IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 35, No. 4, August, 2021, pp. 349-357
Journal homepage: http://iieta.org/journals/ria

Identification of Network Traffic over IOT Platforms

Shilpa P. Khedkar[1,2*], Aroul Canessane Ramalingam[1]

[1] Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai 600119, India
[2] Department of Computer Engineering, M.E.S. College of Engineering, S. P. Pune University, Pune 411001, India

Corresponding Author Email: shilpa.khedkar@mescoepune.org

https://doi.org/10.18280/ria.350410

Received: 19 July 2021
Accepted: 15 August 2021

Keywords:
traffic classification, network traffic, Internet of Things, machine learning, deep learning

ABSTRACT

The Internet of Things (IoT) is a rising infrastructure of 21st century. The classification of traffic over IoT networks is attained significance importance due to rapid growth of users and devices. It is need of the hour to isolate the normal traffic from the malicious traffic and to assign the normal traffic to the proper destination to suffice the QoS requirements of the IoT users. Detection of malicious traffic can be done by continuously monitoring traffic for suspicious links, files, connection created and received, unrecognised protocol/port numbers, and suspicious Destination/Source IP combinations. A proficient classification mechanism in IoT environment should be capable enough to classify the heavy traffic in a fast manner, to deflect the malevolent traffic on time and to transmit the benign traffic to the designated nodes for serving the needs of the users. In this work, adaboost and Xgboost machine learning algorithms and Deep Neural Networks approach are proposed to separate the IoT traffic which eventually enhances the throughput of IoT networks and reduces the congestion over IoT channels. The result of experiment indicates a deep learning algorithm achieves higher accuracy compared to machine learning algorithms.

1. INTRODUCTION

An IoT (Internet of Things) has been evolved from cloud computing and has become a promising infrastructure to suffice the on-demand requirements of the users. An IoT infrastructure comprises three main components, front end devices with sensing capability, back-end storage and computing facility, and a communication network that connects front-end to back-end for communication. IoT is connecting a huge number of diverse devices which are heterogeneous in nature using wired or wireless communication [1, 2]. The devices in IoT environment are highly mobile and cover a wide geographical area while moving from one region to another [3]. Hence, with the advent of IoT, different types of wireless technologies have been researched and employed to provide seamless services to IoT users. However, IoT has transformed the conventional way of connectivity into a high-tech connectivity where everything can be connected anytime and anywhere, but there is a huge risk involved for the connecting devices and users. There are more possibilities for adversaries to attack the IoT devices. It is indeed very difficult to detect and prevent the network abuse with growing volume of traffic over IoT. The IoT devices also make individual and organizations more vulnerable along with their benefits of enormous connectivity and usability. Due to heterogeneous nature of IoT devices interoperability is the major issue in IoT networks. Devices from unknown vendors have different installation and configuration methods. To communicate such smart devices with heterogeneous capabilities required heterogeneous communication technologies. The usage of heterogeneous communication technologies like ZigBee, RF links, Bluetooth, 6LoWPAN, WiFi has given rise to critical issues such as balancing of traffic load, segregation of traffic, storage sharing in devices, response time, throughput, task sharing and so forth [4]. Summary of communication technologies shown in Table 1.

In IoT environment, the devices which are heterogeneous in nature, sense and collect data in the sensing plane, and disburse the data to the gateway [5]. The load of integrated traffic in IoT network may become exhaustively heavy with normal as well as malicious traffic. However, many traffic classification models for IoT environment have been presented and studied by the researchers [6-10] but the use of machine learning algorithms in IoT is still to be explored which will certainly be a great solution to balance the traffic load, for predicting the traffic load and to channelize the traffic to the designated nodes. Hence, in this paper machine learning and deep learning based approaches have been proposed which would improve the network throughput and responsiveness of the network. The contribution to this paper is elaborated below.

1. First, we are considering standard dataset available at the repository of University of California, Irvine to train our ML (machine learning) model. The dataset is divided into training and testing with the ratio of 7:3. The dataset provides information about the packet payload. The state of the session and application information from the content of the packet can be retrieved.
2. The data processing module is then initiated to clean the data and to rescale the data to make it appropriate for machine learning based classifiers. Then we have applied ensembling machine learning approaches such as AdaBoost and Bagging classifiers by modifying them with respect to our problem statement to train the proposed system and to achieve better accuracy. IoT Network classification is made based on normal traffic (periodic, event and query based) and malicious traffic arisen from the Botnets.

3. We then propose novel deep learning using artificial neural networks for traffic classification at the packet-level. The proposed approach classifies the traffic and enhances the classification accuracy by experimenting with no. of hidden layers and other fine-tuning parameters.

4. We have presented performance comparison of machine learning and deep learning-based algorithms on test set by using confusion matrix, F1 Score, AUC score, Precision and Recall Scores.

The proposed model allows the segregation of the traffic generated by IoT devices. The motive is to forward the normal traffic on the IoT channels and deflects the malicious traffic at the earliest to free up the bottled- necked channels occupied by unwanted traffic. This approach would assist in maximizing the throughput; minimize the congestion and maximizing the transmission-rate.

This paper has been divided into different sections as: First section provides introduction and contributions of the paper in brief. The second section covers state-of-the-art works in the aligned field. Next section explains proposed work using machine as well as deep learning approaches and research outcomes. The last section summarizes the proposed work and provides future directions on our area of research.

