JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
http://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Akash, N. S., Rouf, S., Jahan, S., Chowdhury, A., Chakrabarty, A., & Uddin, J. (2022). Botnet detection in IoT devices using random forest classifier with independent component analysis. *Journal of Information and Communication Technology, 21*(2), 201-232. https://doi.org/10.32890/jict2022.21.2.3

Botnet Detection in IoT Devices Using Random Forest Classifier with Independent Component Analysis

[1]Nazmus Sakib Akash, [2]Shakir Rouf,
[3]Sigma Jahan, [4]Amlan Chowdhury,
[4]Amitabha Chakrabarty & *[5]Jia Uddin
[1]Department of Computing & Information System,
Daffodil International University, Bangladesh
[2]Department of Computer Science & Engineering,
BRAC University, Bangladesh
[3]Faculty of Computer Science,
Dalhousie University, Canada
[4]Department of Computer Science & Engineering,
BRAC University, Bangladesh
[5]AI and Big Data Department, Endicott College,
Woosong University, South Korea

sakib.cis0021.c@diu.edu.bd; ext.shakir.rouf@bracu.ac.bd; sigma.jahan@dal.ca; amlan.chowdhury@g.bracu.ac.bd; amitabha@bracu.ac.bd; *jia.uddin@wsu.ac.kr
*Corresponding author

Received: 21/2/2021 Revised: 3/11/2021 Accepted: 23/11/2021 Published: 7/4/2022

ABSTRACT

With rapid technological progress in the Internet of Things (IoT), it has become imperative to concentrate on its security aspect. This paper represents a model that accounts for the detection of botnets through the use of machine learning algorithms. The model examined anomalies, commonly referred to as botnets, in a cluster of IoT devices attempting to connect to a network. Essentially, this paper exhibited the use of transport layer data (User Datagram Protocol - UDP) generated through IoT devices. An intelligent novel model comprising Random Forest Classifier with Independent Component Analysis (ICA) was proposed for botnet detection in IoT devices. Various machine learning algorithms were also implemented upon the processed data for comparative analysis. The experimental results of the proposed model generated state-of-the-art results for three different datasets, achieving up to 99.99% accuracy effectively with the lowest prediction time of 0.12 seconds without overfitting. The significance of this study lies in detecting botnets in IoT devices effectively and efficiently under all circumstances by utilizing ICA with Random Forest Classifier, which is a simple machine learning algorithm.

Keywords: Botnets, distributed denial of service, independent component analysis, internet of things, random forest classifier.

INTRODUCTION

The Internet of Things (IoT) is a network of physical objects equipped with sensors, software, and other technologies to communicate and share data with other devices and systems over the Internet. IoT is a phenomenon that has taken off in recent years, with the number of smart devices being forecasted at around 30.73 billion, which would mark a growth of 15 percent (Statista, 2020). This would encompass various devices such as voice controllers, doorbell cameras, smart TV, security cameras and so on, which add an extra dimension to everyday life. However, as the number of IoT devices grows, the number of security vulnerabilities from the edge to the cloud increases. IoT generally refers to an extensive network, which is practically impossible to visualize from a standalone point of view. This makes the system harder to monitor while leaving loopholes in the security protocols that can be easily bypassed to allow unfettered access to vast amounts of confidential data. The system provides little control with no evident security measures.

With the rising number of IoT based tools, it is noticeable that the devices are prone to various sorts of attacks. This is due to unique constraints like multiple technologies, multiple verticals, and resource limitations that include low memory and low computational power (Rayes & Salam, 2019). Among these attacks, botnet attacks are the most prevalent and have the most impact. Botnets refer to devices hacked by a botmaster used in various nefarious attacks, such as email spam delivery, distributed denial-of-service (DDoS) attacks, password cracking, and keylogging. To detect botnet attacks in real time, the device's network traffic must be monitored, which can be considered as high-volume data. Although it is evident that having a high volume of data allows the machine learning model to master more patterns and extrapolate new data, it is also noteworthy that adding low-quality data and input features haphazardly generates noise and increases computational time.

Nevertheless, determining which characteristics should be extracted from a dataset is challenging, and these features have a cabbalistic effect on the ultimate output of machine learning algorithms (Wang et al., 2018). In this manner, dimension reduction restricts the number of attributes in a dataset while preserving as much heterogeneity as feasible in the actual dataset. It is a part of the data pre-processing procedure that must be completed before the model can be trained, which also helps avoid overfitting and cancelling noise in data. Numerous data-dimensionality reduction strategies are available to determine how meaningful each column is and whether to remove it from the dataset.

There are two fundamental ways of dimension reduction. One way includes feature selection such as Backward Elimination, Forward Selection, and Random Forest (Pramoditha, 2021). Another refers to acquiring a new multitude of features known as feature transformation, which can be later divided into two parts: linear methods like Principal Component Analysis (PCA), Factor Analysis (FA), Linear Discriminant Analysis (LDA), and Independent Component Analysis (ICA); and non-linear methods such as Kernel PCA, t-distributed Stochastic Neighbor Embedding (t-SNE), Multidimensional Scaling (MDS), and Isometric Mapping (Ayesha et al., 2020). Diversified dimension reduction techniques have been utilized effectively to compress datasets and construct a prediction model for botnet detection.

The current research chose the Independent Component Analysis (ICA), a frequently employed linear dimension reduction technique for multi-channel data processing. ICA was deliberately considered for implementation in this particular research since the data were linear and aggregated data from various streams into a singular destination. ICA is a Blind Source Separation (BSS) technique that typically segregates multivariate input into statistically independent source inputs or components (Kaewwit et al., 2017). The approach is an explanatory tool to identify uncorrelated components that are then rotated to become as independent or non-Gaussian. In a study by Nordhausen and Oja (2018), ICA is a data analysis technique that can be thought of as an improvement of PCA, which is a prevalent dimension reduction technique. PCA is a linear transformation of data that reduces duplication measured by covariance and boosts data projected by variance (Velliangiri et al., 2019).

One drawback of PCA is that independent variables become more challenging to comprehend. Researchers have developed dimension reduction approaches as extensions of PCA, such as Kernel PCA, Multilinear PCA, and ICA, to incorporate prior data knowledge into PCA. While PCA seeks mutually orthogonal components, ICA targets mutually independent components. Since ICA reduces mutual data in identified components, it can remove many types of noise. The PCA features are sorted in strict order from the most substantial to the least important, and then some of the variables are discarded to lower the dimension. However, PCA is less appropriate theoretically in this study's dataset because ranking is not relevant. On the other hand, components obtained from ICA are inherently unsorted, equivalent, and cannot be ordered; thus, it is ideal for this research.

Several machine learning propositions have been recommended to detect botnets in IoT devices. Nevertheless, two main approaches exist in applying machine learning algorithms in IoT networks: network-based, which uses metadata from the IoT network, and host-based, which uses data on the device (Zeadally & Tsikerdekis, 2019). This study's implementation will be based on the network, which has a remarkable detection rate due to common extract flow features being independent of botnet categorization (Chen et al., 2017). In a previous study, Naïve Bayes had been implemented for botnet detection with a 97 percent accuracy (Anthi et al., 2018). Although it is considered a fair accuracy, in Naïve Bayes, all features are inherently assumed to be mutually unrelated. In reality, obtaining a collection of totally independent attributes is nearly impossible (Kumar et al., 2019). All three IoT botnet datasets included in this study were huge, high volume, and high dimensional, with a wide range of attributes. Support Vector Machine (SVM) algorithm is not ideal for such massive datasets because it will operate inefficiently with the growing number of features (Kumar, 2019).

Similar concerns may be seen with k-Nearest Neighbor (kNN), which is simple to set up yet slows down as the dataset expands and the quantity of variables increases. One of the most significant issues is that kNN selects neighbors primarily dependent on the distance criterion, which is incredibly susceptible to outliers. Considering all these, this study selected the Random Forest Classifier for detecting botnets. Random Forest is a robust supervised learning technique that identifies crucial data quickly from large datasets, and attempts to increase decision tree accuracy by reducing overfitting. This classifier also handles missing values in the data, unlike other machine learning classifiers. Random Forest's significant benefit is that it relies on a collection of different decision trees to reach any resolution (Team, 2020). Another advantage is the ability to deal with high-dimensional and complex data, which makes it a suitable classification model (Dang et al., 2020).

