JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 07 Issue 01
2024

Research Article
https://doi.org/10.56979/701/2024

Machine Learning-Based Detection of Mirai and Bashlite Botnets in IoT Networks

Fatima Yousaf[1*], Muhammad Arslan[2], Ammar Ahmad Khan[3], Ashir Tanzil[4], Asiya Batool[3], and Muhammad Asad[2]

[1]Department of Computer Science & IT, Institute of Southern Punjab, Multan 60800, Pakistan.
[2]Faculty of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan.
[3]Department of Computer Science, NAMAL University, Mianwali, 42250, Punjab, Pakistan
[4]Department of Computer Science, Abasyn University Islamabad Campus, Islamabad, 44000, Pakistan.
[*]Corresponding Author: Fatima Yousaf. Email: fatimayousaf@isp.edu.pk

Received: February 11, 2024 **Accepted:** May 23, 2024 **Published:** June 01, 2024

Abstract: The growth of IoT devices has caused more botnet attacks, similar the Mirai botnet, which is a major cause of distributed denial of service (DDoS) attacks. Mirai gained notoriety for its involvement in large-scale attacks that compromised numerous IoT devices through weak authentication credentials. Similarly, Bashlite, also known as Bash0day or Lizkebab, targets vulnerable IoT devices by exploiting the Shellshock vulnerability in Linux-based systems. These botnets leverage compromised devices to carry out malicious activities and the propagation of malware. Machine Learning (ML) methods have been proposed to detect botnets, but finding both Mirai and Bashlite botnets at the same time is difficult because their attack patterns are different. The Random Forest (RF), Support Vector Machine (SVM) and Logistic Regression (LR) based detector for Mirai and Bashlite botnets are implemented in our detection method using machine learning. This study used N-BaIoT dataset to train these algorithms in order to detect the best features that distinguish botnet attacks on Internet of Things (IoT) devices. In this research we used two infected devices against five protocols. All machine learning algorithms used are reasonably accurate, as their test validation accuracy was greater than 99%, although Random Forest seemed to work the best.

Keywords: IoT Botnet Detection; Machine Learning; Bashlite Botnet; Mirai Botnet; N-BaIoT Dataset.

1. Introduction

Botnets are usually a network of devices that have been taken over, and that exhibit on external management or control by an outside server [1]. Robots are the software components, commonly well-known as "bots" or "zombies," are usually organized after being compromised by an attacker due to a vulnerability in the software or security settings [2]. An attacker can compromise these devices and then use them to perform a wide range of malicious activities including, but not limited to launching DDoS attacks, spamming users, stealing private data, and so on [3].

IoT devices have also been increasingly manipulated for assembling massive botnets to transfer Distributed Denial of Service (DDoS) attacks. This is a kind of malicious attack, due to the vulnerabilities found on various IoT devices. Botnet attacks, mostly those using the Mirai botnet, pose significant threats to IoT network stability and security through methods like TCP, UDP, and ACK Flooding, DNS attacks, and HTTP Flooding [4]. These attacks exploit IoT vulnerabilities, leading to common interruptions and compromised systems. Botnet attacks detection at specific levels is a hard problem because not much research has been conducted to find bandits in color flow characteristics. Many research papers measured the performance of three classifiers, to classify botnet attacks on nine devices using comprehensive dataset of N-BaIOT having 115 features discussed in literature. However, the low power consumption of IoT

devices makes it challenging to manage and monitor massive amounts of data for the classification of network traffic. These restrictions are due to the storage capabilities and also higher computation costs [5].

Machine Learning and deep learning are being used in every research field of this modern era such as IoT, biomedical images, content based image retrievals, education, development, industry etc [6]–[8]. While existing literature has primarily focused on the detection of botnet attacks, there is no noticeable research proposing effective methods to detect multiple botnets. Consequently, a significant need arises for novel approaches and methodologies.

In this paper, we present a strategy to effectively identify IoT devices targeted by multiple botnet attacks. Our focus is on detecting and stopping malicious traffic from Mirai and Bashlite attacks. The key contributions of our work include:

1. We trained machine learning algorithms using the publicly available N-BaIoT dataset to detect and classify botnet attacks on IoT devices.
2. By employing Random Forest, Support Vector Machine, and Logistic Regression techniques, we achieved a test validation accuracy exceeding 99%. Random Forest demonstrated superior performance compared to the other algorithms.

The remaining portions of the paper are arranged as follows: A comprehensive review of the literature on the methods for detecting and defeating IoT-based botnets is included in Section 2. Section 3 presents our proposed methodology for traffic analysis and the training of machine learning algorithms. The effectiveness of the machine learning models that have been developed in recognizing malicious traffic is illustrated in Section 4. In Section 5, we bring our work to a close by highlighting the importance of our study in creating practical defenses against botnet assaults for IoT devices.

2. Literature Review

In this section, the discussion is made about literature on methods of detecting and mitigating botnets based on IoT. Host-based detection and network-based detection are the two main types of botnet detection approaches. Host-based detection techniques are employed on individual devices, like smartphones and personal computers, whereas network-based detection techniques are implemented on a central machine, like a router or gateway, which handles all network traffic. Host-based detection techniques are frequently employed when a device possesses adequate storage capacity to support many detection algorithms. However, due to the limited resources of IoT devices, storing such resource-intensive detection algorithms becomes challenging. This has led to early detection strategies that mainly focused on network-based detection algorithms as a major.