Table 1. Comparison of communication technologies

Communication Technology	Topology	Bandwidth	Range	Spectrum	Bit Time (μs)
ZigBee	Mesh, Star, Tree	250 Kbps	10-300 m	2.4 GHz	4
RF links	-	18 MHz	<3 m	2.4 GHz	-
Bluetooth	Star	1 Mbps	<30 m	2.4 GHz	1.39
6LoWPAN	Mesh, Star	250 Kbps	800 m	2.4 GHz	-
WiFi	Star	upto 54 Mbps	4-20 m	2.4-5 GHz	0.0185

2. RELATED WORK

Many researchers have proposed techniques for segregation of the traffic [11-14] over the internet for different purposes, but IoT needs a quicker mechanism to classify the network traffic for optimizing the throughput, response-time, transmission rate, bandwidth consumption, network latency and minimizing the congestion rate. The existing traffic classification techniques have been explored in this section prior to presentation of our proposed work in next section.

2.1 Port-number based approach

Traffic classification using port information is the oldest technique. These classifiers gather the information about the port number using the TCP headers and UDP header of the packets. After ascertaining the port number, the comparison is made between the assigned TCP/UDP and extracted port numbers for the classification of the traffic. This classification procedure is the fastest and the simplest method for traffic classification [15]. According to Moore et al. [16] around 30% to 60% of the total traffic can be segregated using port-number based methods. A few applications such as P2P applications (Napster and Kazaa) do not use their own ports. In some cases, the server ports are allotted dynamically as per the requirement. Moreover, the connecting devices on IoT can transmit huge amount of data anytime and this method is not sufficient to suffice the need of the hour. There is a need to explore newer methods for traffic classification.

2.2 Statistical approaches

There exist other approaches, known as statistical approaches, where each application possesses statistical characteristics which are unique in nature with respect to each application. Crotti et al. [17] have proposed protocol based on the probability density function of inter-arrival time and normalized thresholds of the packets. The research study has been conducted for a group of protocols such as POP3, HTTP, and SMTP. The outcome of their proposed work has achieved 91% accuracy. Similar work proposed by Parish and Wang and Parish [18] presented an approach for optimizing the classification of network traffic by making use of multiple classifiers. The classification covers a broader range of protocols FTP, IMAP, TELNET and TCP with 87% accuracy.

2.3 Payload-based segregation of IP traffic

Due to complete dependency on port numbers, many industry products make use of packet payload information. Sen et al. [19] have presented a technique to classify the P2P application traffic by utilizing the application level signatures. Researchers Papagiannaki and Moore have used a mixture of port and payload-based methods to classify the network applications [20]. The classification begins with the examination of port number of a flow. If well-known port is not found, the first packet is examined for known a signature. If the signature is not seen, then the packet is inspected for a well-known protocol. Thus, flows remained unclassified and then needs an inspection of the entire flow payload. By making use of a full payload packet, an attempt has been made to identify the types of errors that may result from port-based classification. However, payload-based approaches avoid reliance on port numbers, but this method does not work well while there is a need to deal with encrypted traffic.

2.4 Machine learning and deep learning-based approaches

Many research endeavors have been made by the researchers to make use of machine learning algorithms for the network traffic classification. Bayesian neural network (BNN) used by Auld et al. for classification of the P2P protocols which includes Bit Torrent, Kazaa and GnuTella [21]. The authors have used a distributed host-based traffic collection platform known as DHTCP to gather traffic data. In Ref. [22] probabilistic neural network was applied for segregating the traffic. Web and P2P traffics were considered for the experimental study. Draperl et al. have studied the effectiveness of flow-based features to detect VPN traffic and to classify the encrypted traffic into diverse categories [23]. They have made use of k-nearest neighbor (k-NN) and C4.5 decision tree algorithms for classification into six classes such as Web browsing, email, chat, streaming, file transfer and VoIP. They have achieved 92% recall using k-NN and 88%

recall using the C4.5. In Ref. [24] the authors have tried to identify end-user applications such as Facebook, Skype and Twitter. They have used UNB ISCX Network Traffic dataset which comprises 14 well-known applications. They have used four ML based algorithms namely Random Forest, J48, Bayes Net and K-NN. K-NN and Random Forest provides better accuracy as compared to other algorithms as 93.94% accuracy and 90.87% respectively.

Deep Learning (DL) based approaches have also used by the researchers for network traffic classification in recent years with promising results. Hwang, et al. have used LSTM model for classification of malicious traffic using packet information. The major advantage of work is that it doesn't require processing of packets into flows which reduces preprocessing time and has achieved classification accuracy of 97% [25]. Lim, et al. have presented two different deep learning models for traffic classification using CNN and ResNet. Using preprocessing techniques, the data is generated for eight application layer-based applications [26].

Aceto et al. have proposed different DL classifiers like MLP, CNN, SAE, etc. for traffic classification [27]. CNN classifier shows best performance with 85.70 % accuracy using android data set. Abbasi et al. [28] have conducted a review on usage of DL for network traffic monitoring and analysis. Wang et al. used sequential feature selection for multilayer perceptron for detection of DDOS attacks from the network traffic. Using this optimal feature selection during training phase, researchers have achieved 98% accuracy [29]. Bendiab et al. proposed ResNet50 for detecting malicious traffic. Dataset consists of 1000 pcap files used for training of the model and finally the model has achieved 94.50% accuracy [30].

There are many existing approaches to classify the traffic on internet, but with the advent of Internet of Things, the task of classifying the traffic has become more tedious and complex. Classical Machine learning algorithms need experts and manual feature extraction methods which are not suitable for modern networks because: (1) mobile traffic increase due to smart phones and tablets, (2) change in mobile traffic along with increasing mobile application, (3) transport layer protocol reduces effectiveness of deep packet inspection method based on ML algorithms. Hence, there is a need to discover newer algorithms rather than depend on classical ML algorithms for the segregation of the IoT traffic. Hence, we are proposing machine learning and deep learning based dynamic approaches to segregate the IoT traffic for minimizing the congestion and to improve the channel utilization for normal traffic. This dynamic approach is called ensemble learning which improved results by combining different models. Applications of ensemble learning are it improve confidence of model to make decisions, selection of optimal features, error correcting and incremental learning. Simple ensemble techniques are max voting, averaging, weighted averaging and advanced are stacking, blending, bagging, and boosting. We used boosting technique in our work.

