2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress

Malicious Network Traffic Detection in IoT Environments Using A Multi-level Neural Network

Menglu Li, Eleonora Achiluzzi, Md Fahd Al Georgy, Rasha Kashef
Electrical, Computer, and Biomedical Engineering Department
Ryerson University
Toronto, Ontario, Canada
{menglu.li@ryerson.ca, eachiluzzi@ryerson.ca, mgeorgy@ryerson.ca, rkashef@ryerson.ca}

***Abstract*— The Internet of Things (IoT) is a system that connects physical computing devices, sensors, software, and other technologies, and data can be collected, transferred, and exchanged with other devices over the network without requiring human interactions. One challenge that the development of IoT is facing is the existence of malicious botnet attacks. Recently, research on botnet traffic detection has become popular. However, most state-of-the-art detection techniques focus on one specific type of device in IoT or one particular botnet attack type. Therefore, we propose a neural network-based algorithm, 2-FFNN, which can detect malicious traffic in the IoT environment and be deployed generally without restricting device or attack types. The proposed model consists of two levels of the Feed Forward Neural Network framework to identify some hard-to-detect botnet attacks. Experimental analysis has shown that the 2-FFNN outperforms the baseline FFNN and some state-of-the-art methods based on the detection accuracy and ROC score.**

Keywords—Anomaly detection, Botnet traffic, Neural networks, IoT

I. INTRODUCTION

The Internet is full of cybersecurity threats; many are mainly suitable indicators but serve the wrong purpose. A typical example is the malicious use of 'Botnets'. A botnet is simply a network of computers coordinated to perform some evil and unauthorized actions. The mischievous activities can range from simply maintaining a chatroom to taking total control of distant computer resources. Botnets have emerged as one of the most critical perils in the modern internet. These "bots" operate under the remote control of a human operator called the "Botmaster". Botmasters take control of host computers by remotely running the individual bots on targeted host computers. According to their annual report, Spamhaus Malware Labs detected and blocked more than 17000 botnet command and control (C&C) servers hosted on 1210 different networks in 2019. The research report states that the number is a massive 71.5% increase from the number of botnet C&Cs found in the previous year. Since 2017, the number of freshly identified botnet C&Cs has skyrocketed from 9,500 to 17,602 [1]. Botnets have emerged as a significant threat vector against cybersecurity. They prove to be a convenient medium to perform many crimes such as Distributed Denial of Service (DDoS) attacks, malware distribution, phishing, and click fraud. As a result, the identification and detection of botnets have been growing as a topic of interest among cybersecurity researchers lately. Both academic and industry researchers have come up with different solutions to prevent botnet attacks. There are primarily two methodologies for botnet detection. The first approach is based on 'honeynets'. A 'honeynet' is a decoy network setup that looks like a real network. Its primary purpose is to invite attacks, to gather more information on attackers' activities and methods. The second approach is built on passive network traffic monitoring and analysis. While honeynets fail to detect botnets, passive network monitoring can detect botnets in the network. Passive network monitoring techniques can be further divided into signature-based, anomaly-based, DNS-based, and mining-based detection [2]. The diffusion of the internet of things (IoT) creates smart gadgets as indispensable elements of modern human lives. However, on the other hand, it is also exposing us to increased threats of severe malware infestation. Botnets are one of the most extensive and hazardous malware, so detection and prevention of Botnets are mission-critical objectives to ensure our cybersecurity. Most of the current botnet attack detection techniques focused on either one type of device or one botnet attack type. Therefore, there is a gap in developing a detection technique that can discover malicious traffic in the IoT environment with high accuracy, without restrictions by the types of devices or attacks.

In this paper, we propose a botnet detection algorithm called 2-FFNN to detect malicious IoT botnet attacks. The model is based on a two-level neural network technique for malicious network traffic detection in IoT environments. The model is a novel anomaly detection technique that can detect malicious traffic in the IoT environment with high accuracy, without restrictions by the types of devices or attacks. In our experimental work, we utilized a publicly available botnet dataset. The dataset contained real-world data gathered from 3 commercial IoT devices. Our experiment results show that the proposed model achieved an average accuracy and ROC score of 97.93% and 0.9838, respectively. The results are then used to validate that the second level of the feed-forward neural network (2-FFNN) model caught the attack traffic data leaked from the first level classification.

The remainder of this paper is organized as follows: Section 2 introduces the related work of botnet attack detection in the IoT environment. Section 3 highlights the proposed multi-level detection model for malicious traffic. Section 4 presents the experiential analysis. Finally, the conclusion and future directions are summarized in Section 5.

