Sonali A. Patil
L. Arun Raj

CLASSIFICATION OF TRAFFIC OVER COLLABORATIVE IOT/CLOUD PLATFORMS USING DEEP-LEARNING RECURRENT LSTM

Abstract *The Internet of Things (IoT) and cloud-based collaborative platforms have emerged as new infrastructures over the recent decades. The classification of network traffic in terms of benign and malevolent traffic is indispensable for IoT/cloud-based collaborative platforms for optimally utilizing channel capacity for transmitting benign traffic and blocking malicious traffic. The traffic-classification mechanism should be dynamic and capable enough for classifying network traffic in a quick manner so that malevolent traffic can be identified at earlier stages and benign traffic can be speedily channelized to the destined nodes. In this paper, we present a deep-learning recurrent LSTM RNet-based technique for classifying traffic over IoT/cloud platforms using the Word2Vec approach. Machine-learning techniques (MLTs) have also been employed for comparing the performance of these techniques with the proposed LSTM RNet classification method. In the proposed research work, network traffic is classified into three classes: Tor-Normal, NonTor-Normal, and NonTor-Malicious traffic. The research outcome shows that the proposed LSTM RNet accurately classifies such traffic and also helps reduce network latency as well as enhance data transmission rates and network throughput.*

Keywords IoT, network traffic, machine learning, deep learning, classification, cloud computing

Citation Computer Science 22(3) 2021: 371–389

1. Introduction

The Internet of Things (IoT) and cloud computing have become a promising collaborative infrastructure to suffice the on-demand requirements of users. The IoT infrastructure is comprised of three main components: front-end devices with sensing capabilities, a back-end storage and computing facility, and a communication network that connects front-end to back-end for communication. As every coin is comprised of two sides, an IoT/cloud collaborative environment similarly allows for seamless connectivity; however, each connected device is at risk for vulnerable attacks. In order to suffice the on-demand resource requirements of IoT users, the IoT must depend upon cloud services. The connectivity among those IoT devices is prone to more possibilities for security threats and adversaries. Hence, there is a need to address the issues that surface in the security and privacy of IoT/cloud-based communications; only then we can take advantage of the enormous benefits brought about by an IoT/cloud collaborative environment.

The IoT connects a huge number of diverse devices that are heterogeneous in nature by using wired or wireless communication [14]. The devices in the IoT environment are highly mobile and cover a wide geographical area while moving from one region to another [6]. Hence, with the advent of the IoT, different types of wireless technologies have been researched and employed to provide seamless services to IoT users. However, the IoT has transformed the conventional way of connectivity into a high-tech connectivity, where everything can be connected anytime and anywhere; however, there is a huge risk involved for connecting devices and users. There are more possibilities for adversaries to attack IoT devices. There has been a rapid upsurge in IoT traffic, and it has become fairly challenging to detect and prevent network abuse. Along with their benefits of enormous connectivity and usability, IoT devices also make individuals and organizations more vulnerable. The use of heterogeneous communication technologies has given rise to various critical issues such as traffic load balancing, traffic channelization, throughout, responsiveness, space sharing among devices, and so forth [17]. The newer concept of software-defined networks holds the capability of more scalable network architectures, which are sorely needed in the IoT environment [18]. The IoT uses newer technologies to make networks more scalable and secure in order to fulfill the needs of IoT users [24], [28].

It is indeed a difficult task to protect network traffic in the IoT environment, where everything is connected seamlessly and multiple protocols are involved in the smooth functioning of the IoT. This requires an inclusive approach for detecting malicious traffic in the flow and deflect this abnormal traffic by segregating it on time. Attacks from malicious traffic are increasing day by day; for example, a Mirai-based DDoS attack impacted major sites such as Amazon, AirBnB, PayPal, Netflix, Visa, and so on.

Several researchers have presented studies in which software-defined networks control the traffic over the IoT environment [1, 2, 15, 19, 22], but the use of machine-learning (ML) and deep-learning (DL) methods in the Internet of Things is still to be researched more in order to classify the traffic, segregate malicious traffic from benign traffic, and predict the traffic load on channels for the optimal utilization of channels. The use of ML methods can improve the performance of IoT platforms in terms of reducing congestion, enhancing the throughput, and optimizing the utilization of the bandwidth. Hence, dynamic techniques that are based on ML and DL are presented in this paper to classify traffic in order to segregate malicious traffic from benign traffic in Tor- and NonTor-based IoT/cloud platforms. This approach eventually improves the network throughout and minimizes network congestion by identifying unwanted traffic.

Artificial intelligence (AI) and machine-learning (ML) methods are able to provide solutions for complex and dynamic problems. AI has transformed the conventional techniques to connect things on the Internet. Nowadays, AI and ML techniques have had a tremendous impact on IoT-enabled sectors. These techniques have the ability to gain knowledge automatically and improve upon previous solutions. In the IoT, devices can join and leave anytime due to their dynamic nature; it is mandatory to devise a mechanism that is suitable for coping up with the dynamic nature of the IoT environment. It is very difficult to identify IoT traffic and ascertain the load of dynamic and enormous traffic. The IoT channel could be bottle-necked with malicious or unwanted data; therefore, it is the need of the hour to classify IoT traffic in a dynamic way and predict channel load to better utilize the IoT network. Conventional techniques are not capable enough to predict loads in such a dynamic environment nor classify traffic in an efficient way in order to enhance the network throughput. Hence, we are proposing a newer technique that is based on deep learning to classify the network traffic in IoT/cloud collaborative platforms.

This paper is structured in four segments. The first segment offers information on the background details of an IoT/cloud-based collaborative platform; it also provides information on the benefits of using machine learning-based techniques in the collaborative IoT/cloud environment. The second part of this paper provides a detailed study of the existing research work that is related to our problem statement. The third section provides a detailed explanation of the proposed techniques for classifying traffic. The last section of the paper concludes our research work and also provides future directions.

