Journal of Computing and Security

Research Article, 2021, Volume 8, Number 2 (pp. 27–42)

http://www.jcomsec.org

Intrusion Detection in IoT With Logistic Regression and Artificial Neural Network: Further Investigations on N-BaIoT Dataset Devices

Fereshteh Abbasi [a], Marjan Naderan [a,*], Seyed Enayatallah Alavi [a]

[a] *Department of Computer Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran.*

ARTICLE INFO.

Article history:
Received: 27 August 2021
Revised: 16 November 2021
Accepted: 4 December 2021
Published Online: 29 December 2021

Keywords:
Internet of Thing, Anomaly Detection, Artificial Neural Network, Logistic Regression, Botnet.

ABSTRACT

Due to the increasing development and applications of the Internet of Things (IoT), detection and prevention of intruders into the network and devices has gained much attention in the past decade. For this challenge, traditional solutions of Intrusion Detection Systems (IDS) are not responsive in IoT environments or at least may not be very efficient. In this article, we deeply investigate the previous methods of using machine learning methods for intrusion detection in IoT, and two methods for feature extraction and classification are proposed. The first method is feature extraction and classification using Logistic Regression (LR) and the second method is to use an Artificial Neural Network (ANN) for classification. To evaluate the performance of the proposed method, six devices of the N_BaIoT dataset, which consists of data samples related to nine devices IoT and several attacks are used according to some criteria for evaluating the performance of the proposed methods. Simulation results in comparison with some other deep learning methods in terms of accuracy, precision, recall and F1-score show that using logistic regression, is more efficient and above 90% classification accuracy is achieved.

1 Introduction

With the advent of the Internet of Things (IoT) paradigm, physical devices are connected to each other and to the World Wide Web in such a way that they can operate automatically. Data connection and identification must be transmitted from one device to the rest of the IoT system, whether they are computing devices or other devices. To have an accurate connection, a device must be able to declare its presence uniquely in the Internet through its IP address. Based on the variations that occur in their environment, these devices show responses completely automatically and can also exchange various data with other network devices without any human intervention. Devices available in the Internet of Things communicate to each other and to the Internet based on wireless networks and their main purposes are to collect data from different places, monitoring, remote control, etc. [1].

One of the most important security challenges in the IoT is malicious actions taken by internal or external attackers. These malicious actions, also known as attacks, try to compromise the target system, mainly

* Corresponding author.
Email addresses: f.abasi1205@gmail.com (F. Abbasi), m.naderan@scu.ac.ir (M. Naderan), se.alavi@scu.ac.ir (S. E. Alavi)

https://dx.doi.org/10.22108/JCS.2021.129807.1077
ISSN: 2322-4460

by infiltrating it. Since IoT devices exchange data with millions of other devices around the world, this type of large-scale open communication makes them particularly attractive to users with malicious intent. In 2017 alone, attacks on IoT devices increased by 600 percent [2].

In many cases, attackers do not directly target the IoT device but use it as a device to attack other devices or websites [3]. As a result, cybercrimes have become the second most reported crime in the world [4]. IoT systems seem to be easy targets for attackers, mostly because their manufacturers often place a lot of emphasis on cost, size, and usability when building such devices, while mainly security aspects are not considered. Lally and Sgandurra stated in [5], that some manufacturers hide the security vulnerabilities of their devices mainly because it ruins their market and the ideal image of their company [5][6].

IoT devices are inherently vulnerable due to insecure design and configuration. Changes in attackers' behaviors, due to the increase in their skills and data heterogeneity in the IoT, have proven that securing IoT devices poses some challenges, including identifying complex and dynamic attacks, data imbalances, data heterogeneity, real-time responses, and predictability. On the other hand, the rapid evolution of the Internet of Things has created billions of Internet-enabled devices in our daily lives, such that it is predicted that the number of connected devices will increase to 45.41 billion by 2023 [7][8]. Thus, manufacturers are more rapidly building many new devices without fundamental security and privacy checks and thus allowing attackers to easily and quickly detect and find vulnerabilities that allow them to be identified [6][9].

Therefore, the need to provide solutions to detect and prevent attacks and intrusions on IoT devices is one of the main areas of security in these networks. In particular, the devices and equipment available in IoT networks are not just computing devices such as computers and include devices such as home appliances, kitchens, doorbells, light bulbs, garden irrigation systems, alarms in buildings, etc., and intrusion and failure in any of them can lead to irreparable damage. In other words, IoT devices are prone to various physical, network, and application-layer attacks [10] that may lead to activity interruptions, privacy violations, or even physical harm. The consequences of these attacks are not limited to the users of these systems because they cause significant problems for other information systems as well. Attacked devices increase the attacking capacity of botnets, especially in denial of service attacks.

Therefore, IoT intrusion detection can be defined as including monitoring of each device and computer system and also the network traffic and analyzing activities to detect possible targeted attacks on the system [11]. For this purpose, a set of tools and mechanisms known as Intrusion Detection Systems (IDS) are used.

The scope of IDS systems usually falls into one of two categories: host-based or network-based, and upon detecting a malicious behavior an alert is created by the system. In terms of the diagnostic method, common types of intrusion detection systems are [12]:

- Signature-based diagnosis
- Anomaly-based diagnosis

These two categories are used either separately or combined to increase the accuracy of the diagnosis. The signature of a template is preset to match a known intrusion pattern. Therefore, signature-based diagnosis is defined as the "process of comparing signatures against observed events to identify potential intrusions" [13]. However, this method is not sufficient to detect unknown intrusions, as their patterns are unfamiliar. In addition, keeping the IDS knowledge database up-to-date is another challenge, as it is a time-consuming and difficult process. In contrast, anomaly-based diagnosis is defined as the "process of comparing normal activities to the observed events to identify significant deviations" [12].

Anomaly-based detection consists of three general modules:

1. Parameterizing: This represents the behavior observed in the profile, which consists of various features that must be considered, such as network connections, hosts, and applications.
2. Training: This processes the parametric profiles to build a classification model that distinguishes between normal and anomaly behaviors.
3. Detection: This uses a built-in classification model to detect new traffic anomalies.

Among the most important solutions of anomaly-based and botnet detection methods are machine learning methods [14] such as SVM, logistic regression, decision trees, etc. [13], [15], [16]. In recent years, there has been a great deal of interest in the use of deep machine learning methods, and some effective methods in detecting intrusion in the Internet of Things have been performed using deep learning methods. We have investigated some of these studies in the next section.

It is shown in [17], the use of deep learning methods mostly leads to more accurate models. In contrast, deep learning techniques require higher computational costs and the results are not easily interpretable compared to other shallow learning cases. However, to overcome the problems of scalability and computational resources in the IoT environment, it is essential

to minimize the features required for classification [18].

An important issue in the previous works is the absence of specific dataset(s) related to IoT environments. Many previous studies can be found in the field of intrusion detection on many various datasets, including KDD99, NSL_KDD, CICIDS2017, CICIDS2018, etc. On the other hand, datasets for IoT devices have not been studied compared to other datasets and there is a gap in this field. By investigating previous works, a novel dataset, namely N-BaIoT [19] is found which is specifically related to IoT devices and it is a good choice for our study. Because this data set contains many samples and a large number of attacks are generated, deeper studies in this field are required.

Another important issue in anomaly detection in IoT is that since devices in an IoT network are so diverse, different features can be used to detect intrusions. Features that are important to one device in detecting intrusion may not have the same level of importance for another device. On the other hand, features that are important in detecting one class of attack may not be very successful in detecting another class of attack. These differences led us to first identify the characteristics of each class (normal or attack) for a specific device and then classify the samples based on these characteristics. We anticipate that classifications based on the characteristics of each class are more accurate than if the properties were not specified in advance. A similar study has been performed in [13], but for the cloud processing environment and on the NSL_KDD data set. Here we intend to apply this method to the N_BaIoT dataset.