In the research conducted by Alrashdi et al. (2019), an anomaly detection-based fog network was used while having Random Forest Classifier as the backbone. However, the F1 score finding, which refers to the harmonic mean of precision and recall, in their proposed model was 86 percent. The lack of proper dimension reduction techniques in conjunction with Random Forest Classifier was the reason for this subpar performance. For better forecasting results, low correlations between models in Random Forest need to be ensured. With that being in mind, the current study chose ICA as the preferred dimension reduction technique. Simultaneously, ICA ensures that the components are mutually independent, making the data have a low correlation, thus, ensuring that the Random Forest Classifier model will provide more accurate results. It is to be mentioned that a model with a high correlation would, in turn, have high variance of weights, which would make the model sensitive to data and be unstable (Srinivasan, 2019). With more correlation between data, the noise and complexity of the system will increase, as would the prediction time (Chatterjee, 2018). This is primarily the reason that having low correlation in the data would ensure more accurate results. This paper reflects on taking real data from infected IoT devices in benign and attack states to derive meaningful insights.

In this paper, the problem of botnets in IoT had been targeted for finding solutions, and in the process of doing so, raw data were divided into two parts. Between these two parts, the benign data represented the state of a device where it was not under attack, while the attack data signified the opposite. The dimension reduction techniques helped decide which features would prove to be prolific in determining the outcome. This paper proposed Random Forest Classifier with Independent Component Analysis for detecting botnets in a real-world setting, which also achieved the best result in the shortest time. The research was done based on three different high-volume IoT-botnet datasets.

The research insights can be extended to other devices since the model performed well in all three IoT datasets. To sum up, this study's contributions can be divided into four key parts. Firstly, ICA was recommended as a data dimension reduction technique to characterize data without discarding components in enormous high-volume IoT-botnet datasets with high dimensions. Secondly, an intelligent botnet detection system based on Random Forest Classifier with the optimal number of trees generated through Out-of-Bag (OOB) error rate rather than using default tree numbers to avoid overfitting was proposed, which effectively achieved the highest accuracy up to 99.99 percent with the lowest prediction time of 0.12 seconds. This model was adopted because of its versatility to operate with massive datasets while ensuring a high computational pace. In addition, the aforesaid model was evaluated on three separate datasets (Ecobee_Thermostat from N-BaIoT Dataset, Provision_PT_737E_Security_Camera from N-BaIoT Dataset, and Aposemat IoT-23 Dataset) to validate the proposed approach while attaining state-of-the-art results. As a final contribution, this paper demonstrated an additional comparative result analysis of implementing several other machine learning classifiers such as k-Nearest Neighbor, Support Vector Machine, and Naïve Bayes with Independent Component Analysis for detecting botnets.

RELATED WORKS

Internet of Things (IoT) provides a massive amount of personal information and extreme detail without the user's active participation.

It has many drawbacks in terms of security and privacy. The most widely recognized security dangers are gate-crashers, mainly known to the world as infection, malware, and anomaly. These are accompanied with the threats of communicating individual information to the world and causing more digital assaults like Denial of Service (DoS), Remote to Local (R2L), Probe, and botnet. A massive botnet attack has previously ensued on Imperva, an online streaming application with 4,00,000 IoT devices to propagate a DDoS attack (Newman, 2017).

A practical example of the multiple vectors of IoT botnet attacks would be the 2016 DDoS attack on the Internet Service Provider (ISP) organization known as Dyn, which resulted in the unavailability of the system (Jain et al., 2020). Among IoT botnets, the Mirai botnet is perhaps the most recognized and feared for its ability to infect a wide array of devices while boasting a steady state of over 200,000 infected systems at its peak and being used in several high-profile DDoS attacks (Chandler et al., 2020). These IoT-related botnet attacks are directly related to the devices' lack of security infrastructure due to cost-related reasons. Therefore, the detection of IoT botnet attacks is essential with the rise of technology.

Machine learning applications with many features make training on high-dimensional data extremely sluggish and prone to overfitting. In the research of Akkalkotkar and Brown (2017), a unique method called Mixed ICA/PCA via Reproducibility Stability (MIPReSt) was developed. This used an incremental forecasting model to order diverse sources to locate the dimensions of non-Gaussian subspaces utilizing a combination of data (Akkalkotkar & Brown, 2017). However, using PCA in conjunction with ICA would not be much effective for this particular research. After looking into the limitations of the aforementioned research, the method would perform well but at the expense of a huge computational burden and an increased prediction time.

Moving on, Doshi et al. (2018) proposed that a packet-level machine learning DoS detection can precisely recognize typical DoS attack traffic from consumer IoT devices. Their research tested various machine learning classifiers such as kNN, Support Vector Machine with linear kernel (LSVM), Decision Tree using Gini impurity scores, Random Forest using Gini impurity scores (RF), and Neural Network (NN) on a dataset of ordinary and DoS attack traffic (Doshi et al., 2018). Still, no specific model has been suggested for real-world settings. Moreover, the literature lacks a thorough discussion for detecting attacks that are subtler than DoS floods. In a study from McDermott et al. (2018), the execution of deep learning in this sector with the Bidirectional Long Short-Term Memory Recurrent Neural Networking in conjunction with Word embedding for botnet detection was demonstrated. The proposal included categorizing flows into similar groups, which would decrease the complexity of training data yet risking the prospect of adding overhead, leading to a substantial rise in processing time.

According to Meidan et al.'s (2018) research on an IoT device with various functionalities, a novel network-based anomaly detection technique for the IoT was suggested to separate the network's conduct previews and use deep neural network-based autoencoders for detection. Nonetheless, the paper's definition and investigation of the subject of traffic predictability was ambiguous. Additionally, Apruzzese and Colajanni (2018) showed the vulnerability of network intrusion detection systems based on Random Forest Classifiers to adversarial attacks to enhance network intrusion detection technologies predicated on machine learning approaches.

In the comparative study executed by Brady et al. (2018), kNN, Logistic Regression (LR), Linear Discriminant Analysis (LDA), Decision Tree (CART), and Naïve Bayes (NB) were the machine learning classifiers that were considered. For both with and without feature reduction, the maximum accuracy achieved in real-time detection was 80 percent. This was impressive in its regard; however, recent research have surpassed this accuracy. Timcenko and Gajin (2018) proposed using algorithms from the SVM category: Sequential Minimal Optimization (SMO) and LibSVM Cost-Sensitive Support Vector Machine (CSVM) classifier, whereby the receiver operating characteristic (ROC) curve scores were 89 percent and 55 percent, respectively. The prediction time needed for SMO and LibSVM CSVM was 17,716 seconds and 51,512 seconds, respectively, which were not ideal for real-time detection.

From another point of view, the Interruption Identification Framework (IDF) and Interruption Anticipation Framework (IAF) are utilized to screen and recognize abnormalities or any suspicious conduct. As different conditions and most recent advances are inclined to be malignantly assaulted, machine learning calculations can identify, break down, and group gate-crashers precisely and rapidly (Dey, 2019). In the research from Su et al. (2019), they incorporated Multi-Cluster Feature Selection (MCFS) into their proposed scheme to conform to the online feature selection setting and record sensor data correlation changes in an attempt to maximize IoT equipment anomaly identification. Nevertheless, the proposed approach was not validated on a larger dataset with high dimensionality in order to demonstrate the widespread nature of sensor correlation changes. Along with machine learning classifiers, Nguyen et al. (2020) suggested implementing a Poly Software International (PSI) graph, which refers to a scientific and engineering data analysis plot, to extract the novel features of their dataset, allowing them to achieve a high true positive rate only on the condition of less complex data samples.

Although multiple approaches such as Neural Networking, Deep Learning, and various machine learning approaches have already been accounted for, this study worked with classifiers like kNN (Zhang et al., 2017), Naïve Bayes (Kolpe & Kshirsagar, 2021), and SVM (Pisner & Schnyer, 2020) for comparison with the proposed model consisting of ICA (Nordhausen & Oja, 2018) and Random Forest (Apruzzese & Colajanni, 2018), focusing on the problem of classification of compromised IoT devices. The proposed method aggravated the chances of being successful even if the data were complicated and obscure. At the same time, it could avoid categorization errors due to multi-featured IoT devices. Most researchers used real-time imbalanced datasets, risking the chances of the results being deceptive. The current study countered this obstacle through data normalization and decrease in high dimensionality, which allowed to eradicate data redundancy and increase accuracy. Furthermore, to validate the suggested approach more logically, the methodology was implemented on three different datasets and generated state-of-the-art results from all of them, thus making the work more explicable. Furthermore, the proposed model achieved the expected results with optimal efficiency in terms of data processing time.