3. PROPOSED METHODOLOGY

Machine learning and deep neural networks-based procedures are presented in this section to segregate the IoT traffic in an accurate manner. The generic view of machine learning based model is shown in Figure 1. A standard dataset with 14000 records has been used to train our ML model the dataset is available for experimental usage at the repository of University of California, Irvine. The proposed traffic classification model has been trained on 70% of dataset and testing has been performed on 30% of data respectively. IoT Network classification is a multivariate classification problem where the segregation of traffic is made based on four classes: periodic traffic, event-based traffic and query-based traffic and malicious traffic. Periodic traffic is generated by periodic sensing devices who senses temperature, light, and humidity periodically and send it to the controller. Event driven traffic are created after occurrence of any specific event. This type of traffic is irregular and random. As a result of query, a random and irregular traffic are created which is called query-based traffic. Malicious traffic is created by suspicious link, files, and connection created or received on network.

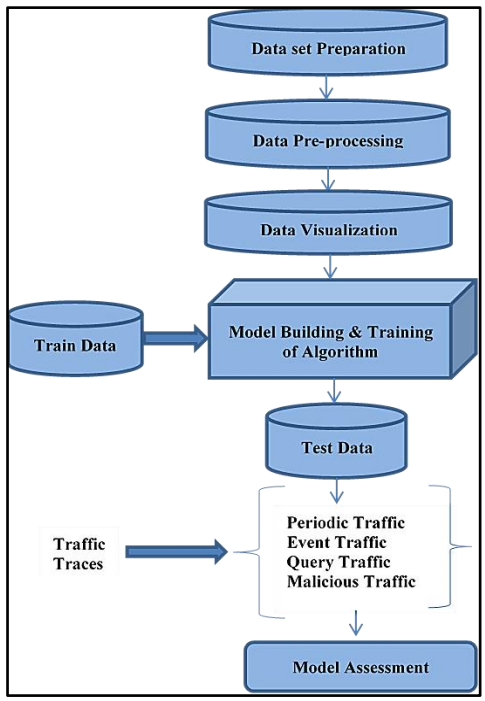

Figure 1. ML based model to classify the network traffic in IoT environment

3.1 Parameters of dataset for ML based model

The parameters considered for study are shown in Table 2. The data should be balanced and equally distributed before applying machine learning based algorithms. Therefore, rescaling methods have been applied on the dataset to make the dataset appropriate for our classification model. Min-max scaling is used which is the simplest way to rescale the range between [0, 1]. The generic formula for min-max scaling is as shown in Eq. (1).

$$a' = \frac{a - \min(a)}{\max(a) - \min(a)} \quad (1)$$

Here a is representing original value where as a' represent normalized value. We have rescaled a range between arbitrary values (x, y) by using the formula as shown in Eq. (2).

$$a' = x + \frac{(a - \min(a))(x - y)}{\max(a) - \min(a)} \quad (2)$$

where, (x, y) represents the range of min-max values.

Table 2. Dataset attributes for classification of the IoT traffic

Sr. No.	Attribute	Description
1.	Utilized Bandwidth Rate	Information about the used Bandwidth
2.	Node	Nodes that are transmitting data
3.	Packet rcvd rate	Packets collected per second by each node
4.	Assigned Bandwidth	Initially assigned bandwidth to each node
5.	End-to-End delay time per sec	Depicts End-to-end delay time per second
6.	% of lost Packet rate	The rate of lost packets at the time of transmission
7.	% of lost Byte rate	The rate of the Bytes lost between source to destination
8.	Packet Drop Rate	Packet drop rate as a normalized value
9.	Consumed bandwidth	Bandwidth consumed
10.	Lost Bandwidth	Bandwidth lost during transmission
11.	Packet size byte	Packet size in Bytes
12.	Packets Transmitted	Packets transmitted per second
13.	Packets Received	Packets collected by each node per second
14.	Packets lost	Lost packets per second
15.	Transmitted Byte	Transmitted bytes by each node per second
16.	Received Byte	Bytes received by each node per second
17.	10 iterations Avg Drop Rate	Avg drop rate of packets for 10 iterations
18.	10 iterations Avg Bandwidth Use	Avg of utilized bandwidth for 10 consecutive iterations
19.	10 iterations Delay	Avg latency for 10 consecutive iterations
20.	Status Node	Depicts classification of nodes
21.	Packet Payload	Unencrypted information about the packet, 0 -malicious, 1-Query, 2-Periodic, 3-Event

Heatmap: A heat map is a 2D representation of the given data which makes use of the colors to highlight the correlation between the variables. Correlation is used to represent dependency between two variables. It also shows the statistical measure of linear relationship between two variables. For multiple variables they are store in matrix and then relation is shown using correlation matrix. Correlation are measure using the following:

1) Correlation coefficient / Pearson correlation coefficient: This coefficient measures variation of values of two variables with respect to each other. This coefficient is calculated as shown in Eq. (3).

$$\frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}} \tag{3}$$

where, x & y are variables, x bar is mean of x,y bar is mean of y ,xi and yi are different values of x and y.

2) Rank correlation coefficient metric such as Spearman correlation coefficient is used to measure the extent to which one variable increases / decreases as the other variable increases / decreases.

The correlation coefficient can take any values from -1 to 1. If the value is 1 it indicates positive correlation, -1 indicates negative correlation and 0 indicates no relation between two variables. Heat map assists in visualizing the relationships between the variables. It helps to identify whether the variables are co-related to each other and if correlation is there then how strong is the correlation between the variables. The heatmap shown in Figure 2 depicts that more co-related variables are brighter in color. For example, the correlation between percentage of lost packet rate and packet drop rate, packet transmitted, and full bandwidth is very strong whereas correlation between packet transmitted and delay time, delay time, packet drop rate and transmitted byte is very weak. The entire heatmap can be studied with the brightness and dullness of colors even without looking at the respective values.

