# Anomaly detection in Internet of Things using feature selection and classification based on Logistic Regression and Artificial Neural Network on N-BaIoT dataset

2021 5th International Conference on Internet of Things and Applications (IoT) | 978-1-6654-4448-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/IoT52625.2021.9469605

Fereshteh Abbasi
M.Sc. student of Artificial Intelligence,
Department of Computer Engineering,
Shahid Chamran University of Ahvaz,
Ahvaz, Iran
f-abasi@scu.ac.ir

Marjan Naderan
Associate Professor, Department of
Computer Engineering, Shahid Chamran
University of Ahvaz, Ahvaz, Iran
m.naderan@scu.ac.ir

Seyed Enayatallah Alavi
Assistant Professor, Department of
Computer Engineering, Shahid Chamran
University of Ahvaz, Ahvaz, Iran
se.alavi@scu.ac.ir

*Abstract*— **According to the paradigm of the Internet of Things (IoT), physical devices are connected to each other and to the Internet such that they operate automatically. One of the major challenges in the IoT is to detect and prevent intruders into the network and devices, a challenge that traditional solutions of Intrusion Detection Systems (IDS) are not responsive for it or at least not very efficient to use in IoT. In this article, we address the problem of using machine learning methods for anomaly detection 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, the N_BaIoT dataset, which consists of data samples related to nine devices IoT and several attacks is used according to a number of criteria for evaluating the performance of the proposed methods. Simulation results in comparison with other four deep learning methods in terms of F1-score and precision show that using logistic regression, is more efficient and the highest classification accuracy (equivalent to 99.98%) can be achieved.**

***Keywords: Internet of Thing; anomaly detection; Artificial Neural Network; Logistic Regression; Botnet;***

## I. Introduction

With the advent of Internet of Things (IoT) paradigm, physical devices are connected to each other and to the World Wide Web in such a way that they have the ability to 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 by infiltrating it. 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 and etc., and intrusion and failure in any of them can lead to irreparable damage. 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 [2]. For this purpose, a set of tools and mechanisms known as an 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 [3]:

- Signature based diagnosis
- Anomaly based diagnosis

These two categories are used either separately or combined with each other 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” [4]. However, this method is not sufficient to detect unknown intrusions, as their patterns are unfamiliar. In addition, keeping 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” [3].

Anomaly-based detection consists of three general modules:

1. Parameterizing: which 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: which is processing the parametric profiles to build a classification model that

978-1-6654-4448-4/20/$31.00 ©2021 IEEE

distinguishes between normal and anomaly behaviors.

3. Detection: which is using a built-in classification model to detect new traffic anomalies.

Among the most important solutions of anomaly-based methods are machine learning methods, which we can mention SVM, logistic regression, decision trees, etc. [4], [5], [6]. In recent years, there has been a great deal of interest in the use of deep machine learning methods, and a number of 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.

An important issue in the previous works is the absence of specific dataset(s) related to IoT environments. In fact, 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 [13] is found which is specifically related to IoT devices and it is a good choice for our study. Due to the fact that 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. In fact, 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 [4], 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 [9] and [10].

The rest of this paper is organized as follows: in section II the related works are presented in brief. In section III the proposed method is explained in details. In section IV, evaluation criteria and simulation results are presented and finally in section V conclusion and directions for future research is given.

## II. Related Work

In this section, we briefly 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.

In [7] the researchers examined the AutoEncoder (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 [8] 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. Both studies in [7] and [8], have not investigates a dataset related to IoT devices.

In [9], the researchers used the deep auto-encoder network on the N_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 [10], 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 [11], researchers examined the LSTM method on the NSL_KDD dataset, which has an accuracy of 97.5%. In [12], the researchers examined the DBN method with the Softmax classifier on the KDD9910% dataset, which has an accuracy of 97.9%.

It can be seen from the previous methods that since the N-BaIoT dataset is the newest and the only database on IoT tools, only references [9] and [10] have used this dataset. Furthermore, this dataset contains data from several devices and samples for each device is separate from other ones. Therefore, deeper investigations for each device and for each class of attack is required for this dataset. We have considered this dataset and samples related to one of its devices in this study.

## III. 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. In fact, 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. Fig. 1 shows the flowchart of the proposed method. In the following, we describe each of the steps in more detail.

