Kim, Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey. IEEE Access, 10, (2022) 121173-121192. https://doi.org/10.1109/ACCESS.2022.3220622

[2] M. Yair, B. Michael, M. Yael, M. Yisroel, B. Dominik, S. Asaf, E. Yuval, N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders. IEEE Pervasive Computing, 13(9), (2018) 12-22. https://doi.org/10.1109/MPRV.2018.03367731

[3] M. Elrawy, A. Awad, H. Hamed, Intrusion detection systems for IoT-based smart environments: a survey. Journal of Cloud Computing, 7(1), (2018) 1-20. https://doi.org/10.1186/s13677-018-0123-6

[4] J. King, A.I. Awad, A distributed security mechanism for resource-constrained IoT devices. Informatica (Slovenia), 40(1), (2016) 133–143.

[5] M. Weber, M. Boban (2016) Security challenges of the internet of things In: 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, Croatia. https://doi.org/10.1109/MIPRO.2016.7522219

[6] A.A. Gendreau, M. Moorman (2016) Survey of intrusion detection systems towards an end to end secure internet of things. In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, Austria. https://doi.org/10.1109/FiCloud.2016.20

[7] A. Ahmadian Ramaki, A. Rasoolzadegan, A. Javan Jafari, A systematic review on intrusion detection based on the hidden markov model. Statistical Analysis and Data Mining: The ASA Data Science Journal, 11(3), (2018) 111–134. https://doi.org/10.1002/sam.11377

[8] G. Kumar, K. Kumar, M. Sachdeva, The use of artifcial intelligence based techniques for intrusion detection: a review. Artificial Intelligence Review, 34(4), (2010) 369–387. https://doi.org/10.1007/s10462-010-9179-5

[9] R. McKay, B. Pendleton, J. Britt, B. Nakhavanit Machine learning algorithms on botnet traffc: ensemble and simple algorithms. In: Proceedings of the 2019 3rd International Conference on Compute and Data Analysis. ACM, (2019) 31–35. https://doi.org/10.1145/3314545.3314569

[10] A. Patcha, J.M. Park, An overview of anomaly detection techniques: existing solutions and latest technological trends. Computer Networks, 51(12), (2007) 3448–3470. https://doi.org/10.1016/j.comnet.2007.02.001

[11] M. Sabhnani, G. Serpen, Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. In: Proc. of International Conference on Machine Learning: Models, Technologies, and Applications, 1, (2003) 209–215.

[12] A. Jain, R. Duin, J. Mao, Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), (2000) 4–37. https://doi.org/10.1109/34.824819

[13] S. Saraswathi, G.R. Suresh, J. Katiravan, False alarm detection using dynamic threshold in medical wireless sensor networks. Wireless Networks, 27, (2021) 925–937. https://doi.org/10.1007/s11276-019-02197-y

[14] C.M. Nalayini, K. Jeevaa, Detection of DDoS Attack Using Machine Learning Algorithms. SSRN, 9(7), (2022) 4173187.

[15] J. Katiravan, A Two level Detection of Routing layer attacks in Hierarchical Wireless Sensor Networks using learning based energy prediction. KSII Transactions on Internet and Information Systems, 9(11), (2015) 4644-4661. https://doi.org/10.3837/tiis.2015.11.022

[16] N. Dharini, R. Balakrishnan and A. P. Renold, "Distributed detection of flooding and gray hole attacks in Wireless Sensor Network," 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Avadi, India, 2015. https://doi.org/10.1109/ICSTM.2015.7225410

[17] Dharini, N., Duraipandian, N. Katiravan, J. ELPC-Trust Framework for Wireless Sensor Networks. Wireless Pers Commun 113, 1709–1742 (2020). https://doi.org/10.1007/s11277-020-07288-0

[18] A. Allhusen, I. Alsmadi, A. Wahbeh, M. Al-Ramahi, A. Al-Omari, (2021) Dark Web Analytics: A Comparative Study of Feature Selection and Prediction Algorithms. SSRN, *3949786*.

[19] S. Nomm, B. Hayretdin, (2018) Unsupervised Anomaly Based Botnet Detection in IoT Networks. Proceedings of IEEE International Conference on Machine learning and applications, IEEE, USA. https://doi.org/10.1109/ICMLA.2018.00171

[20] A. Almomani, (2023). Darknet traffic analysis and classification system based on modified stacking ensemble learning algorithms. Information Systems and e-Business Management, 1-32. https://doi.org/10.1007/s10257-023-00626-2

[21] J. Yousra, R. Navid, Multi-Layer Perceptron Artificial Neural Network Based IoT Botnet Traffic Classification. Proceedings of the future technology conferences*,* 1*,* (2019) 973-984. https://doi.org/10.1007/978-3-030-32520-6_69