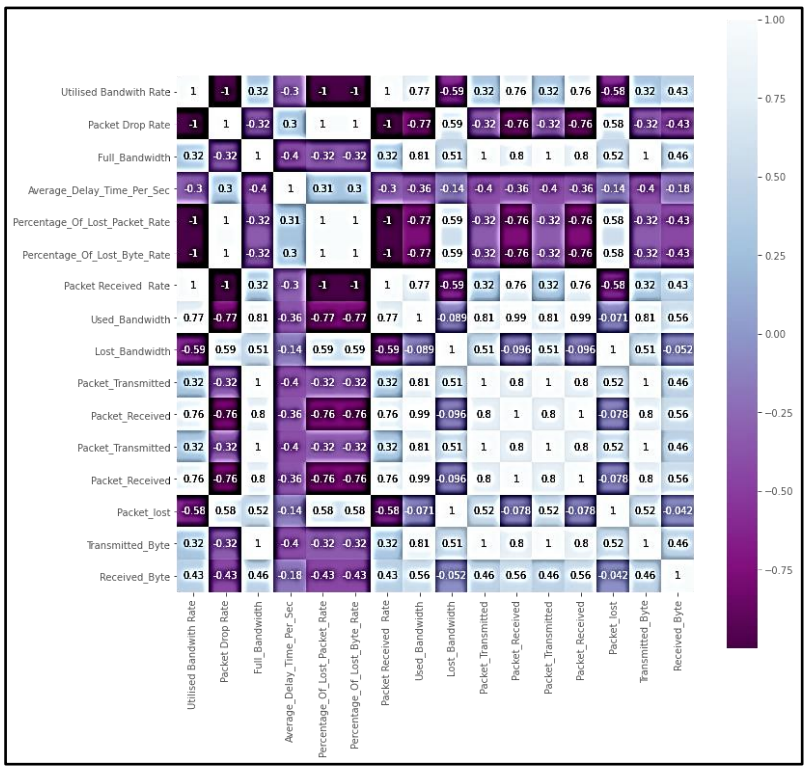

Figure 2. Heat map for variable considered for the study

3.2 Machine and deep leaning approaches to classify malicious traffic

This subsection describes about the network traffic classifiers. We have applied modified AdaBoost, Bagging classifier and deep neural networks to classify the network traffic into four classes, i.e. periodic traffic, event based traffic, query based traffic and malicious traffic in IoT environment. The outcome of each algorithm is measured by using the performance metrics such as classification accuracy Score, confusion matrix, precision score, sensitivity score, and F1 Score. The simulation environment is created on a desktop with an Intel Core i7-6900K processor, 64GB RAM, RHEL (Ootpa) 8.0 platform with kernel 4.18, and Nvidia Titan Xp GPU. The link bandwidth is assumed to be 64 MbP/s and the switch buffer size is set to 5 MB. We have reasonably restricted the simulation to a small sized network to demonstrate the proof-of-concept of the proposed methodology. The onion routers have been generated in NS-3 based simulated environment to add the mechanism of classifying the IoT traffic. The mechanism of classification is presented in Table 3 before the explanation of the algorithms used for the research work.

Table 3. Traffic segregation algorithm used by onion router in IOT environment

Algorithm 1: Classification algorithm for IoT traffic
Input: IoT devices - info, ready-queue, Onion router $_{count}$,Dataset IoT traffic
1. Begin
2. while Ready flow in not empty do
3. T← IoT traffic from the ready – queue of Onion Router
4. Run the ML or DL classifier for segregation of IoT traffic T
5. Deflect the malicious traffic
6. Check the priorities of the normal traffic
7. Determine the nearest Onion routers from the routing table
8. Forward the normal traffic to the next Onion router based on priority of traffic
9. end while
10. end
11. PROCEDURE: Select the ML or DL based classifier
12. Input: Training Dataset
13. Begin
14. index←0
15. while T [index]≠ null do
16. t← T [index]
17. r← Select the ML or DL algorithm for classification of traffic T
18. Transmit normal traffic t to selected router r
19. Update status of traffic t
20. index← index + 1
21. end while
22. end

3.2.1 AdaBoost Classifier

AdaBoost, also known as adaptive boosting, is the highly effective algorithm which is used for classification nowadays. AdaBoost is a kind of gradient boosting with built-in functionality of cross-validation. It allows the user to run a cross-validation at each iteration of the boosting process and thus this property makes the process of getting the exact optimum number of boosting iterations in a single run quite easy. Hence, we are using AdaBoost in our research work for promising results. The idea of boosting has evolved from the concept of learning where a weak learner can be trained well to become a better learner. In AdaBoost, the weak learners are the decision trees with a single split, also called decision stumps due to their shortness. The Adaboost algorithm has been applied into three steps on our problem statement for classifying the IoT based traffic into four aforesaid classes:

1) A weak learner is allowed to make classifications, as we have applied greedily constructed decision tree.
2) An additive model has been applied to add weak learners to optimize the loss function.
3) Newer weak learners have been merged to our Machine learning classification model to correct the residual errors made by all the former trees.

For AdaBoost model creation most important parameters are base_estimator, n_estimators, and learning rate.

1) Base_estimator: -Learning algorithm to train weak model is called base_estimator. Decision tree is the bydefault value for this parameter.
2) n_estimator: -is the number of models to train iteratively. We used this value as 10.
3) Learning rate: - is the contribution of each model to the weights. This value is set to 0.1 for forcing model to be train slowly.