II. RELATED WORK

There has been a significant increase in Botnet attacks on IoT devices in recent years. Therefore, researchers are

978-1-6654-2174-4/21/$31.00 ©2021 IEEE
DOI 10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00039

2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech) | 978-1-6654-2174-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/DASC-PICOM-CBDCOM-CYBERSCITECH52372.2021.00039

concentrating more on the detection system for the IoT environment. In a study by Soe et al. [3], the researchers suggested a botnet attack detection framework based on a sequential detection architecture. They adopted a feature selection approach to implementing a lightweight detection system with high efficiency. According to their research result, the overall detection performance was around 99% for the botnet attack detection. Soe et al. used three Machine Learning algorithms: Artificial Neural Network (ANN), J48 Decision Tree, and Naïve Bayes algorithms. The experiment results showed that the proposed setup could effectively identify botnet-based occurrences and be stretched to detect new attacks. Shafiq et al. [4] proposed a framework to identify malicious activities in the IoT environment. They used naïve Bayes as an effective algorithm for the anomaly detection system in their research. The researchers used a publicly available BoT-IoT identification dataset and selected 44 useful features from several ML algorithms' features. Then they selected five ML algorithms for malicious traffic identification. The algorithms were Bayes Net, C4.5 decision tree, Naïve Bayes, Random Forest, and Random Tree, and selected the most widely used ML algorithm performance evaluation metrics. The researchers adopted a ' bijective soft set approach' to determine which ML algorithm is useful and should be used. Based on the Bijective soft, the researchers proposed an algorithm. Their result showed that the Naïve Bayes ML algorithm effectively detects anomaly and intrusion detection in the IoT environment. Cervantes et al. [5] proposed a cyber-attack detection system called INTI (Intrusion detection of Sinkhole attacks on 6LoWPAN for the IoT). Their approach is based on a 'dynamic clustering' mechanism used to support data transmission and examine the behaviour of router nodes in the forwarding task. The maliciousness of the suspicious nodes is identified by 'reputation' and 'trust mechanism'. As a result of the work, the study reports a minimum of 90% detection accuracy on fixed devices and 72% on mobile scenarios. The aim of INTI was 'prevent, detect, and isolate' the adverse impact of the attack sinkhole in the routing. In another study, Soe et al. [6] suggested a system that helps to detect multiple specific attacks successively. Each kind of specific attack is identified by utilizing a designated classifier rather than a common one. An artificial neural network (ANN) architecture was trained and used for detecting botnet attacks. The sequential detection scheme is built as a lightweight detection architecture, mainly due to its configuration, such as a single hidden layer and one output node. The experiment results indicated that the ANN architecture was sufficient to detect botnet attacks with the proposed sequential architecture. Song et al. [7] proposed a deep learning-based approach for IoT botnet detection. The researchers used 'damped incremental statistics' to mine features of the IoT devices. Then the authors applied the 'Z-Score' method to normalize the features. Afterward, the 'triangle area maps' (TAM) based 'Multivariate Correlation Analysis (MCA)' algorithm is engaged to generate the dataset. An 8-layer 'Convolutional Neural Network' (CNN) was designed to learn the dataset, and the trained CNN was then used to classify the traffic. The final experiments effectively distinguished benign traffic and different kinds of attacks. The proposed method worked with an accuracy level of 99.57%. In another research, Dutta et al. [8] presented a technique that capitalized on the Deep Neural Network (DNN), Long Short-Term Memory (LSTM), and meta classifiers such as logistic regression. The model was built on the principles of stacked generalization. The proposed method utilized a two-step methodology to process the detection of network anomalies. The first stage is called data pre-processing, in which a Deep Sparse AutoEncoder (DSAE) method is deployed for feature engineering. While in the second phase, a stacking ensemble learning method is applied for classification.

Van et al. [9] proposed to use deep learning algorithms to implement network intrusion detection systems. Two deep learning models are mentioned: the Restricted Boltzmann Machines (RBM) and Autoencoder (AE). The authors constructed a stacked RBM and a stacked AE as two Deep Belief Network (DBN) structures and compared their performance on detection intrusion. A dataset that contains four types of network attacks was used to test the ability of two DBM models. The results show the Stacked AE outperforms the Stacked RBM on the accuracy of intrusion detection; however, the training time and execution time of the stacked RBM are much longer than the Stacked AE model. Similarly, Almiani et al. [10] also proposed a deep learning-based intrusion detection system in the IoT environment. The proposed intelligent intrusion detection model contains two major components: traffic analysis and classification engines. First, the traffic analysis engine is used to pre-process the traffic data, such as symbolic-to-numeric transformation, feature reduction, and normalization. Then, the processed data is fed into the classification engine, which adopts two deep recurrent neural networks (RNN). A sample dataset was used to test the effectiveness of the proposed model. Almiani et al. [10] compare their proposed model with other baseline models for anomaly detection accuracy and execution time, which indicates the proposed RNN model has a high sensitivity to detect abnormal attacks in a competitive computational overhead.