In this paper, two methods are used to detect intrusion in IoT. In the first method, the logistic regression method is first used to select more effective features, which has not been conducted in previous IoT-related work. Features are weighted using this linear and low-cost algorithm. Then, using the given weights, ineffective features are removed and more effective features are maintained. In the second method, a neural network is used for classification. To evaluate the proposed method, the typical criteria for evaluating intrusion detection methods are used and the proposed method is compared with some other deep learning methods in [17] and [20].

The rest of this paper is organized as follows: in Section 2 the related works are presented in brief. In Section 3 the proposed method is explained in detail. In Section 4, evaluation criteria and simulation results are presented, and finally in Section 5 conclusion and directions for future research are given.

2 Related Work

In this section, we investigate some of the previous works related to anomaly detection in IoT environments. It is worth mentioning that many studies exist for intrusion detection systems but few of them are related to IoT environments. Specifically, intrusion and anomaly detection in IoT differs from that of custom ones due to the diversity in devices, protocols, and standards in IoT environments. We start with works with traditional machine learning methods on older datasets (KDD 99 and NSL-KDD) and finally reach newer studies with deep learning algorithms conducted on IoT datasets.

Intrusion detection on KDD 99 and NSL-KDD datasets has been well investigated in previous work, and researchers have proposed single, hybrid, and ensemble classifiers increase accuracy [21]. Javaid et al in [22] introduced a deep learning approach with two-stage classification, which involves a good feature representation learning of unlabeled data and applying it to labeled data for classification. The authors used sparse auto-encoders to learn unsupervised features and softmax regression to classify, using the NSL-KDD dataset of five classes with a test accuracy of 79.1%.

Tang et al. in [23] presented a deep neural network model with three hidden layers on the NSL-KDD database. Using only six basic features, their model reached 75.75% accuracy. In [24], Zhang and Zulkernine proposed a random forest-based anomaly detection model. Their combined framework is a combination of misuse and anomaly detection on the KDD 99 database. They converted the attacks into two classes and achieved 94.7% detection accuracy.

In [25], the Naive Bayes classifier was used to solve the intrusion detection problem on the KDD 99 dataset and focused on two cases: four-class classification and two attack classes. The authors also implemented a decision tree classifier, achieving an accuracy of 91.28% and 91.47% for the decision tree and Naive Bayes for the four classes, respectively. For the two-class classification, the decision trees had a classification accuracy of 93.02 while the Naive Bayes method had an accuracy of 91.45.

Mukkamala in [26] used SVM and multi-layer Neural Networks (NN) with feed-forward network and with four and three layers, for the intrusion detection problem. The models were evaluated on KDD 99 data set and both SVM and NN reached high accuracies (almost 100%) for binary classification. However, the evaluation showed a significant difference in training times between NN and SVM.

Deep learning methods have been also used for intrusion detection during the previous years which

some of which are described in the following. In [27] the researchers examined the Auto-Encoder (AE) method with the softmax classifier on the KDD99 dataset and showed that the AE method had an accuracy of 94.71% in detection. In [28] the authors used the Stacked NSAE method and the Random Forest classifier to classify the KDD99 and NSL_KDD datasets. The results on the KDD99 dataset were 97.85% accurate and on the NSL_KDD dataset were 85.42% accurate.

In [29], researchers examined the LSTM method on the NSL_KDD dataset, which has an accuracy of 97.5%. In [30], the researchers examined the DBN method with the Softmax classifier on the KDD9910% dataset, which has an accuracy of 97.9%.

Trying to use datasets specific for IoT, some works can be found on these datasets. In [17], the researchers used the deep auto-encoder network on the N7_BaIoT dataset, which is an IoT-specific data set. They showed with the AE method they reached TPR = 100% and FPR = 0.007 which is the lowest number of false alarms and also the execution time of the algorithm was lower than other compared methods (SVM, Isolation Forest, LOF).

In [20], the authors examined several deep neural networks, including CNN, RNN, and LSTM methods. These methods were tested on the N_BaIoT dataset. Their experimental results show that the CNN method has an accuracy of 91%, the RNN method has an accuracy of 41%, and the LSTM method has an accuracy of 62%.

In [18], researchers investigated the intrusion detection problem using decision tree and kNN methods on various features of the N-BaIoT dataset. The accuracy obtained using the decision tree method on the number of features 2, 3, and 10 is approximately 98% and the K-NN method on the same number of features is 98, 97.24, and 94.97%, respectively.

In [31], the researchers used SVM and Isolation Forest methods to investigate the detection of anomalies on different models of N-BaIoT datasets. The authors have used the Fisher score for feature selection after some normalization of the data and showed that this measure is efficient for selecting the more important features compared to other methods in their simulation results. The accuracy obtained by the SVM method for Baby monitor Philips (B120N/10) device was 37.95% and by the Isolation Forest method for Doorbell (Damini), Thermastat (Ecobee), and four security cameras (Prevision PT-737E, Prevision PT-838, SimpleHome XCS7-1002-WHT and SimpleHome XCS7-1003-WHT) was 95%, 94.94, 66.88, 92.34, 12.95 and 65.95%, respectively.

It can be seen from the previous methods that

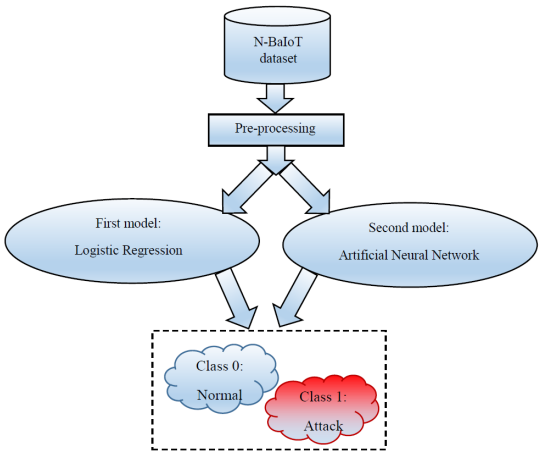


Figure 1. Sample Map of the Ground Floor of the Faculty.

since the N-BaIoT dataset is the newest and the only database on IoT tools, only references [17], [20], and [31] have used this dataset. Furthermore, this dataset contains data from nine devices, and samples for each device are separate from other ones. Therefore, deeper investigations for each device and each class of attack are required for this dataset. We have considered this dataset and samples related to six devices in this dataset in this study. From the viewpoint of feature selection, the method in [31] has used the Fisher score while the methods in [17] and [20] have used the deep neural network for both feature selection and classification. Thus, in this study, we have investigated a method for feature selection and a neural network for classification to obtain comparable results.

3 Proposed Method

As mentioned in the previous sections, we intend to use two different methods for feature extraction and classification of normal and attack samples on an IoT dataset. To this end, the first method uses logistic regression for feature selection and classification, which is a linear algorithm and does not have much computational load. The logistic regression method is used to select a subset of features that are more suitable for identifying each class, and then the classification is performed using the sigmoid function. The second method used for classification is a neural network that consists of five layers and the Relu activator function and in the last layer, the sigmoid function is used for classification. Figure 1 shows the flowchart of the proposed method. In the following, we describe each of the steps in more detail.

3.1 The Dataset

In this study, the N-BaIoT database with 115 different features has been used [19],[32]. Comparing the num-

ber of features of this data set with previous datasets such as NSL-KDD and CICIDS, it can be seen that the number of features is much higher and therefore the power of the feature selection method will be more evident. This database is also provided in 2018 and has been used in a few previous works. This data set generally consists of samples from nine devices (Table 1) and two types of attack and normal data. There is also data on the types of attacks that are not currently used in this phase of the research, and if the feature selection method is effective, they will be used for the next step, which is the classification of attacks. In this dataset for the following features:

- Packet size (outbound/both inbound and outbound)
- Packet count
- Amount of time between packet arrivals
- Packet jitters

and for each value, one or more statistics are calculated which include mean, variance, integer, radius, covariance, and correlation coefficient. The fact that which statistics are used for each attribute are shown in Table 2, and a total of 23 attributes are created from this table. Then for each of these 23 features, five-time windows (100 milliseconds, 500 milliseconds, 1.5 seconds, 10 seconds, and 1 minute) are considered and a total of 115 features are extracted. All 115 features have been used in this study similar to [20].