2. State of the art

Numerous approaches and techniques have been presented by researchers to classify network traffic [5, 11, 13, 16, 25], but the Internet of Things requires newer approaches for handling the dynamic environment to classify IoT traffic. We have surveyed existing techniques in this section to provide insights into the research contributions made by others in the area of our research work.

Port-based techniques: Traffic identification with a port number is the oldest technique for classifying traffic. Port-based identifiers use TCP or UDP packet headers to attain information about port numbers. The comparison is done by matching the assigned TCP/UDP with the extracted port numbers. It is the fastest and oldest method for traffic identification [20, 23]. This method has certain limitations, as applications like Napster and Kazaa do not register their port numbers. Such an application may access other port numbers to avoid the access-control restrictions imposed by operating systems. In other cases, server port numbers are allocated dynamically. IoT devices can transmit an enormous amount of data anytime; this method is not suitable for IoT-based applications.

Payload-based classification: Many applications use the session and application information of a packet rather than a port number [18]; these techniques analyze the available information in the application layer payload of a packet. In [26], a method for utilizing application-level signatures was presented for identifying the traffic of P2P application by looking into traces of packets. Later, the identified signatures are used to develop online filters. High-speed network links can be efficiently tracked by using these filters. In [21], a combination of payload and port-based techniques are presented to classify network applications. The procedure begins with identifying a port number and then locating the signature. In the case of the absence of a signature, the packet is examined for specified protocols. This technique allows for the identification of errors (if any). In [3], the authors proposed a deep packet inspection (DPI) system that could examine an encrypted payload; however, it can only process HTTPS traffic. These approaches avoid a dependency on port numbers, but these techniques are not sufficient enough when dealing with encrypted traffic.

Statistical techniques: In [10], the authors used a probability density function for protocol fingerprints; this function considered inter-arrival time and threshold time for the normalization of the packets.Groups of protocols such as HTTP, POP3, and SMTP were considered for the research study. The accuracy achieved by their proposed work was 91%. Wang and Parish [30] presented a similar method where they used multiple classifiers for the identification of network traffic. The experimental outcome achieved an 87% accuracy. Protocols such as IMAP, FTP, TELNET, and TCP were considered for the experimental study.

Machine learning-based approaches: In [32], the authors proposed an end-to-end traffic-classification mechanism in the IoT which made use of deep learning-based capsule networks for forming an integrated classification model which extract features and classifies the traffic into classes. In [29], the authors applied an encrypted traffic classification in their work. The validation of the method is made on the basis of public non-VPN and ISCX VPN traffic datasets to achieve better accuracy than the techniques that existed at that time. In [4], the authors applied a BNN (Bayesian neural network) to classify P2P-based protocols; they achieved good accuracy. In [12], the authors identified VPN-based traffic and classified the traffic into different classes by using k-NN and C4.5 ML classifiers. In [31], the authors attempted to identify the traffic (such as Facebook, Twitter and Skype)at application layer and aligned the applications by using Random Forest, J48, Bayes Net, and k-NN. In [27], the authors collected and synthesized network traffic traces from diverse IoT devices such as cameras, appliances, lights, and health-monitors. Then, they analyzed the traces by classifying them on the basis of statistical attributes such as burstiness in the data rates, signaling patterns, and activity cycles. Their classification approach distinguished IoT traffic from non-IoT traffic and achieved a 95% accuracy.

Many research endeavors were made by researchers to classify network traffic, but traffic segregation on the Internet of Things is still an unexplored area where traffic is huge and dynamic. It is inappropriate to use traditional techniques to segregate IoT-based traffic and protect the data from unauthorized access. It is the need of the hour to put research endeavors toward newer AI-based intelligent techniques that are suitable for the IoT environment. Hence, we propose deep-learning and machine-learning methods for classifying IoT-based traffic.

3. Proposed work

IoT traffic is growing rapidly at the present time due to the emergence of newer technologies and applications. Machine learning can certainly help classify traffic and predict traffic load in order to provide seamless services to IoT users. Most of the traffic is to be channelized on the cloud to exploit cloud services for fulfilling the resource or service requirements of users. We are proposing deep-learning and machine-learning techniques in this paper to accurately classify network traffic. The traffic in an IoT/cloud collaborative environment is categorized into three classes: Tor-Normal, NonTor (NT)-Normal, and NonTor (NT)-Malicious. Tor-based tiny networks allow users to exploit Internet services in a secure way over IoT/cloud platforms by using a special line of Onion routers that are integrated with secured protocols. NonTor traffic is comprised of both benign and malicious data. The malicious data that was considered for our research study was comprised of non-human, distributed denial-of-service (DDoS), and MCA (malicious cyber attacker) traffic. Non-human traffic is the IoT traffic that is generated by scripts, bots, and implicitly programmed codes for surfing the web without any human intervention. In an IoT/cloud environment, malicious traffic also refers to URLs that are used by MCAs to host malware, viruses, or phishing scams that can potentially harm IoT networks and devices. A DDoS attack is also considered to be a malicious attempt to interrupt the services of network servers by overfeeding the servers with a huge amount of false traffic from thousands of source locations. Malicious traffic is to be identified at the earliest possible time and blocked or deflected to free up the channel for normal traffic.

Our motive is to classify traffic using machine-learning and our proposed deep-learning techniques to enhance the security and integrity of IoT data. The contribution of the paper has been summarized below and also depicted in Figure 1.

1. At the beginning, we took online data that was available at 'Amazon Cloud' for training our machine learning-based system. A standard dataset is considered to train the ML- and DL-based models for classifying the traffic into Tor-based benign, NonTor-based benign, or NonTor-based malicious traffic classes.