The scalability and distribution of resources are the characteristics of the Internet of Things. Therefore, any anomaly detection model that depends on a centralized cloud will fail to handle the IoT requirements [11]. N.g. et al. [11] proposed Vector Convolutional Deep Learning (VCDL) model to detect anomalies in IoT traffic, which applies an emerging distributed intelligence approach called "fog computing". The proposed model contains three layers of components. The first layer is IoT devices which are distributed. The second layer is the fog layer—multiple work fog nodes connected to the IoT devices and train each VCDL model in a distributed manner. The master fog node in the fog layer will collect and share the best set of parameters with the worker nodes. Therefore, the traffic data will be fed into the corresponding worker node and be classified as either normal or attack. The classification result will be passed to the cloud layer, the third layer of the proposed framework. Finally, the cloud layer is used to validate the information from the entire fog layer. The experiment results indicate that the proposed distributed VCDL framework can detect anomaly traffic data with high accuracy and less detection time than the centralized detection model.

Hou et al. [12] proposed to detect anomalies in the monitoring images or videos. The proposed framework is to identify personnel and detect fire smoke in the monitoring video footage. For each detection purpose, Hou et al. [12] proposed a

different deep learning-based detection algorithm. In the experiment phase, the proposed framework demonstrated the ability to detect personnel and fire smoke with high accuracy and low false alarm rate from the monitoring images. Fatemifar et al. [13] proposed training one-class client-specific classifiers with deep CNN techniques to improve the performance of face presentation attack detection. When applied to unseen attack scenarios, this work proves to outperform two-class approaches to facial anti-spoofing. Furthermore, Duo et al. [14] developed an anomaly detection technique to detect intrusion attacks on the train communication network. The authors proposed a model based on SVM and a two-attack classification model based on random forest. The results show that the SVM-based model can detect abnormal data in train real-time Ethernet, while the random forest-based model can identify the types of attacks to allow better responses. Finally, Duessel et al. [15] proposed a payload-based anomaly detection model to detect application-specific attacks, otherwise known as zero-day attacks. In addition, a novel data representation was introduced to allow for a context-aware detection of network intrusions. The results show that the proposed model effectively increases the detection accuracy of web application attacks, such as XSS and SQL injections.

III. The Proposed Two-level Feed Forward Neural Network (2-FFNN) Model

In this paper, we propose a multi-level neural network-based technique called 2-FFNN for malicious network traffic detection in the IoT environment. The 2-FFNN is not limited by any specific device or restricted to detecting any particular type of malicious traffic attacks. The architecture and design principles of our proposed model are presented in this section.

A. The Model: An Overview

The overall flow diagram of the proposed model, 2-FFNN, is displayed in Figure 1. The 2-FFNN algorithm uses two stages to discover malicious traffic. Each stage contains a 3-layer Feed-Forward neural network (FFNN) framework, including an input layer, a hidden layer, and an output layer. The traffic data will need to pass through the two stages of the classification process to receive a binary label, either "benign" or "attack." The purpose of setting up the second level of the Feed-Forward neural network is to catch the attack traffic data which leak out from the classification of the first level. Our model applies the supervised learning technique, so this model's training data is labelled "benign" or "attack". For training the first level of FFNN, all training data are used. Then, the data with the "attack" label in the training set are removed. Only the traffic data with "benign" labels are used to train the second level of FFNN. Therefore, even though the two levels of Feed Forward Neural Network contain the same structure and use the same back-propagation technique to update the weights, the two levels of networks will learn different traffic data characteristics and receive different parameters and hyperparameters. The first classification layer is used to learn the difference between benign traffic and attack traffic, while the second classification layer focuses on fetching the characteristics of benign traffic. For example, some tricky botnet attack traffic may not follow the classical patterns of common attack, and then it may be classified as benign traffic by the first classification layer. However, when this tricky attack is passed into the second classification layer, the second FFNN will compare it with the criterion of benign data. Therefore, the proposed model has a second chance to identify and label some hard-to-detect botnet attacks. To use the trained 2-FFNN model to classify a traffic dataset should pass the following procedure. The to-be-tested traffic should be passed into the first level of FFNN first and gets an initial classification label. If the initial label is "attack", this label will be the final label for this data, which means this traffic has been identified as a botnet attack. If the initial label is "benign", the to-be-tested data will be fed into the second level of the FFNN. The classification result from the second level of FFNN will be the final label of this particular traffic data.