The outcome is a powerful classification modelling algorithm which can classify the IoT traffic into 4 classes. The intention is to segregate the malicious traffic from the normal traffic. Periodic, event and query-based traffic is considered as normal traffic in this research study. The performance of the proposed algorithm is evaluated using confusion matrix as presented in Figure 3, ROC curve in Figure 4 and the evaluation matrix as in Tables 4 & 5. The accuracy score is 80.11 % for training dataset and the accuracy score is 78.99% for testing dataset.

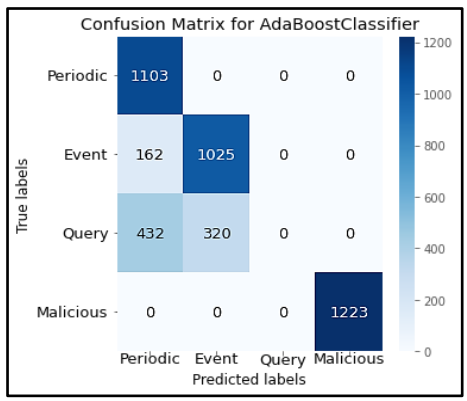

Figure 3. Confusion matrix Adaboost based classification

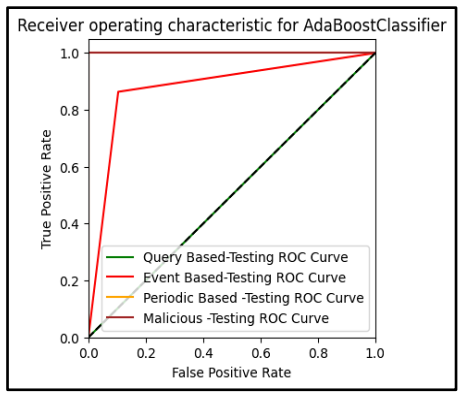

Figure 4. ROC curve for Adaboost classifier

3.2.2 Bagging Classifier

Bagging Classifier is one of the most powerful and more popular machine learning algorithms. It is an ensembling algorithm that not only classifies the data but also makes predictions [31] Bagging stands for bootstrap aggregation and is a powerful statistical technique for estimating a quantity from the given dataset. Bagging attempts to use similar learners on the sampling dataset and then it aggregates all the results. It uses boot- strap method of sampling to fetch the subsets of datasets for the training of base learners. An ensemble method in bagging classifier combines the predictions/results from multiple machine learning algorithms altogether to make better and accurate predictions than any individual algorithm. For aggregating the outputs of base learners, bagging uses voting for classification. In our research work, we have modified the original bagging classifier to make it suitable to our problem statement and to add novelty. we have created many random sub-samples of our dataset with replacement. Then we have trained our CART model (classification and regression trees) on each sample. We then have calculated the average prediction from each model. While bagging with decision trees, lesser concern has been shown to individual trees that are overfitting the training data. The individual decision trees are grown deep, and the trees are not pruned for better efficiency. The only parameters we have considered during bagging is the number of samples (number of trees) to include. Bagging classifier has worked well on our problem statement to classify the traffic. The Confusion matrix in Figure 5, ROC curve in Figure 6 and performance evaluation matrix in Tables 4 & 5 are evaluating the performance of Bagging Classifier.

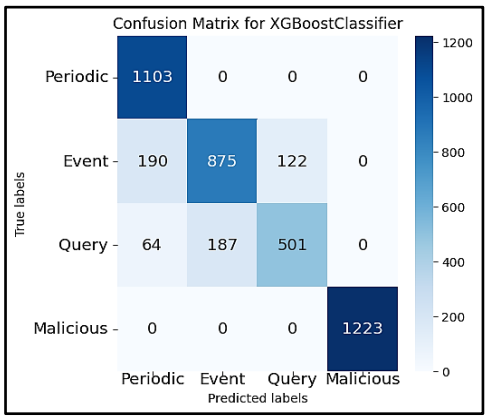

Figure 5. Confusion matrix – Bagging

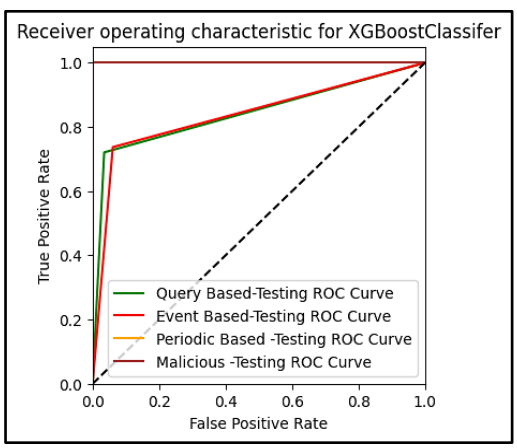

Figure 6. ROC curve for bagging classifier

3.2.3 Deep neural networks (DNN) based classification approach

For classifying IoT traffic, neural networks-based models work well with small no. of packets as well as for larger no. of packets. Our objective is to classify the incoming packets into normal (Periodic, Event based, or Query based) or malicious traffic without pre-processing the incoming packets into a particular flow. Deep neural networks have been reported as more appropriate approach for classifying larger and diverse traffic. We have also made use of deep neural networks (DNN) for classifying the traffic over IoT environment. During the training of DNN, the inputs are supplied to each Onion router as labelled data to train deep learning model. The controller deflects the malicious data after segregation of data is made using our deep learning model. The training method is using greedy layer-to-layer training after the initialization, chased by the back propagation method. First, the deep learning model with only one hidden layer is trained. Once the training with one hidden layer is attained, the training with second hidden layer will take place, and so on. At each iteration, the previous trained (n1) hidden layer of deep learning model is taken as an input and the n^{th} hidden layer is added. The back propagation is used to adjust the weights within the layers. When the training phase is over, the output is forwarded to the controller router.