The dataset is generated by injecting different attacks using Bashlite and Mirai botnets. Bashlite is used to infect Linux-based IoT devices for DDoS attacks. Mirai, which is used to carry out large-scale attacks using IoT devices, was also discovered in August 2016 and is now available as open-source [33]. Since 2016, botnets have evolved significantly and are more dangerous [34],[35]. Table 3 shows 10 specific types of Bashlite and Mirai attacks as also used in [20]. In this study, samples related to six devices: the Doorbell (Danmini), Thermostat (Ecobee), Baby Monitor (PhilipB120N/10), Security Camera (PT-737E), Security Camera (PT-838), and Security Camera (XCS7-1002) data were used.

3.2 Feature Selection and Classification Based on Logistic Regression

The first method used to select features and categories in this article is the logistic regression method, which lies under the category of supervised learning methods [13],[36]. Feature weighting is also used to remove or select a feature and, the sigmoid function in (1) is used for classification, which ensures that the output is in the range [0-1].

Table 1. All Nine Deviced Used in the N-Baiot Dataset.

Device Type	Device Model Name
Doorbell	Danmini Ennio
Thermostat	Ecobee
Baby monitor	Philips B120N/10
Security camera	Prevision PT-737E Prevision PT-838 SimpleHome XCS7-1002-WHT SimpleHome XCS7-1003-WHT
Webcam	Samsung SNH 1011 N

$$h_\theta(x) = g\left(\theta^T x\right) = \frac{1}{1+e^{-\theta^T x}} \tag{1}$$

The input of this function, according to the 115 features of the dataset, are as:

$$\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_{115} x_{115} \tag{2}$$

$$\text{Cost}\left(h_\theta(x), y\right) = \left\{ \begin{array}{ll} -\log\left(h_\theta(x)\right) & \text{if } y = 1 \\ -\log\left(1 - h_\theta(x)\right) & \text{otherwise} \end{array} \right\} \tag{3}$$

in which y is the class label which is either 0 or 1 for binary classification. The cost function in (3) can be re-written as:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}\left(h_\theta\left(x^i\right), y^i\right) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^i \log h_\theta\left(x^i\right) + \left(1-y^i\right)\log\left(1-h_\theta\left(x^i\right)\right)\right] \tag{4}$$

such that the input to this function is the unknown parameters and m is the number of samples. To calculate the unknown parameters of each class, the label of that class is set to 1, and labels of other classes are set to 0. The values of unknown parameters must be calculated such that the cost function in (4) is minimized. This cost function is convex and differentiable. According to the convexity of the objective function,

Table 2. Details of Features Calculated in the N-Baiot Dataset [32].

Aggregated by	Value	Statistic	Total No. of Features
Source IP	Packet size (only outbound)	Mean, variance	3
	Packet count	Integer	
Source MAC-IP	Packet size (only outbound)	Mean, variance	3
	Packet count	Integer	
Channel	Packet size (only outbound)	Mean, variance	10
	Packet count	Integer	
	Amount of time between packet arrivals	Mean, variance, integer	
	Packet size (both inbound and outbound)	Magnitude, radius, covariance, correlation coefficient	
Socket	Packet size (only outbound)	Mean, variance	7
	Packet count	Integer	
	Packet size (both inbound and outbound)	Magnitude, radius, covariance, correlation coefficient	

Table 3. Botent and Attack Tyoes Used in This Study.

Botnet	Attack	Explanation
Bashlite	Scan	Scans the network for vulnerable devices
	Junk	Sending spam data
	UDP	UDP flooding
	TCP	TCP flooding
	COMBO	Sends spam data and open connection of IP, port
Mirai	Scan	Automatic scanning for vulnerable devices
	Ack	ACK flooding
	Syn	SYN flooding
	UDP	UDP flooding
	Plain	Less of an option of UDP flooding for higher
	UDP	packet per second

the gradient descent method can be used, therefore:

$$\theta_j = \theta_j + \alpha \left(y^i - h_\theta \left(x^i\right)\right) x_\theta(i) \qquad (5)$$

After classifying the data set by logistic regression algorithm, 115 values or coefficients for Θs are obtained for 115 features in the N-BaIoT dataset. The larger the value of a coefficient, the more important the corresponding feature with that coefficient. Next, to achieve the most important properties for the normal class, we do the following procedure: first coefficient values which represent the weights of features are sorted in descending order (due to the lack of space we have not presented in this table). Next, larger coefficient values are added to the ROC graph sooner. This means first the largest coefficient is selected and values of TP and FP are calculated. Next, the second-largest coefficient is added and TP and FP are again

calculated based on these two features which have these two coefficients. This process of adding features and calculating the TP-FP values is repeated until the ROC value reaches nearly the value of 1 and adding another feature does not increase the value of ROC greatly. This indicates that no longer adding features has many effects on the accuracy of the classification. This result is presented in Section 4, the simulation results, and according to that results, 19 features that are more important to the normal class in the N-BaIoT dataset are shown in Table 4. As the final note for this section, it is worth mentioning that in our experiments these 19 features were the same for all six devices, which was interesting. This means that for all the six devices selected, the same 19 features of Table 4 were selected by the logistic regression method, despite their exact values being different. For more certainty, this experiment was repeated three times for each device, and the same 19 features were selected by the logistic regression method.

3.3 Classification Using Artificial Neural Network

The second method used for classification in this paper is an artificial neural network that consists of three parts: input, output, and processing. Each part contains one or more layers, and each layer contains a group of nerve cells (neurons) that are generally associated with all neurons in other layers unless the user restricts communication between neurons, but the neurons in each layer have no connection with other neurons in the same layer. A neuron is the smallest unit of information processing that forms the basis of the function of neural networks. A neural network is a collection of neurons that, being located in different layers, form a special architecture based on the connections between neurons in different layers. Neurons can be a nonlinear mathematical function, therefore, a neural network made up of a community of these neurons can also be a complete complex, nonlinear system. In the neural network, each neuron operates independently, and the overall behavior of the network is the result of the behavior of multiple neurons. In other words, neurons correct each other in a cooperative process.

In this study, a 5-layer neural network consisting of three hidden layers is investigated. The number of neurons in the three hidden layers was tested for several values and finally, these values were reached: 10, 40, 10. The Relu activator function is used in the hidden layers and the sigmoid function is used in the last layer. Experiments were also performed with two and four hidden layers, the results of which are given in Section 3, but the best results were obtained with three hidden layers and the number of neurons mentioned. The advantage of using the Relu function is that it has a less computational cost and the weights are updated better, which results in faster network training. This function, in (6), maps inputs smaller than zero to zero and inputs larger than zero to themselves [37].

$$R(z) = \left\{ \begin{array}{ll} z & z > 0 \\ 0 & z <= 0 \end{array} \right\} \tag{6}$$

In the next section, we present evaluation metrics and simulation results of the experiments conducted.

4 Evaluation and Simulation Results

To simulate the proposed method, Python software and Jupiter Notebook [38], which is one of the most important environments for Python development, have been used. A preprocessing step is also performed on the dataset samples so that the values in the N-BaIoT dataset are normalized according to (7) such the final values are between 0 and 1.

$$\text{Norm\_value} = \frac{realvalue - minvaluedataset}{maxvaluedataset - minvaluedataset} \tag{7}$$

Next, the samples of this study are divided into two groups of the train (70% of data) and the test (30% of data), and the data of the first group are used in the learning process. In addition, we have conducted validation experiments in which we divide the dataset into three subsets: 80% for train, 10% for the test, and 10% for validation. The results of this step are also presented in the last section.