B. The FFNN Framework

The Feed Forward Neural Network is one of the simplest Neural Network types, also called a multi-layer perceptron. In the FFNN, the information only moves in the forward direction, which means it does not contain any cycles or feedback loops in the network [16]. The primary purpose of using Feed Forward Neural Network is to extract normal or malicious traffic data patterns to approximate a classifier [17]. The FFNN can estimate a nonlinear function because of its structure, which implies the FFNN-based classifier may be more powerful than the linear-based classifier. The FFNN we choose to use in the proposed model comprises three layers, as Figure 2 indicates. The number of neurons in the input layer is equal to the number of features in the dataset. The number of neurons in the output layer is one because the output of the FFNN is binary, which is either benign or attack. Furthermore, we set the number of neurons in the hidden layer to be 50 to let it be less than twice the input layer's size [18]. The activation function applied to both the input and hidden layers is the ReLU function, defined as Eq.1. Then, the weights in the FFNN are updated using the back-propagation techniques.

$$ReLU(x) = max(0, x) \qquad (1)$$

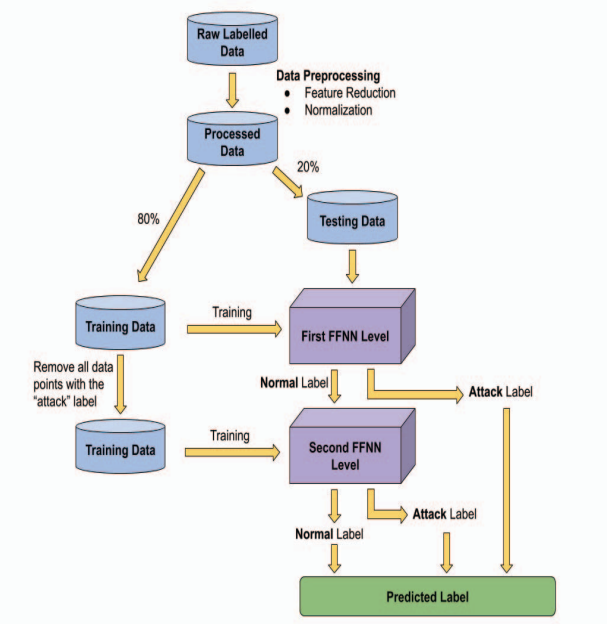

Figure 1. The overall flow diagram of the proposed model.

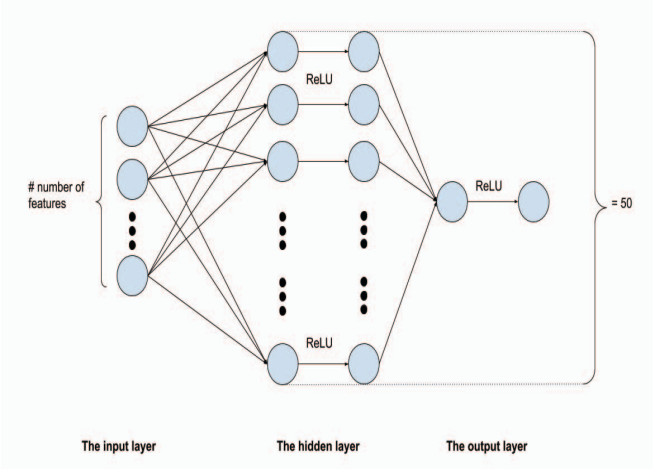

Figure 2. The Proposed FFNN architecture

For the proposed model, we implement the FFNN framework based on Keras, a widely used Python library for artificial neural networks.

IV. Experimental Analysis

In this section, we introduce the datasets used to evaluate the performance of the proposed neural network-based model. The data preprocessing and training process are also outlined. The evaluation criteria and the baseline models used for performance validation and comparison are also discussed in this section. Finally, a thorough evaluation of the experimental results and performance is provided when malicious network data is applied to the proposed 2-FFNN model.

A. Datasets

For the experiment, we used a public botnet dataset available in Kaggle [19]. They are real traffic data gathered from 9 commercial IoT devices. We used traffic data from three devices: Danmini Doorbell, Ecobee Thermostat, and Ennio Doorbell. Each device has different traffic data, and all traffic data are labelled as benign traffic or attack traffic. The summary of the three chosen datasets is presented in Table 1. The attack traffic data in the dataset are not limited to one type, including UDP flood attacks, TCP attacks, or other kinds of botnet attacks. Each traffic data contains 115 features related to four types of stream aggregation in five different time frames. The full description of the features is shown in Table 2.

Table 1. The summary of three chosen datasets for the experiment.

Dataset	Size	Benign Traffic	Network Attack
Danmini_Doorbell	27851	18906	8945
Ecobee_Thermostat	17023	11638	5390
Ennio_Doorbell	25583	17468	8115

Table 2. The description of features in the dataset [19].