2. We extracted the parameters to train the data and test the proposed machine learning-based model.
3. The pre-processing of the data was done to make the data usable for ML-based classifiers.
4. We also made use of data visualizations to provide a better understanding to the readers regarding the variables that were considered for the study.
5. Next, we proposed an LSTM (long short-term memory) recurrent deep-learning model for classifying network traffic. In the first step, each packet was transformed into a sequence of n-grams. Then, the n-grams were consumed to create a dictionary where the n-grams were mapped to integer identifiers. Then, the Word2Vec technique was employed to generate word embeddings from the n-grams. These word embeddings assisted in the creation of feature vectors for the LSTM RNet classifier to accurately classify the traffic.
6. We also applied machine learning-based algorithms to train the model for classifying IoT traffic.
7. Then, the evaluation of the performance of the proposed technique with ML-based methods was done using a confusion matrix, accuracy score, F1 score, precision, and sensitivity score.

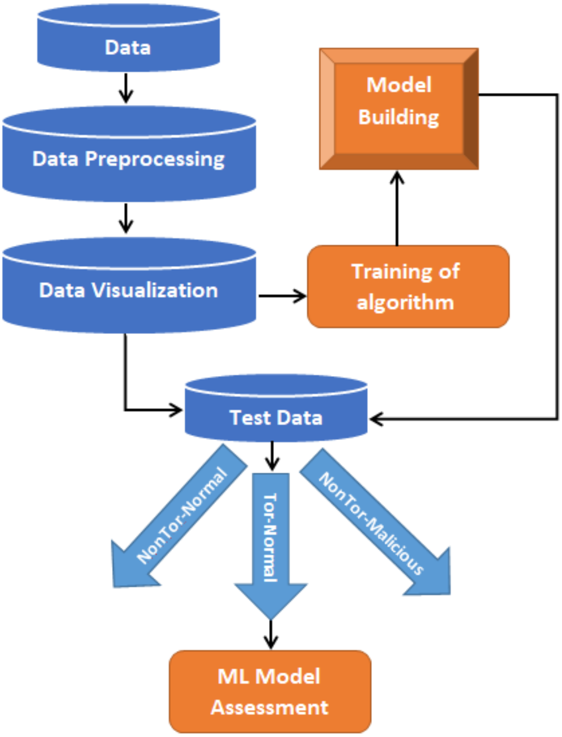


Figure 1. Machine learning-based model for classifying traffic

3.1. Dataset parameters considered for ML-based classifiers

The parameters that were considered for this study are shown in Table 1.

Table 1

Attributes considered for training dataset

Field Title	Description
Source IP Add	Source IP of flow
Traffic-Type	Depicts whether traffic is Tor or NonTor
Source Port No.	Source port number
Dest IP Add	Destination IP of packet
Dest Port No.	Destination port number of packet
Protocol No.	Transport layer protocol number identifier (i.e., TCP = 6, UDP = 17).
Packet Id	Header information of packet
Flow Duration	Duration of flow
Flow Bytes	Number of bytes per second in flow
Flow Packets	Number of packets per second in flow
Flow Mean IAT	Mean value of inter-arrival time (IAT) of flow (bi-directional)
Flow Std Dev IAT	Standard deviation of IAT of flow (bi-directional)
Flow Max IAT	Maximum value of IAT of flow (bi-directional)
Flow Min IAT	Minimum value of IAT of flow (bi-directional)
Fwd Mean IAT	Mean of IAT in forward direction
Fwd Std IAT	Standard inter-arrival time in forward direction
Fwd Max IAT	Maximum value of IAT in forward direction
Fwd Min IAT	Minimum value of IAT in forward direction
Bwd Mean IAT	Mean of IAT in backward direction
Bwd Std IAT	Standard inter-arrival time in backward direction
Bwd Max IAT	Maximum IAT in backward direction
Bwd Min IAT	Minimum IAT in backward direction
Active Mean	Mean time of active flow before flow becomes idle
Active Std	Standard deviation time of active flow before flow becomes idle
Active Max	Maximum time of active flow before flow becomes idle
Active Min	Minimum time flow was active before becoming idle
Idle Mean	Mean time of active flow before flow becomes idle
Idle Std	Standard deviation time flow was idle before becoming active
Idle Max	Maximum time flow was idle before becoming active
Idle Min	Minimum time flow was idle before becoming active

3.2. Classification of network traffic over IoT/cloud environment

This subsection gives detailed information about our proposed classification system. We have proposed an LSTM RNet-based deep classifier for classifying network traffic into three classes: i.e., Tor-based benign traffic, NonTor-based benign traffic, and NonTor-based malicious traffic. We have made a comparison of the LSTM RNet method with a multi-variate SVM (support vector machine) and hybrid random forest classifier.

LSTM RNet: The recurrent network (RNet) organizes hidden state vectors h_t^d in a 2D matrix with time step index t ranging from $t = 1 \ldots T$, and $d = 1 \ldots D$ is the depth. The bottom row of vectors $h_t^0 = a_t$ at a depth of zero carries input vector a_t, and each vector in the uppermost h_t^D row is used to forecast an output vector o_t. The rest of the intermittent vectors h_t^d are calculated with a recurrence formula that is based on h_t^d and h_t^{d-1}. Each output o_t at time stamp t through hidden vectors becomes a function of all of the input vectors up to $t\{a_1, \ldots, a_t\}$. The mathematical modulation of recurrence $\left\{h_t^d, \ldots, h_t^{d-1} \rightarrow h_t^d\right\}$ changes precisely from model to model.