There are specific signature-based approaches for detecting IoT botnets such as in research paper [9]. Although these approaches are primarily network-based, they incorporate Mirai signatures into the detection algorithm to enhance its effectiveness. Additionally, approaches like [10] focus on detecting malicious domains within an IoT network to prevent malicious activities originating from specific domains. A unique graph-based method is carried out in article [11] to identify botnets that are compatible with a range of device designs. The process entails identifying the botnet lifecycle by analyzing binary files that contain both harmful and benign traffic. The proposed approach involves creating function call graphs and printable string information (PSI) graphs from botnet lifecycle related functions and using a convolutional neural network to classify benign and malicious samples with an accuracy of over 95%.

This Paper [12] suggests a method for utilizing power usage trends to identify the Mirai botnet and its variations on Internet of Things devices (PCP). The methodology involves collecting PCP of each device during different stages and training a CNN model on the preprocessed dataset. The model achieved 90% accuracy, but the proposed methodology faces several implementation challenges, such as expensive power consumption tools and lack of standard datasets of (PCP). The authors of research paper [9] suggest a network-based approach to find bots in IoT networks while they're scanning. Having looked at all the Mirai malware signatures, they chose to search for bots that use the port scanning signature. While it would do little to mitigate botnet attacks in general, the authors suggested that known misbehaving bots could be blocked or made unable to communicate once they were identified. A method for discovering and quarantining vulnerable devices in an Internet of Things (IoT) network is proposed by the authors of [13], preventing malware from infecting them and enrolling them into a Botnet. The technique starts by looking for open ports on Internet of Things devices, specifically HTTP, SSH, and Telnet. Then, firewall rules are

set to blacklist internet access for vulnerable devices. In paper [14], a policy descriptor mechanism is proposed to detect any abnormal activities in Internet of Things (IoT) devices by comparing their current rules with the saved policies at first-time entry, for communication, usage and access. The proposed method provides an effective way to detect and prevent the spread of Mirai virus in the Internet of Things network. Like that, [15] recommended solving botnet recognition with Deep learning from network flow data. The approach involves capturing network traffic flows, converting them into connection records and training classifiers to recognize the difference between malicious or benign traffic. Hence the suggested method used to identify botnet activity will be proven to be highly efficient.

A Deep Learning Botnet Detection method using LAE and BLSTM Techniques: A Recent study of [16] proposed deep learning heterogeneous botnets detection method for LAE and BLSTM techniques. LAE is employed to reduce the data dimensionality and BLSTM for detecting long-term temporal dependency present between the low-dimensional features set spitted out by LAE in order to categorize a given attack. On the BOT-IoT data set, the researchers brought the amount of data down by 91.89 percent with LAE [17]. Provides a BLSTM-dependent RNN approach to IoT botnet detection at the packet level. Using a sandbox instance, the authors performed a Mirai attack against IoT devices and collected their dataset. This dataset was then classified using the BLSTM-RNN network. In research paper [18], authors suggested a machine learning technique that evaluated data accuracy through the use of an artificial neural network. Although they only used data from one device in their investigation, they used the N-BaIoT dataset from nine distinct devices. The accuracy of the model was found to be 92%. The authors of [19] introduced a brand-new feature selection technique called corrauc, which computes the area under the ROC curve and assesses correlation qualities. The four steps of the methodology are as follows: first, features with adequate information are chosen; next, feature selection algorithms are applied; finally, selected features are validated; and finally, the methodology is assessed using four distinct machine learning algorithms on the BOT-IoT dataset, yielding an accuracy of more than 96%.

A binary and multiclass classification approach was introduced by the authors of paper [20] to distinguish between normal and malicious Internet of Things communications. Three machine learning algorithms and two feature selection techniques were used for classification. Results show that all classifiers work well and with a high level of accuracy in binary a multi-class classification. Finally, the authors recommended a density-based classification of green IoT devices outlier identification in network data by applying DBSCAN: it is considered as the most widely used clustering algorithm-based method. Zero-Day proof low-density clusters were marked as Malicious and High-Density Clusters were categorized as normal Trash. Three machine learning algorithms were also used to better identify the named clusters; over 90% accuracy was achieved for every attack. Keeping this in view, such a technique can be beneficial probably to identify and prevent malicious communication on green IoT devices.

The paper [21] offered a combination of BO-GP and DT machine learning methods for detecting malicious traffic in the BOT-IoT dataset. The authors addressed the issue of unbalanced datasets by employing the SMOTE algorithm and the min-max normalization method. They did not, however, apply any feature selection strategy or ideal model hyper-parameters to enhance detection performance with the Bayesian Optimization Gaussian Process. The model achieved 99 percent accuracy, but because it lacks a feature selection strategy, it might not be suitable for Internet of Things applications. The complete dataset was not used by the authors to assess the model. An unsupervised machine learning algorithm-based domain name detection technique is proposed by the study in article [22] to distinguish between malicious and normal domains. After gathering both normal and malicious domains, the writers extracted 204 variables. However, during preprocessing, they only kept 20 features. They tested nine different algorithms and found out that LAC, ANN, K-Means, AP, DBSCAN, Hierarchical Clustering, K-medians GMM and Mini Batch K-means were the four that reached 99% accuracy level. According to the study's findings, unsupervised machine learning methods can successfully distinguish between normal and malicious domains.

Botnet traffic classification using supervised machine learning algorithms is a major research area. The BOT-IoT dataset was used in conjunction with chi2 as the feature selection method, respectively [23]. The three supervised machine learning algorithms that were employed included Multilayer Perceptron Artificial Neural Network (MLP ANN), K nearest Neighbor (KNN) and Gaussian Naïve Bayes (GNB). It has been concluded from research that KNN is the most accurate of these three algorithms.