Header	Description
Stream aggregation	
H	Statistics of recent traffic from this packet's host
HH	Statistics of traffic going from this packet's host to the packet's destination host
HpHp	Statistics traffic going from this packet's host and port to the packet's destination host and port.
HH_jit	Statistics of the jitter of the traffic from the packet's host to the packet's destination host.
Time Frame	
Lj	Within j minutes
Statistics	
weight	The weight of the stream
mean	The weight of the stream
radius	The root squared sum of the two streams' variances
magnitude	The root squared sum of the two streams' means
cov	An approximated covariance between two streams
pcc	An approximated covariance between two streams

B. Preprocessing

We calculated the correlation among all 115 features and drew a heatmap based on the correlation coefficient, as Figure 3 shows. Several pure white or pure black regions in the heatmap indicate several features are highly correlated or negatively correlated. Therefore, we performed a feature reduction to three datasets. For example, we found out that the timeframe values of L5, L3, and L1 are significantly positively correlated. In addition, the values in the timeframe of L0.1 and L0.01 are positively correlated, so we only keep the features of L3 and L0.1 to represent different timeframes. After the process of feature reduction, we reduced the number of features to 28. The heatmap of the selected 28 features is displayed in Figure 4. Figure 4 contains more regions with red colour and fewer regions with pure or black than Figure 3, which means the selected features are more independent. In addition to the feature reduction, we also performed a normalization on the three dataset's values by using the preprocessing function of the scikit-learn Python library.

C. Parameter Settings

We used 80% of the data as a training set for each selected dataset. The rest of the traffic data is set to be the testing data. During the training process, 10% of the training set will be used as a validation set to validate the training performance. The batch size and epochs of both levels of FFNN are set to be 1000 and 10, respectively. The learning rates for both FFNN are also the same, which is 0.1.

D. Evaluation Criteria

The proposed method's performance on identifying malicious network traffic data from benign traffic data was evaluated by accuracy and Receiver Operating Characteristic (ROC) score. In this work, there are two predicted classes: positive or negative. A positive class indicates the data is malicious, and a negative class labels the data as benign. In this context, a true positive (TP) refers to a case predicted to be malicious and confirmed to be malicious. Similarly, a true negative (TN) refers to a predicted case as benign traffic, and it is confirmed as benign by the testing dataset. A false positive (FP) occurs when data is predicted to be malicious when it is benign. A false negative (FN) occurs when data is predicted to be benign; however, it is malicious. Accuracy is a measure to identify how many objects in the testing dataset were correctly classified, and it is defined as follows:

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

While accuracy is a good measure of performance, the ROC score can identify the separability measure for a two-class classification problem. A ROC curve for a predictive model can be constructed to illustrate the trade-off between the true-positive rate (TPR) and the false-positive rate (FPR). TPR refers to the proportion of correctly classified objects or predicted as positive out of all positives. In contrast, FPR refers to the proportion of incorrectly labelled objects as positive out of all negatives. TPR and FPR have been used in various research studies related to attacks and anomaly detection [21]-[26] as:

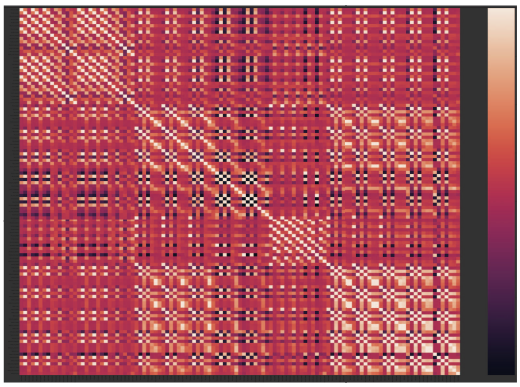

Figure 3. The heatmap of 115 features.

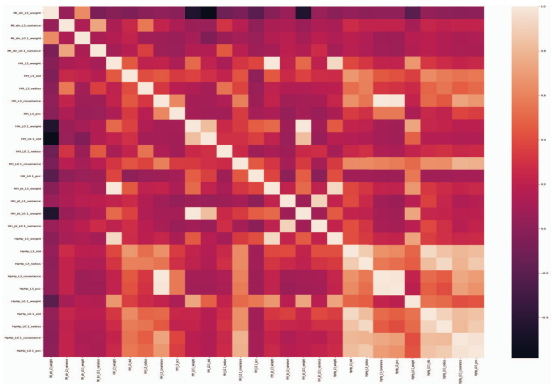

Figure 4. The heatmap of 28 selected features

$$TPR = \frac{TP}{TP+FN} \quad (3)$$

$$FPR = \frac{FP}{TN+FP} \quad (4)$$

The ROC score is computed as the area under the ROC curve and ranges from 0 to 1. A ROC score close to 1 indicates that the predictive model has a good measure of separability. It is highly capable of distinguishing between classes in a two-level classification problem.

E. The Baseline Models

To assess the performance of the proposed multi-level FFNN to detect malicious network traffic, the results were compared to a simple 3-layer FFNN. Figure 5 illustrates the structure of a typical 3-layer FFNN. The baseline model of the 3-layer FFNN consisted of an input layer, a single hidden layer, and an output layer. The hidden layer consisted of 50 nodes, where the activation function implemented at each node was a rectified linear activation function. The output layer consisted of one output node corresponding to a binary class: malicious or benign. The baseline model was trained with a batch size of 1000, referring to the number of training examples propagated through the network in each iteration. The model was trained by running ten epochs or iterations, corresponding to the number of weight updates. After each epoch was completed, the errors equivalent to the difference between the predicted output and the expected output were utilized to update the bias and weights during back-propagation. The bias value and weights were updated using a learning rate of 0.1. Once the training datasets were applied to the 3-layer FFNN, the accuracy and ROC score were determined compared to the proposed 2-FFNN model in this paper.