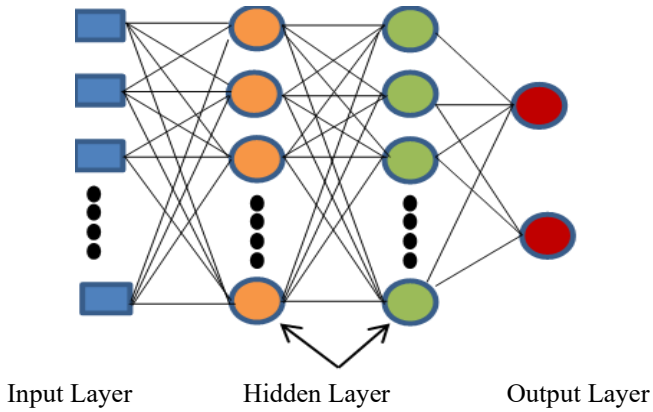

Figure 7. Deep neural network classifier

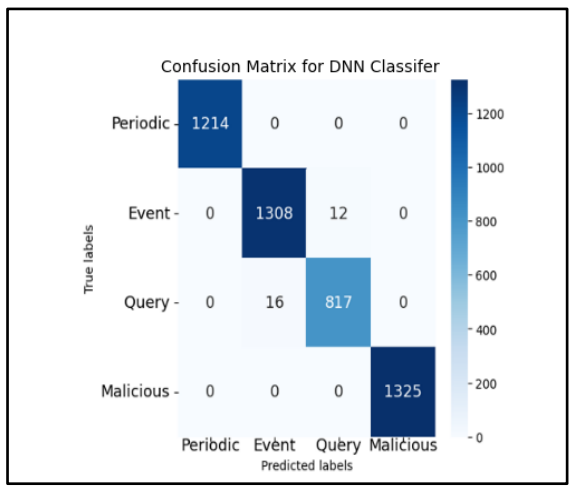

Figure 8. Confusion matrix for Deep neural network classifier

The objective is to segregate the traffic between normal and malicious traffic. Later, the malicious traffic is to be deflected by the controller and normal traffic is to be forwarded to the

intended onion routers. The deep neural networks have been trained to classify the traffic on IoT environment in a more accurate manner even if the network channel is bottled-necked with huge amount of traffic. Our model is making use of a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input. Our model is shown in Figure 7.

The evaluation of DNN based classifier is made using ROC curve (Figure 9), confusion matrix (Figure 8), and other performance parameters as mentioned in Table 3.

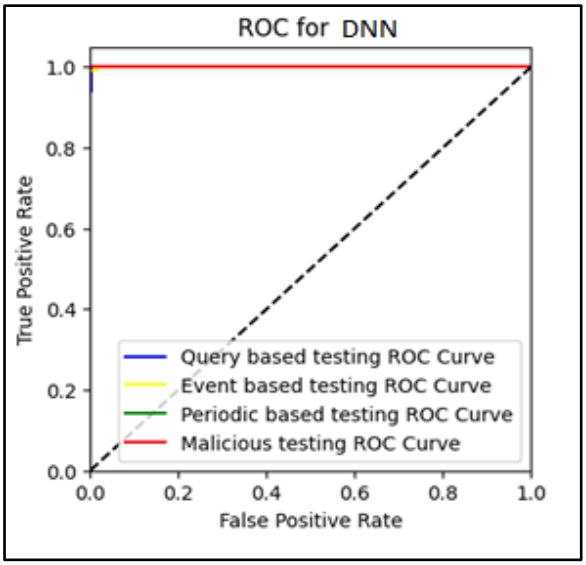

Figure 9. ROC curve for deep neural network classifier

4. RESULTS AND DISCUSSION

In this paper, machine learning and deep learning approaches are presented to identify the traffic in IoT environment. The evaluation of the performance of the algorithms is presented in Table 4 & 5. From Tables 4 & 5 it is observed that the deep learning-based classifier produces the most accurate results, bagging classifier is next to deep learning-based approach to classify the traffic on IoT environment accurately. AdaBoost is third in ranking of classifiers to produce accurate results. The accurate classification in IoT environment is required to deflect the abnormal traffic, to provide on demand services to the users, to use them. The data in IoT based networks is huge and to consume the bandwidth with normal data is essential by blocking the malicious data. The normal traffic can be prioritizing the 'event based' traffic gets more priority over other traffics in general, but the priorities can be defined and changed as per the user needs.

The machine learning module helps the router to segregate the traffic. After the classification of the traffic, abnormal traffic is deflected, and the normal traffic is channelized to next Onion router. The packets of normal traffic are forwarded to next routers to see the impact on throughput and network latency of the proposed approach. The generic approach is also employed which handles the data without any segregation and forwards the entire data (normal plus abnormal) to the next routers. It is observed that the segregation of malicious traffic from the normal traffic enhances end-to-end throughput as shown in Figure 10. Congestion generated from unwanted traffic leads to degradation of the entire network performance in real time scenario. It is observed from Figure 11 that the proposed segregation method is effective in reduction of network latency as well.