RNet is comprised of three parameter matrices (P, W, V) with activation functions, where P represents the input hidden matrix, W represents the hidden-hidden matrix, and R represents the hidden-output matrix. h_t represents the hidden states, a_t represents the input vector (as shown in Eq. 1), and $\widehat{o}$ is an output vector (as shown in Eq. 2). The tangent hyperbolic function is represented by $tanh(\cdot)$ in RNet. $\gamma(\cdot)$ is an output transformation function that can be selected for any kind of task or target data. This feature enables RNet to model anything without any constraints.

$$h_t = tanh\left(Pa_t + Wh_{t-1}\right) \tag{1}$$

$$\widehat{o} = \gamma(Vh_t) \tag{2}$$

We present an LSTM-based RNet approach for traffic classification that uses packet information in the flow. The proposed classification mechanism separates each incoming packet into malicious and benign traffic. The packets are considered in the CBOR (concise binary object representation) format. CBOR is a kind of binary data serialization format that is loosely based on JSON. CBOR permits the transmission of data objects that contain name-value pairs in a concise manner. Tor-based networks use Onion routers with embedded security software and protocols to detect and deflect malicious traffic; however, when traffic moves from a source to a destination, it may go through Tor as well as NonTor networks. Hence, classifying the traffic is vital for maintaining the integrity of the data. Tor transmits IoT traffic through an overlay-based network that is comprised of more than 7,000 relays to conceal the location of the sender and hide the information from anonymous users who perform traffic analyses or network surveillance. NonTor traffic may contain malicious traffic along with normal traffic. Hence, this research work focuses on classifying the traffic into three categories (Tor-Normal, NT [NonTor]-Normal, and NT [NonTor]-Malicious) without much pre-processing done to the packets.

To attain this goal, the following steps have been considered:

1. All packets are transformed into n-grams;
2. A dictionary is generated to add the words and then the n-grams are transformed into numerical or integer identifiers;
3. The vectors of the integer vectors are prepared and passed to an embedding layer of LSTM RNet;

4. The Word2Vec method is utilized for embedding the layer where the n-gram embeddings are aggregated using the CBOW (common bag of words) model of Word2Vec;
5. Next, the aggregated n-gram embeddings are utilized to create feature vectors;
6. Finally, the feature vectors derived from the embeddings assist in accurately classifying the traffic.

The order of the columns in each packet assists in resembling the grammar rules that are conclusive in constructing sentence patterns for NT-Malicious and Tor- or NonTor-based benign traffic. The Word2Vec-based approach can considerably accelerate the classification of IoT traffic, as the characteristics of the packets can ultimately divulge whether the flow contains malicious or benign traffic. After applying the Wor2Vec technique, the vector encodes the words using the CBOW model. The CBOW model considers each word as an input and attempts to forecast the word that corresponds to the input context. The input is encoded as a vector of size V. The hidden layer is comprised of N neurons, and the output is a V-length vector, represented as softargmax values.

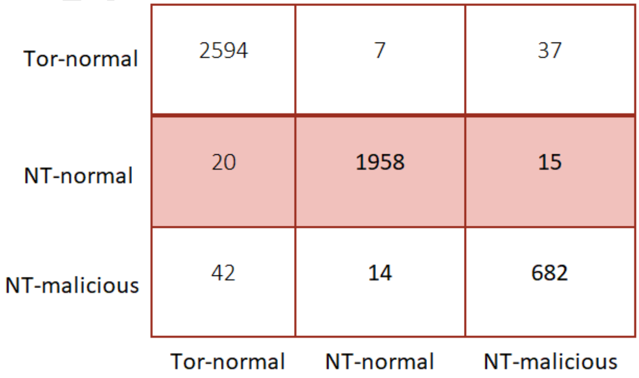
The pre-processing of the data is necessary to make the data appropriate for the classification algorithms. The parsing of the data packet is performed, and then a word translation is done. The converted dataset of translated words is represented in the form of integer numbers. The converted dataset is the segregated into two parts; i.e., training and testing (at a ratio of 7:3). Out of the available data,seventy percent data is used for training and remaining data is used for testing purposes. The first thing in LSTM RNet is to make a decision about which information is to be removed from the cell state. The next thing is to decide which information is to be stored in the cell state. A relu layer generates a vector of newer candidate values $(\widehat{c_t})$, which can be added to the existing state. Input gates i_t determine the value of new cell states $\widehat{c_t}$ to be concatenated with the existing cell states. Finally, a decision must be made regarding the desired output. The output classes will be based on the filtered version.

The training stage begins with the training data and executing the LSTM RNet three-layered model. The dropout rate can also be flexibly adjusted (this was tuned to 0.3 in our case study). The loss function uses categorical cross entropy. We made use of an Adam optimizer to enhance the learning process of our training model. A Softargmax-based dense output layer (also known as Softmax) is added to the model. Finally, the proposed model was tested on the testing dataset, and the efficacy of the model was judged on the basis of the accuracy score, recall score, F1-score, precision, and loss. The proposed algorithm is presented in Table 2. Our proposed LSTM RNet model was implemented using Keras, and the output was produced by Softargmax/Softmax.

The outcome/result is a powerful predictive-modeling algorithm. The evaluation of the proposed algorithm is represented using a confusion matrix (as shown in Fig. 2) and performance matrix (Table 3).

Table 2
Traffic-classification algorithm for IoT/cloud platforms

```
Algorithm 1: Classification algorithm by LSTM RNet
Input: Sequential supply of data packets from flow
Output: classification of packets
-----------------------------------------------------------------
begin
    $n - grams$ = Each packet is transformed into sequence of $n - grams$
    n-gram-translation = null; # where n-grams are mapped to integer identifiers
    dictionary = array () # Feed integer identifiers into index array
       while true do
          Parsing of packets is performed
          Each byte of packet data is parsed as sequence of n-grams
       for n = 1; n < wordcount; n++ do
          if word is available in dictionary dictionary (words[n]), then
          index = dictionary (words[n]) # Fetching index of words
       else
          dictionary[] = words[n]; # Adding new word to dictionary
          index = dictionary (words[n])
          end if
          Concatenate (wordtranslation,index)
       end for
       Embeddings () # Word2Vec technique is employed to generate embeddings from n-grams
       Feature vectors () # Embeddings are used with integer vectors to form feature-vectors
    end while
       Distribute training and testing data a ratio of 7:3
       Train and validate model
       Feature vector supplies packet data in deep-learning understandable format
       Input Feature vector to LSTM RNet and use relu function (alternative of tanh)
       Dropout
       Feedfwd to second layer of LSTM RNet
       Dropout
       Feedforward to third layer of LSTM RNet
       Dropout
       Formulate input for small-batch (say, 100 packets)
       Adam optimizer is applied for fast learning of model;
       Use Softargmax function for output of model
       Apply categorical cross-entropy as loss function ;
    for (epoch = 1; epoch < 300; epoch++) do
    Evaluate loss and evaluate accuracy
    end for
end
```