BDS anti-botnet defense system is a collection of four components: worm launcher, monitoring module, policy planner and command control used to research botnets and their strategy [24]. First, authors develop a method named Few Elite Launch for the PetriNet 2 simulator in order to repell white-capped insects according to both its life cycle and network structural density. The paper [25] describes a Mirai-based scanner in order to obtain vulnerability data for IoT devices, identify the vulnerabilities and produce reports on which it can assist network administrators or home users with. This is tool mimics Mirai infection like attacks on IoMT devices that gives a solution to better the security of IoMT networks. In this paper, the authors developed an algorithm of Hybrid Strawberry African Buffalo Optimizer (HSABO) [26] based on collective behavior of strawberry plant and African buffalo for identifying IoT botnet attack to enhance security in these types of devices. In [27], by calculating hash code of each device in the IoT network, it presents a way for mitigating of Mirai botnet activity and has advised implementing this method through analysis module and an application monitor whitelist. By verifying the security properties of their code, an Invincea virtual machine running on each device prevents Mirai botnet attacks by limiting what can run and denying rogue applications access to sensitive data.

On the other hand, limited research has taken place in preventing IoT devices from being part of botnet. Also, previous research has shown limited solutions for removing malicious activity from compromised IoT devices, with one method being botnet-vs-botnet, which requires significant time, battery power, and storage. Installing IDS on every device is an alternate method, but it uses up storage and isn't the best for IoT devices because of their short battery lives. Another approach is to prevent devices from becoming bots by addressing three primary vulnerabilities: all-time online availability, open ports, and weak credentials. However, these measures are not entirely effective due to user awareness issues and Mirai's ability to compromise devices through brute force attacks.

3. Methodology

This section presents a methodology for the detection and prevention of botnets in IoT networks using a machine learning approach. The general machine learning process is shown under Figure 1. This method analyzes network traffic using machine learning techniques to identify trends that demonstrate botnet activity. The detected botnets are then removed from the network. The machine learning algorithms are trained on publicly available N-BaIoT dataset. The dataset description and preprocessing are described in the following section.

3.1. Dataset description

The N-BaIoT dataset, which was first presented in paper [27] and is freely accessible at the UCI Machine Learning Repository, was used in this investigation. Nine commercial Internet of Things (IoT) devices—four security cameras, one webcam, two doorbells, and one thermostat—were used to collect real-time traffic data to create the dataset. We used two devices in this study. Each device was deliberately infected with two types of malwares, namely Mirai and BASHLITE, and subjected to various types of attacks, as mentioned under Table 1.

After each device is installed, normal traffic is intercepted to ensure that malicious samples are removed from the training dataset. The dataset contains a total of 115 objects, of which 23 objects were recorded five times, each with a duration of 100 ms, 500 ms, 1.5 s, 10 s and 1 min.

Table 1. Attack Types in N-BaIoT Dataset

IoT Devices	Malware	Attacks Considered
Damini_Doorbell, Provision_PT_737E_Security_Camera,	Bashlite	Scan, TCP, UDP, Junk, Combo
	Mirai	Scan, Ack, UDP, Syn, UDP plain

Furthermore, several statistics are computed from each packet which include covariance, size and radius of two flow sources, Pearson correlation coefficient and mean standard deviation and variance of packet sizes.

3.2. Data preprocessing

Data preprocessing is a fundamental step in the field of machine learning as it prepares the data before feeding it to the machine learning algorithm to ensure optimal performance. Data preparation involves several steps such as converting data into a numeric representation, eliminating null and duplicate samples, balancing the data set, and normalizing values. For instance, in the case of N-BaIoT dataset, class balancing, and dataset normalization are required due to high ratio of attack samples as compared to corresponding benign data samples. Consequently, it is necessary to choose and implement the right

preprocessing techniques for proper preparation of datasets for machine learning algorithms that will be used to obtain accurate reliable results.