[22] Q. Abu Al-Haija, M. Krichen, W. Abu Elhaija, Machine-learning-based darknet traffic detection system for IoT applications. Electronics, 11(4), (2022) 556. https://doi.org/10.3390/electronics11040556

[23] B. Hayretdin, N. Sven, B. Fabio, (2018) Dimensionality Reduction for Machine Learning Based IoT Botnet Detection. Proceedings of the IEEE International Conference on Control, Automation, Robotics and Vision, IEEE, Singapore. https://doi.org/10.1109/ICARCV.2018.8581205

[24] H. Mohanty, A.H. Roudsari, A.H. Lashkari, Robust stacking ensemble model for darknet traffic classification under adversarial settings. Computers & Security, 120, (2022) 102830. https://doi.org/10.1016/j.cose.2022.102830

[25] T. Hasan, J. Malik, I. Bibi, W.U. Khan, F.N. Al-Wesabi, K. Dev, G. Huang, (2022). Securing industrial internet of things against botnet attacks using hybrid deep learning approach. IEEE Transactions on Network Science and Engineering, 10(5), (2022) 2952-2963. https://doi.org/10.1109/TNSE.2022.3168533

[26] N. Rust-Nguyen, M. Stamp, (2022) Darknet traffic classification and adversarial attacks. arXiv. https://doi.org/10.48550/arXiv.2206.06371

[27] A. Hasan, H. Theyazn, H. Aldhyani. Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications. Security and Communication Networks, 2021, (2021). https://doi.org/10.1155/2021/3806459

[28] S. Bharath, S. Dineshkumar, P. Pankesh, G.B. John, I.A. Muhammad, (2021) Edge2Guard: Botnet Attacks Detecting Offline Models for Resource-Constrained IoT Devices. Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, IEEE, Germany. https://doi.org/10.1109/PerComWorkshops51409.2021.9431086

[29] A. Mahi, M. Hassan, M.B.I. Mohamed, IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection. *Sensors.* 20(21), (2020) 1-21.https://doi.org/10.3390/s20216336

[30] A.O. Prokofiev, Y.S. Smirnova, V.A. Surov, (2018). A method to detect Internet of Things botnets. In 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), IEEE, Russia. https://doi.org/10.1109/EIConRus.2018.8317041

[31] M.Y. Alzahrani, A.M. Bamhdi, Hybrid deep-learning model to detect botnet attacks over internet of things environments. Soft Computing, 26, (2022) 164–175. https://doi.org/10.1007/s00500-022-06750-4

[32] K. Jiyeon, W. Hyerin, S. Minsun, H. Seungah, C. Eunjung Feature Analysis of IoT Botnet Attacks based on RNN and LSTM, International Journal of Engineering Trends and Technology, 68(4), (2020) 43-47. https://doi.org/10.14445/22315381/IJETT-V68I4P208S

[33] D. Cullen, J. Halladay, N. Briner, R. Basnet, J. Bergen, T. Doleck, Evaluation of synthetic data generation techniques in the domain of anonymous traffic classification. IEEE Access, 10, (2022) 129612-129625. https://doi.org/10.1109/ACCESS.2022.3228507

[34] I. Ullah, Q.H. Mahmoud, (2020) A scheme for generating a dataset for anomalous activity detection in iot networks. In Canadian conference on artificial intelligence, Springer International Publishing.

[35] K. Alissa, T. Alyas, K. Zafar, Q. Abbas, N. Tabassum, S. Sakib, Botnet attack detection in iot using machine learning. Computational

Intelligence and Neuroscience, 2022, (2022). https://doi.org/10.1155/2022/4515642

[36] M. Almseidin, M. Alkasassbeh, An Accurate Detection Approach for IoT Botnet Attacks Using Interpolation Reasoning Method. Information, 13(6), (2022) 300. https://doi.org/10.3390/info13060300

[37] N. Dharini, S.P. Shakthi, S.S. Shruthi, (2023) Botnet Attack Detection in IoT-Based Security Camera Device Using Principal Component Analysis with Various Machine Learning Algorithms. Proceedings of the 2nd International Conference on Cognitive and Intelligent Computing. ICCIC 2022. Cognitive Science and Technology. Springer, Singapore. https://doi.org/10.1007/978-981-99-2746-3_65

Authors Contribution Statement

Dharini N: Conceptualization, Methodology, Investigation, Data Curation, Formal Analysis, Writing—Original Draft. Jeevaa Katiravan: Data Curation, Visualization, Formal Analysis, Writing—Review & Editing. Shakthi S P: Software, Validation, Writing—Review & Editing. All the authors read and approved the final version of the manuscript.

Funding

This research was conducted without the aid of any financial grants.

Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes