Automated diagnosis of attacks in internet of things using machine learning and frequency distribution techniques

Toufik Ghrib[1], Mohamed Benmohammed[2], Purnendu Shekhar Pandey[3]
[1]University of Mohamed Khider Biskra, Algeria
[1]Laboratory of Valorization and Promotion of Saharan Resources, University of Kasdi Merbah, Ouargla, Algeria
[2]Department of Software Technologies and Information Systems, Faculty of New Technologies of Information and Communication, University Constantine2, Algeria
[3]BML Munjal University, India

Article Info

Article history:

Received Nov 11, 2020
Revised Jan 14, 2021
Accepted Feb 6, 2021

Keywords:

Internet of things (IoT)
Intrusion detection system
ML techniques
MQTT
Security

ABSTRACT

The internet of things (IoT) is the interconnection of things around us to make our daily process more efficient by providing more comfort and productivity. However, these connections also reveal a lot of sensitive data. Therefore, thinking about the methods of information security and coding are important as the security approaches that rely heavily on coding are not a strong match for these restricted devices. Consequently, this research aims to contribute to filling this gap, which adopts machine learning techniques to enhance network-level security in the low-power devices that use the lightweight MQTT protocol for their work. This study used a set of tools and, through various techniques, trained the proposed system ranging from ensemble methods to deep learning models. The system has come to know what type of attack has occurred, which helps protect IoT devices. The log loss of the ensemble methods is 0.44, and the accuracy of multi-class classification is 98.72% after converting the table data into an image set. The work also uses a convolution neural network, which has a log loss of 0.019 and an accuracy of 99.3%. It also aims to implement these functions in IDS.

Corresponding Author:

Toufik Ghrib
University of Mohamed Khider 07000 Biskra, Algeria
Email: gharib.toufik2006@gmail.com

1. INTRODUCTION

In our digital world, security is of the utmost importance. The idea of the internet of things (IoT) is based on connecting objects around us to make our everyday lives more efficient and thus provide more comfort and productivity in our business and personal life. But these connections also expose sensitive data. It is evident that the requirement for security is undoubtable. The internet of things displays new usage challenges. As the quantity of associated devices in our lives develops and the measure of information (data) that is collected everyday skyrockets, security a progressively crucial.

The objective of MQTT is to give a lightweight and simple-to-use communication protocol for internet of things. The protocol itself indicates just a few security mechanisms. At the network level, the intrusion detection system (IDS) is used to detect various anomalies and protect our IoT systems. The objective of this research is to adopt machine learning techniques to enhance network level security in power constrained devices that use the lightweight MQTT protocol for their functioning. This work is interested in testing various machine learning techniques that could help improve intrusion detection systems. The multiclass classification models are provided with data containing frames under various types of attacks and normal frames labelled respectively. The pertained model can then be utilized to identify and thus prevent unwanted attacks or intrusion in the IoT system. In this research, we were also able to classify the DDoS attack even if the there is a low rate of attack.

2. RELATED WORK

To detect the anomalies in traditional types of networks there are many approaches in machine learning. The dataset used for that are KSL-KDD and KDD99 [1]. This data set contains different attacks followed on TCP protocols (traffic is analysed). Balancing the dataset was indeed an important task to increase the accuracy of the model, which was done by balancing classes [2-5]. Fuzziness based on supervised learning has also been implemented to improve the accuracy of recognition of attack by [6] whereas sequential extreme learning has improved the machine learning detection of attack by [7]. Thus, the obtained results clearly depict that these algorithms (machine learning approaches) are the appropriate approaches to enhance the detection of intrusion at the network layer. There are also approaches in deep learning used to detect the intrusion and any anomalies [8]. One way of doing that is to apply DBM (deep belief network), for selecting various features on the KDD dataset and then applying SVM over that. This predicts the type of attack [9-10]. Another approach is adopted to find the fisher score using deep learning approaches. In this case, a classical statistic approach is used along with auto encoder to decrease the number of features and exact the features of the highest importance [11]. Deep learning is not only used for intrusion detection, but it is also used for classifiers. As the Temporal data sequence of intrusion detection is also useful, so LSTM (long short term memory) has also been used to find the attack (using the KDD dataset) [12, 13]. With respect to the IoT (internet of things) and IDS (intrusion detection system), there are many approaches that use edge computing and fog computing. Using fog computing and edge computing, a simulation is created for the NSL-KDD dataset. This approach to detecting IDS has exhibited good time dependence performance and good accuracy [14]. Over the KDD dataset, the rules of IDS detection are also modified using machine learning approaches such as SVM and KNN [15]. As IoT as a field is still developing, the given solutions cover fewer aspects of IoT attacks [16]. There are other datasets such as AWID [17], in which TCP frames for the WLAN network is collected and analysed for attacks on 802.11. In this case, study of Wi-Fi intrusion was done by a neural network classifier [18]. For training purposes, the CICIDS dataset [19] is used to validate the intrusion detection algorithm with recurrent neural networks [20, 21]. The present research paper is mainly based on IoT scenarios and help in detecting vulnerability concerning the IoT. The IoT data traffic is analysed, which uses MQTT protocol for communication between the publisher and various clients.

Here we have used the KDD dataset, but the problem with the KDD data set is that it was not clean (not pre-processed) and has redundant values. Second, the parameters (variables) of the experiment were not clear. Third, the important variables, which clearly help in spotting the attack, were not used in the KDD dataset. In order to overcome these problems, we have created our scenario, as mentioned in the KDD data set links. We even generated the data set and found the parameters, such as sequence number, MAC address, socket number and others, which greatly helps in finding the attack, and even reduced the dimension of the KDD dataset (which was not important). This paper then used the combined data set of KDD and the scenario data that we have generated, which has ultimately given us better results for this given research.

3. PROBLEM STATEMENT

While there is no doubt that many kinds of security are a requirement, including information security in our day-to-day life, we face a lot of implementation challenges when it comes to the Internet of Things, which demands high usability. Security has always presented a trade-off off between the degree of insurance and the level of ease of use. This trade-off gets significantly more intriguing with the Internet of Things. Normally, IoT devices create very low memory capacities and little computing power. High-security cryptographic algorithms require considerably more assets than small IoT devices can have. So, until we find that Holy Grail of compact energy sources, we must search for different lightweight ways to provide high-level security to our tiny but crucial IoT devices that make our lives easier, more productive and ergonomic.

Security approaches that depend vigorously on encryption are not a solid match for these constrained devices, since they are not equipped to perform complex encryption and decryption rapidly enough to have the option to transmit information safely, progressively and securely in real-time [22-25]. IoT frameworks systems should make use of multiple layers of safeguard and defence to make up for these device impediments. Applying “security intelligence” for detecting, recognizing and mitigating attacks as they occur is an IoT security challenge. One approach to this, as used in our work, is to use multiclass classification of frames using standard machine learning techniques to correlate frame characteristics with frame type. A longer-term goal may include unfolding threats by applying AI to predict adaptively modified security method, applied depending on the viability of past activities and previous actions.

As can be seen in the preceding Figure 1, the setup uses a. There are three levels in the hierarchy-level 1, 2, and 3. Each component in the different levels plays a different role. The base units work at the base level and collect data from the sensors and appliances and transfer it to the intermediary node. This transferred data, received from all leaf nodes, and is aggregated at the intermediary node. From there, it is forwarded to the server via MQTT protocol. The Server collects data from all the intermediary nodes and then processes and analyzes it. MQTT is indeed an application layer protocol, whereas at transport layer TCP is used to increase the reliability in communication. At network layer we have IP and at data link layer the sensors works, whereas, physical layer used to send the data in binary form. But, as we can see that there is no security aspect in IoT architecture so we need ways to protect the IoT network. Too overcome this problem the machine learning algorithm has been embedded in the application layer of the IoT architecture.

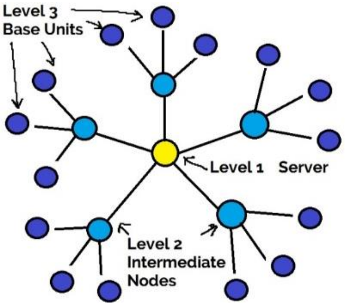


Figure 1. Hybrid star tree topology

Security approaches that depend vigorously on encryption are not a solid match for these constrained devices, since they are not equipped of performing complex encryption and decryption rapidly enough to have the option to transmit information safely progressively and securely in real-time. IoT frameworks systems should make utilize of multiple layers of safeguard and defence to make up for these device impediments. Applying “security intelligence” for detecting, recognizing and mitigating attacks as they occur is an IoT security challenge. One approach to this as used in our work is to use Multiclass classification of frames using standard machine learning techniques to correlate frame characteristics with frame type. A longer-term goal may include unfolding threats by applying AI to predict adaptively modify security method applied dependent on the viability of past activities and previous actions

4. EXPERIMENTS

Various software tools that are used are Low Orbit Ion Cannon (LOIC), Ettercap, and Wireshark. Wireshark is used to store the traffic of data moving from my system to the internet and also data coming from the internet to our system. For attacking, the system LOIC and Ettercap tools are used. Once the data is stored, we have applied a machine learning algorithm using python. As far as hardware is concerned, this research paper has used other os ar9271 Wi-Fi adapter to capture the packets in monitor mode. We have used different types of tools in order to perform different types of attacks. These tools are LOIC (for performing DoS attack) and Ettercap (for performing MitM attack) [9, 14], as shown in Figure 2(a)-(b). LOIC is a windows application used to perform different types of DDoS attacks and DoS attacks using protocols like TCP, UDP, HTTP etc.

Timeout 9001
HTTP Subsite /
TCP / UDP message A cat is fine too. Desudesudesu~
Port 80
Method
Threads 10

(a)

(b)

Figure 2. (a) LOIC tool, (b) Ettercap tool

Ettercap [20, 26], which is depicted in Figure 2, is mainly used in Kali Linux Devices. It is used to perform different types of attacks on our device that is running the Ettercap, used as a malicious node to perform the man-in-the-middle attack. First, we need to have all the target sources and the malicious node in the same network, then the malicious node opens Ettercap and clicks on scan for hosts that will scan and give the results of available nodes/devices in that network. While performing the attack, we need to collect the data for our analysis. We are using the Wireshark data from the victim's device. That data is converted to CSV format and labelled according to the attack type.

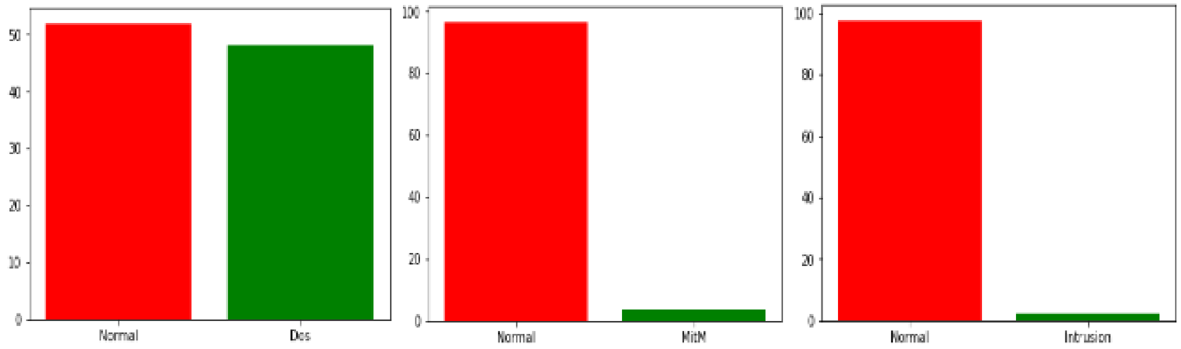
Now, after completing the above process, we have three datasets, which are the DDoS, MitM and the normal dataset. These datasets need to be cleaned and then combined to form one dataset, then used for the classification process. With DDoS, we will follow a different approach, as it is hard to distinguish from usual traffic. DDoS is a type of denial of service attack where multiple compromised nodes that are distributed over the globe attack a server that offers service. Mainly, DDoS attacks are one of two types. High volume attacks are also known as Brute Force attacks. They can be easily detected because of the sudden high traffic. The other attack type is vulnerability attacks that attack weaknesses in the protocol. As the traffic graph is similar to normal traffic, they are difficult to detect. Also, high volume attacks are no longer viable for attackers even if the servers with cloud providers are now in a large quantity. Additionally, since multiple tenants share a single cloud server, it can impact multiple services in a cascading manner. Low rate DDoS attacks are instead periodic in nature and come from multiple machines, trying to exploit deficiencies in different protocols in a computer network. Due to them not being easily distinguishable from normal traffic, they are more difficult to detect and can be more harmful. Thus, to find DDoS we will follow a different approach given in a later section.

4.1. DataSet

We have generated the datasets as mentioned in the section above, but those are unorganized. Now comes the pre-processing that needs to be performed before applying the model. After completion of the pre-processing of the dataset, it is clean and organized. We need to build a model that performs the classifications based on the label given in the dataset. We need to prepare our training and testing data using sklearn.model_selection() we used train_test_split() to split our data into 70% training and 30% testing.

4.2. Binary classification

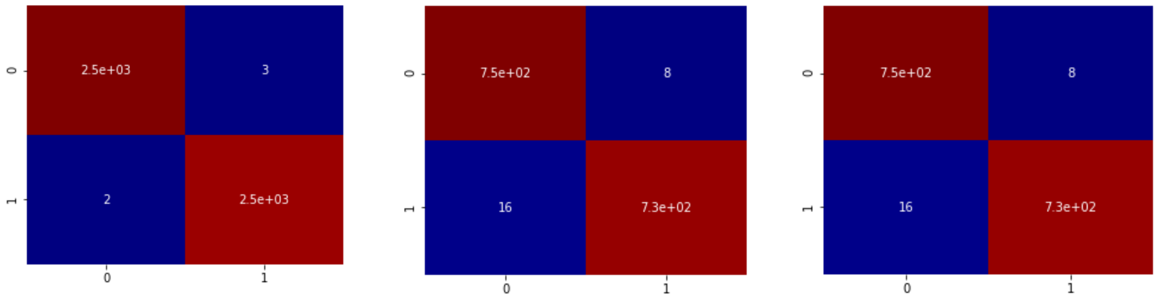
Binary classification of the three datasets as shown in Figure 3, was done as a base to view distinctions between normal frames and frames under attack. Initially the data sets are shown in the figure below:



(a) (b) (c)

Figure 3. Initial three data sets without processing, (a) DoS data, (b) MitM data, (c) Intrusion data

Two of the three datasets are visibly imbalanced vis-a-vis MitM and intrusion data. Also, the three data sets feature a different number of samples. The work uses standard implementation of XGBoost for classification for the reasons that are explained above. Quite satisfactory scores are obtained for binary classification. The three data sets lead to following confusion matrices for DoS, MitM, and intrusion attack, as shown in Figures 4(a)-(c) respectively.



(a) (b) (c)

Figure 4. Confusion matrices for, (a) DoS attack data, (b) MitM attack data, (c) Intrusion attack data

The fast algorithm of XGBoost was used to develop the trees for gradient boosting. In this tree, growing method just a subset of conceivable split values is considered. The classifiers classify 5 samples incorrectly out of a total of 5000 samples in the DoS dataset, whereas it misclassifies 24 samples out of 1400 samples in MitM, and intrusion dataset. The binary classification, although very good, is not of much use as using different filters will increase load on the devices, slowing down real-time communication. Thus, a multiclass classifier needs to be developed. For better finding DDoS attacks, we have used the following ways to gather the data set: As a very first step in our process, we will be grouping data by frequencies / 10 milliseconds. Our current dataset has the timestamp and the address. We need to combine all those timestamps and group them by 10 seconds. This will allow us to construct a frequency chart. When we use the frequencies over a grouped interval of 10 milliseconds, and create a power spectrum distribution of them, using normal fast Fourier transform (FFT), we get the following for a normal Non DDoS scenario and a DDoS scenario respectively. This will be done only after splicing the different types of data. After this, we will be using Matplotlib to plot the data.

4.3. Multiclass classification (ensemble methods)

Combined data sets look as shown in Figure 5. The data is imbalanced, as can be seen in the bar graph of Figure 5. Different numbers of samples and imbalance in individual datasets makes the combined dataset imbalanced. The combined data contains (X and Y axis) 17500, 10000, 2500, and 5000 samples of frames with normal, DoS, intrusion and MitM labels respectively. This can be dealt with using oversampling techniques, like SMOTE and random oversampling, if needed. For the time being, however, the research leaves it to the model to perform under this imbalance. We take care of this fact in the coming sections as needed. This paper used the standard parameters, which are provided by Sklearn’s random forests, for this result. A grid search was performed later to decide upon the best parameters of the model.

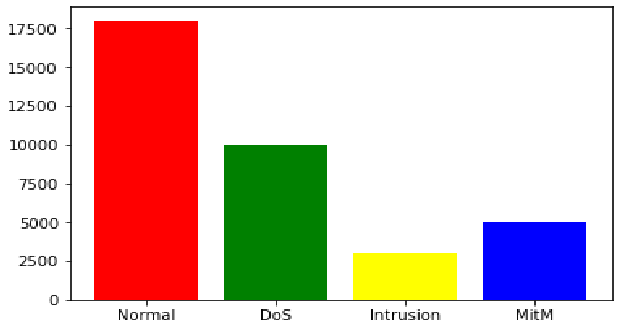


Figure 5. Combined data sets and attack classification

4.4. Multiclass classification (deep learning methods)

Convolutional neural networks have transformed computer vision. Their automatic feature engineering capabilities have allowed the release of the load from hard-coding or manual feature engineering.

Vanilla neural networks don’t give very good results when applied to images because they lose the spatial relation between pixels, which is not only preserved but also enhanced by computer vision algorithms. However, in this moment, the researcher tends to adopt a different method. Tabular row data is converted into an image to process and thus use the benefits of CNN, as depicted in Figure 6.

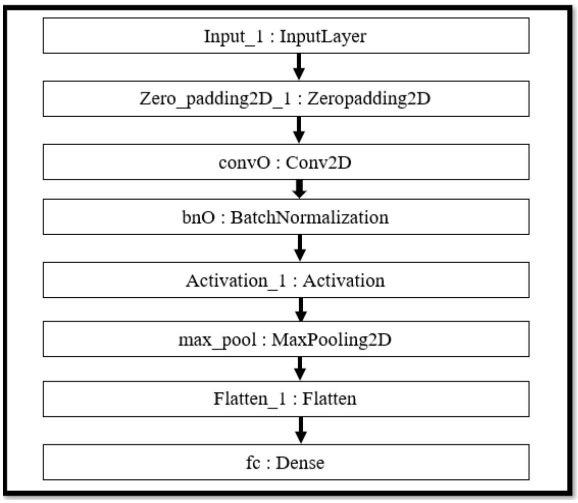


Figure 6. Following layers are used in this architecture as can be seen in the model plot

The attacks are either high rate or low rate. High rate attacks are easy to detect due to their high volume, while low rate attacks are difficult to detect as the traffic spikes are not observed. Low rate attacks normally attack deficiencies in the low level protocols. In the coming sections we will also see how we can predict whether it is a DDoS attack in a low traffic attack. For finding the DDoS attack, we have used the algorithm showed in Figure 7. In this implementation, as shown in Figure 7, we will first analyze multiple PSD transformations on attack data on different protocols. Further, we will be analyzing which transformations are more susceptible to a DDoS attack and thus are better to detect one as it happens. We will be using multiple PSDs and seeing the periodicity observed in them with respect to normal traffic and traffic under a DDoS attack. Additionally, we will be analyzing the time taken to process an input to analyze computational efficiency.

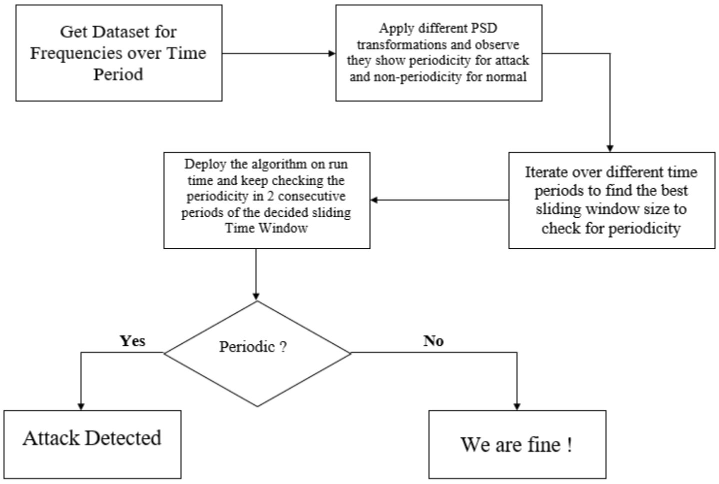

Get Dataset for Frequencies over Time Period
Apply different PSD transformations and observe they show periodicity for attack and non-periodicity for normal
Iterate over different time periods to find the best sliding window size to check for periodicity
Deploy the algorithm on run time and keep checking the periodicity in 2 consecutive periods of the decided sliding Time Window
Periodic ?
Yes
No
Attack Detected
We are fine !

Figure 7. Finding the DDoS attack

5. RESULTS AND DISCUSSION

The training followed by testing data led to the following outcomes.

5.1. Ensemble methods

Both random forest and XGBoost perform quite similarly. XGBoost results are shown in Table 1. Even though the classification accuracy is 98%, the classification isn't as good as desired because of the accuracy paradox described above. The imbalance of the dataset leads us to this problem. The confusion matrix proves to be a good metric to test classification correctness. Other measures, Precision and recall provide good insight into the correctness of classification. These are determined as entireties and proportions of various pieces of a confusion matrix, as depicted in Figure 8.

Table 1. Ensemble scores

N°	Metric	Score
0	Accuracy	0.987225
1	Recall	0.812599
2	Precision	0.985884
3	F1 Score	0.879643
4	F beta Score	0.879643
Log_loss		0.44122046213658767

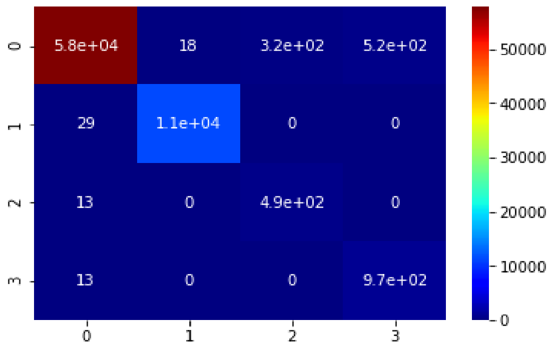
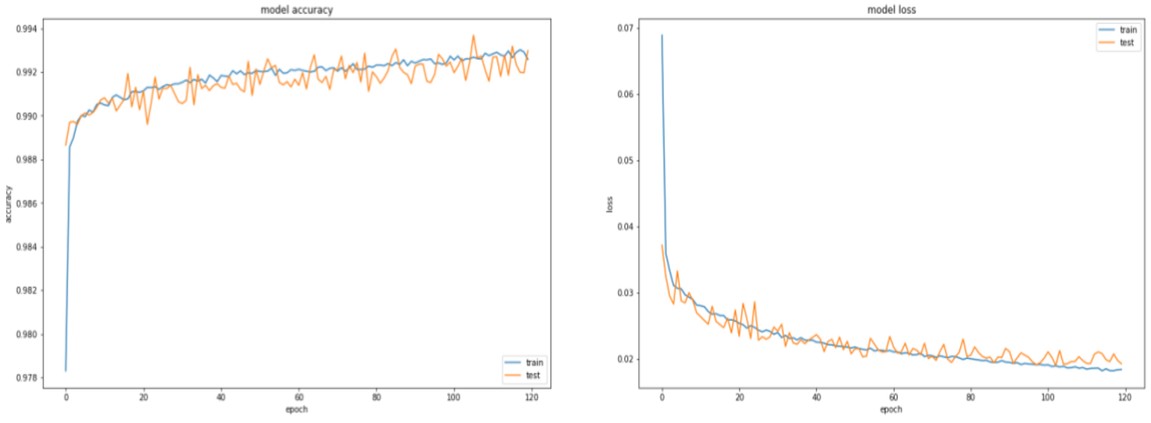


Figure 8. Ensemble confusion matrix

The confusion matrix points out that MitM and intrusion data frames were quite frequently misclassified as normal samples. 320 and 520 samples were misclassified out of 810 and 1490 samples of intrusion and MitM datasets, respectively. This indicates inefficient classification. Multiclass classification problems tend to be more complex than binary problems, which makes getting better results more difficult for these problems. Although the imbalance was dealt with, there were enormous contrasts between classes. This may have influenced the precision in a few models negatively.

5.2. Deep learning methods

The CNN model outperforms both ensemble methods after enough training. The training curves for the model can be seen in Figure 9. With increasing of epochs (iteration over the defined batch size) the training and testing accuracy of the model increases, i.e. it learns the parameters (improve over initial random prediction basically through back propagation) that describes our data perfectly and helps in better classification. The logarithmic loss (multiclass log loss) plot shows a gradual decrease of epochs both for training and testing, signifying that the model is making less and less errors in classifying training and testing samples shown by the blue and orange curve respectively.



(a)

(b)

Figure 9. (a) Accuracy plot for training and testing, (b) Loss plot for training and testing

The CNN model performed better than ensemble methods on all parameters, as mentioned in Table 2. The reason for that lies that CNN resulted in low variance as compared to ensemble methods. Even the parameters such as Accuracy, f1_score, recall score of CNN model has outperformed ensemble methods. As shown in Figure 10, again, the confusion matrix is used to see how well the samples labelled MitM and intrusion (minority samples) are classified by our new model, because, as discussed in previous sections, this data faces the accuracy paradox due to imbalance. The confusion matrix shows that the deep learning model outperforms ensemble models when classifying minority samples. The following confusion matrix is obtained:

Table 2. CNN scores

Metric	Score
Accuracy	0.9932771464911178
f1_Score	0.9271314804803563
f bita score	0.9271314804803563
recall_score	0.9366207730622080

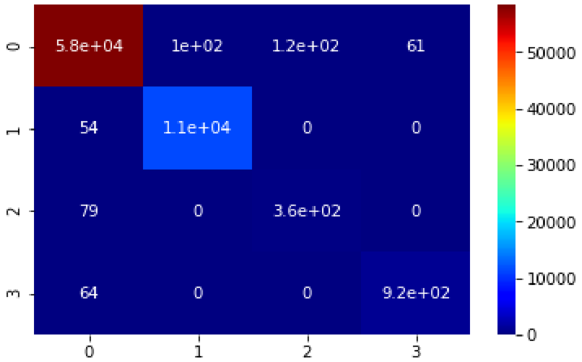


Figure 10. CNN confusion matrix

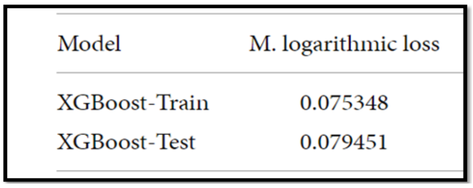
The CNN model misclassifies only 100 out of 11000, 120 out of 360 and 61 out of 920 samples of DoS, Intrusion and MitM respectively of the validation dataset. Table 3 summarizes the model that gave the above confusion matrix:

Table 3. Model summary

Layer (type)	Output shape	Param #
Input_1 (InputLayer)	(None, 8, 8, 1)	0
Zero_padding2d_1 (ZeroPadding2D)	(None, 14, 14, 1)	0
conv∅ (conv2D)	(None, 8, 8, 32)	1600
bn∅ (BatchNormalization)	(None, 8, 8, 32)	128
activation_1 (Activation)	(None, 8, 8, 32)	0
max_pool (MaxPooling2D)	(None, 4, 4, 32)	0
flatten_1 (Flatten)	(None, 512)	0
fc (Dense)	(None, 8)	2052
Total params : 3780		
Trainable params : 3716		
Non-trainable params : 64		

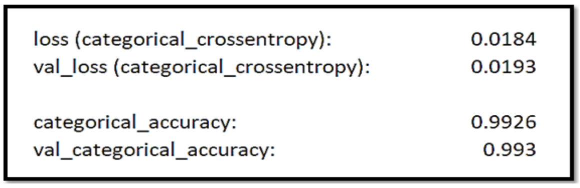
5.3. Result comparison

The XGBoost model is compared in terms of Model logarithmic loss in Figure 11, which clearly shows that the data is balanced. This work achieved a log loss of 0.0193, 0.0184 improving upon the previous works log loss of 0.079, 0.0753 for the validation set and training sets respectively, as shown in Figure 12.



Model	M. logarithmic loss
XGBoost-Train	0.075348
XGBoost-Test	0.079451

Figure 11. Multiclass classification procedure for detecting attacks on MQTT-IoT protocol



```
loss (categorical_crossentropy):          0.0184
val_loss (categorical_crossentropy):      0.0193

categorical_accuracy:                     0.9926
val_categorical_accuracy:                  0.993
```

Figure 12. Comparison of log loss values and accuracy for XGBoost training and testing data

Based on DFT or discrete wavelet transform the normal TCP flow possesses periodic property, this is not the same for attack flows. Periodicity can be estimated using the power spectral density. But as a downside even if the TCP flow is not periodic due to legitimate reasons, it can be marked as DDoS attack. To overcome this, frequency domain can be used as depicted in Figures 13-15 respectively; these attacks have high energy in low frequency bands. Other than relying on frequency or power domains, collaborative filtering using routers or template matching can be used. These allow using the previous attack characteristics so as to find what might be a DDoS attack. Special hardware systems such as FPGA PSD converters can be used. Also, as eventually a DDoS attack leads to congestion in the network, the traffic with a higher congestion participation rate can be termed as malicious traffic. So, given below are few algorithms which clearly shows how we can detect malicious attack by change in frequency.

Here in Figure 13, it clearly shows the change in frequency is sharp as soon as attack happens, while protocol used here is LDAP (lightweight dictionary access protocol), sharp rise in frequency can be observed during the duration of the attack. In Figure 14, it clearly shows the change in frequency is sharp as soon as attack happens, while protocol used here is UDP (user datagram protocol), sharp rise in frequency can be observed during the duration of the attack. In Figure 15, it clearly shows the change in frequency is sharp as soon as attack happens, while protocol used here is MySQL. The graph always looks uniform as the attack is low rate, and thus frequency techniques are not sufficient to detect the presence of an attack