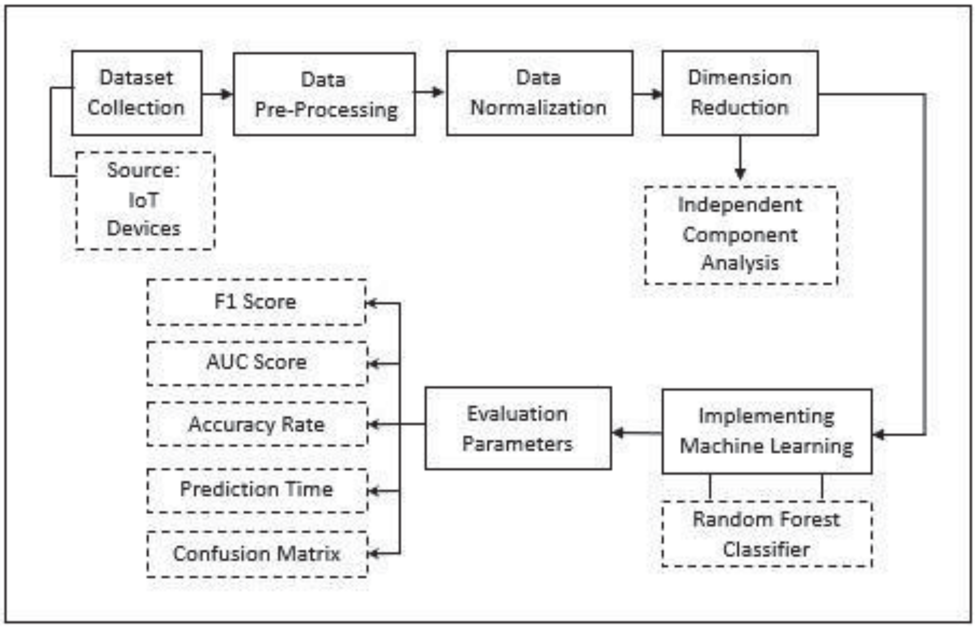
Numerous research papers employed ensemble classifiers based on Random Forest because of its impressive results, which have been empirically displayed to surpass many other machine learning methods for network intrusion detection processes. However, these methods are typically more complex and time-consuming. Random Forest classification has diversified usage in various sectors. The ensemble classifier with Random Forest can outperform any standalone classifier to solve multi-class imbalance learning problems if the data are not manipulated (Sainin et al., 2021). On the other hand, the present research indicated that with the use of a simple form of Random Forest, which utilized an optimal number of trees generated through OOB error rate along with ICA, it is possible to obtain high accuracy with minimal processing time.

PROPOSED MODEL

This research paper proposed an optimized model for botnet detection in IoT devices using machine learning classifier, and its workflow can be divided into multiple steps as shown in Figure 1.

Figure 1

Proposed Model for Botnet Detection in IoT Devices



From Figure 1, the sequential steps of the proposed model for botnet detection in IoT devices can be illustrated as shown in the block diagram:

1. Collecting the dataset (from IoT devices)
2. Data pre-processing
3. Data normalization
4. Dimension reduction (Independent Component Analysis)

5. Implementing machine learning (Random Forest Classifier)
6. Evaluation parameters: F1 score, area under the ROC curve (AUC) score, accuracy, prediction time, and confusion matrix.

Collecting the Dataset

An adequately labeled dataset for botnet detection in IoT is very rare to find. This paper looked for datasets with benign and malicious dataflows, and primarily targeted the User Datagram Protocol (UDP) dataflows because of their inherent vulnerabilities. Upon searching for such datasets, the N-BaIoT dataset was found from the University of California Irvine (UCI) Repository for Machine Learning and the Aposemat IoT-23 dataset at the Stratosphere Laboratory to fit the criteria.

From the N-BaIoT dataset, this study focused on the data of two separate devices: a security camera, labeled as Provision_PT_737E_Security_Camera, and a thermostat, labeled as Ecobee_Thermostat. The creator of the dataset was Yair Meidan and dated back to March 2018. There were two different datasets for the benign dataset labeled as benign.csv, where the data represented a phase in which the IoT, as mentioned earlier, was attack-free. Additionally, the attack data were labeled as udp.csv, which meant the stage where the device was being attacked. There were 115 features for each dataset, which had been further processed using the ICA dimension reduction technique. The number of entries in the Provision_PT_737E_Security_Camera dataset consisted of 62,154 entries for the benign dataset and 156,248 entries for the attack dataset, while the number of entries in the Ecobee_Thermostat numbered at 13,113 entries for the benign dataset and 151,481 entries for the attack dataset. The UCI Machine Learning Repository number 00442 consisted of nine different device datasets, with around 7,062,606 numbers. This study preferred to choose the security camera and thermostat datasets since they suited the research (Meidan et al., 2018).

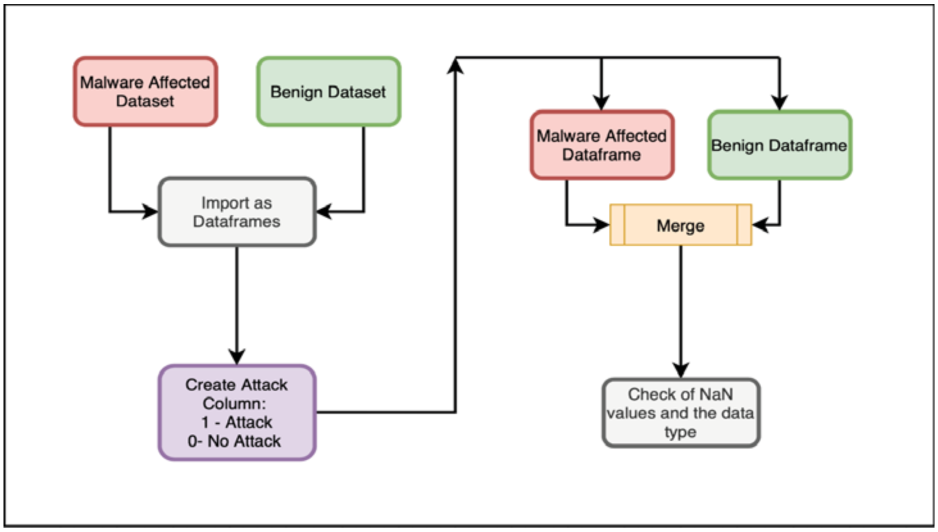
From the Aposemat IoT-23 dataset, eleven datasets were selected in total, among which eight datasets were malicious, whereby the IoT devices were being attacked, and three datasets were benign, whereby the devices in question were not being attacked. This dataset was created in 2020 as part of the Avast AIC Laboratory with the funding of Avast Software (Garcia, 2020). The combined entries for the three benign datasets amounted to 1,187 entries, and the combined entries for the eight attack datasets were numbered at 42,2021 entries while considering only the UDP protocol dataflows.

Data Pre-processing

Pre-processing the data is crucial to build any detection system, especially when using large datasets with high dimensions. A dataset of multi-various representations and sizes that has redundant features can severely impact computational performance and the detection system's accuracy (Jabbar & Mohammed, 2020). Therefore, the data need to be processed before the learning phase by following steps such as dataset cleaning, removing null attributes, and labeling.

Figure 2

N-BaIoT Dataset



- **N-BaIoT Dataset**

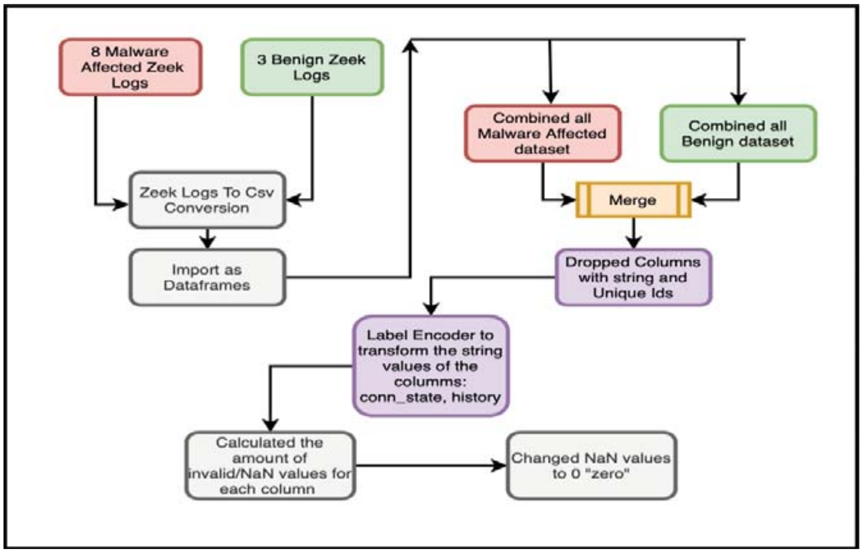
The N-BaIot dataset characteristics were multivariate and sequential. There were 115 attributes. The data type was similar for all the features, which was float64. First, the udp.csv and benign.csv datasets were read into different data frames and differentiated by introducing another attribute, labeled as 'Attack'. The 'Attack' column for the benign data frame was filled with 0s, representing no attack, while the ‘Attack’ column for the attack dataset was filled with 1s, describing an attack. Both data frames were then concatenated into a single data frame, which was named df_UDP. To eliminate repetitive and unnecessary values in the concatenated dataset, Not a Number (NaN) and infinity values were deleted from the dataset. Finally, the zero attributes (an attribute with a single value equal to zero) were removed for all the records to increase the model’s accuracy. This pre-processing methodology allowed to create data frames suitable for applying machine learning classifiers from the unprocessed Provision_PT_737E_Security_Camera and Ecobee_Thermostat data. The whole process is summarized in Figure 2.