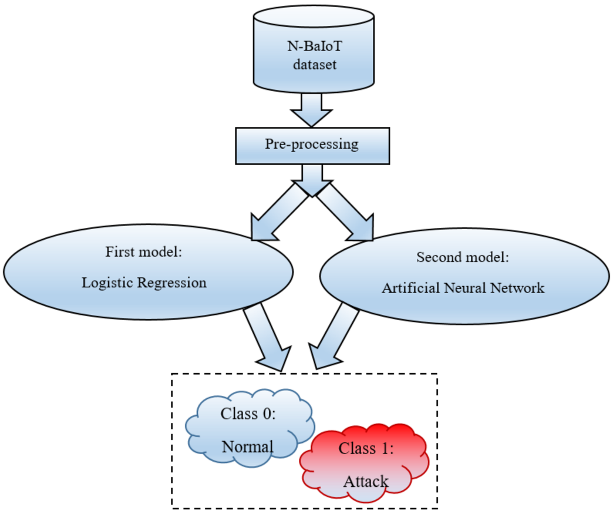

Figure 1. Flowchart of the proposed method

### *A. The dataset*

In this study, the N-BaIoT database with 115 different features has been used [13][19]. Comparing the number 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 I) 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 is used for each attribute are shown in Table II, 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 [10].

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 [14]. Since 2016, botnets have evolved significantly and are more dangerous [15][16]. Table III shows 10 specific types of Bashlite and Mirai attacks as also used in [10]. In this study, samples related to the Doorbell device data was used.

TABLE I. ALL NINE DEVICES USED IN THE N-BAIOT DATASET

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

TABLE II. DETAILS OF FEATURES CALCULATED IN THE N-BAIOT DATASET [19]

| 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 | |
| **Total** | | | 23 |

TABLE III. BOTNET AND ATTACK TYPES 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 UDP | Less of an option of UDP flooding for higher packet per second |

### B. 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 [4][17]. 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].

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

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

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

in which $\theta_0$ is the bias parameter and other $\theta_i$ which $1<=i<=115$, are the un-known parameters which are to be calculated. $x_i$ values are the feature values and the cost function is defined as:

$$\text{Cost}\ (h_\theta(x), y) = \begin{cases} -\log \ (h_\theta(x)) & if\ y = 1 \\ -\log(1 - h_\theta(x)) & otherwise \end{cases} \qquad (3)$$

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:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost\ (h_\theta(x^i), y^i) =$$

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

such that the input to this function are the un-known parameters and $m$ is the number of samples. To calculate the un-known 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 un-known 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, the gradient descent method can be used, therefore:

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

After classifying the data set by logistic regression algorithm, 115 values or coefficients for $\theta$s 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 this table). Next, coefficient values which are larger 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 coefficient. 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 much effect on the accuracy of the classification. This result is presented in Section 3, 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 IV.

### C. 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.

TABLE IV. 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 |

In this study, a 5-layer neural network consisting of three hidden layers is investigated. The number of neurons in the three hidden layer were 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 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 [18].

$$R(z) = \begin{cases} z & z > 0 \\ 0 & z <= 0 \end{cases} \qquad (6)$$

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

## IV. EVALUATION AND SIMULATION RESULTS

To simulate the proposed method, Python software and Jupiter Notebook [20], 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.

$$Norm\_value = \frac{realvalue - minvaluedataset}{maxvaluedataset - minvaluedataset} \qquad (7)$$

Next, the samples of this study are divided into two groups of train (70% of data) and test (30% of data) and the data of the first group are used in the learning process.

### A. 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 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 attack.

Using these basic criteria, the criteria used are:

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

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (8)$$

- 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): which 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).

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

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

$$Recall=\frac{TP}{TP+FN} \quad (10)$$

- 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).

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

### B. 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 2.2, features with larger coefficient values are added to the ROC chart sooner. This diagram is shown in Fig. 2. 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 are selected, which according to Table IV, are 19 features. Fig. 3 shows the confusion matrix, 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.

### C. Performance evaluation of the neural network method

To evaluate the performance of the proposed neural network we used two metrics: loss function [19] and accuracy. In figures 4 and 5 the horizontal axis represents the number of epochs and the vertical axis represents the loss function and accuracy, respectively. In these diagrams the red line is related to the train subset and the blue line is related to the test subset. From figures 4 and 5 it can be deduced that the proposed model converges and no signs of over/under-fitting is seen, despite some oscillations 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.

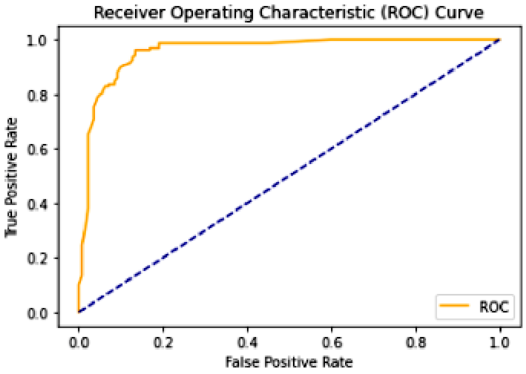

Figure 2. The ROC diagram according to the most important features selected by Logistic Regression for the N-BaIoT dataset