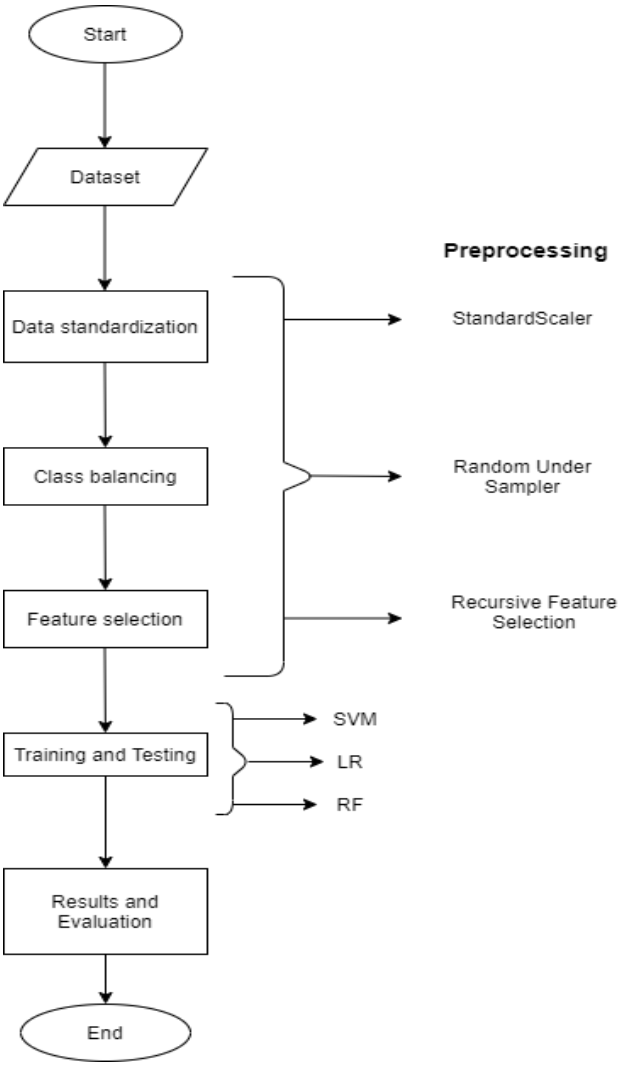

Figure 1. Botnet Detection Research Methodology

3.3. Data Standardization

We use standard scalar methods for normalizing values in a dataset so as to enhance the model's performance. With this method, the mean value of each feature is subtracted followed by scaling all values to unit variance by dividing by their corresponding standard deviation [24]. The transformed data has a mean of 0 and a standard deviation of 1, defined as;

$$z = \frac{x-u}{\sigma} \quad (1)$$

Where σ represents the standard deviation, μ is the mean of the samples in the data set, and x is the original value of the trait. This normalization process is performed using standard scalar library modules.

3.4. Class Balancing

There is a clear class imbalance in the binary classification dataset, with significantly more samples belonging to the attack category than to the benign threat category. This type of imbalance can be caused by bias and overfitting in favor of certain categories. In addition, feature selection techniques can also support features that are very important for overrepresented classes. It is known that all these problems are caused by data imbalance. We ensure that each device contains an equal number of samples from both aggressive and benign classes to balance the data set and minimize these issues. This is achieved by using the RandomUnderSampler module in the imblearn library, which randomly compresses the majority class to match the size of the minority class.

3.5. Feature Selection

Dimensionality reduction is a useful machine learning preprocessing step that helps remove redundant and irrelevant data, improving the accuracy of the learning process and the understanding of the results

[25]. In our study, we performed feature selection by selecting the top 10 features from the combined attack and benign datasets. There are three main feature selection methods, including the packing method, the filtering method, and the embedding method. These methods are used to determine the most important features by assessing the impact of different features on the performance of the model. The proposed method involves selecting the main features based on their statistical significance (determined by their correlation with the target variable).

3.6. Recursive feature elimination

We use recursive feature elimination (wrapper method) and logistic regression to select the best subset of features for binary classification. This method, as illustrated in Figure 2, takes all dataset features as input and removes features that have the least impact on classification accuracy. Starting with all 115 features in dataset, the features are evaluated and removed 20 features with each iteration until reaching the optimal subset of size *k*, aiming to identify the most informative features for model accuracy.

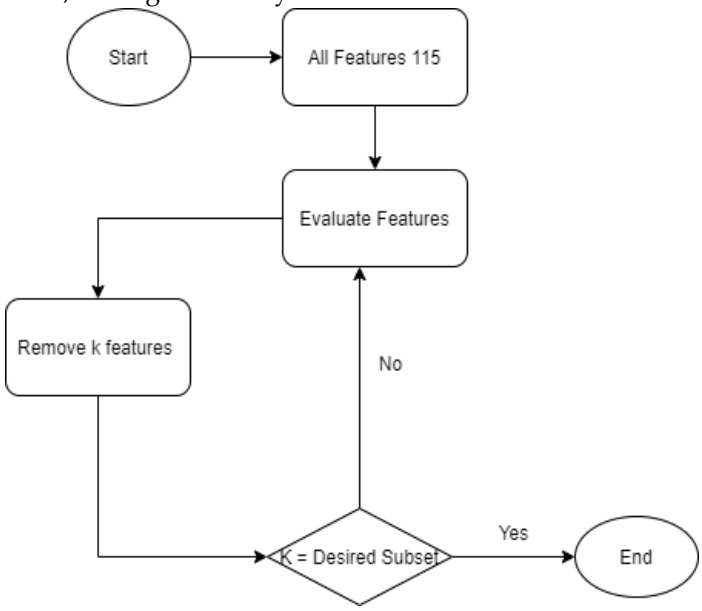

Figure 2. Recursive Feature Examination

3.7. Classification

We use a binary classification approach using three classifiers (logistic regression, support vector machine, and random forest) to identify malicious traffic. These classifiers are trained on selected features to differentiate between benign and malicious traffic. By using LR, we leverage its ability to model nonlinear relationships and make accurate predictions based on selected features. Additionally, we use SVM to accurately classify benign and malicious traffic by identifying optimal solution boundaries in high-dimensional data. Finally, when in-depth data analysis is required, choose Random Forest (RF) to perform regression and classification tasks. It consists of decision trees, and the predictions of each decision tree are combined to produce the final result. The accuracy and efficiency of RF is affected by the number of decision trees.

3.8. Machine Learning Models for Identifying Malicious Traffic

The methodology of this study aims to classify malicious traffic in IoT devices using machine learning techniques, specifically random forest, support vector machine (SVM), and logistic regression. The N-BaIOT dataset was obtained from nine affected devices that used five different protocols. The models under review are trained and tested with this dataset to find out the information signatures for effective detection of botnet attacks on IoT devices.

3.8.1 Random Forest

A cluster of decision trees is generated during training, where they form a class by averaging classes predicted by every individual tree using random forest ensemble learning method. It is capable of handling large multidimensional datasets and inferring intricate associations between attributes.