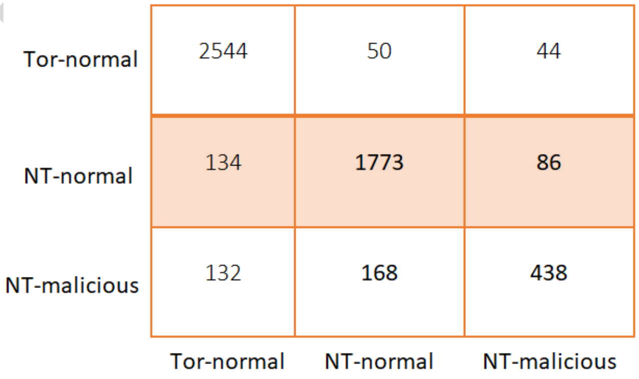
	Tor-normal	NT-normal	NT-malicious
Tor-normal	2594	7	37
NT-normal	20	1958	15
NT-malicious	42	14	682

Figure 2. Confusion matrix for LSTM-RNet classifier

The following are the observations from Figure 2:

- In the testing dataset, there were a total of 2,638 records with target variable 'Tor-Normal,' Of these, 2,599 were correctly classified, and 39 were misclassified;
- In the testing dataset, there were a total of 1,993 records with target variable 'NT-Normal.' Of these, 1,852 records were correctly classified, and the remaining 141 were misclassified;
- The testing dataset contained 738 records with 'NT-Malicious'; 655 of these were recognized correctly, and 83 were classified incorrectly.

MV-SVM Classifier: The first algorithm that we applied for traffic classification is a multi-variate (MV) support vector machine (SVM) [8]. SVM is a a supervised ML classifier [9]. In MV-SVM, each data item is referred to as a point in n-dimensional space with a value of each feature. Then, the classification is performed by finding the hyper-plane that is able to effectively differentiate the classes. Multi-class SVMs were implemented in our research work by combining several binary SVMs. Our objective was to test the robustness of various kind of kernels for the multi-class SVM classifier and to find a plane that had the maximum margin from the hyper-plane. Several kernels (namely, linear, rbf, poly, and sigmoid) were tried; the accuracy obtained by the linear kernel was 81%, the poly kernel was 71%, the rbf kernel was 83%, and the sigmoid kernel was 79%. On the basis of each kernel, a hyper-plane is decided. The support vectors impact the orientation as well as the position of the hyper-plane; basically, they represent data points that are nearer to the hyper-plane. In our case study, the multi-class SVM classifies the data into three classes. The assessment of MV-SVM is made using a confusion matrix (as depicted in Fig. 3) as well as other performance matrices (as depicted in Table 3).



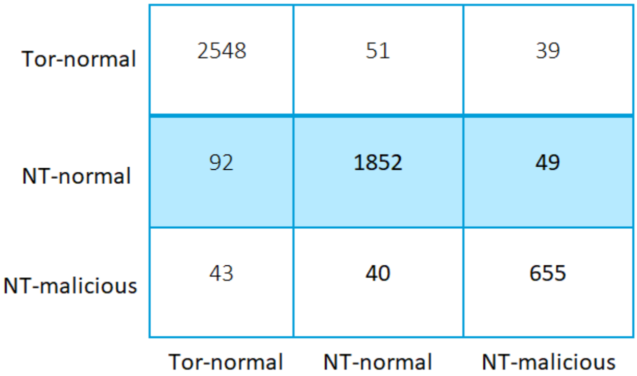
	Tor-normal	NT-normal	NT-malicious
Tor-normal	2544	50	44
NT-normal	134	**1773**	**86**
NT-malicious	132	**168**	**438**

Figure 3. Confusion matrix for MV-SVM classifier

The observations from the confusion matrix (Fig. 3) are as follows:

- The testing dataset contained 2,638 records as 'Tor-Normal.' Of these, 50 records were misclassified as NT-Normal, and 44 records were misclassified as NT-Malicious;
- There were a total of 1,993 'NT-Normal' records; 1,773 records were classified accurately, whereas 220 records were misclassified;
- Out of the 738 'NT-Malicious' records, 438 were predicted correctly, 168 were misclassified as NT-Normal, and 132 were misclassified as NT-Malicious.

Random forest-based classifier: RF is one of the most powerful ML classifiers; it is an ensembling algorithm that is used for classification as well as prediction [7]. The classifier uses bootstrap aggregation and is a powerful statistical technique for estimating a quantity from a given dataset. It attempts to deploy similar learners on small samples and then takes a mean or aggregated value of all of the results. An ensemble method in the RF classifier aggregates the predictions from multiple ML algorithms all together for making more-accurate predictions. Combining the predictions from diverse algorithms works better if the outcome from the sub-models are weakly correlated or uncorrelated. In our research work, we created random sub-samples of the dataset; then, we ascertained the mean of each sub-sample. Next, we aggregated the collected means and projected the result as a predicted mean for the data. During the formation of the decision trees, the evaluation of the error function was done for a variable at each split point. These drops in error were averaged across all of the decision trees. The greater the drop when a variable was chosen, the greater the importance. The individual decision trees were grown deep, and the trees were not pruned for better efficiency. The only parameters we considered during the classification using the RF classifier was the number of trees to include for the sub-samples. The RF classifier worked well on our problem statement to classify IoT traffic. The performance of the RF classifier was represented using the confusion matrix (as depicted in Fig. 4) as well as in the evaluation matrix (as shown in Table 3).