- **Aposemat IoT-23 Dataset**

A different approach had to be taken to process the data from the Aposemat IoT-23 dataset. This dataset had 23 captures in total, from which 11 were selected for this research, among them three captures represented benign IoT device traffic, and eight captures represented various infected IoT traffic. Firstly, these selected captures had to be converted from Zeek log files to .csv format to be read as data frames. Then, the benign scenarios were taken as input in the form of data frames and concatenated as one data frame that held all the benign dataflow entries. Similarly, the remaining eight attack scenarios were read as data frames and concatenated as a single attack data frame. Similar to the aforementioned BaIot dataset pre-processing, an extra attribute labeled ‘Attack’ was introduced to the benign and attack data frames, which had all 0 values in the benign data frame to represent no attack and all 1 values in the attack data frame to illustrate an attack. These data frames were merged, and only the UDP protocol dataflow was selected to create the ‘df_UDP’ data frame. The zero attributes were removed using the label encoder. This study transformed the meaningful string values to numerical values that could be used for classification. Then, for each attribute, the amount of NaN values was calculated, and it came to notice that three columns consisted of 95 percent NaN values. Therefore, instead of dropping the NaN valued rows, they were replaced with ‘0’ so that the dataset did not lose meaningful values (Song & Szafir, 2019). The whole process is summarized in Figure 3.

Figure 3

Aposemat IoT-23 Dataset


8 Malware Affected Zeek Logs
3 Benign Zeek Logs
Zeek Logs To Csv Conversion
Import as Dataframes
Combined all Malware Affected dataset
Combined all Benign dataset
Merge
Dropped Columns with string and Unique Ids
Label Encoder to transform the string values of the columns: conn_state, history
Calculated the amount of invalid/NaN values for each column
Changed NaN values to 0 "zero"

Data Normalization

The single data frames (df_UDP) for Provision_PT_737E_Security_ Camera, Ecobee_Thermostat, and Aposemat IoT-23 datasets were normalized because all three contained attributes of different scales. The data frames were normalized to eliminate the measurement units for the data and better compare the data from various attributes. The data were rescaled using a standard scaler so that they centered around 0. This is known as feature scaling, and the formula is in Equation 1:

$$x_{new} = \frac{x-\mu}{\sigma} \tag{1}$$

where, σ refers to Standard Deviation and μ denotes Mean.

Dimension Reduction

The more features available in the dataset, the harder it becomes to interpret the data. Therefore, dimension reduction techniques were performed on the data frames from three different datasets to allow better interpretability and eliminate attributes that were not needed in the prediction. The methodology was to test various dimension reduction approaches and compare the results to find the most appropriate approach for the datasets. The main target behind this approach was to reduce the dimensions using the most appropriate dimension reduction technique to ensure a better machine learning classification. Consequently, ICA (Nordhausen & Oja, 2018) was applied to the data frame to reduce the dataset's dimension during the botnet detection process.

The distinguishing factor in the case of ICA is that it searches for components that are both statistically independent and non-Gaussian (Hyvärinen et al., 2001). In the three datasets that were dealt with for this research, the data were multivariate, and thus, ICA was a natural choice for reducing the dimensions. In the security camera and thermostat datasets collected from the N-BaIot dataset, ICA was performed to derive two datasets of six independent components from the training and test datasets that consisted of 115 attributes after they were normalized. In the case of the Aposemat IoT-23 dataset, ICA was conducted on the training and test datasets found after pre-processing and normalization to find six independent components from 13 independent attributes. The *n*_components in ICA were selected to be six after they proved to provide the best results in terms of confusion matrix and accuracy through rigorous experimentation of other possible values.

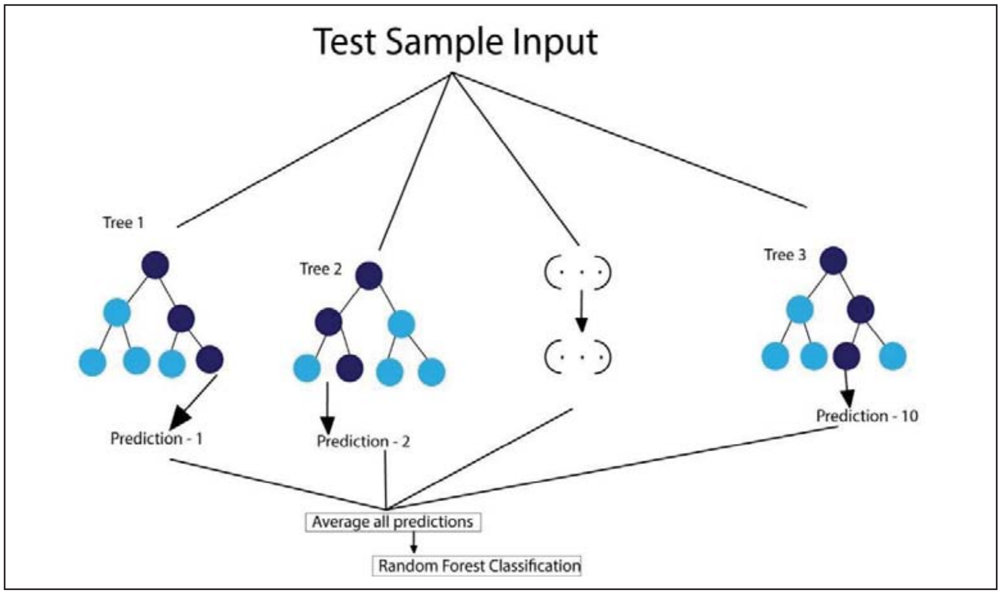
Implementing the Machine Learning Classifier

There has already been some significant investigation regarding botnet detection, with prevailing multiple deep learning, neural networking, and machine learning approaches. Nevertheless, this study explored a few more classifiers with some recurrent strategies, focusing mainly on Random Forest Classifiers and classified the three datasets by using them to compare results.

The Random Forest Classifier is termed as an ensemble algorithm as it encapsulates more than one algorithm, which might be of the same or different kind. It comprises a large number of individual decision trees that operate together at the same time. Each tree produces a class prediction, and the class with the most votes is chosen as the model's prediction by Random Forest. Each tree derives its input from the dataset's sampled data and the features available; a subset of features is selected for each node (Yiu, 2019). The trees do not involve any pruning. The advantages of Random Forest are its capability to work on large databases and maintain a fast-computational speed efficiently; however, it is prone to overfitting. Nevertheless, this research used ICA for reducing the dimension of datasets and choosing the optimal number of trees generated through the OOB error rate instead of using an arbitrary or default value of tree number in order to avoid overfitting. For this reason, the current study primarily focused on this algorithm for the given dataset. The algorithm is visualized in Figure 4 in terms of test inputs.

Figure 4

Random Forest Classification



Firstly, the classifier worked on the dataset by creating multiple different decision trees. Then, it trained each decision tree on multiple divergent samples where the sampling was done through replacements. The process, in turn, aided in having a better interpretation of the bias and variance. Since the dimension of the dataset was already reduced using the ICA dimension reduction approach, feature importance was not calculated to drop any more additional attributes. Nevertheless, there were three medium-sized datasets that had varying data with different attributes. The OOB error evaluated the accuracy of Random Forest and selected optimal values for tuning parameters. Random Forest is beneficial since it usually produces reasonably good results with the default hyperparameters: number of trees and number of variables available for splitting at each tree node or known as m_{try}. Even with more features, the classifier is unlikely to overfit the model if there are enough trees in the forest. The biggest drawback of Random Forest is that it can become too sluggish and ineffective for real-time forecasts if there are too many trees. For this reason, finding out the optimal number of trees was very crucial.