3.8.2. Support Vector Machine (SVM)

Support vector machine (SVM) is a type of supervised learning model which splits up the class into segments through constructing a hyperplane based on data analysis. For classification purposes, it maps

new instances to high-dimensional feature spaces using decision thresholds. SVM works well with both linear and nonlinear separable data.

3.8.3. Logistic Regression

The statistical model called logistic regression predicts the probability of binary outcome given the input variables. Logistic functions are used in analyzing the relationship between dependent variable(s) and one or more independent variables to estimate probabilities. On classification problems involving binary outcomes, logistic regression is commonly applied.

During a given time frame the data set is examined to identify qualities (e.g. weights or biases) that can signify particular attributes of malicious traffic. These qualities are used as training data for chosen machine learning models with labeled data where all instances of malicious traffic have labels. The model's performance is assessed using measures like testing accuracy rate. The aim here is to examine how well random forest, SVM and logistic regression can classify attack from benign IoT traffic.

4. Results and Evaluations

This study uses five different evaluation metrics to assess how well the trained model performs. To evaluate the model, four parameters (TP, TN, FP and FN) are obtained from the confusion matrix, which is the first measure. For example, if the predicted result is malicious and the actual value is also malicious, this is called true positive (TP) [28]. If the actual value is favorable but the predicted result is harmful, it is called a false negative (FN). Similarly, if the predicted result is favorable and the actual value is also favorable, it is called true negative (TN). However, if the actual value is malicious, it is called a False Positive (FP). All other metrics are calculated using these four parameters.

The classifier's accuracy is assessed using accuracy. However, precision, recall, and f1-score are also used because accuracy alone is insufficient for evaluating performance [29] .

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Precision is useful when the FP rate increases because it double checks the accuracy of the TP by determining how often the predictions are correct.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Recall is used to determine how often our model correctly predicts a class, which is used to estimate the true positive rate.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Recall is used to determine how often our model correctly predicts a class, which is used to estimate the true positive rate.

$$F1 - score = \frac{2(precision)(recall)}{precision+recall} \quad (5)$$

This research primarily focuses on outcome obtained by using three classifiers: Support Vector Machine (SVM) [30], Logistic Regression (LR) [31] and Random Forests (RF) [32] in identification of benign and botnet traffic. Out of 115 features, we selected top ten most informative ones per attack type. All the selected attack-related features are listed by their sequence numbers from 1 to 115.The end part examines it critically regarding the test results obtained employing each type classifier against different attacks made through some device. Confusion metrics, Accuracy, Precision, F1-score, and Recall score are among the evaluation metrics that were employed to evaluate the classifiers.

The IoT device dataset's benign file is combined and balanced with each attack dataset file independently, and features are then retrieved utilizing wrapper techniques. After that, the chosen features are put through to the categorization techniques.

Tables 2 and 3 show the outcomes of the classification techniques for each of two IoT devices and attacks. For every attack, the evaluation metrics are computed individually. For instance, three methods were used to classify the results of a combination attack, and Figures 3 and 4 display the evaluation metrics for each algorithm.

4.1. Danmini_Doorbell Dataset Results

Table 2. Damini_Doorbell Dataset Results

Botnet	Attacks	Algorithm	Accuracy	Precision	Recall	F1 Score	TP	FP	TN	FN
Bashlite	g_combo	RF	1.0	1.0	1.0	1.0	9910	0	9910	0

		LR	1.0	1.0	1.0	1.0	9910	0	9910	0
		SVM	1.0	1.0	1.0	1.0	9910	0	9910	0
		RF	0.9999	1.0	0.9998	0.9999	5814	0	5813	1
	g_junk	LR	0.9996	1.0	0.9993	0.9996	5814	0	5810	4
		SVM	0.9996	1.0	0.9993	0.9996	5814	0	5810	4
		RF	1.0	1.0	1.0	1.0	5970	0	5970	0
	g_scan	LR	1.0	1.0	1.0	1.0	5970	0	5970	0
		SVM	1.0	1.0	1.0	1.0	5970	0	5970	0
		RF	1.0	1.0	1.0	1.0	9910	0	9910	0
	g_tcp	LR	0.9998	0.9997	0.9998	0.9998	9908	2	9909	1
		SVM	0.9998	0.9997	0.9998	0.9998	9908	2	9909	1
		RF	1.0	1.0	1.0	1.0	9910	0	9910	0
	g_udp	LR	0.9996	1.0	0.9993	0.9996	9910	0	9904	6
		SVM	0.9998	1.0	0.9996	0.9998	9910	0	9907	3
		RF	1.0	1.0	1.0	1.0	9910	0	9910	0
	m_ack	LR	1.0	1.0	1.0	1.0	9910	0	9910	0
		SVM	1.0	1.0	1.0	1.0	9910	0	9910	0
		RF	1.0	1.0	1.0	1.0	9910	0	9910	0
	m_scan	LR	0.9999	1.0	0.9998	0.9999	9910	0	9909	1
		SVM	0.9999	1.0	0.9998	0.9999	9910	0	9909	1
		RF	1.0	1.0	1.0	1.0	9910	0	9910	0
Mirai	m_syn	LR	0.9999	0.9998	1.0	0.9999	9909	1	9910	0
		SVM	1.0	1.0	1.0	1.0	9910	0	9910	0
		RF	1.0	1.0	1.0	1.0	9910	0	9910	0
	m_udp	LR	1.0	1.0	1.0	1.0	9910	0	9910	0
		SVM	1.0	1.0	1.0	1.0	9910	0	9910	0
		RF	1.0	1.0	1.0	1.0	9910	0	9910	0
	m_udp plain	LR	1.0	1.0	1.0	1.0	9910	0	9910	0
		SVM	1.0	1.0	1.0	1.0	9910	0	9910	0