	Tor-normal	NT-normal	NT-malicious
Tor-normal	2548	51	39
NT-normal	92	1852	49
NT-malicious	43	40	655

Figure 4. Confusion matrix for random-forest classifier

The observations from the confusion matrix (Fig. 4) are as follows:

- Out of the 2,638 'Tor-Normal' records in the testing dataset, 2,548 were classified correctly, 51 were misclassified as NT-Normal, and 39 were misclassified as NT-Malicious;
- Out of the 1,993 'NT-Normal' records, 1,852 were correctly classified, and the remaining 141 records were misclassified;
- Out of the 738 'NT-Malicious' records in the testing dataset, 655 records were correctly classified, and the remaining 83 were classified incorrectly.

4. Results and discussion

We used three different methods for classifying the traffic on an IoT/cloud collaborative environment to segregate the abnormal traffic from the benign traffic. A performance evaluation of the ML and DL algorithms is shown in Table 3.

Table 3
Matrix presenting performance of ML-based classifiers

Algorithms	Training Accuracy	Test Accuracy	F1 Score	Precision	Sensitivity
SVM Classifier	73.08%	72.57%	82.08%	92%	0.75
Random Forest	83.19%	82.96%	84.17%	94.11%	0.97
LSTM RNet Classifier	96.8%	96.70%	98.02%	97.45%	0.98

The results presented in Table 3 reveal that the LSTM RNet-based classifier produces the most accurate results; the RF-based classifier is second to the LSTM RNet approach in accurately classifying the traffic on an IoT/cloud collaborative environment. An accurate classification over the Tor- and NonTor-based IoT environments is required to provide uninterrupted services to IoT users. The data in an IoT/cloud-based collaborative environment is huge; deflecting the unwanted traffic at the earliest stages is very important for the network's health. Normal traffic is also categorized as Tor and NonTor traffic. Generally, Tor-based traffic is considered to be secured traffic because it uses Onion routers with embedded protocols and software to provide safety for normal data. NonTor traffic is comprised of both normal and abnormal (or malicious) traffic. Hence, we classified the traffic into three classes: Tor-Normal, NT-Normal, and NT-Malicious. Tor-based traffic is generally assigned a higher priority than NonTor traffic, but the priorities can be defined and changed as per the user's needs. The existing traffic-classification techniques are insufficient to satisfy the on-demand needs of IoT users by categorizing the traffic accurately. The ML- and DL-based techniques are exploited to categorize the traffic at early stages, which eventually enhances the transmission rate and throughput. It also enhances the security of the data by deflecting the unwanted data. The proposed ML- and DL-based techniques reduce the network latency during transmission of data (as depicted in Fig. 5).

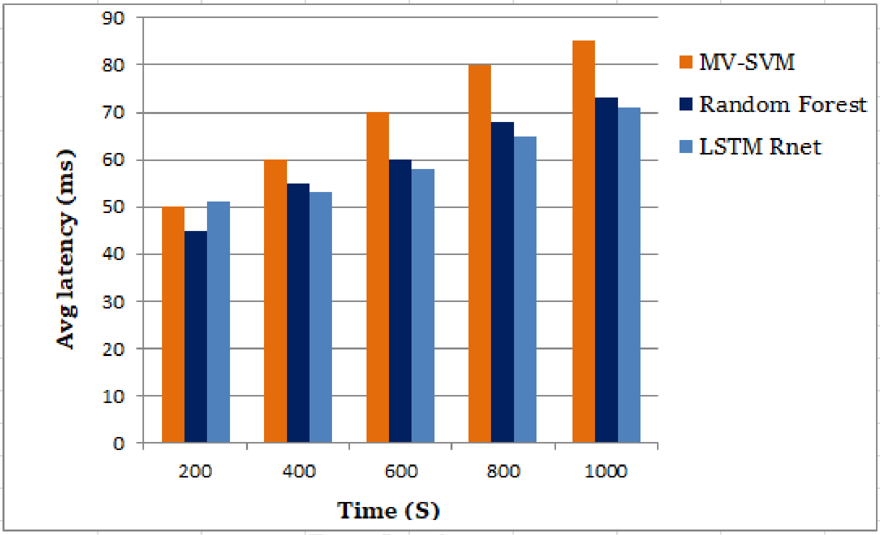


Figure 5. Average latency achieved by ML- and DL-based models

The readings were interpreted iteratively after each 200 seconds to ascertain the network's latency. High network latency becomes problematic, as IoT traffic begins to grow during peak hours. The latency issue is significant to consider while designing any model, as commercial businesses are connected to cloud servers. Normal users also exploit cloud servers for diverse applications, and delays in responses from cloud servers can hamper one's business; also, they hamper the quality of services to IoT users. Hence, our motive is to classify the traffic and deflect the malicious traffic after its early identification to improve the latency rate. Malicious traffic is blocked as soon as it is identified using our proposed mechanism. We are focused on traffic classification and are not going deeper on the deflection of malicious traffic, as this is beyond the scope of the paper. The timely segregation of traffic can certainly reduce network congestion and also assist in forwarding normal traffic to its intended nodes.

Our proposed techniques also improves the throughput of a network. When malicious traffic is identified and blocked from traveling further through the channels, it eventually decreases the bandwidth consumption of the channels and improves the transmission rate of the IoT data (as shown in Fig. 6). It also improves network latency and throughput (as presented in Fig. 7).