Therefore, the OOB score was calculated to find the best possible *n*_estimators value, which would specify the number of trees generated. The OOB score was generated for each dataset, having the value of *n*_estimators iterate between a range of 10 to 200. The lowest number of trees to generate was found, whic would have less variance and, in turn, not be prone to overfitting (Kunchhal, 2020). Furthermore, the lowest number of trees were selected to maintain less computation time during the actual prediction. The least OOB errors with the number of *n*_estimators are shown in Table 1 for all three datasets.

Table 1

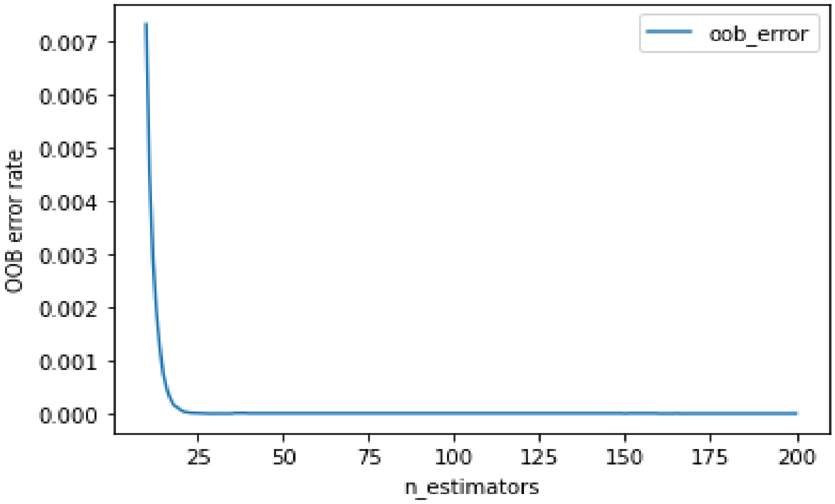
OOB Error of Three Datasets

Dataset	*n*_estimators	OOB Error (%)
Ecobee_Thermostat	26	0
Provision_PT_737E_Security_Camera	28	0
Aposemat IoT-23	69	0

In Figure 5, the curve of the number of estimators (shown in x-axis) versus the OOB error rate (shown in y-axis) are displayed for the Provision_PT_737E_Security_Camera dataset.

Figure 5

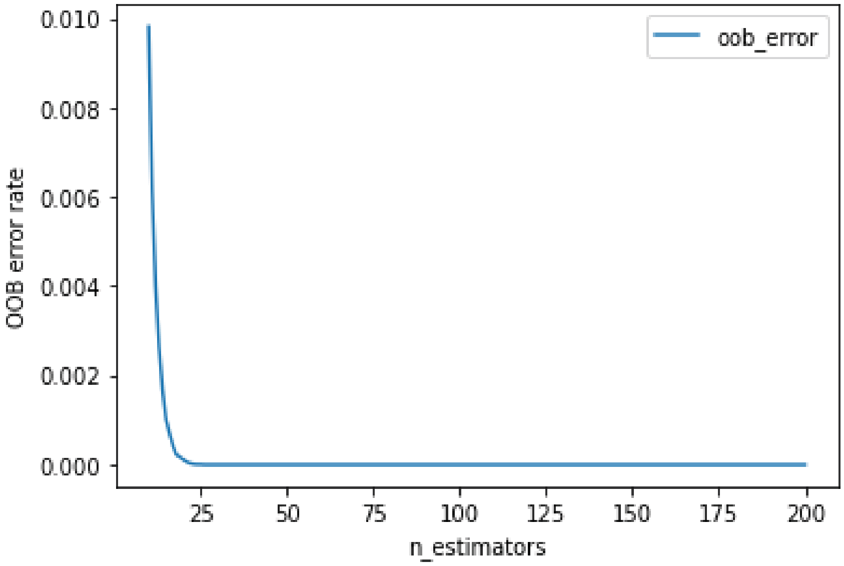
OOB Error Rate of Provision_PT_737E_Security_Camera



In Figure 6, the curve of the number of estimators (shown in x-axis) versus the OOB error rate (shown in y-axis) are displayed for the Ecobee_Thermostat dataset.

Figure 6

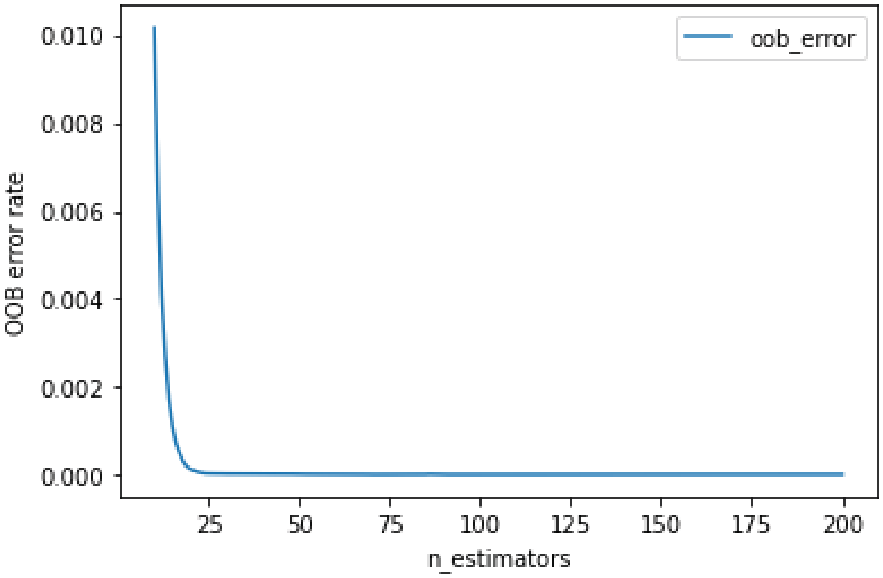
OOB Error Rate of Ecobee Thermostat



In Figure 7, the curve of the number of estimators (shown in x-axis) versus the OOB error rate (shown in y-axis) are displayed for the Aposemat IoT-23 dataset.

Figure 7

OOB Error Rate of Aposemat IoT-23



Using the lowest *n*_estimators that produced low variance, Random Forest Classifier was conducted on the normalized test set.

Evaluation Parameters

Evaluation parameters were used to compare the results of models implemented on the dataset. There were several performance metrics as evaluation parameters for the detection system. The following parameters were used for analyzing the results: F1 score, Accuracy, AUC score, prediction time, and confusion matrix.

- F_1 Score: The F_1 score is an accuracy metric based on a confusion matrix's sensitivity and precision. Sensitivity shows how much of the actual positives the model has captured out of all the actual positives. While precision is the calculation that determines how much of the predicted positive is actually positive.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2}$$

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

$$F_1 score = 2 \times \frac{Precision \times Sensitivity}{TPrecision + Sensitivity} \tag{4}$$

- AUC (Area under ROC curve): This is another performance measurement metric for classifiers. ROC (Receiver Operating Characteristics) curve is a probability curve that distinguishes between the true positive rate and false positive rate (Narkhede, 2019). AUC determines the fraction of area that falls underneath the ROC curve.
- Accuracy: Accuracy is one of the most common metrics that are used to compare classification models. It determines the fraction of correct predictions made by a model to the total number of predictions it has made (*Classification: Accuracy | Machine Learning Crash Course*, 2020). The following equation shows how it is calculated:

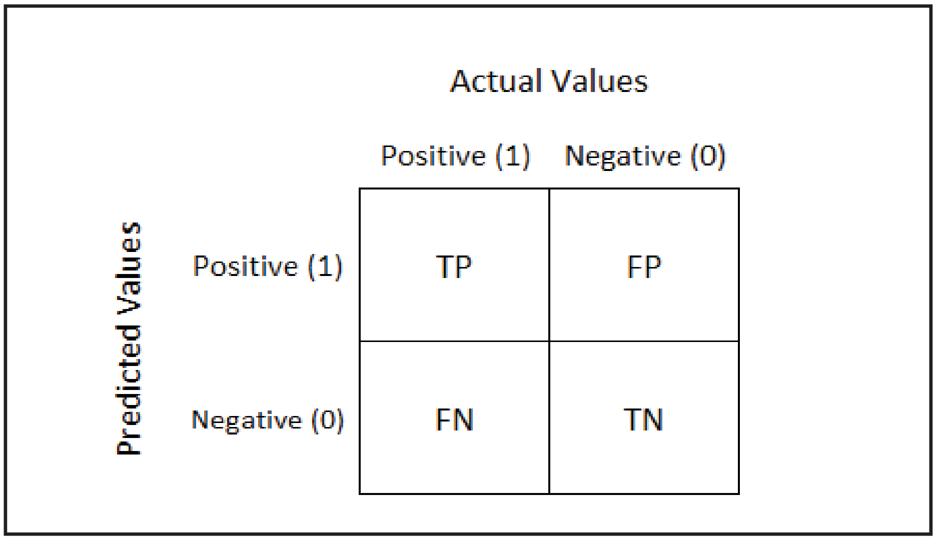
$$Accuracy = \frac{True\ Positive + True\ Negatives}{Total\ predictions} \tag{5}$$

- Prediction time: Its functions involve measuring or timing each classifier's prediction phase. It is assumed that, when put in a real-life situation, the speed at which the model predicts an outcome will prove to be most beneficial.