4.1 Evaluation Metrics

To evaluate the effectiveness of the proposed methods, several machine learning criteria are used including accuracy, the ROC Curve, True Positive Rate (TPR), False Positive Rate (FPR), specificity, recall, and F1-score. These criteria are based on the following four basic values:

- True Positive (TP): The number of normal samples that have been correctly classified as normal.
- True Negative (TN): The number of attack samples that have been correctly classified as an attack.
- False Positive (FP): The number of attack samples that have been classified as normal.
- False Negative (FN): The number of normal samples that have been detected as an attack.

Using these basic criteria, the criteria used are:

Table 4. Number of Important Features Selected by the Logistic Regression Method for the N-Baiot Dataset.

Feature name	Number of features
MI-dir-L5-weight, MI-dir-L5-mean, MI-dir-L5-variance, MI-dir-L3-weight, MI-dir-L3-mean, MI-dir-L3-variance, MI-dir-L1-weight, MI-dir-L1-mean, MI-dir-L1-variance, MI-dir-L0.1-weight, HpHp-L0.1-pcc, HpHp-L0.1-covariance, HpHp-L0.01-weight, HpHp-L0.01-mean, HpHp-L0.01-std, HpHp-L0.01-magnitude, HpHp-L0.01-radius, HpHp-L0.01-covariance, HpHp-L0.01-pcc	19

- Accuracy: which refers to the percentage of correct classification over the whole classification on the test set, as in (8).

$$\text{Accuracy} = \frac{TP}{TP + FP + TN + FN} \tag{8}$$

- ROC Curve: which is used to differentiate data in given classes (such as normal and attack). The goal is to determine the division point for the classifier that achieves the maximum number of true positives and the lowest number of false positives.
- Specificity (precision): This means the ratio of the number of correct samples classified by the classifier to the total number of samples (which the classifier has either correctly or incorrectly classified as normal) and is calculated by (9).

$$Specificity(precision) = \frac{TP}{TP + FP} \tag{9}$$

- Recall (sensitivity): which is also called the true negative response rate and calculated as in (10).

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

- F1-score: which is the combination of specificity and recall (sensitivity) and has a better measure of mistakenly classified samples compared to the accuracy measure. It is calculated as in (11).

$$\text{F1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{11}$$

In addition, we have also reported the confusion matrices for different devices. The confusion matrix shows the results of classification based on real available information and it shows the performance of the classification. It is mostly used for supervised learning algorithms and each column of the matrix shows the predicted value while each row represents the real (true) value.

4.2 Performance Evaluation of the Logistic Regression Method

To evaluate the performance of the logistic regression method, first, the diagram of the number of features is presented in terms of the value of ROC and the confusion matrix. As mentioned in Section 3.2, features with larger coefficient values are added to the ROC chart sooner. These diagrams are shown in Figure 2 for each of the six devices separately. In the ROC diagram, the horizontal axis represents the FP rate and the vertical axis represents the TP rate. According to this diagram, the least number of features that have an acceptable ROC value is selected, which according to Table 4, are 19 features (for all six devices). Figure 3 shows the confusion matrices for each of the six devices, which is the result of the classification based on the total classification available (TP, TN, FP, and FN). Each column of this matrix represents a sample of the predicted value and each row represents a correctly classified sample.

4.3 Performance Evaluation of the Neural Network Method

To evaluate the performance of the proposed neural network we used two metrics: accuracy and loss function [32]. In Figure 4 the horizontal axis represents the number of epochs and the vertical axis represents the

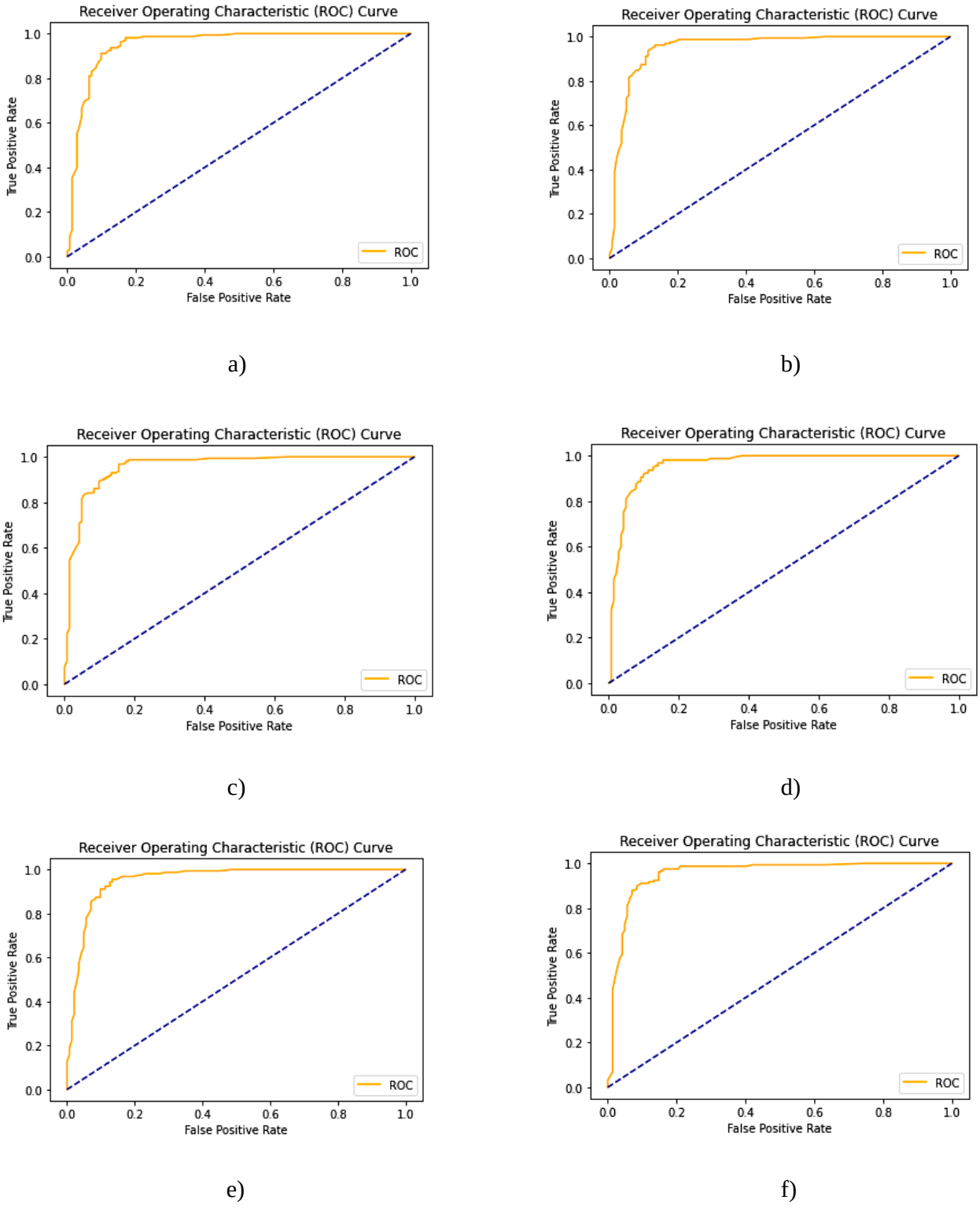
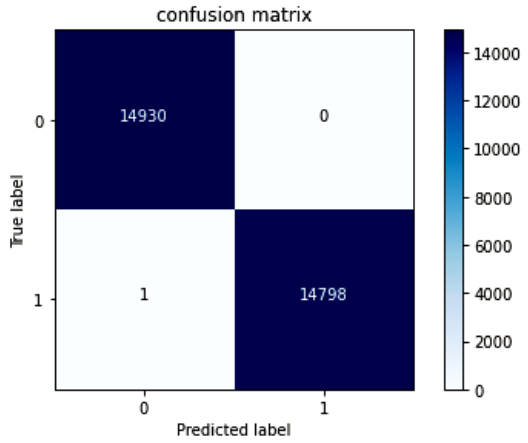
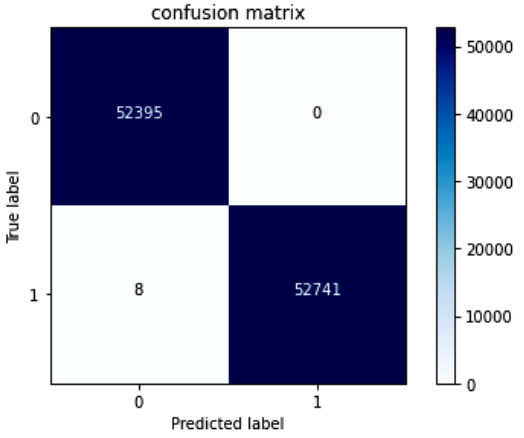