Table 4. Tabular presentation of performance evaluation parameters of classifiers

Algorithms	Training Accuracy	Test Accuracy	Training AUC score	Testing AUC Score
DNN	99.4%	99.45%	97%	98%
AdaBoost	79.44%	78.57%	80.11%	82.11%
Bagging Classifier	87.57%	86.8%	90.92%	89.45%

Table 5. Performance evaluation based on F1 Score, Precision, Recall of classifiers

Algorithms	F1 Score	Precision	Recall
DNN	0.99	0.99	0.99
AdaBoost	0.93	0.95	0.91
Bagging Classifier	0.9	0.89	0.91

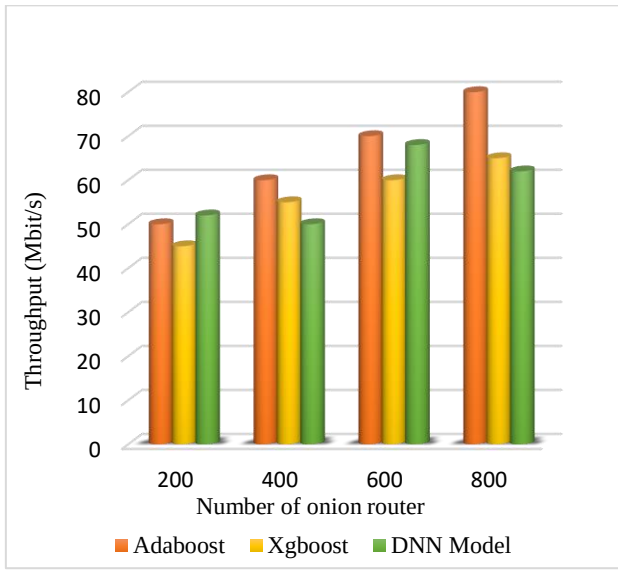

Figure 10. Throughput of the IoT network

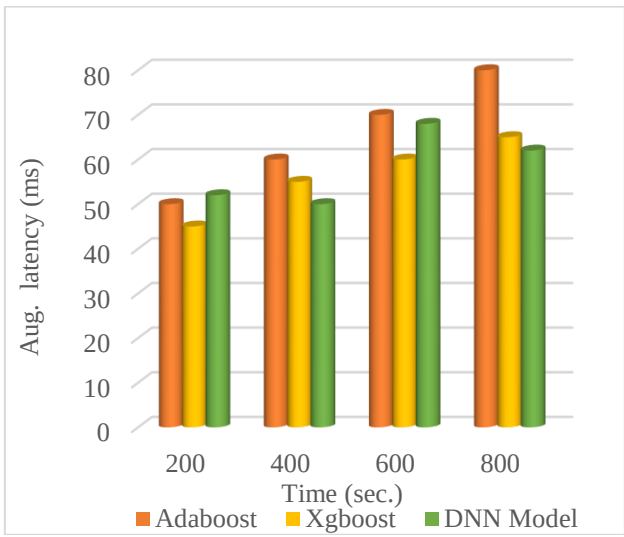

Figure 11. Average latency projected by the IoT network

The proposed work can be implemented in SDN (Software Defined Networks) enabled IoT environment and in normal IoT environment for segregation of traffic to deflect the unwanted traffic and to channelize the normal traffic. The timely classification certainly provides solution to reduce the congestion by deflecting the malicious traffic and by forwarding the normal traffic to the intended nodes.

5. CONCLUSIONS

The advanced technologies have improved the means of traffic transmission over IoT channels but the malicious data can degrade the performance of any fast network by consuming the bandwidth and other resources. In order to resolve this issue, this manuscript presented machine learning and deep learning-based methods for segregating the IoT traffic. The IoT traffic has been classified into four classes (periodic traffic, query-based traffic, malicious traffic and event-based traffic). The segregation allows the normal traffic to reach at the destined nodes in a quicker manner and deflects the abnormal traffic to free up the bandwidth consumed by the malicious traffic. The segregation of traffic and early detection of malicious traffic helps to reduce the congestion over the channels, enhances the network throughput and increases the transmission rate of the IoT traffic. The performance of machine learning and deep learning approaches has been evaluated using performance metrics such as accuracy score, F1 score, confusion matrix, precision and recall scores. Deep learning method provided the best accuracy in classification of the traffic, followed by Bagging classifier and then AdaBoost classifier for showing accuracy in identification of IoT traffic. The huge traffic in IoT environment requires dynamic algorithms for segregation of the traffic. Hence an attempt has been made in this paper to suffice the need of IoT environment by proposing machine learning based dynamic algorithms for quicker segregation of IoT traffic. The proposed approach helps to segregate the malicious traffic from the normal traffic and also classify different types of normal traffic. The scope of this work will be extended to explore the techniques to handle the malicious traffic and channelize the normal traffic in a fast manner.

REFERENCES

[1] Gubbi J., Buyya R., Marusic S., Palaniswami M. (2013) Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7): 1645-1660. https://doi.org/10.1016/j.future.2013.01.010

[2] Bashar, A. (2019). Review on sustainable green Internet of Things and its application. Journal on Sustainable Wireless Systems, 1(4): 256-264. https://doi.org/10.36548/jsws.2019.4.006

[3] Botta, W., Donato D., Persico V., Pescap A. (2016). Integration of cloud computing and internet of things: A survey. Future Generation Computer Systems, 56: 684-700. http://dx.doi.org/10.1016/j.future.2015.09.021

[4] Lee, I., Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4): 431-440. https://doi.org/10.1016/j.bushor.2015.03.008

[5] Lu, Y.F., Ling, Z., Zhu, S.H., Tang, L. (2017). SDTCP: Towards datacenter TCP congestion control with SDN for IoT applications. Sensors (Basel), 17(1): 109. https://doi.org/10.3390/s17010109

[6] Abdelmoniem, A., Bensaou, B., Abu, A. (2017). SICC: SDN-based incast congestion control for data centers. IEEE International Conference on Communications (ICC), pp. 1-6. https://doi.org/10.1109/ICC.2017.7996826

[7] Bensaou, B., Abdelmoniem, A., Abu, A. (2018). Mitigating incast-TCP congestion in data centers with SDN. Annals of Telecommunications, 73: 263-277. https://doi.org/10.1007/s12243-017-0608-1

[8] Jouet, S., Perkins, C., Pezaros, D. (2016). OTCP: SDN-managed congestion control for data center networks. IEEE/IFIP Network Operations and Management Symposium, pp. 171-179. https://doi.org/10.1109/NOMS.2016.7502810

[9] Petri, I., Zou, M., Zamani, A., Diaz-Montes, J., Rana, O., Parashar, M. (2015). Integrating software defined networks within a Cloud federation. 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 179-188. https://doi.org/10.1109/CCGrid.2015.11

[10] Mechtri, M., Houidi, I., Louati, W., Zeghlache, D. (2013). SDN for Inter Cloud Networking. IEEE SDN for Future Networks and Services (SDN4FNS), pp. 1-7.