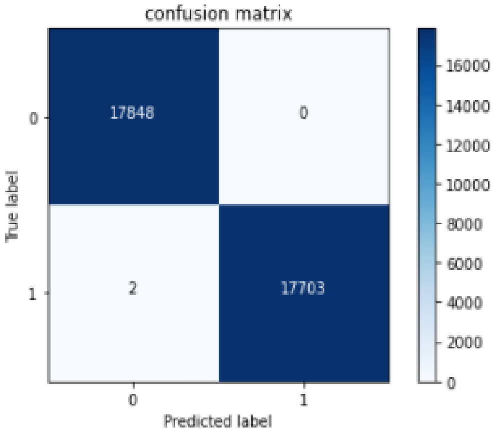

Figure 3. The confusion matrix resulted by classification by the Logistic Regression method for the N-BaIoT dataset

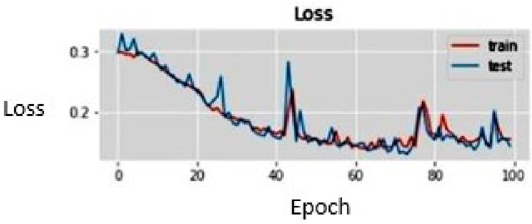

Figure 4. Diagram of the loss function vs. number of epochs

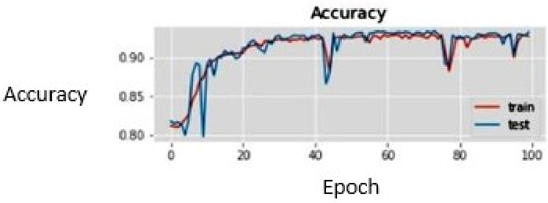

Figure 5. Diagram of accuracy vs. number of epochs

Figures 6, 7, 8 and 9 show the loss and accuracy functions with four and two hidden layers in the proposed neural network, respectively. As can be seen in the diagrams, the model is over-fitted. In fact, 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.

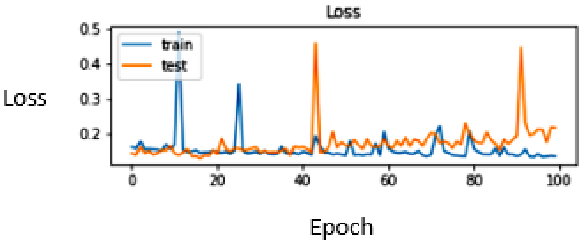

Figure 6. Diagram of the loss function for four hidden layers in the proposed neural network

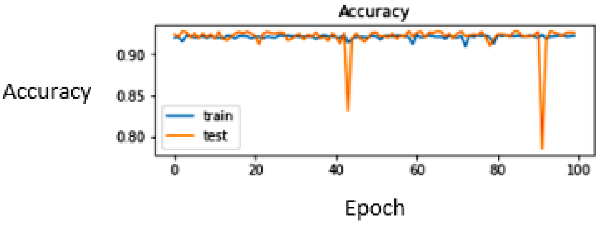

Figure 7. Diagram of accuracy for four hidden layers in the proposed neural network

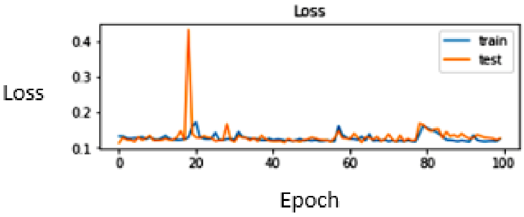

Figure 8. Diagram of the loss function for two hidden layers in the proposed neural network

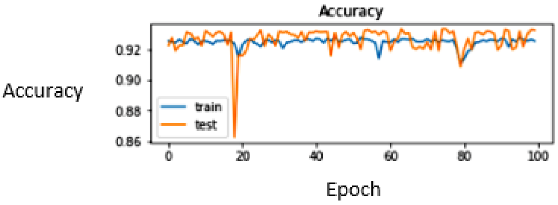

Figure 9. Diagram of accuracy for two hidden layers in the proposed neural network

Finally, Table V represents the results of several measures of the proposed methods (logistic regression and neural network) with two measures of the study in [9]. These results are the average of 10 execution for each measure. Table VI also compares the results of the proposed methods with that of the three other deep learning methods presented in [10]. As seen from these 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. These results are much better than that of the three other deep learning networks in [10] and comparable with the AE method in [9].

## V. Conclusion and Future Work

In this paper, the problem of IoT intrusion detection was addressed and for this purpose, a number of 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 one of the devices in the N-BaIoT dataset, which was the Doorbell, show that the feature selection method based on logistic regression leads to a 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 does not have over/under-fitting issues.

As for future work, it is recommended to use all the 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 specially the GAN neural network, which has not been used in this field so far.