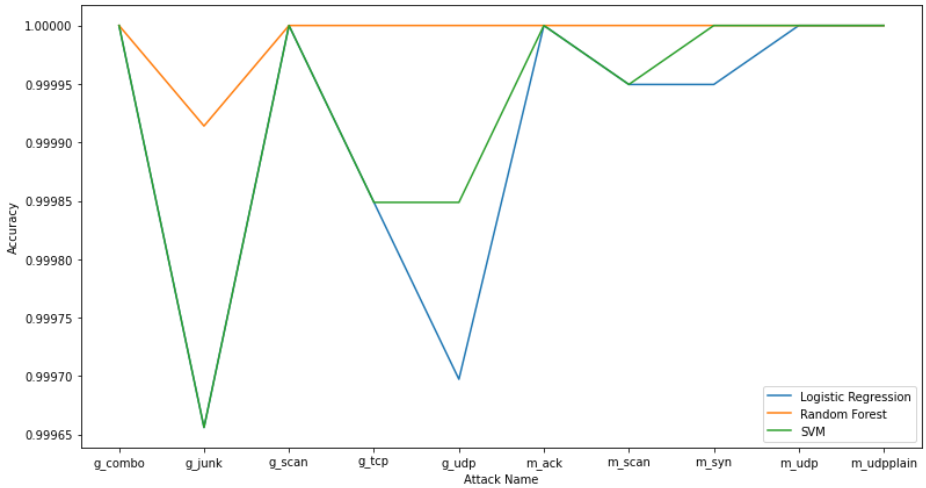

Figure 3. Damini_Doorbell Dataset Accuracy

The comparison of all the chosen algorithms' accuracy against all kinds of attacks is displayed in Figure 3. Overall accuracy ranges from 99 to 100%, with RF accuracy ranking highest and LR accuracy ranking lowest.

4.2. Provision_PT_737E_Security_Camera Dataset Results

Table 3. Provision_PT_737E_Security_Camera Dataset Results

Botnet	Attack	Algorithm	Accuracy	Precision	Recall	F1 Score	TP	FP	TN	FN
Bashlite	g_combo	LR	0.9999	1.0	0.9998	0.9999	12276	0	12274	2
		RF	0.9999	1.0	0.9998	0.9999	12276	0	12274	2
		SVM	0.9999	1.0	0.9998	0.9999	12276	0	12274	2
	g_junk	LR	0.9996	1.0	0.9993	0.9996	6180	0	6176	4
		RF	0.9999	1.0	0.9998	0.9999	6180	0	6179	1
		SVM	0.9995	0.9998	0.9993	0.9995	0.9999	1	6176	4
	g_scan	LR	0.9999	1.0	0.9998	0.9999	5859	0	5859	1
		RF	1.0	1.0	1.0	1.0	5859	0	5860	0
		SVM	0.9999	1.0	0.9998	0.9999	5859	0	5859	1
	g_tcp	LR	0.9993	0.9988	0.9999	0.9993	12417	14	12430	1
		RF	1.0	1.0	1.0	1.0	12431	0	12431	0
		SVM	0.9998	0.9998	0.9999	0.9998	12429	2	12430	1
	g_udp	LR	0.9992	0.9984	1.0	0.9992	12412	19	12431	0
		RF	0.9999	1.0	0.9999	0.9999	12430	1	12431	0
		SVM	0.9995	1.0	0.9991	0.9995	12420	11	12431	0
Mirai	m_ack	LR	0.9999	1.0	0.9998	0.9999	12109	0	12109	2
		RF	1.0	1.0	1.0	1.0	12112	0	12111	0
		SVM	0.9999	1.0	0.9998	0.9999	12111	0	12109	2

m_scan	LR	0.9998	1.0	0.9997	0.9998	12431	0	12428	3
	RF	0.9998	1.0	0.9997	0.9998	12431	0	12428	3
	SVM	0.9998	1.0	0.9997	0.9998	12431	0	12428	3
m_syn	LR	0.9998	1.0	0.9996	0.9998	12431	0	12427	4
	RF	0.9999	1.0	0.9999	0.9999	12431	0	12430	1
	SVM	0.9998	1.0	0.9996	0.9998	12431	0	12427	4
m_udp	LR	0.9998	1.0	0.9997	0.9998	12431	0	12428	3
	RF	0.9998	1.0	0.9997	0.9998	12431	0	12428	3
	SVM	0.9998	1.0	0.9997	0.9998	12431	0	12428	3
m_udp plain	LR	0.9999	1.0	0.9999	0.9999	11336	0	11336	1
	RF	1.0	1.0	1.0	1.0	11336	0	11337	0
	SVM	0.9999	1.0	0.9999	0.9999	11336	0	11336	1