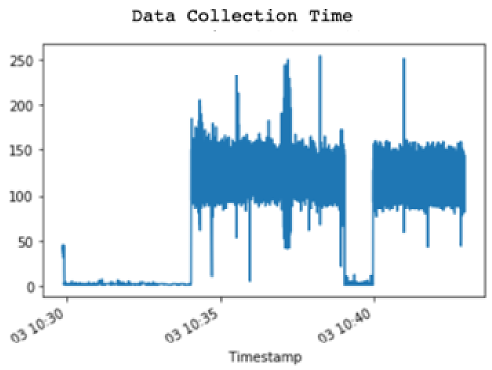


Figure 13. Frequency plot of data of LDAP protocol

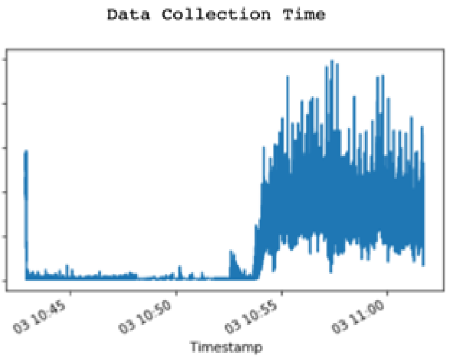


Figure 14. Frequency plot of data of UDP protocol

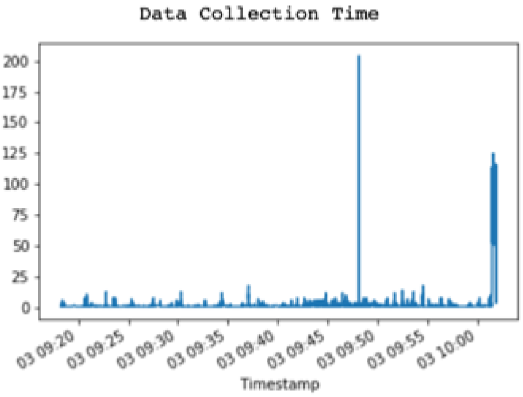
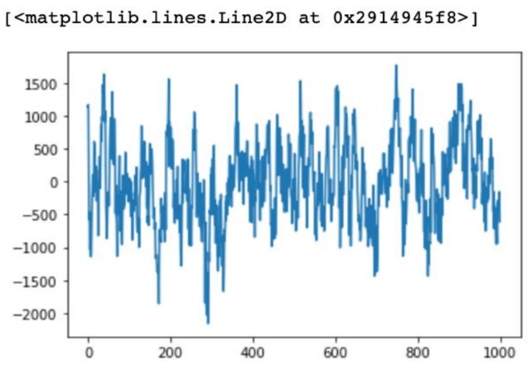
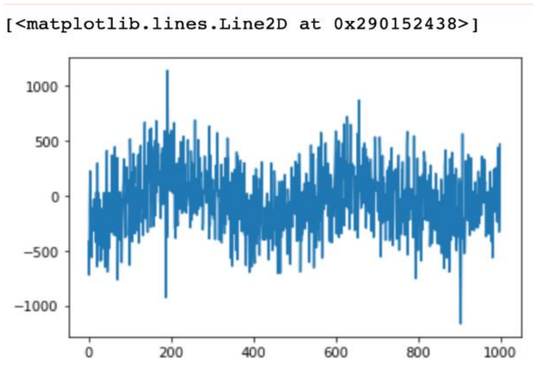


Figure 15. Frequency plot of data of MySQL protocol

But anomalies also exist in some types of data like, which results in randomness, as shown in Figure 16. and it is sometimes hard to find i.e no specific pattern is observable in any subset of the data due to the inherent randomness of genuine traffic. Figure 17, depicts that in an attack scenario, the PSD is showing periodic properties, while no periodicity is observed in a normal scenario, as shown in Figure 16. This is the way the attack is programmed; the traffic tends to be periodic. While that is not true for normal traffic, where the frequencies might increase or decrease. Here, a strong repetition of pattern is observed and thus, an attack scenario is created and hence, is difficult to be completely random.



```
[<matplotlib.lines.Line2D at 0x2914945f8>]
```



```
[<matplotlib.lines.Line2D at 0x290152438>]
```

Figure 16. The power spectral distribution for the non attack scenario of the frequencies of input traffic

Figure 17. The power spectral distribution (PSD) for the attack scenario of the frequencies of input traffic

In Figure 18, similar graphs are observed for other PSDs such as correlogram, covariance, and Yule-Walker to predict the DDoS attack scenarios. Here it is clearly observed that while the degree of periodicity might be different, each of these algorithms does show uniformity in an attack scenario. While correlogram is observed to be depicting periodicity here as well, as shown in Figure 19, none of the other PSD algorithms show any uniformity for a non-attack scenario, validating our claim. But, as a few still show periodicities, it cannot be taken as a general claim that PSD will never be uniform for non-attack cases.

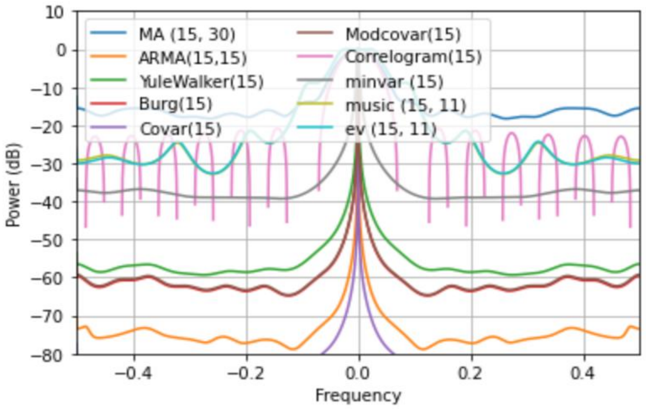


Figure 18. Plots for several different power spectral distribution algorithms like correlogram, covariance, and Yule-Walker for the DDoS attack scenario of a single protocol

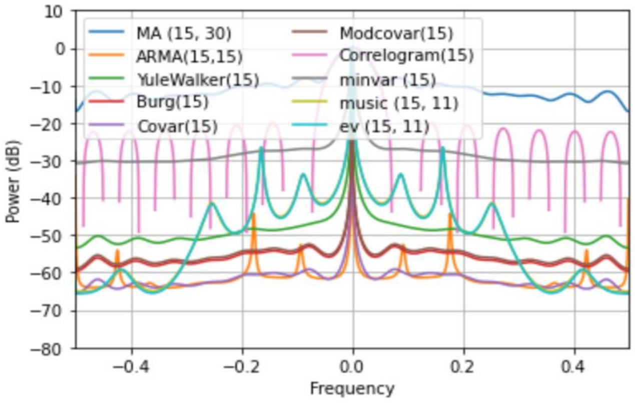


Figure 19. Plots for several different power spectral distribution algorithms like correlogram, covariance, and Yule-Walker for the non-attack scenario

6. CONCLUSION

The internet of things has been a link between the physical and digital world for a long time. Today, everything we think of as "smart" can be attributed to IoT systems. The heterogeneous nature of these systems makes their security challenging. Intrusion detection systems (IDS) will be the frameworks (systems) that monitor network traffic for suspicious action or activity and issues alerts when such activity is discovered. Network intrusion detection systems have provided security in these devices for a long time. These detection systems are trained with datasets containing attacked labels of various attacks under MQTT protocol, which is used for communication between IoT devices. The present research adopted various techniques for training our system, ranging from ensemble methods to deep learning models. It used random forests and XGBoost under the category of ensemble methods. CNN was opted for under the deep learning category.