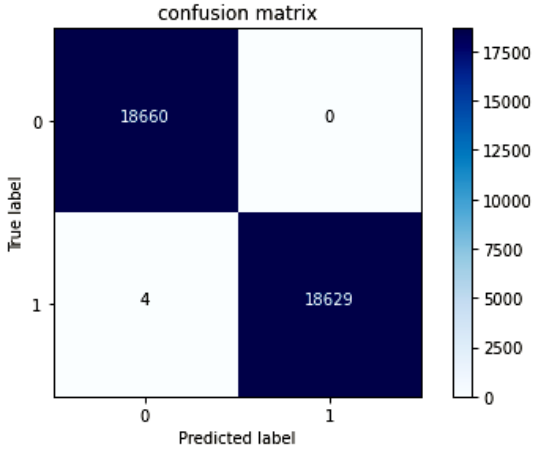
Figure 2. The ROC diagrams according to the most important features selected by Logistic Regression for a) the Doorbell (Danmini) device, b) the Thermostat (Ecobee), c) Baby Monitor (PhilipB120N/10), d) Security Camera (PT-737E), e) Security Camera (PT-838) and f) Security Camera (XCS7-1002) of the N-BaIoT dataset.



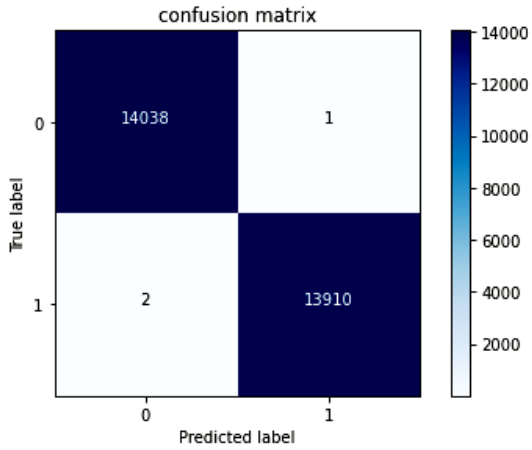
a)



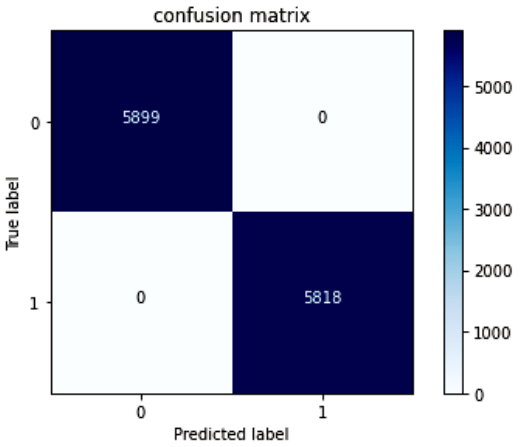
b)



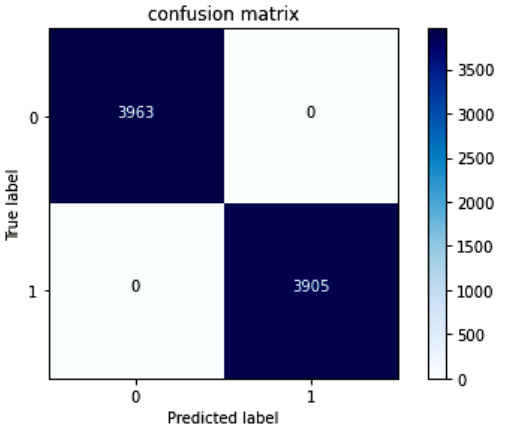
c)



d)



e)



f)

Figure 3. Confusion matrices according to the most important features selected by Logistic Regression for a) the Doorbell (Danmini) device, b) the Thermostat (Ecobee), c) Baby Monitor (PhilipB120N/10), d) Security Camera (PT-737E), e) Security Camera (PT-838) and f) Security Camera (XCS7-1002) of the N-BaIoT dataset.