TABLE V. COMPARISON OF THE RESULTS OF LOGISTIC REGRESSION METHOD AND THE PROPOSED NEURAL NETWORK

| **Performance measures** | **Auto-encoder [9]** | **Proposed ANN** | **Logistic regression** |
|---|---|---|---|
| Accuracy | - | 96.4% | 99.98% |
| Precision | 99.30% | 93.9% | 99.9% |
| Recall | 99.993% | 95.1% | 99.96% |
| F1-score | - | 99.13% | 99.92% |

TABLE VI. COMPARISON OF THE F1-SCORE MEASURE FOR THE PROPOSED METHODS AND THREE DEEP LEARNING NETWROKS IN [10]

|  | **F1-score** |
|---|---|
| **CNN [10]** | %91 |
| **RNN [10]** | %41 |
| **LSTM [10]** | %62 |
| **Proposed ANN** | %95.13 |
| **Logistic regression** | %99.92 |

## References

[1] D. Mendez Mena, I. Papapanagiotou, B. Yang, “Internet of things: Survey on security,” Information Security Journal: A Global Perspective, Vol. 27, No. 3, pp. 162-182, 2018.

[2] J. Hou, L. Qu, W. Shi, “A Survey on Internet of Things Security from Data Perspectives,” Computer Networks, Vol. 148, pp. 295-306, 2018.

[3] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, S. C. de Alvareng, “A survey of intrusion detection in Internet of Things,” Journal of Network and Computer Applications (JNCA), Vol. 84, pp. 25-37, 2017.

[4] E. Besharati, M. Naderan, E. Namjoo, “LR-HIDS: Logistic Regression Host-based Intrusion Detection System for Cloud Environments,” Journal of Ambient Intelligence and Humanized Computing, Vol. 10, No. 9, pp. 3669-3692, 2019.

[5] W.G. Hatcher, W.E.I. Yu, “A Survey of deep learning: platforms, applications and emerging research trends,” IEEE Access, Vol. 6, pp. 24411-24432, 2018.

[6] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, K. Kim, “A survey of deep learning-based network anomaly detection,” Cluster Computing, Vol. 22, pp. 949-961, 2019.

[7] F. Farahnakian, J. Heikkonen, “A deep auto-encoder based approach for intrusion detection system,” 20th Int. Conf. on Advanced Communication Technology (ICACT), Chuncheon, Korea (South), pp. 178-183, 11-14 Feb. 2018. DOI: 10.23919/ICACT.2018.8323688

[8] N. Shone, T. N. Ngoc, V. D. Phai, Q. Shi, “A Deep learning approach to network intrusion detection,” IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 2, No. 1, pp. 41–50, Feb. 2018.

[9] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, Y. Elovici, “N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders,” IEEE Pervasive Computing, Vol. 17, No. 3, 2018.

[10] J. Kim, M. Shim, S. Hong, Y. Shin, E. Choi, “Intelligent Detection of IoT Botnets Using Machine Learning and Deep Learning,” Applied Sciences, Vol. 10, Issue 19, 2020.

[11] Y. Fu, F. Lou, F. Meng, Z. Tian, H. Zhang, F. Jiang, “An intelligent network attack detection method based on RNN,” IEEE 3rd Int. Conf. on Data Science in Cyberspace, Guangzhou, China, pp. 483–489, 18-21 July 2018. DOI: 10.1109/DSC.2018.00078

[12] K. Alrawashdeh, C. Purdy, “Toward an online anomaly intrusion detection system based on deep learning,” 15th IEEE Int. Conf. on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, pp. 195-200, 18-20 Dec. 2016. DOI: 10.1109/ICMLA.2016.0040

[13]The N-BaIoT dataset: https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

[14] Mirai source code: https://github.com/jgamblin/Mirai-Source-Code (accessed on 8 October 2020).

[15] M. Antonakakis, et. al, “Understanding the Mirai Botnet,” In Proceedings of the 26th USENIX Security Symposium, Vancouver, BC, Canada, pp. 1093–1110, 2017.

[16] A. Marzano, et. al, “The evolution of Bashlite and Mirai IoT Botnets,” In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, pp. 00813–00818, 25-28 June 2018. DOI: 10.1109/ISCC.2018.8538636

[17] Z. Khandezamin, M. Naderan Tahan, 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), Vol. 9, No. 2, pp. 115-123, 2020.

[18] C. E. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning”, 2nd International Conference on Computational Sciences and Technology (INCCST), Jamshoro, Pakistan, pp. 124-133, 17-19 Dec. 2020.

[19] Y. Mirsky, T. Doitshman, Y. Elovici, A. Shabtai, “Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection,” Network and Distributed System Security Symposium (NDSS’18), San Diego, CA, USA, 18-21 Feb. 2018. DOI: 10.14722/ndss.2018.232

[20] https://jupyter.org/