- Confusion matrix: While the accuracy measuring metrics provide a general overview, the confusion matrix shows the prediction's actual results into four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Figure 8 showcases the position of TP, TN, FP, and FN in the matrix.

Figure 8

General Confusion Matrix



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

True Positive and True Negative show the fraction of test data that have been correctly predicted. While False Positive and False Negative indicate the segment of test data that the classifier has wrongly predicted. Generating a confusion matrix by plotting with the package becomes more comfortable to visualize.

RESULTS AND ANALYSIS

In this chapter, the results will be reviewed based on the aforementioned evaluation parameters. Finally, this chapter will end by comparing the results with machine learning algorithms from previous related research in botnet detection and presenting the comparative analysis to illustrate why the chosen model of Random Forest Classifier with Independent Component Analysis (ICA) provides the best possible results in the shortest amount of time. The experiment used Spyder

(Python 3.8) in a Windows OS environment on 16GB RAM and 2.6 GHz Intel Core i7. Moreover, two labels were introduced for the transport data, ‘0’ and ‘1’, whereby 0 refers to Normal/Non-attack, and 1 refers to Abnormal/Attack. It is also noteworthy that the entire dataset was split into 7:3, whereby 70 percent of the data were used to train the classifiers, and 30 percent of the information were used to test the trained classifiers.

In Table 2, the accuracy score, F1 score, AUC score, and prediction time found for the Provision_PT_737E_Security_Camera dataset are listed for the four classifiers while ICA is used as the dimension reduction technique.

Table 2

Result of Provision_PT_737E_Security_Camera Dataset

Sl No.	Classifiers	Parameters	ICA
1.	kNN	F1 Score (%)	99.98
		AUC (%)	99.97
		Accuracy (%)	99.97
		Prediction Time (sec.)	6.55
2.	Naïve Bayes	F1 Score (%)	96.77
		AUC (%)	91.67
		Accuracy (%)	95.22
		Prediction Time (sec.)	0.02
3.	Random Forest Classifier	F1 Score (%)	99.99
		AUC (%)	99.99
		Accuracy (%)	99.99
		Prediction Time (sec.)	0.16
4.	Support Vector Classifier	F1 Score (%)	99.92
		AUC (%)	99.81
		Accuracy (%)	99.89
		Prediction Time (sec.)	314.63

Figure 9 portrays the confusion matrix generated for the Provision_ PT_737E_Security_Camera dataset while using Random Forest Classifier along with ICA.

Figure 9

Confusion Matrix of Provision_PT_737E_Security_Camera Dataset

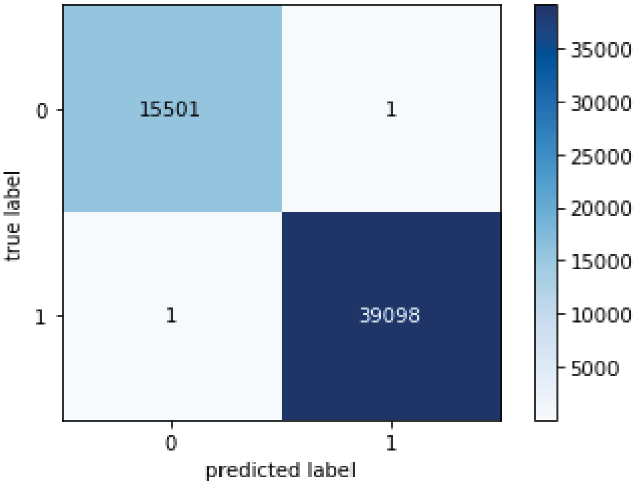
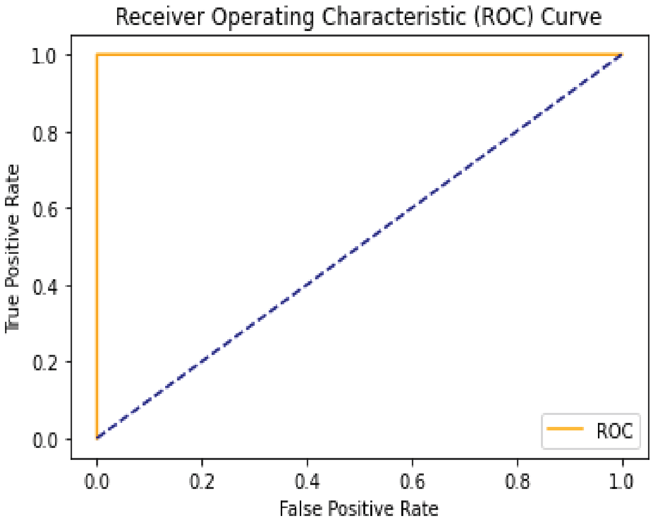


Figure 10 depicts the ROC curve generated for the Provision_ PT_737E_Security_Camera dataset while using Random Forest Classifier along with ICA. using Random Forest Classifier along with ICA.

Figure 10

ROC Curve of Provision_PT_737E_Security_Camera Dataset



In Table 3, the accuracy score, F1 score, AUC score, and prediction time found for the Ecobee_Thermostat dataset are listed for the four classifiers while ICA is used as the dimension reduction technique.

Table 3

Result of Ecobee_Thermostat Dataset

Sl No.	Classifiers	Parameters	ICA
1.	kNN	F1 Score (%)	99.98
		AUC (%)	99.82
		Accuracy (%)	99.97
		Prediction Time (sec.)	3.69
2.	Naïve Bayes	F1 Score (%)	99.57
		AUC (%)	100
		Accuracy (%)	100
		Prediction Time (sec.)	0.01
3.	Random Forest Classifier	F1 Score (%)	100
		AUC (%)	100
		Accuracy (%)	100
		Prediction Time (sec.)	0.12
4.	Support Vector Classifier	F1 Score (%)	97.07
		AUC (%)	64.63
		Accuracy (%)	94.44
		Prediction Time (sec.)	53.09

Figure 11 portrays the confusion matrix generated for the Ecobee_ Thermostat dataset while using Random Forest Classifier along with ICA.

Figure 11

Confusion Matrix of Ecobee_Thermostat Dataset

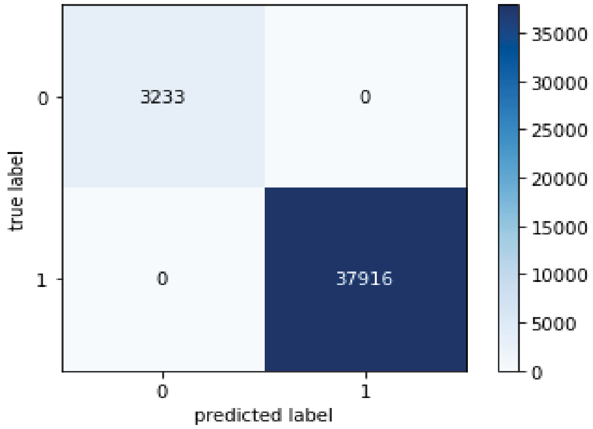
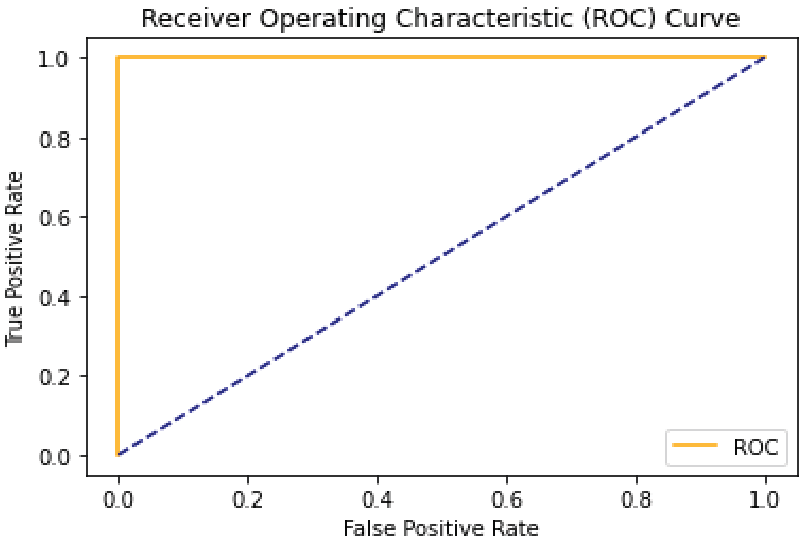


Figure 12 depicts the ROC Curve generated for the Ecobee_Thermostat dataset while using Random Forest Classifier along with ICA.