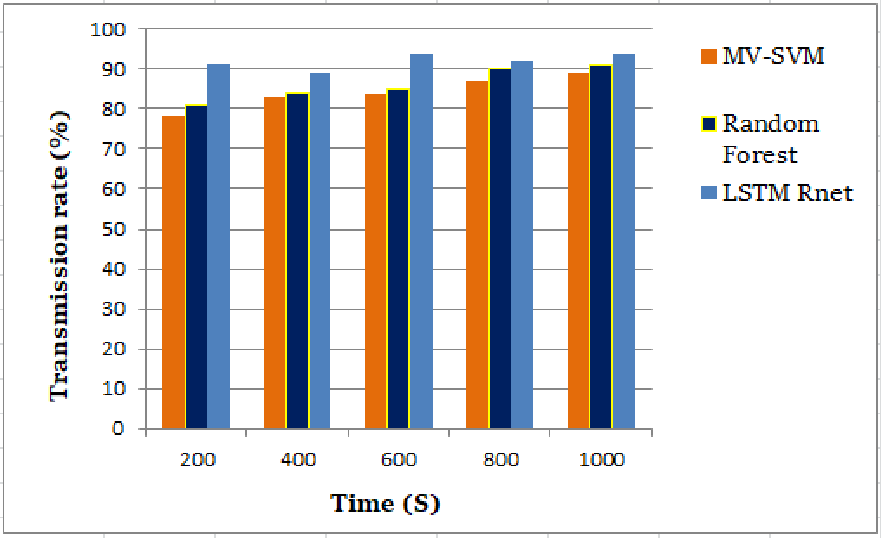


Figure 6. Transmission rate achieved by ML- and DL-based models

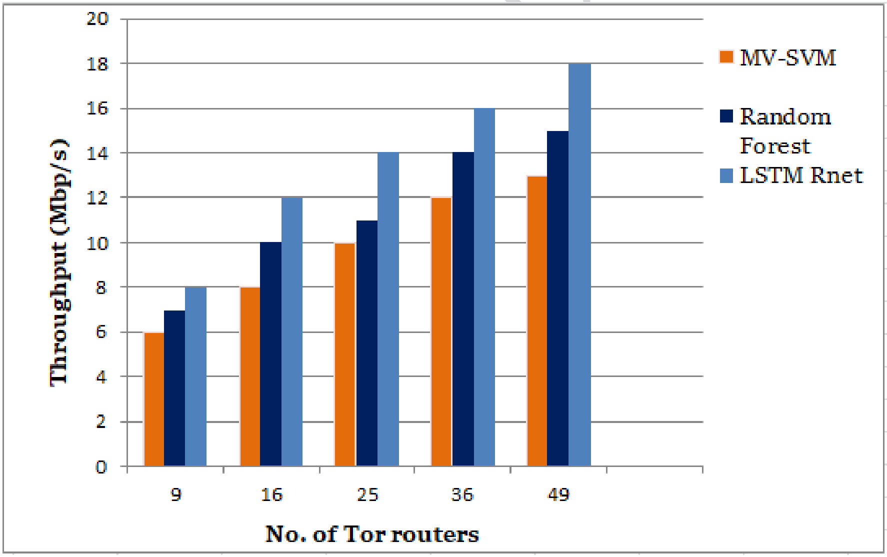


Figure 7. Throughput achieved by ML- and DL-based models

5. Conclusion

The growth in sensor based data and quick response requirements of the collaborative platforms of IoT and cloud have created a demand for high-speed transmission of network traffic. However, IoT/cloud-based technologies are attempting to improve user services tremendously, but deep LSTM RNet- and ML-based techniques are to be employed to meet the high computational needs of users over IoT/cloud collaborative platforms. Network traffic has been classified into three classes: Tor-Normal, NT-Normal, and NT-Malicious. The motive behind the classification of traffic is to segregate normal traffic from malicious traffic so that the malicious traffic can be blocked at its earliest occurrence to reduce congestion on a channel and assure that the normal traffic is forwarded to the intended nodes. Hence, an idea has been proposed to utilize LSTM RNet, which is capable of extracting packet information and identifying traffic accurately in a quick manner. In future work, we will research mechanisms that deal with malicious data and channelize normal traffic according to the priorities of the traffic defined by the underlying protocols.

Acknowledgements

We would like to express our gratitude to B.S. Abdur Crescent Institute of Science & Technology, Tamil Nadu, India for allowing us to conduct this research study and for providing the necessary resources to complete this research work. The proposed IoT traffic detector will be available to third-party users after obtaining the appropriate licenses.

References

[1] Abdelmoniem A., Bensaou B., Abu A.A.: Mitigating incast-TCP congestion in data centers with SDN, *Mitigating incast-TCP congestion in data centers with SDN, Annals of Telecommunications*, vol. 73(3), pp. 263–277, 2018.

[2] Abdelmoniem A.M., Bensaou B., Abu A.J.: SICC: SDN-based incast congestion control for data centers. In: *SICC: SDN-based incast congestion control for data centers, 2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[3] Akram H., Aniruddha G., Pascal B., C. S.D., Thierry G.: Software-defined networking: challenges and research opportunities for future internet, *Comput Networks*, vol. 75, pp. 453–471, 2014.

[4] Auld T., Moore A.W., Gull S.F.: Bayesian Neural Networks for Internet Traffic Classification, *Bayesian Neural Networks for Internet Traffic Classification, IEEE Transactions on Neural Networks*, vol. 18(1), pp. 223–239, 2007.

[5] Bermolen P., Mellia M., Meo M., Rossi D., Valenti S.: Abacus: Accurate behavioral classification of P2P-TV traffic, *Computer Networks*, vol. 55(6), pp. 1394 – 1411, 2011.

[6] Botta A., de Donato W., Persico V., Pescapé A.: Integration of Cloud computing and Internet of Things: A survey, *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016. doi: 10.1016/j.future.2015.09.021.

[7] Breiman L.: Bagging Predictors, *Machine Learning*, vol. 24(2), pp. 123–140, 1996.

[8] Chamasemani F.F., Singh Y.P.: Multi-class Support Vector Machine (SVM) Classifiers – An Application in Hypothyroid Detection and Classification. In: *2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 351–356, 2011.