F. Experimental Results and Performance Analysis

To validate the performance of the proposed 2-FFNN model in malicious network traffic detection, three datasets were used to compare the performance of the proposed and baseline models: Danmini_Doorbell dataset, Ecobee_Thermostat dataset, and Ennio_Doorbell dataset. The classification performance of the models was evaluated in terms of detection accuracies and ROC scores. The classic 3-layer FFNN and 2-FFNN methods were tested with the Danmini Doorbell dataset to detect malicious network traffic. The dataset consists of

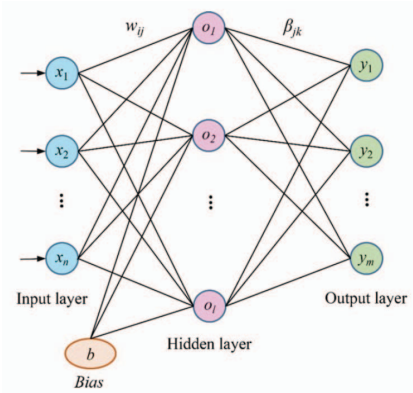

Figure 5. The Structure of a 3-layer Baseline FFNN [20].

objects that are either labeled as benign or malicious. For each method, the predicted detection labels were compared to the true labels to determine the number of TP, TN, FP, and FN, which were utilized to validate the proposed model's performance. For measuring the baseline and proposed models' performance, the detection accuracy, and ROC score were calculated. When tested on the Danmini Doorbell dataset, the baseline FFNN method had an accuracy of 82.58%. However, the proposed 2-FFNN was able to achieve a detection accuracy of 98.82%. Furthermore, the ROC score of the baseline and proposed techniques were determined to be 0.7280 and 0.9908, respectively. This indicates that the proposed 2-FFNN is more capable of distinguishing malignant data from benign data. Figure 6 illustrates the performance of attack detection of each technique when the Danmini Doorbell dataset is applied. As seen in Figure 6, the proposed model outperformed the baseline model when tested on the Danmini Doorbell dataset.

The resulting accuracy and ROC scores were calculated when the FFNN and 2-FFNN models were tested on the Ecobee Thermostat dataset. The baseline model correctly predicted 83.23% of the objects in the testing dataset, which is inadequate. Applying the 2-FFNN increased the detection accuracy of the model to 97.62%. Besides, the FFNN model yielded a ROC score of 0.7241; however, the 2-FFNN model achieved a ROC score of 0.9818. In terms of accuracy and ROC score, the performance of each technique is illustrated in Figure 7. The 2-FFNN model achieved more optimal results when tested on the Ecobee Thermostat dataset compared to the baseline FFNN model. Figure 8 outlines the calculated detection accuracy and ROC score for FFNN and 2-FFNN methods when applied to the Ennio Doorbell dataset. The detection accuracy of the FFNN model was 82.94%; however, the 2-FFNN model resulted in a detection accuracy of 97.36%. The ROC score of the FFNN and 2-FFNN models were calculated as 0.7304 and 0.9787, respectively. The results indicate that the 2-FFNN correctly detected more attacks in the dataset and better-measured separability in the two-class classification problem presented in this work than the baseline model. In addition, the 2-FFNN model was more capable of distinguishing between classes. The results validate that the proposed 2-FFNN model is favorable to the classic 3-layer FFNN model.

As stated in [14], the authors show that the particle swarm optimization-support vector machine (PSO-SVM) model improved the intrusion detection accuracy to 95.75%.

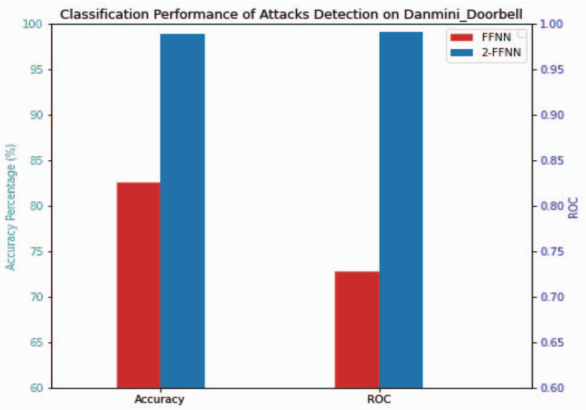

Figure 6. The Detection Performance (Danmini Doorbell)