These models can be exploited for future work in which an intrusion detection system IDS is reinforced with a model. This paper mainly deals with intrusion detection, i.e. once the attack has happened, it will help in knowing whether the attack has happened and, if so, what type of attack, but it does not deal with intrusion prevention. Thus, the future work will mainly deal with not only detection of attack, but also prevention of various types of attacks, so this work can be the future work of this research work. The main focus of this research came to contribute to filling this gap, which adop ts machine learning techniques to enhance network-level security in the low-power devices that use the lightweight MQTT protocol for their work. This study used Low Orbit Ion Cannon (LOIC), Ettercap, Wireshark tools and through various techniques to train the proposed system ranging from ensemble methods to deep learning models. The system has come to know what type of attack has occurred, which helps protect IoT devices. The log loss of the ensemble methods is 0.44, and the accuracy of multi-class classification is 98.72%. After converting the table data into an image set. The work also uses a convolution neural network, which has a log_loss of 0.019 and an accuracy of 99.3%. It also aims to implement these functions in IDS.

ACKNOWLEDGEMENTS

This work is supported by the Ministry of Energies, the Ministry of the Environment and Renewable Energies and MESRS Algeria, under the agreement number (A05N01UN300120180003).

REFERENCES

[1] I. S. Arora, G. K. Bhatia, and A. P. Singh, "Comparative analysis of classification algorithms on KDD'99 data set," *International Journal of Computer Network and Information Security*, vol. 9, pp.34-40, 2016.

[2] B. Chakrabarty, O. Chanda, and Md. Sinful, "Anomaly based intrusion detection system using genetic algorithm and k-centroid clustering," *International Journal of Computer Application*s, vol. 163, no. 11, p. 13-17, 2017.

[3] N. A. Hussein and M. I. Shujaa, "DNA computing based stream cipher for internet of things using MQTT protocol," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 1035-1042, 2020.

[4] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLOS ONE*, vol. 11, no. 6, p. e0155781, 2016.

[5] J.-H. Seo and Y.-H. Kim, "Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection," *Computational Intelligence and Neuroscienc*e, vol. 2018, p. 1-11, nov. 2018.

[6] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484-497, 2017.

[7] R. Singh, H. Kumar, and R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8609-8624, 2015.

[8] 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 Computer*, vol. 22, p. 949-961, 2019.

[9] Y. Xiao and X. Xiao, "An intrusion detection system based on a simplified residual network," *Information*, vol. 10, no. 11, p. 356, 2019.

[10] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *International Journal of Security and Its Applications*, vol. 9, no. 5, pp. 205-216, 2015.

[11] H. A. Tran, D. Tran, L. G. Nguyen, Q. T. Ha, V. Tong, and A. Mellouk, "SHIOT: A novel SD*N*-based framework for the heterogeneous Internet of Things," *An International Journal of Computing and Informatics*, vol. 42, pp. 313-323, 2018.

[12] A. Bonguet and M. Bellaiche, "A survey of denial-of-service and distributed denial of service attacks and defenses in cloud computing," *Future Internet*, vol. 9, no. 43, pp. 1-19, 2017.

[13] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, NSW, pp. 1-6, 2018.

[14] X. An, X. Zhou, X. Lü, F. Lin, and L. Yang, "Sample selected extreme learning machine based intrusion detection in fog computing and MEC," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1-10, 2018.

[15] L. Deng, D. Li, X. Yao, D. Cox, and H. Wang, "Mobile network intrusion detection for IoT system based on transfer learning algorithm," *Cluster Computing*, vol. 22, no. 4, p. 9889-9904, 2019.

[16] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25-37, 2017.

[17] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184-208, Firstquarter 2016.

[18] M. E. Aminanto, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Wi-Fi intrusion detection using weighted-feature selection for neural networks classifier," *2017 International Workshop on Big Data and Information Security (IWBIS)*, Jakarta, pp. 99-104, 2017.

[19] Z.-Q. Qin, X.-K. Ma, and Y.-J. Wang, "Attentional payload anomaly detector for web applications," *International Conference on Neural Information Processing*, Springer, Cham, Switzerland, pp. 588-599, 2018.

[20] T. Halabi et M. Bellaiche, "How to evaluate the defense against DoS and DDoS attacks in cloud computing: A survey and taxonomy," *International Journal of Computer Science and Information Security (IJCSIS0)*, vol. 14, no. 12, pp. 1-10, 2016.

[21] M. A. Naagas, E. L. Mique Jr, T. D. J. D. Palaoag, and Dela Cruz, "Defense-through-deception network security model: Securing university campus network from DOS/DDOS attack," *Bulletin of Electrical Engineering and Informatics*, vol. 7, no. 4, pp. 593-600, 2018.

[22] C. Vijayakumaran, B. Muthusenthil, and B. Manickavasagam, "A reliable next generation cyber security architecture for industrial internet of things environment," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 387-395, 2020.

[23] S. Bravo and D. Mauricio, "Systematic review of aspects of DDoS attacks detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 1, pp.162-176, 2019.

[24] M. Turkanovic, "Authentication and key agreement protocol for Ad Hoc networks-based on the Internet of Things paradigm," *An Interntional Journal of Computing and Science*, vol. 40, no. 1, pp. 153-154, 2016.

[25] P.S Juwita, R. Fadhil, T. N. Damayanti, and D. N. Ramadan, "Smart parking management system using SSGA MQTT and real-time database," *TELKOMNIKA Telecommunication, Computing, Electronics and Contro*l, vol. 18, no. 3, pp. 1243-1251, 2020.

[26] X. Tao, D. Kong, Y. Wei, and Y. Wang, "A big network traffic data fusion approach based on fisher and deep auto-encoder," *Information*, vol. 7, no 20, pp. 1-10, 2016.