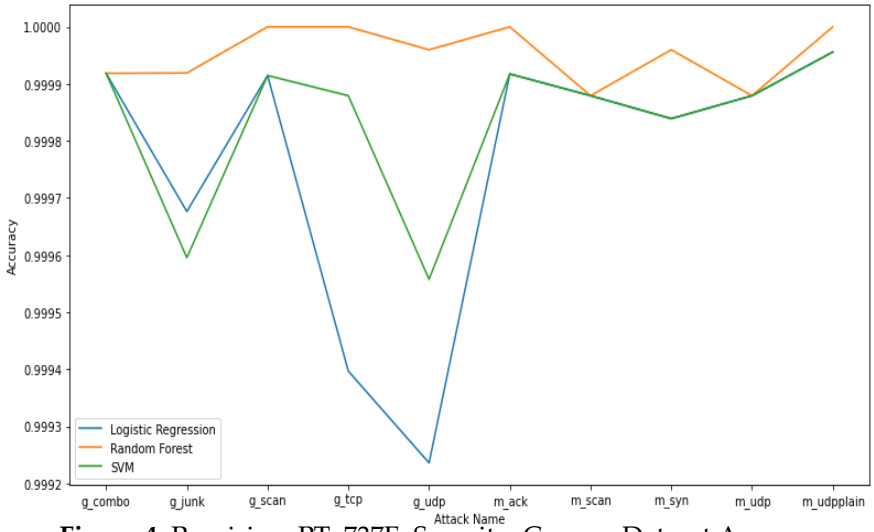

Figure 4. Provision_PT_737E_Security_Camera Dataset Accuracy

5. Conclusion and Future Directions

In this research, we aimed to achieve two objectives, namely dimensionality reduction of the N-BaIoT dataset for quick yet efficient malicious traffic identification and evaluating performance of various ML algorithms for malicious traffic using selected features from multiple IoT device-to-device communications.

The N-BaIoT dataset included normal traffic data as well as network traffic from two different IoT devices that were infected with the ten attack types of Bashlite and Mirai. We used Random Forest, Support Vector Machine, and Logistic Regression techniques to perform binary classification after reducing the dataset's feature set from 115 to 10. Our results showed that the selected subset of features achieved above 99% accuracy in detecting botnet traffic.

References

1. R. S. S. Moorthy and N. Nathiya, "Botnet Detection Using Artificial Intelligence," *Procedia Comput. Sci.*, vol. 218, pp. 1405–1413, 2023, doi: 10.1016/j.procs.2023.01.119.
2. J. Velasco-Mata, V. González-Castro, E. Fidalgo, and E. Alegre, "Real-time botnet detection on large network bandwidths using machine learning," *Sci. Rep.*, vol. 13, no. 1, p. 4282, Mar. 2023, doi: 10.1038/s41598-023-31260-0.
3. F. S. Gharehchopogh, B. Abdollahzadeh, S. Barshandeh, and B. Arasteh, "A multi-objective mutation-based dynamic Harris Hawks optimization for botnet detection in IoT," *Internet of Things*, vol. 24, p. 100952, Dec. 2023, doi: 10.1016/j.iot.2023.100952.
4. C.-H. Yang, J.-P. Wu, F.-Y. Lee, T.-Y. Lin, and M.-H. Tsai, "Detection and Mitigation of SYN Flooding Attacks through SYN/ACK Packets and Black/White Lists," *Sensors*, vol. 23, no. 8, p. 3817, Apr. 2023, doi: 10.3390/s23083817.
5. A. Woodiss-Field, M. N. Johnstone, and P. Haskell-Dowland, "Examination of Traditional Botnet Detection on IoT-Based Bots," *Sensors*, vol. 24, no. 3, p. 1027, Feb. 2024, doi: 10.3390/s24031027.
6. A. Imran, K. R. Malik, A. H. Khan, M. Sajid, and M. Arslan, "Methodology for Ensuring Secure Disease Prediction using Machine Learning Techniques | Journal of Computing & Biomedical Informatics," vol. 07, no. 01, 2024, [Online]. Available: https://jcbi.org/index.php/Main/article/view/435
7. M. Abbas, M. Arslan, R. A. Bhatty, F. Yousaf, A. A. Khan, and A. Rafay, "Enhanced Skin Disease Diagnosis through Convolutional Neural Networks and Data Augmentation Techniques," vol. 07, no. 01, 2024.
8. A. Ijaz *et al.*, "Innovative Machine Learning Techniques for Malware Detection | Journal of Computing & Biomedical Informatics," 2024, [Online]. Available: https://jcbi.org/index.php/Main/article/view/508
9. A. Kumar and T. J. Lim, "Early Detection of Mirai-Like IoT Bots in Large-Scale Networks through Sub-sampled Packet Traffic Analysis," 2020, pp. 847–867. doi: 10.1007/978-3-030-12385-7_58.
10. C. OKUR and M. DENER, "Detecting IoT Botnet Attacks Using Machine Learning Methods," in *2020 International Conference on Information Security and Cryptology (ISCTURKEY)*, IEEE, Dec. 2020, pp. 31–37. doi: 10.1109/ISCTURKEY51113.2020.9307994.
11. H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, "A novel graph-based approach for IoT botnet detection," *Int. J. Inf. Secur.*, vol. 19, no. 5, pp. 567–577, Oct. 2020, doi: 10.1007/s10207-019-00475-6.
12. W. Jung, H. Zhao, M. Sun, and G. Zhou, "IoT botnet detection via power consumption modeling," *Smart Heal.*, vol. 15, p. 100103, Mar. 2020, doi: 10.1016/j.smhl.2019.100103.
13. C. Dietz *et al.*, "IoT-Botnet Detection and Isolation by Access Routers," in *2018 9th International Conference on the Network of the Future (NOF)*, IEEE, Nov. 2018, pp. 88–95. doi: 10.1109/NOF.2018.8598138.
14. S. M. Sajjad and M. Yousaf, "UCAM: Usage, Communication and Access Monitoring Based Detection System for IoT Botnets," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, IEEE, Aug. 2018, pp. 1547–1550. doi: 10.1109/TrustCom/BigDataSE.2018.00221.
15. S. Sriram, R. Vinayakumar, M. Alazab, and S. KP, "Network Flow based IoT Botnet Attack Detection using Deep Learning," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, Jul. 2020, pp. 189–194. doi: 10.1109/INFOCOMWKSHPS50562.2020.9162668.
16. S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4944–4956, Mar. 2021, doi: 10.1109/JIOT.2020.3034156.
17. C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet Detection in the Internet of Things using Deep Learning Approaches," in *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Jul. 2018, pp. 1–8. doi: 10.1109/IJCNN.2018.8489489.
18. T. G. Palla and S. Tayeb, "Intelligent Mirai Malware Detection in IoT Devices," in *2021 IEEE World AI IoT Congress (AIIoT)*, IEEE, May 2021, pp. 0420–0426. doi: 10.1109/AIIoT52608.2021.9454215.
19. M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242–3254, 2021, doi: 10.1109/JIOT.2020.3002255.
20. Z. Alothman, M. Alkasassbeh, and S. Al-Haj Baddar, "An efficient approach to detect IoT botnet attacks using machine learning," *J. High Speed Networks*, vol. 26, no. 3, pp. 241–254, Nov. 2020, doi: 10.3233/JHS-200641.
21. M. N. Injadat, A. Moubayed, and A. Shami, "Detecting Botnet Attacks in IoT Environments: An Optimized Machine Learning Approach," *Proc. Int. Conf. Microelectron. ICM*, vol. 2020-Decem, 2020, doi: 10.1109/ICM50269.2020.9331794.
22. A. W. Regis, R. Anne, G. Kirubavathi, and U. K. Sridevi, "Detection of IoT Botnet using Machine learning and Deep Learning Techniques," 2023, [Online]. Available: https://doi.org/10.21203/rs.3.rs-2630988/v1