[9] Cortes C., Vapnik V.: Support Vector Networks, *Machine Learning*, vol. 20(3), pp. 273–297, 1995.

[10] Crotti M., Dusi M., Gringoli F., Salgarelli L.: Traffic Classification Through Simple Statistical Fingerprinting, *SIGCOMM Comput Commun Rev*, vol. 37(1), pp. 5–16, 2007.

[11] Dainotti A., Pescapé A., Sansone C.: Early classification of network traffic through multi-classification, *Traffic Monitoring and Analysis, Lecture Notes in Computer Science*, vol. 6613, pp. 122–135, 2011.

[12] Draper-Gil G., Lashkari A.H., Islam-Mamun M.S., Ghorbani A.A.: Characterization of Encrypted and VPN Traffic using Time-related Features. In: *Characterization of Encrypted and VPN Traffic using Time-related Features, ICISSP*, 2016.

[13] Finamore A., Mellia M., Meo M., Rossi D.: KISS: Stochastic Packet Inspection Classifier for UDP Traffic, *IEEE/ACM Transactions on Networking*, vol. 18(5), pp. 1505–1515, 2010.

[14] Gubbi J., Buyya R., Marusic S., Palaniswami M.: Int. of Things (IoT): A vision, arch. elements, and fut. directions, *Future Generation Computer Systems, Elsevier*, vol. 29(7), pp. 1645 – 1660, 2013.

[15] Jouet S., Perkins C., Pezaros D.: OTCP: SDN-managed congestion control for data center networks. In: *OTCP: SDN-managed congestion control for data center networks, NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 171–179, 2016.

[16] Kim H., Claffy K., Fomenkov M., Barman D., Faloutsos M., Lee K.: Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices. In: *Proceedings of the 2008 ACM CoNEXT Conference*, pp. 11:1–11:12, CoNEXT '08, ACM, 2008.

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

[18] Li X., Freedman M.J.: Scaling IP Multicast on Datacenter Topologies. In: *Scaling IP Multicast on Datacenter Topologies, Proc. of the 9th ACM Conf. on Emerging Networking Experiments and Technologies*, pp. 61–72, 2013.

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

[20] Moore A., Zuev D., Crogan M.: Packet Classification Algorithms: From Theory to Practice. In: *Packet Classification Algorithms: From Theory to Practice, Department of Computer Science Research Reports,Queen Mary, Univeristy of London*, pp. 1–16, 2013.

[21] Moore A.W., Papagiannaki K.: Toward the Accurate Identification of Network Applications. In: *PAM*, 2005.

[22] Petri I., Zou M., Zamani A.R., Diaz-Montes J., Rana O., Parashar M.: Integrating Software Defined Networks within a Cloud Federation. In: *Integrating Software Defined Networks within a Cloud Federation, 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 179–188, 2015.

[23] Qi Y., Xu L., Yang B., Xue Y., Li J.: Discriminators for use in flow-based classification. In: *Discriminators for use in flow-based classification,IEEE INFOCOM 2009*, pp. 648–656, 2009.

[24] RIFAI M.: Next-Generation SDN Based Networks, *PhD Thesis*, pp. 1–178, 2017.

[25] Salman O., Nunes B., Mayoral A., Jararweh Y., Abdelmoniem: IoT survey: An SDN and fog computing perspective, *Computer Networks*, vol. 143(6), pp. 221–246, 2018.

[26] Sen S., Spatscheck O., Wang D.: Accurate, Scalable In-network Identification of P2P Traffic Using Application Signatures. In: *Accurate, Scalable In-network Identification of P2P Traffic Using Application Signatures, Proceedings of the 13th International Conference on World Wide Web*, pp. 512–521, WWW '04, 2004.

[27] Sivanathan A., Sherratt D., Gharakheili H.H., Radford A., Wijenayake C., Vishwanath A., Sivaraman V.: Characterizing and classifying IoT traffic in smart cities and campuses. In: *Characterizing and classifying IoT traffic in smart cities and campuses, 2017 IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 559–564, 2017.

[28] Son J., Buyya R.: A Taxonomy of Software-Defined Networking (SDN)-Enabled Cloud Computing, *A Taxonomy of Software-Defined Networking (SDN)-Enabled Cloud Computing, ACM Comput Surv*, vol. 51(3), pp. 59:1–59:36, 2018.

[29] Wang W., Zhu M., Wang J., Zeng X., Yang Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: *End-to-end encrypted traffic classification with one-dimensional convolution neural networks, 2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 43–48, 2017.

[30] Wang X., Parish D.J.: Optimised Multi-stage TCP Traffic Classifier Based on Packet Size Distributions. In: *2010 Third International Conference on Communication Theory, Reliability, and Quality of Service*, pp. 98–103, 2010.

[31] Yamansavascilar B., Guvensan M.A., Yavuz A.G., Karsligil M.E.: Application identification via network traffic classification. In: *Application identification via network traffic classification, 2017 International Conference on Computing, Networking and Communications (ICNC)*, pp. 843–848, 2017.

[32] Yao H., Gao P., Wang J., Zhang P., Jiang C., Han Z.: Capsule Network Assisted IoT Traffic Classification Mechanism for Smart Cities, *Capsule Network Assisted IoT Traffic Classification Mechanism for Smart Cities, IEEE Internet of Things Journal*, vol. 6(5), pp. 7515–7525, 2019.

Affiliations

Sonali A. Patil
Department of Computer Sc. & Engg, B.S.Abdur Crescent Institute of Science & Technology, Tamil Nadu, India, psonali119@gmail.com

L. Arun Raj
Department of Computer Sc. & Engg, B.S.Abdur Crescent Institute of Science & Technology, Tamil Nadu, India, arunraj@crescent.edu

Received: ??.??.2021
Revised: ??.??.2021
Accepted: ??.??.2021