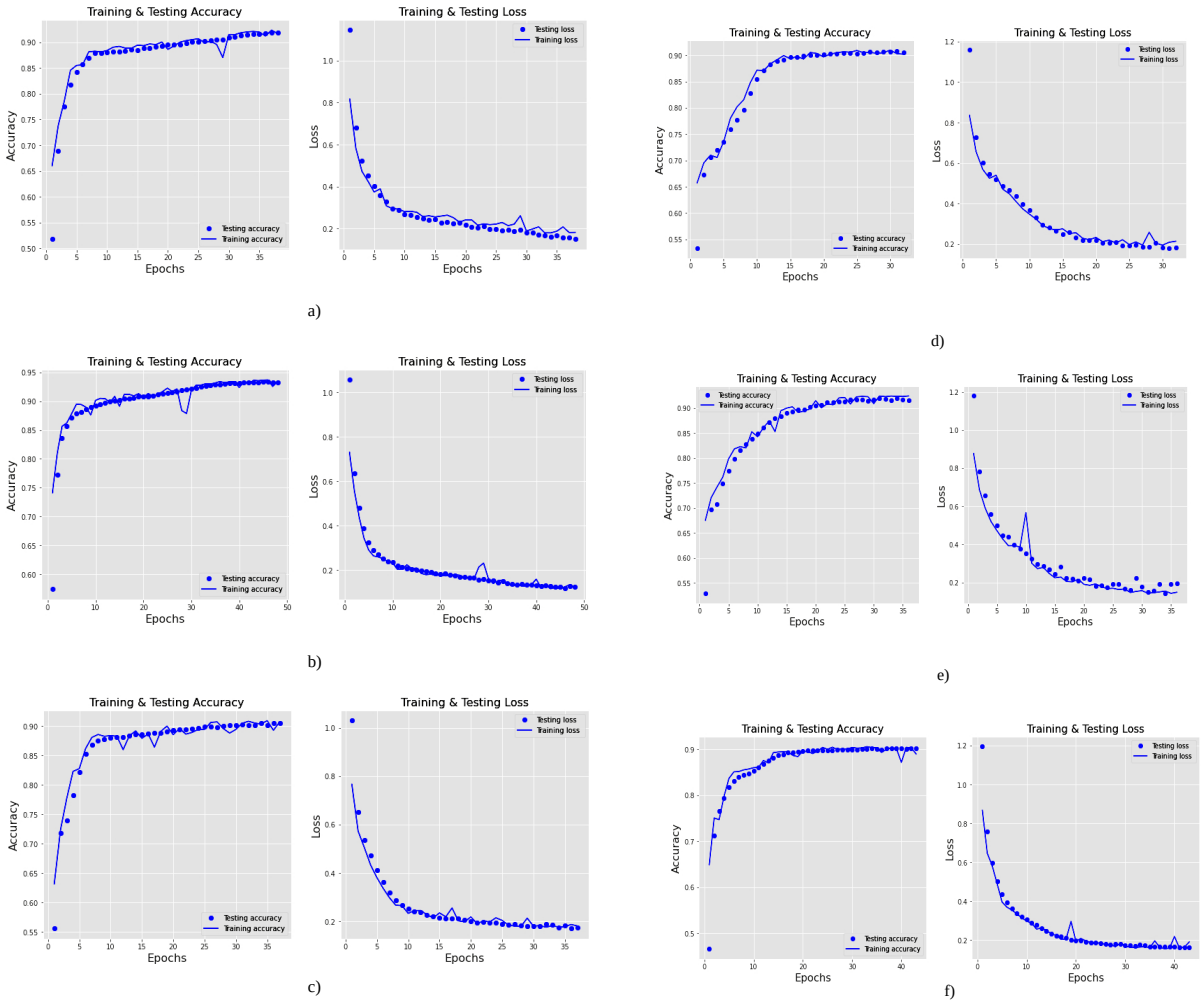


Figure 4. Diagrams of accuracy and loss function for the proposed neural network with three hidden layers for a) the Doorbell (Danmini) device, b) the Thermostat (Ecobee), c) Baby Monitor (PhilipB120N/10), d) Security Camera (PT-737E), e) Security Camera (PT-838) and f) Security Camera (XCS7-1002) of the N-BaIoT dataset.

accuracy and loss function for each of the six devices. In these diagrams, the continuous line is related to the training subset and the circled line is related to the test subset. From Figure 4 it can be deduced that the proposed model converges and no signs of over/under-fitting are seen for none of the devices, despite some oscillations that can be seen in the diagrams. In addition, it can be seen that the more the number of epochs, the better the model is trained since the loss function is reduced and accuracy is increased.

The results of Figure 4 are conducted on the proposed neural network with three hidden layers. Since the number of layers inside a neural network depends on the relationship patterns among the data, and the simpler the relation patterns, the fewer the number of layers in the neural network. On the other hand, more complex relationship patterns among data need more layers for better classification. According to our simulations, since a few layers suffices the desired accuracy, we deduce that the relationship patterns among the data samples are simple and can be classified well.

We have also tested the network with two and four hidden layers for one of the devices (the doorbell) to show that the best results were deduced with three layers. Figure 5 shows these results for the neural network with two and four hidden layers. As can be seen in the diagrams, the model is over-fitted. Over-fitting indicates that the model is well-trained but not well-generalized. This can happen when the data set is too small or when it is too large and complex or contains noisy data. That is why it is said that the machine can not predict the new test samples correctly. The concept of under-fitting occurs when the model is too simple and not suitable for learning.

In addition, we have provided the cross-validation results performed on three subsets of the dataset by dividing them into 80%-10%-10% for train, test, and

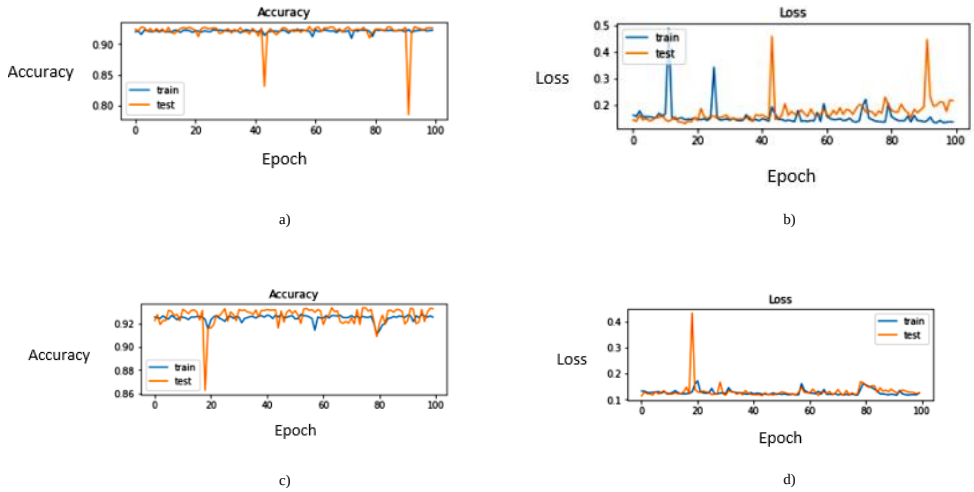


Figure 5. Diagrams of a) accuracy and b) loss function for four hidden layers and c) accuracy and d) loss function for two hidden layers of the proposed neural network.

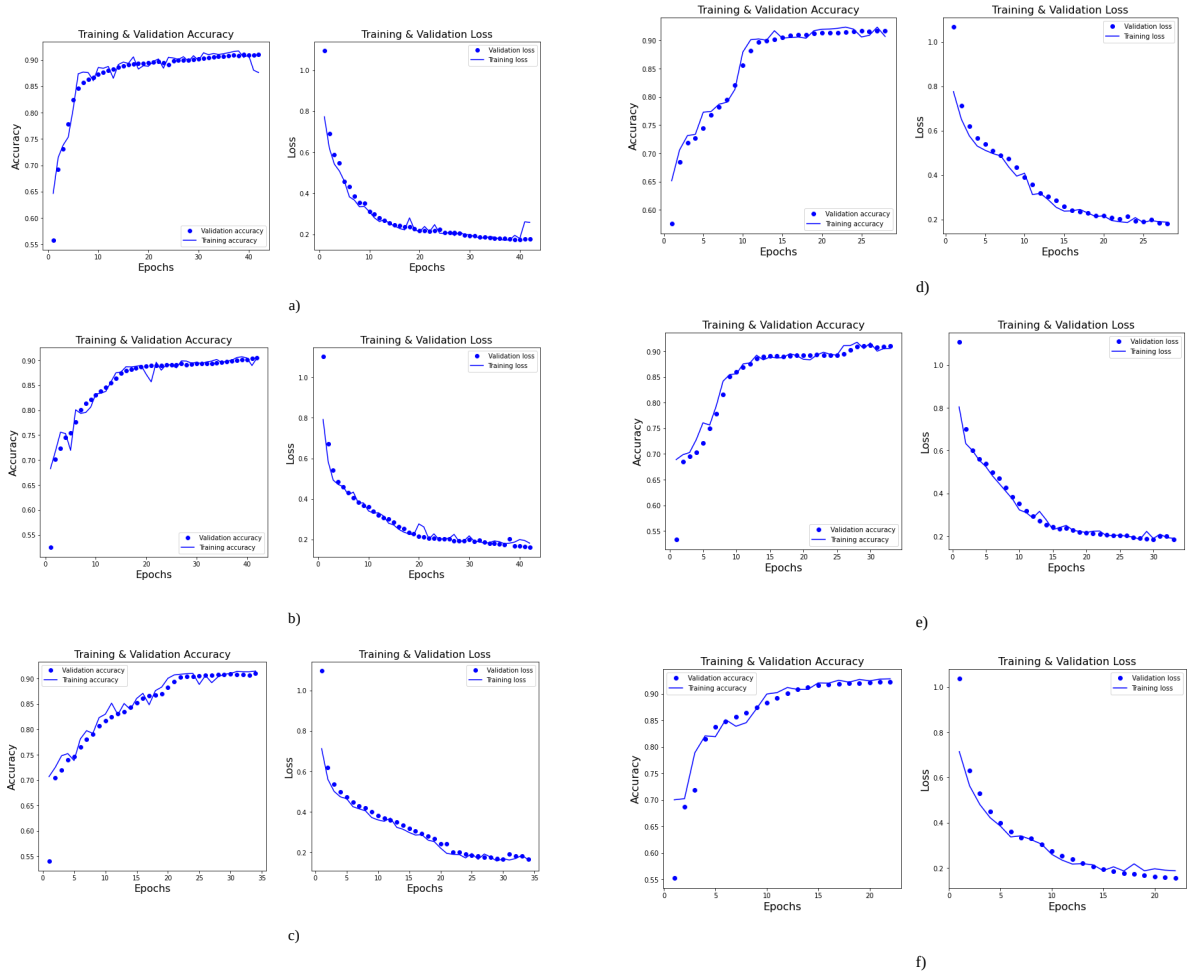


Figure 6. Diagrams of accuracy and loss function for the validation and test results for the proposed neural network with three hidden layers for a) the Doorbell (Danmini) device, b) the Thermostat (Ecobee), c) Baby Monitor (PhilipB120N/10), d) Security Camera (PT-737E), e) Security Camera (PT-838) and f) Security Camera (XCS7-1002) of the N-BaIoT dataset.