Figure 12

ROC Curve of Ecobee_Thermostat Dataset



In Table 4, the accuracy score, F1 score, AUC score, and prediction time found for the Aposemat IoT-23 dataset are listed for the four classifiers while ICA is used as the dimension reduction technique.

Table 4

Result of Aposemat IoT-23 Dataset

Sl No.	Classifiers	Parameters	ICA
1.	kNN	F1 Score (%)	100
		AUC (%)	100
		Accuracy (%)	100
		Prediction Time (sec.)	8.49
2.	Naïve Bayes	F1 Score (%)	98.73
		AUC (%)	93.24
		Accuracy (%)	97.50
		Prediction Time (sec.)	0.09

(continued)

Sl No.	Classifiers	Parameters	ICA
3.	Random Forest Classifier	F1 Score (%)	99.99
		AUC (%)	99.64
		Accuracy (%)	99.99
		Prediction Time (sec.)	0.65
4.	Support Vector Classifier	F1 Score (%)	99.87
		AUC (%)	50.18
		Accuracy (%)	99.87
		Prediction Time (sec.)	9.74

Figure 13 portrays the confusion matrix generated for the Aposemat IoT-23 dataset while using Random Forest Classifier along with ICA.

Figure 13

Confusion Matrix of Aposemat IoT-23 Dataset

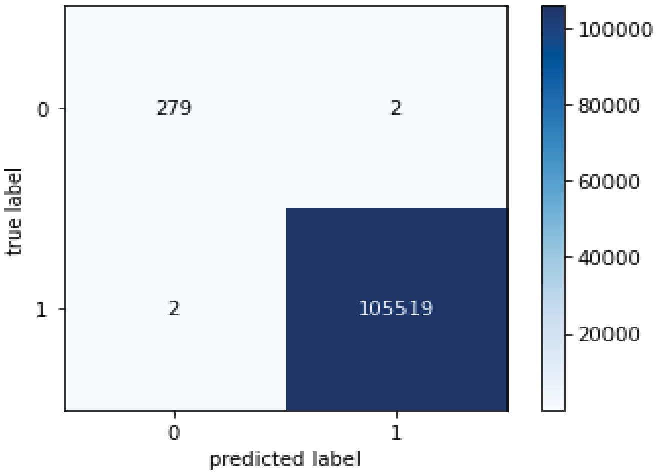
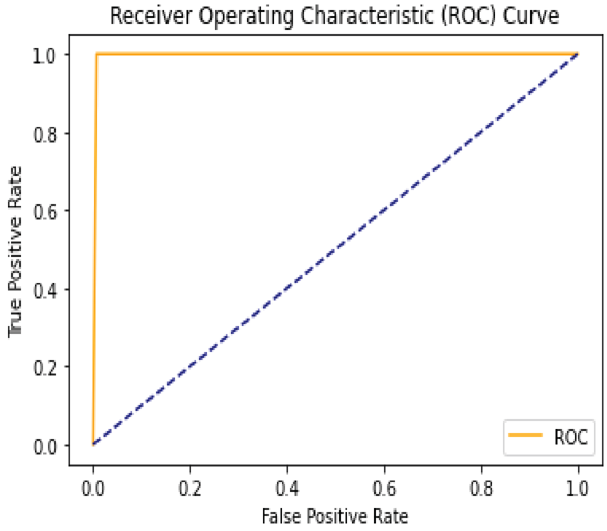


Figure 14 depicts the ROC Curve generated for the Aposemat IoT-23 dataset while using Random Forest Classifier along with ICA.

Figure 14

ROC Curve of Aposemat IoT-23 Dataset



The combination of Random Forest Classifier with ICA dimension reduction technique provided some of the best results while detecting botnets in the three datasets, as shown in Tables 2, 3, and 4. This study worked with three different botnet datasets in the experiments. The results demonstrated a similar pattern throughout the datasets with the initially proposed model, which validated the proposed model beyond question. Table 2 shows the comparative analysis of multiple classification algorithms for the Provision_PT_737E_Security_Camera dataset. Similarly, Tables 3 and 4 display the same analysis for the 23 datasets from the Ecobee_Thermostat and Aposemat IoT-23 datasets.

First and foremost, prediction time is an essential metric to evaluate a model's performance in classification, especially when it comes to security issues. From the experimental results, the kNN classifier achieved a fairly good accuracy score in all three datasets. However, the efficiency and speed of computation declined drastically with a larger dataset. Additionally, the kNN classifier did not work well with datasets that had a non-uniform distribution of classes. Looking toward the experimental results, it can be seen that the kNN classifier had a prediction time of 6.55 seconds for the Provision_PT_737E_Security_Camera dataset, 3.69 seconds for Ecobee_Thermostat, and 8.49 seconds for the Aposemat IoT-23 dataset.

Similar to the kNN classifier, the SVM classifier was inefficient for large datasets. With an exceeding number of features, SVM tended to perform poorly at classification. The experimental output depicted that SVM took the longest to predict, taking 53.09 seconds to predict in the Ecobee Thermostat dataset. This was the worst prediction time encountered in the experiment throughout all three datasets and four classifiers. The lengthy prediction time could be attributed to SVM's complicated algorithm structure. Since prediction time was crucial in the real-time botnet detection system, both kNN and SVM would be ineffective as a proper classification model. In contrast, the proposed model of Random Forest Classification provided the shortest prediction time, which was 0.16 seconds for the Provision_PT_737E_Security_Camera dataset, 0.12 seconds for the Ecobee_Thermostat dataset, and 0.65 seconds for the Aposemat IoT-23 dataset without compromising accuracy.

In case of the result, it was found that for the Naïve Bayes classifier, the prediction time, accuracy, F1 score, and AUC score were all acceptable. However, this model had a massive shortcoming because it could not handle a dataset that included complex features, which is ever-present in the network traffic of IoT devices. Furthermore, Naïve Bayes avoided noise to the extent that it might be problematic in the specified scenario of botnet detection. While this study implemented the Random Forest Classifier on the three datasets, the OOB error was ensured to be at a minimum, and thus, the variance among the data was lessened. This provided a model that would not overfit in any scenario, which is why this is a suitable model for anomaly detection.

In the research conducted by Stoian (2020), using the same Aposemat IoT-23 dataset to predict botnets, the researcher found Naïve Bayes classifier, Artificial Neural Network, and Adaboost to have an F1 score of 25 percent, 52 percent, and 83 percent, respectively. The proposed model of Random Forest Classifier in tandem with ICA dimension reduction surpassed all these results, giving an F1 score of 99.99 percent on the same dataset. Upon observing all the results of the experiment of three different datasets and the experiment of another researcher on one of those three datasets, it can be summated that Random Forest Classifier with ICA dimension reduction approach provided the best solution to the IoT botnet detection problem.

CONCLUSION

To conclude this research, the model of Random Forest Classifier with Independent Component Analysis dimension reduction technique provided the best results for real-time botnet detection. This statement refers to the high accuracy in prediction that this model had shown in all the datasets it was run on. The outstanding results came from the multivariate datasets being broken down into six statistically independent components and classified using the best possible number of *n*_estimators in Random Forest Classification. Similar research work found on IoT botnet datasets also suggested Random Forest Classification to be one of the better models for intrusion detection and thus supported this study's claim. Introducing ICA as the dimension reduction technique brought novelty to the whole research as this is a combination that was previously not worked on much, yet it yielded great results and thus, it can be called the best overall choice.

As for future work direction, this model is planned to be enhanced and to develop an intrusion prevention system in tandem with botnet attack detection. Real-time detection and immediate response to botnet attacks are challenging. The Mirai source code can be altered in multiple ways to bypass the security protocols. Still, there is a wide scope for this research, and the increasing amounts of botnet-related attacks compel researchers to carry on this research.