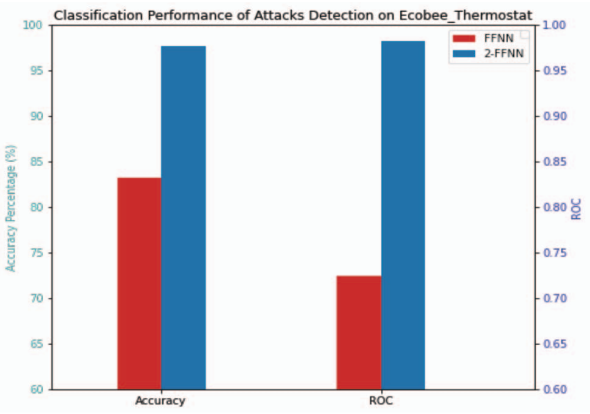

Figure 7. The Detection Performance (Ecobee Thermostat)

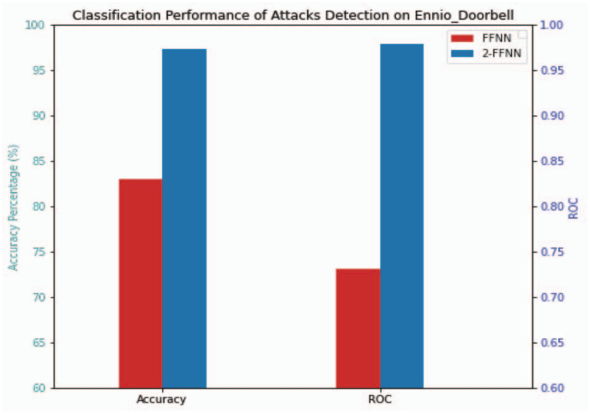

Figure 8. The Detection Performance (Ennio Doorbell)

Furthermore, the genetic algorithm-support vector machine (GA-SVM) model achieved a detection accuracy of 95.85%. Thus, we can see that our proposed model and architecture outperform some of the state-of-the-art methods.

V. Conclusion and Future Directions

Identifying malicious network traffic in the IoT environment is essential for the security and privacy of connected devices and reduces compromised data risk. After reviewing the current state-of-the-art techniques that have been developed to detect malicious botnet attacks, it was noted that most of the methods focused on either one type of device or one type of botnet attack. For this reason, this work develops a technique that can detect malicious traffic in the IoT environment with high accuracy, without restrictions by the types of devices or attacks. This paper proposes a 2-FFNN model to detect malicious IoT botnet attacks. This approach includes data preprocessing to reduce data dimensionality and complexity and computational time. The processed datasets were split into 80% training and 20% testing sets. The testing datasets were applied to two levels of FFNN, where each FFNN level exhibits a 3-layer FFNN structure. The second level FFNN was implemented to remove malicious traffic data misclassified from the first level FFNN. Therefore only benign traffic data was used to train the second level FFNN. The Danmini Doorbell, Ecobee Thermostat, and Ennio Doorbell datasets were utilized to validate the detection

performance of the proposed method. The proposed 2-FFNN model achieved the best classification performance in terms of detection accuracy and ROC score on all datasets compared to the classical 3-layer FFNN. The average detection accuracy and ROC score of the single baseline FFNN model were 82.92% and 0.7275, respectively. The proposed model achieved an average accuracy and ROC score of 97.93% and 0.9838, respectively. The results validated that utilizing one level of the Feed-Forward neural network was insufficient to identify the presence of botnet traffic attacks accurately. The second level FFNN was able to identify the majority of the misclassified benign objects from the output of the first level FFNN, as intended. The proposed 2-FFNN model proved to correctly classify data and distinguish between classes with a high accuracy level. This model can be used in time-sensitive IoT applications as a tool to detect harmful botnet attacks with high efficiency in various IoT devices. It could be useful to explore increasing the number of levels used in the FFNN framework. Future work will focus more on applying this 2-FFNN model on Edge devices in an IoT network and improving its timely responsiveness.

REFERENCES

[1] Spamhaus Malware Labs, Spamhaus Botnet Threat Report 2019,. [Online]. Available: https://www.spamhaus.org/news/article/793/spamhaus-botnet-threat-report-2019. [Accessed: 24-Nov-2020].

[2] Feily, M., Shahrestani, A., & Ramadass, S., 2009. "A Survey of Botnet and Botnet Detection." Proceedings - 2009 3rd International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009: 268–73.

[3] Soe, Y., Feng, Y., Santosa, P., Hartanto, R., & Sakurai, K., 2019. "A Sequential Scheme for Detecting Cyber Attacks in IoT Environment." Proceedings - IEEE 17th International Conference on Dependable, Autonomic and Secure Computing, IEEE 17th International Conference on Pervasive Intelligence and Computing, IEEE 5th International Conference on Cloud and Big Data Computing, 4th Cyber Science and Technology Congress, 2019 324: 238–44.

[4] Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M., 2020. "Selection of Effective Machine Learning Algorithm and Bot-IoT Attacks Traffic Identification for Internet of Things in Smart City." Future Generation Computer Systems 107: 433–42.