[11] Finamore, A., Mellia, M., Meo, M., Rossi, D. (2010). KISS: Stochastic packet inspection classifier for UDP Traffic. IEEE/ACM Transactions on Networking, 18(5): 1505-1515. https://doi.org/10.1109/TNET.2010.2044046

[12] Bermolen, P., Mellia, M., Meo, M., Rossi, D., Valenti, S. (2011). Abacus: Accurate behavioral classification of P2P-TV traffic. Computer Networks, 55(6): 1394-1411. https://doi.org/10.1016/j.comnet.2010.12.004

[13] Dainotti, A., Pescape, A., Sansone, C. (2011). Early classification of network traffic through multi-classification. Traffic Monitoring and Analysis, Lecture Notes in Computer Science, pp. 122-135. https://doi.org/10.1007/978-3-642-20305-3_11

[14] Salman, O., Nunes, B., Mayoral, A., Jararweh, Y., Abdelmoniem, A. (2018). IoT survey: An SDN and fog computing perspective. Computer Networks, 143(6): 221-246. https://doi.org/10.1016/j.comnet.2018.07.020

[15] Qi, Y., Xu, L., Yang, B., Xue, Y., Li, Y. (2009). Packet classification algorithms: From theory to practice. IEEE INFOCOM, pp. 648-656. https://doi.org/10.1109/INFCOM.2009.5061972

[16] Moore, A., Zuev, D., Crogan, M. (2005). Discriminators for use in flow-based classification. Technical Report. http://qmro.qmul.ac.uk/xmlui/handle/123456789/5050.

[17] Crotti, M., Dusi, M., Gringoli, F., Salgarelli, L. (2007). Traffic classification through simple statistical fingerprinting. SIGCOMM Computer Communication Review, 37(1): 5-16. https://doi.org/10.1145/1198255.1198257

[18] Wang, X.M., Parish, D.J. (2010). Optimised Multi-stage TCP Traffic Classifier Based on Packet Size Distributions. Third International Conference on Communication Theory, Reliability, and Quality of Service, pp.98-103. https://doi.org/10.1109/CTRQ.2010.24

[19] Sen, S., Spatscheck, O., Wang, D.M. (2004). Accurate, scalable in-network identification of P2P traffic using

application signatures.13th International Conference on World Wide Web, pp. 512-521. https://doi.org/10.1145/988672.988742

[20] Papagiannaki, K., Moore, A. (2005). Toward the accurate identification of network applications. International Workshop on Passive and Active Network Measurement, pp. 41-54. https://doi.org/10.1007/978-3-540-31966-5_4

[21] Auld, T., Moore, A., Gull, S. (2007). Bayesian neural networks for internet traffic classification. IEEE Transactions on Neural Networks, 18(1): 223-239. https://doi.org/10.1109/TNN.2006.883010

[22] Sun, R.Y., Yang, B., Peng, L.Z., Chen, Z.X., Zhang, L., Jing, S. (2010). Traffic classification using probabilistic neural networks. International Conference on Natural Computation, pp. 1914-1919. https://doi.org/10.1109/ICNC.2010.5584648

[23] Draperl, G., Lashkari, A., Mamun, M., Ghorbani, A. (2016). Characterization of encrypted and VPN traffic using time-related features. 2nd International Conference on Information Systems Security and Privacy - ICISSP, pp. 407-414. https://doi.org/10.5220/0005740704070414

[24] Yamansavascilar, B., Guvensan, M., Yavuz, A., Karsligil, M. (2017). Application identification via network traffic classification. International Conference on Computing, Networking and Communications (ICNC), pp. 843-848. https://doi.org/10.1109/ICCNC.2017.7876241

[25] Hwang, R., Peng, M., Nguyen, V., Chang, Y. (2019). An LSTM-based deep learning approach for classifying malicious traffic at the packet level. Applied Sciences, 9(16): 3414. https://doi.org/10.3390/app9163414

[26] Lim, H., Kim, J., Heo, J. (2019). Packet based network traffic classification using deep learning. International Conference on Artificial Intelligence in Information and Communication, pp. 046-051. https://doi.org/10.1109/ICAIIC.2019.8669045

[27] Aceto, G., Ciuonzo, D., Montieri, A., Pescape, A. (2019). Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. IEEE Transaction Network Service Management, 16(2): 445-458. https://doi.org/10.1109/TNSM.2019.2899085

[28] Abbasi, M., Shahraki, A., Taherkordi, A. (2021). Deep learning for network traffic monitoring and analysis (NTMA): A survey. Computer Communications, 170: 19-41. https://doi.org/10.1016/j.comcom.2021.01.021

[29] Wang, M., Lu, Y., Qin, J. (2020). A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Computer and Security, 88: 101645. https://doi.org/10.1016/j.cose.2019.101645

[30] Bendiab, G., Shiaeles, S., Alruban, A., Kolokotronis, N. (2020). IoT Malware network traffic classification using visual representation and deep learning. IEEE International Conference on Network Softwarization (NetSoft), pp. 444-449. https://doi.org/10.1109/NetSoft48620.2020.9165381

[31] Breiman, L. (1996). Bagging predictors. Machine Learning, 24: 123-140. https://doi.org/10.1007/BF00058655