ACKNOWLEDGMENT

This research paper is an extension of a bachelor's degree thesis from BRAC University titled "Botnet detection In IoT devices using machine learning" (http://hdl.handle.net/10361/14069). This research received no specific grant from any agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Akkalkotkar, A., & Brown, K. S. (2017). An algorithm for separation of mixed sparse and gaussian sources. *Plos One*, *12*(4), e0175775. https://doi.org/10.1371/journal.pone.0175775

Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., & Ming, H. (2019, January). AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. In

Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, (pp. 305–310). IEEE. https://doi.org/10.1109/CCWC.2019.8666450

Anthi, E., Williams, L., & Burnap, P. (2018, March). Pulse: An adaptive intrusion detection for the internet of things. In *Proceedings of the Living in the Internet of Things: Cybersecurity of the IoT – 2018* (pp. 4). https://digital-library.theiet.org/content/conferences/10.1049/cp.2018.0035

Apruzzese, G., & Colajanni, M. (2018, November). Evading botnet detectors based on flows and random forest with adversarial samples. In *Proceedings of the 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, USA, (pp. 1–8). IEEE. https://doi.org/10.1109/NCA.2018.8548327

Ayesha, S., Hanif, M. K., & Talib, R. (2020). Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion*, *59*, 44–58. https://doi.org/10.1016/j.inffus.2020.01.005

Brady, S., Magoni, D., Murphy, J., Assem, H., & Portillo-Dominguez, A. O. (2018, November). Analysis of machine learning techniques for anomaly detection in the internet of things. In *Proceedings of the 2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Gudalajara, Mexico, (pp. 1–6). IEEE. https://doi.org/10.1109/LA-CCI.2018.8625228

Chandler, J., Fisher, K., Chapman, E., Davis, E., & Wick, A. (2020, January). Invasion of the botnet snatchers: A case study in applied malware cyberdeception. In *Proceedings of the 53rd Hawaii International Conference on System Sciences, Cyber Deception for defense, Digital Government*, (pp. 1–10). https://doi.org/10.24251/HICSS.2020.229

Chen, R., Niu, W., Zhang, X., Zhuo, Z., & Lv, F. (2017, April). An effective conversation-based Botnet detection method. *Mathematical Problems in Engineering*, *2017* (pp. 1–9). https://doi.org/10.1155/2017/4934082

Dang, X., Cao, Y., Hao, Z., & Liu, Y. (2020). WiGId: Indoor group identification with CSI-based random forest. *Sensors*, *20*(16), 4607. https://doi.org/10.3390/s20164607

Dey, A. (2019, May 10). *Internet of things (IoT)-security, privacy, applications & trends*. Medium. https://medium.com/@arindey/internet-of-things-iot-security-privacy-applications-trends-3708953c6200

Doshi, R., Apthorpe, N., & Feamster, N. (2018, May). Machine learning ddos detection for consumer internet of things devices. In *Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA (pp. 29–35). IEEE. https://doi.org/10.1109/SPW.2018.00013

Garcia, S., Parmisano, A., & Erquiaga, M. J. (2020, January 20). *IoT-23: A labeled dataset with malicious and benign IoT network traffic*. Zenodo. https://doi.org/10.5281/zenodo.4743746

Genesis (2018, September 25). *Pros and cons of K-nearest neighbors.* https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors

Hyvärinen, A., Karhunen, J., & Oja, E. (2004). *Independent component analysis* (1st ed.). John Wiley & Sons. https://www.wiley.com/en-au/Independent+Component+Analysis-p-9780471405405

Jabbar, A. F., & Mohammed, I. J. (2020, November). Development of an optimized botnet detection framework based on filters of features and machine learning classifiers using cicids2017 dataset. In *IOP Conference Series: Materials Science and Engineering*, 928 032027. https://doi.org/10.1088/1757-899x/928/3/032027

Jain, L. C., Tsihrintzis, G. A., Balas, V. E., & Sharma, D. K. (Eds.). (2020b). *Data communication and networks: Proceedings of GUCON 2019*. Springer. https://doi.org/10.1007/978-981-15-0132-6

Kaewwit, C., Lursinsap, C., & Sophatsathit, P. (2017). High accuracy EEG biometrics identification using ICA and AR model. *Journal of Information and Communication Technology*, *16*(2), 354–373. https://doi.org/10.32890/jict2017.16.2.8236

Kumar, D. (2019, June 14). *Top 4 advantages and disadvantages of support vector machine or SVM*. Medium. https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107

Kumar, N. (2019, March 2). *Advantages and disadvantages of naive bayes in machine learning*. The Professionals Point. http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of-naive.html

Kunchal, R. (2020, December 11). *Out-of-bag (OOB) score in the random forest algorithm*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/12/out-of-bag-oob-score-in-the-random-forest-algorithm/

Machine Learning Crash Course. (2020, February 10). *Classification: accuracy*. Google Developers. https://developers.google.com/machine-learning/crash-course/classification/accuracy

McDermott, C. D., Majdani, F., & Petrovski, A. V. (2018, July). Botnet detection in the internet of things using deep learning approaches. In *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). https://doi.org/10.1109/ijcnn.2018.8489489

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—Network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, *17*(3), 12–22. https://doi.org/10.1109/mprv.2018.03367731

Narkhede, S. (2019, May 26). *Understanding AUC-ROC curve-towards data science*. Medium. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Newman, L. H. (2017, June 3). *Friday's east coast internet outage is a major ddos attack.* Wired. https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/

Nguyen, H. T., Ngo, Q. D., Nguyen, D. H., & Le, V. H. (2020). PSI-rooted subgraph: A novel feature for IoT botnet detection using classifier algorithms. *ICT Express*, *6*(2), 128–138. https://doi.org/10.1016/j.icte.2019.12.001

Nordhausen, K., & Oja, H. (2018). Independent component analysis: A statistical perspective. *Wiley Interdisciplinary Reviews: Computational Statistics*, *10*(5), e1440. https://doi.org/10.1002/wics.1440

Pramoditha, R. (2021, May 5). *11 dimensionality reduction techniques you should know in 2021-towards data science*. Medium. https://towardsdatascience.com/11-dimensionality-reduction-techniques-you-should-know-in-2021-dcb9500d388b

Rayes, A., & Salam, S. (2019). *Internet of things from hype to reality: The road to digitization* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-99516-8

Sainin, M. S., Alfred, R., & Ahmad, F. (2021). Ensemble meta classifier with sampling and feature selection for data with imbalance multiclass problem. *Journal of Information and Communication Technology*, *20*(2), 103–133. https://doi.org/10.32890/jict2021.20.2.1

Security. (2019, October 22). *First three quarters of 2019: 7.2 billion malware attacks, 151.9 million ransomware attacks*. Security Magazine. https://www.securitymagazine.com/articles/91133-first-three-quarters-of-2019-72-billion-malware-attacks-1519-million-ransomware-attacks

Song, H., & Szafir, D. A. (2019). Where's my data? Evaluating visualizations with missing data. *IEEE Transactions on Visualization and Computer Graphics*, *25*(1), 914–924. https://doi.org/10.1109/tvcg.2018.2864914

Statista Research Departement. (2020, November 26). *Internet of things-number of connected devices worldwide 2015-2025*. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

Stoian, N. A. (2020). *Machine learning for anomaly detection in IoT networks: Malware analysis on the IoT-23 data set*. (Essay (Bachelor), University of Twente). http://essay.utwente.nl/81979/

Su S, Sun Y, Gao X, Qiu J, Tian Z. (2019). A correlation-change based feature selection method for IoT equipment anomaly detection. *Applied Sciences*, *9*(3), 437. https://doi.org/10.3390/app9030437

Timcenko, V., & Gajin, S. (2018). Machine learning based network anomaly detection for IoT environments. In L. Moutinho & X. Yang (CC), *Proceedings of the International Conference on Intelligent Science and Technology*. ResearchGate. https://www.researchgate.net/publication/327652075_Machine_Learning_based_Network_Anomaly_Detection_for_IoT_environments

Trehan, D. (2020, July 2). *Why choose random forest and not decision trees*. Towards AI. https://towardsai.net/p/machine-learning/why-choose-random-forest-and-not-decision-trees

Velliangiri, S., Alagumuthukrishnan, S., & Thankumar Joseph, S. I. (2019). A review of dimensionality reduction techniques for efficient computation. *Procedia Computer Science*, *165*, 104–111. https://doi.org/10.1016/j.procs.2020.01.079

Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, *48*(C), 144–156. https://doi.org/10.1016/j.jmsy.2018.01.003

Yiu, T. (2019, August 14). *Understanding random forest-towards data science*. Medium. https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Zeadally, S., & Tsikerdekis, M. (2019). Securing internet of things (IoT) with machine learning. *International Journal of Communication Systems*, *33*(1). https://doi.org/10.1002/dac.4169