[5] Christian, C., Poplade, D., Nogueira, M., & Santos, A., 2015. "Detection of Sinkhole Attacks for Supporting Secure Routing on 6LoWPAN for Internet of Things." Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015: 606–11.

[6] Soe, Y., Feng, Y., Santosa, P., Hartanto, R., & Sakurai, K., 2020. "Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture." Sensors (Switzerland) 20(16): 1–15.

[7] Song, F., Liu, J., & Pannu, H., 2012. "A Hybrid Anomaly Detection Framework in Cloud Computing Using One-Class and Two-Class Support Vector Machines." In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),.

[8] Vibekananda, D., Choraś, M., Pawlicki, M., & Kozik, R., 2020. "A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection." Sensors (Switzerland) 20(16): 1–20.

[9] Van, N. T., Thinh, T. N., & Sach, L. T. (2017). An anomaly-based network intrusion detection system using deep learning. Paper presented at the 210-214. https://doi.org/10.1109/ICSSE.2017.8030867

[10] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. *Simulation Modelling Practice and Theory, 101*, 102031.

[11] N.G, B. A., & S, S. (2020). Anomaly detection framework for internet of things traffic using vector convolutional deep learning approach in fog environment. *Future Generation Computer Systems, 113*, 255-265.

[12] Hou, R., Pan, M., Zhao, Y., & Yang, Y. (2019). Image anomaly detection for IoT equipment based on deep learning. *Journal of Visual Communication and Image Representation, 64*, 102599.

[13] Fatemifar, S., Arashloo, S. R., Awais, M., & Kittler, J. (2021). Client-specific anomaly detection for face presentation attack detection. *Pattern Recognition, 112*, 107696. https://doi.org/10.1016/j.patcog.2020.107696

[14] Duo, R., Nie, X., Yang, N., Yue, C., & Wang, Y. (2021). Anomaly detection and attack classification for train real-time ethernet. *IEEE Access, 9*, 22528-22541. https://doi.org/10.1109/ACCESS.2021.3055209

[15] Duessel, P., Gehl, C., Flegel, U., Dietrich, S., & Meier, M. (2017). Detecting zero-day attacks using context-aware anomaly detection at the application-layer. *International Journal of Information Security, 16*(5), 475-490. https://doi.org/10.1007/s10207-016-0344-y

[16] Andreas, Z., (1994). Simulation Neuronaler Netze [Simulation of Neural Networks.Addison-Wesley. p. 73. ISBN 3-89319-554-8.

[17] Gupta, T., "Deep Learning: Feedforward Neural Network [Online]. Available: https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7. [Accessed: 23-Nov-2020].

[18] K. Mohammed, A. H., Jebamikyous, H., Nawara, D., & Kashef, R. (2021, April). IoT Cyber-Attack Detection: A Comparative Analysis. In *International Conference on Data Science, E-learning and Information Systems 2021* (pp. 117-123).

[19] Naveed, K., "N-BaIoT Dataset to Detect IoT Botnet Attacks," Kaggle, 18-Jan-2020. [Online]. Available: https://www.kaggle.com/mkashifn/nbaiot-dataset?select=README.md. [Accessed: 23-Nov-2020].

[20] Ebtehaj, I., Bonakdari, H., Zaji, A. H., & Sharafi, H. (2018). Sensitivity analysis of parameters affecting scour depth around bridge piers based on the non-tuned, rapid extreme learning machine method. Neural Computing & Applications, 31(12), 9145-9156.

[21] Kashef, Rasha F. "Ensemble-Based Anomaly Detetction using Cooperative Learning." *KDD 2017 Workshop on Anomaly Detection in Finance*. PMLR, 2018.

[22] Close, L., & Kashef, R. (2020). Combining Artificial Immune System and Clustering Analysis: A Stock Market Anomaly Detection Model. *Journal of Intelligent Learning Systems and Applications*, *12*(04), 83.

[23] Li, M., Kashef, R., & Ibrahim, A. (2020). Multi-Level Clustering-Based Outlier's Detection (MCOD) Using Self-Organizing Maps. *Big Data and Cognitive Computing*, *4*(4), 24.

[24] Kashef, R., Gencarelli, M., & Ibrahim, A. (2020). Classification of Outlier's Detection Methods Based on Quantitative or Semantic Learning. In *Combating Security Challenges in the Age of Big Data* (pp. 45-59). Springer, Cham.

[25] Ebrahimian, M., & Kashef, R. (2020). Detecting Shilling Attacks Using Hybrid Deep Learning Models. *Symmetry*, *12*(11), 1805.

[26] Ebrahimian, M., & Kashef, R. (2020, December). Efficient Detection of Shilling's Attacks in Collaborative Filtering Recommendation Systems Using Deep Learning Models. In 2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 460-464). IEEE.