23. H. J. Hadi, S. M. Sajjad, and K. un Nisa, "BoDMitM: Botnet Detection and Mitigation System for Home Router Base on MUD," in *2019 International Conference on Frontiers of Information Technology (FIT)*, IEEE, Dec. 2019, pp. 139–1394. doi: 10.1109/FIT47737.2019.00035.
24. S. Yamaguchi, "Botnet Defense System: Concept, Design, and Basic Strategy," *Information*, vol. 11, no. 11, p. 516, Nov. 2020, doi: 10.3390/info11110516.
25. S. Vysakh and P. K. Binu, "IoT based Mirai Vulnerability Scanner Prototype," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, Aug. 2020, pp. 97–101. doi: 10.1109/ICSSIT48917.2020.9214099.
26. M. Karthik and M. Krishnan, "Securing an Internet of Things from Distributed Denial of Service and Mirai Botnet Attacks Using a Novel Hybrid Detection and Mitigation Mechanism," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 1, pp. 113–123, Feb. 2021, doi: 10.22266/ijies2021.0228.12.
27. Y. Meidan *et al.*, "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul. 2018, doi: 10.1109/MPRV.2018.03367731.
28. M. Mohy-eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection," *Multimed. Tools Appl.*, vol. 82, no. 15, pp. 23615–23633, Jun. 2023, doi: 10.1007/s11042-023-14795-2.
29. O. Kornyo *et al.*, "Botnet attacks classification in AMI networks with recursive feature elimination (RFE) and machine learning algorithms," *Comput. Secur.*, vol. 135, p. 103456, Dec. 2023, doi: 10.1016/j.cose.2023.103456.
30. F. E. Ayo, J. B. Awotunde, S. O. Folorunso, M. O. Adigun, and S. A. Ajagbe, "A genomic rule-based KNN model for fast flux botnet detection," *Egypt. Informatics J.*, vol. 24, no. 2, pp. 313–325, Jul. 2023, doi: 10.1016/j.eij.2023.05.002.
31. D. V C, S. Rajakrishnan, S. S, S. K. T, S. T S, and S. Vajipayajula, "Enhanced Botnet Attack Detection using Machine Learning and Neural Networks," in *2024 International Conference on Inventive Computation Technologies (ICICT)*, IEEE, Apr. 2024, pp. 814–821. doi: 10.1109/ICICT60155.2024.10544628.
32. S. Afrifa, V. Varadarajan, P. Appiahene, T. Zhang, and E. A. Domfeh, "Ensemble Machine Learning Techniques for Accurate and Efficient Detection of Botnet Attacks in Connected Computers," *Eng*, vol. 4, no. 1, pp. 650–664, Feb. 2023, doi: 10.3390/eng4010039.
33. Shaker, B., Ullah, K., Ullah, Z., Ahsan, M., Ibrar, M., & Javed, M. A. (2023, November). Enhancing grid resilience: Leveraging power from flexible load in modern power systems. In 2023 18th International Conference on Emerging Technologies (ICET) (pp. 246-251). IEEE.
34. Munir, A., Sumra, I. A., Naveed, R., & Javed, M. A. (2024). Techniques for Authentication and Defense Strategies to Mitigate IoT Security Risks. Journal of Computing & Biomedical Informatics, 7(01).
35. Ali, H., Iqbal, M., Javed, M. A., Naqvi, S. F. M., Aziz, M. M., & Ahmad, M. (2023, October). Poker Face Defense: Countering Passive Circuit Fingerprinting Adversaries in Tor Hidden Services. In 2023 International Conference on IT and Industrial Technologies (ICIT) (pp. 1-7). IEEE.