Table 5. Results of the Logistic Method and the Proposed Ann According to Some Measures.

	Precision		Recall		F1-score		AUC
	Logistic regression	Proposed ANN	Logistic regression	Proposed ANN	Logistic regression	Proposed ANN	Logistic regression
Doorbell (Danmini)	1	0.9358	0.9999	0.9313	0.9999	0.9335	0.96
Thermostat (Ecobee)	1	0.9411	0.9998	0.9402	0.9998	0.9406	0.95
Baby Monitor (PhilipB120N/10)	1	0.9099	0.9997	0.9090	0.9998	0.9093	0.95
Security Camera (PT-737E)	0.9999	0.9091	0.9998	0.9089	0.9998	0.9089	0.94
Security Camera (PT-838)	1	0.9195	1	0.9180	1	0.9187	0.96
Security Camera (XCS7-1002)	1	0.9000	1	0.8989	1	0.8994	0.95

Table 6. Comparison of the Logistic Regression Method and the Proposed Ann With Some Other Methods.

	Accuracy					Precision			Recall			F1-score				
	Proposed ANN	Logistic regression	SVM [31]	Isolation Forest [31]	Decision Tree [18]	Auto-encoder [17]	Proposed ANN	Logistic regression	Auto-encoder [17]	Proposed ANN	Logistic regression	CNN [20]	RNN [20]	LSTM [20]	Proposed ANN	Logistic regression
Doorbell (Danmini)	0.9298	0.9999	-	0.9500	0.9896	0.9930	0.9358	1	0.99993	0.9313	0.9999	%91	%41	%62	%95.13	%99.92
Thermostat (Ecobee)	0.9438	0.9999	-	0.9494	0.9929	-	0.9411	1	-	0.9402	0.9998	-	-	-	-	-
Baby Monitor (PhilipB120N/10)	0.9160	0.9998	0.8618	-	0.9994	-	0.9099	1	-	0.9090	0.9997	-	-	-	-	-
Security Camera (PT-737E)	0.9112	0.9998	-	0.8866	-	-	0.9091	0.9999	-	0.9089	0.9998	-	-	-	-	-
Security Camera (PT-838)	0.9283	1	-	0.9234	0.9930	-	0.9195	1	-	0.9180	1	-	-	-	-	-
Security Camera (XCS7-1002)	0.9089	1	-	0.9512	-	-	0.9000	1	-	0.8989	1	-	-	-	-	-

validation, respectively. The resulting diagrams for the six devices are presented in Figure 6 and as it is shown the validation values are very close to the test results which confirms the accuracy values for the test samples.

Table 5 represents the results of several measures of the proposed methods (logistic regression and neural network) for each of the six devices and with two measures of the study in [17]. These results are the average of 10 execution for each measure.

Finally, Table 6 compares the results of the two proposed methods with that of some other methods. As seen from these last two tables, the proposed method has an accuracy of more than 95% which is promising, especially the logistic regression method which has a better feature selection method reaches 99% for accuracy. The logistic regression technique has good accuracies for simple datasets and we also showed by the neural network, that the relationship patterns among the data are simple. In addition, since the logistic regression method needs the labels for each data sample and the N-BaIoT is a labeled dataset, we expected that this method will reach acceptable results which are confirmed by these simulation results.

Furthermore, the proposed ANN for the Thermostat (Ecobee) device has the best accuracy with 94%, and the logistic regression method for two security

cameras PT-838 and XCS7-1002 devices have the best accuracies with 100%. The decision tree method in [18] also has 99% accuracies for the Thermostat (Ecobee), Baby Monitor (PhilipB120N/10), and Security Camera (PT-838) devices which are also good results.

These results are much better than that of the three other deep learning networks in [20] and comparable with the AE method in [17].

5 Conclusions and Future Work

In this paper, the problem of IoT intrusion detection was addressed and for this purpose, some datasets were investigated. Among them, the N-BaIoT database was identified and selected as a novel dataset dedicated to IoT devices. The proposed method for feature extraction and classification is the logistic regression and a neural network for just classification, which the logistic regression method has not been observed in previous works in this field. The simulation results on the data of six of the devices in the N-BaIoT dataset show that the feature selection method based on logistic regression leads to better accuracy criteria compared to other methods. Furthermore, the results of the neural network simulation for three different cases for the number of hidden layers, which were 2, 3, and 4 layers, show that the best results are obtained for 3 hidden layers and do not have over/under-fitting issues.

As for future work, it is recommended to use all nine devices in the N-BaIoT database and to classify the attack samples according to the 10 classes of attack. This can be achieved through deep neural networks and especially the GAN neural network, which has not been used in this field so far. Furthermore, using other feature selection methods: i) feature selections by the deep neural networks, and ii) separate feature selection methods (scalar methods) are also advised for more investigations. Finally, exploring other similar datasets like Bot-IoT and CICIDS is also suggested for future research.

References

[1] D. Mendez Mena, I. Papapanagiotou, and B. Yang. Internet of things: Survey on security. *Information Security Journal: A Global Perspective*, 27(3):162–182, 2018. doi:10.1080/19393555.2018.1458258.

[2] Broadcom Inc. Symantec. internet security threat report (istr). `https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf`, Date Accessed: 8 October 2020.

[3] S. Alabdulsalam, K. Schaefer, T. Kechadi, and N. Le-Khac. Internet of Things Forensics – Challenges and a Case Study. In *IFIP International Conference on Digital Forensics*, pages 35–48. Springer, 2018. ISBN 978-3-319-99276-1. doi:10.1007/978-3-319-99277-8_3.

[4] Pwc's global economic crime and fraud survey. `https://www.pwc.com/gx/en/forensics/global-economic-crime-and-fraud-survey-2018.pdf`, Date Accessed: 8 October 2020.

[5] G. Lally and D. Sgandurra. Towards a Framework for Testing the Security of IoT Devices Consistently. In *International Workshop on Emerging Technologies for Authorization and Authentication*, pages 35–48. Springer, 2018. ISBN 978-3-030-04371-1. doi:10.1007/978-3-030-04372-8_8.

[6] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis. A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues. *IEEE Communications Surveys & Tutorials*, 22(2):1191–1221, 2020. doi:10.1109/COMST.2019.2962586.

[7] IoT security market. IoT security market watchkey market needs and solution providers in the IoT landscape. In *Global Digital Transformation Research Team at Frost & Sullivan*, 2017.

[8] A. H. M. Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan, and Y. Park. A Survey on Trend and Classification of Internet of Things Reviews. *IEEE Access*, 8:111763 – 111782, 2020. ISSN 2169-3536. doi:10.1109/ACCESS.2020.3002932.

[9] E. Tsogbaatar, M. H. Bhuyan, Y. Taenaka, D. Fall, K. Gonchigsumlaa, E. Elmroth, and Y. Kadobayashi. DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT. *Internet of Things*, 14:100391, 2021. doi:10.1016/j.iot.2021.100391.

[10] I. Andrea, C. Chrysostomou, and G. Hadjichristofi. Internet of things: Security vulnerabilities and challenges. In *2015 IEEE Symposium on Computers and Communication (ISCC)*, pages 180–187. IEEE, 2015. ISBN 978-1-4673-7194-0. doi:10.1109/ISCC.2015.7405513.

[11] J. Hou, L. Qu, , and W. Shi. A Survey on Internet of Things Security from Data Perspectives. *Computer Networks*, 148:295–306, 2019. doi:10.1016/j.comnet.2018.11.026.

[12] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and de S. C. Alvarenga. A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, 84:25–37, 2017. doi:10.1016/j.jnca.2017.02.009.

[13] E. Besharati, M. Naderan, and E. Namjoo. LR-HIDS: Logistic Regression Host-based Intrusion Detection System for Cloud Environments. *Journal of Ambient Intelligence and Humanized Computing*, 10(9):3669–3692, 2019.

doi:10.1007/s12652-018-1093-8.

[14] A. L. Buczak and E. Guven. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications surveys & tutorials*, 18(2):1153 – 1176, 2015. doi:10.1109/COMST.2015.2494502.

[15] W. Hatcher Grant and W. Yu. A Survey of deep learning: platforms, applications, and emerging research trends. *IEEE Access*, 6:24411 – 24432, 2018. doi:10.1109/ACCESS.2018.2830661.

[16] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim. A survey of deep learning-based network anomaly detection. *Cluster Computing*, 22 (1):949–961, 2019. doi:10.1007/s10586-017-1117-8.

[17] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici. N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018. ISSN 1536-1268. doi:10.1109/MPRV.2018.03367731.

[18] H. Bahşi, S. Nõmm, and F. B. La Torre. Dimensionality Reduction for Machine Learning Based IoT Botnet Detection. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1857–1862. IEEE, 2018. ISBN 978-1-5386-9583-8. doi:10.1109/ICARCV.2018.8581205.

[19] UCL Machine Learning Repository: *detection_of_IoT_botnet_attacks_N_BaIoT* Data Set. The n-baiot dataset. `https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT`, Date Accessed: 8 October 2020.

[20] J. Kim, M. Shim, S. Hong, Y. Shin, and E. Choi. Intelligent Detection of IoT Botnets Using Machine Learning and Deep Learning. *Applied Sciences*, 10(19):7009, 2020. doi:10.3390/app10197009.

[21] M. M. U. Chowdhury, F. Hammond, G. Konowicz, C. Xin, H. Wu, and J. Li. A few-shot deep learning approach for improved intrusion detection. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, pages 456–462. IEEE, 2017. ISBN 978-1-5386-1105-0. doi:10.1109/UEMCON.2017.8249084.

[22] A. Javaid, Q. Niyaz, W. Sun, and M. Alam. A Deep Learning Approach for Network Intrusion Detection System. *9th EAI International Conference on Bio-inspired Information and Communications Technologies*, 3(9), 2016. doi:10.4108/eai.3-12-2015.2262516.

[23] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho. Deep learning approach for Network Intrusion Detection in Software Defined Networking. In *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 258–263. IEEE, 2016. ISBN 978-1-5090-3938-8. doi:10.1109/WINCOM.2016.7777224.

[24] J. Zhang and M. Zulkernine. A hybrid network intrusion detection technique using random forests. In *First International Conference on Availability, Reliability and Security (ARES'06)*. IEEE, 2006. ISBN 0-7695-2567-9. doi:10.1109/ARES.2006.7.

[25] N. B. Amor, S. Benferhat, and Z. Elouedi. Naive Bayes vs decision trees in intrusion detection systems. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 420–424. IEEE, 2004. doi:10.1145/967900.967989.

[26] S. Mukkamala, G. Janoski, and A. Sung. Intrusion detection using neural networks and support vector machines. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, pages 1702–1707. IEEE, 2002. ISBN 0-7803-7278-6. doi:10.1109/IJCNN.2002.1007774.

[27] F. Farahnakian and J. Heikkonen. A deep auto-encoder based approach for intrusion detection system. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 178–183. IEEE, 2018. ISBN 978-1-5386-4688-5. doi:10.23919/ICACT.2018.8323688.

[28] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi. A Deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41 – 50, 2018. doi:10.1109/TETCI.2017.2772792.

[29] Y. Fu, F. Lou, F. Meng, Z. Tian, H. Zhang, and F. Jiang. An Intelligent Network Attack Detection Method Based on RNN. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pages 483–489. IEEE, 2018. ISBN 978-1-5386-4211-5. doi:10.1109/DSC.2018.00078.

[30] K. Alrawashdeh and C. Purdy. Toward an Online Anomaly Intrusion Detection System Based on Deep Learning. In *2016 15th IEEE international conference on machine learning and applications (ICMLA)*, pages 195–200. IEEE, 2019. ISBN 978-1-5090-6168-6. doi:10.1109/ICMLA.2016.0040.

[31] S. Nõmm and H. Bahşi. Unsupervised Anomaly Based Botnet Detection in IoT Networks. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1048–1053. IEEE, 2018. ISBN 978-1-5386-6806-1. doi:10.1109/ICMLA.2018.00171.

[32] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. In *Network and Distributed System Security Symposium*, 2018.

doi:10.14722/ndss.2018.23211.

[33] GitHub. Mirai-source-code. `https://github.com/jgamblin/Mirai-Source-Code`, Date Accessed: 8 October 2020.

[34] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou. Understanding the Mirai Botnet. In *26th {USENIX} security symposium ({USENIX} Security 17)*, pages 1093–1110, 2017.

[35] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. H. Chaves, Í. Cunha, D. Guedes, and W. Meira. The Evolution of Bashlite and Mirai IoT Botnets. In *2018 IEEE Symposium on Computers and Communications (ISCC)*, pages 00813–00818. IEEE, 2018. ISBN 978-1-5386-6951-8. doi:10.1109/ISCC.2018.8538636.

[36] Z. Khandezamin, M. N. Tahan, and M. J. Rashti. Intelligent detection of breast cancer with feature selection based on logistic regression and support vector machine Classification. *Journal of Soft Computing and Information Technology (JSCIT)*, 9(2):115–123, 2020.

[37] C.Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. In *2nd International Conference on Computational Sciences and Technology, (INCCST)*, 2020.

[38] Project jupyter. `https://jupyter.org/`, Date Accessed: 8 October 2020.

Fereshteh Abbasi received her M.Sc. and B.Sc. degrees in Computer Engineering, major in Artificial Intelligence from Shahid Chamran University of Ahvaz (SCU), Ahvaz, Iran in 2021 and 2018, respectively. Her research interests include artificial intelligence, deep learning algorithms and libraries, intrusion detection, and machine learning methods in networks.

Marjan Naderan received her B.Sc. degree in Computer Engineering in 2004 and the M.Sc. degree in Information Technology in 2006 both from Sharif University of Technology (SUT), Tehran, Iran. She received a Ph.D. degree in Computer Engineering, major in computer networks in Feb. 2012, from Amirkabir University of Technology (AUT), Tehran, Iran. Dr. Naderan joined the Computer Engineering department of Shahid Chamran University of Ahvaz (SCU) in Ahvaz, Iran in Sep. 2012. Her research interests include computer networks, wireless and mobile networks, IoT and social networks, network optimization, simulation of network protocols, and bio-inspired and intelligent methods in networks.

Seyyed Enayatallah Alavi received his B.Sc. degrees in Computer Engineering, major in Hardware, from Isfahan University of Technology, Isfahan, Iran, in 1992 and his M.Sc. degree in Computer Engineering major in Artificial Intelligence and Robotics from Shiraz University, Shiraz, Iran, in 1995. He also received his Ph.D. degree in Computer Engineering from the National Belarusian University of Technology, Minsk, Belarus, in 2012. Since 2012, Dr. Alavi joined the Computer Engineering department at Shahid Chamran University of Ahvaz, Ahvaz, Iran. His research interests include Fuzzy logic, Neural networks, Bio-inspired algorithms, and Image Processing.