Future Generation Computer Systems 113 (2020) 255–265

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment

Bhuvaneswari Amma N.G. [a,*], Selvakumar S. [a,b]

[a] Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India
[b] Indian Institute of Information Technology, Una, Himachal Pradesh, India

ARTICLE INFO

Article history:
Received 15 July 2019
Received in revised form 30 January 2020
Accepted 8 July 2020
Available online 11 July 2020

Keywords:
Anomaly detection
Big data
Convolutional neural network
Deep learning
Fog computing
IoT

ABSTRACT

The proliferation of Internet of things (IoT) devices has lured hackers to launch attacks. Therefore, anomalies in IoT traffic must be detected to mitigate these attacks and protect services rendered by smart devices. The lacuna in the existing anomaly detection techniques is the nonscalable nature of anomaly detection systems, resulting in the mishandling of large-scale data generated from IoT devices. The issue of scalability is addressed and an anomaly detection framework in a fog environment is proposed herein using vector convolutional deep learning (VCDL) approach. The anomaly detection system could be scalable if the traffic can be distributed to the nodes in the fog layer for processing. This is effectively captured in the VCDL approach in which the training of IoT traffic is distributed and computations are performed in the fog nodes. The parameters required for training are shared by the master node in the fog layer. Further, the proposed anomaly detection algorithm classifies IoT traffic as either normal or attack and then passes it to the cloud for attack mitigation. Experiments were conducted on UNSW's Bot-IoT dataset and the results indicate that the proposed distributed deep learning approach can efficiently handle scalable data compared with the existing centralized deep learning approaches. Experimental results show that the proposed approach is significantly better in terms of accuracy, precision, and recall compared with the state-of-the-art anomaly detection systems.

1. Introduction

Currently, Internet of things (IoT) devices are extensively used in smart applications, such as smart cities, health care, and transportation. The proliferation of IoT devices with significant data generation from these devices contribute to IoT security issues [1]. IoT devices can be effortlessly targeted by attackers as they are connected to the Internet and do not possess appropriate security mechanisms. An attacker can easily hack IoT devices by acquiring control over smart devices that can be used maliciously to hack other devices connected to the IoT [2].

According to the A10 Networks security report, smart devices connected to the Internet may reach 20.4 billion by 2020 [3]. As the usage of IoT devices increases, cyber attacks may also increase, e.g., distributed denial of service (DDoS) attacks, malwares, keylogging, collusion attacks, and identity theft. Therefore, cyber attacks are emerging as an obstacle to the wide spread provision of services rendered by smart devices [4]. Therefore, the detection of anomalies in IoT traffic is vital to protect IoT devices.

The anomaly detection mechanisms in IoT are classified into statistical and machine learning approaches [5]. Statistical approaches use only normal IoT traffic to build a trained model [6]. Meanwhile, machine learning approaches use both normal and malicious traffic to build a trained model. These approaches are categorized into supervised, unsupervised, and semi-supervised based on the learning process [7]. The supervised learning process maps the traffic features to traffic class, i.e., normal or attack. This learning process is applied only on labeled datasets. The unsupervised learning process learns the traffic features without knowing the traffic class by obtaining interesting structures in the data. The semi-supervised learning groups similar data using unsupervised learning and uses labeled data to classify unlabeled data. In this study, the class labels are known for IoT traffic records; and hence, supervised learning was used to train the model.

Traffic generated from IoT devices is significant, which renders it an ultimate application of big data [8]. Therefore, it is crucial to employ techniques that can handle large volumes of data in real time. One technique is deep learning (DL) which learns data representation by identifying correlations automatically. Widely used DL techniques include the convolutional neural network (CNN), auto encoder (AE), recurrent neural network (RNN), and long short term memory (LSTM). The CNN, RNN, and LSTM are

* Corresponding author.
E-mail addresses: ngbhuvaneswariamma@gmail.com (Bhuvaneswari Amma N.G.), ssk@nitt.edu, director@iiitu.ac.in (Selvakumar S.).

https://doi.org/10.1016/j.future.2020.07.020
0167-739X/© 2020 Elsevier B.V. All rights reserved.

supervised learning techniques, and the AE is an unsupervised learning technique. These techniques require significant amounts of data and high computational power [9].

The existing detection mechanisms for anomaly detection primarily depend on a centralized cloud failing to handle IoT requirements, i.e., the scalability and distribution of resources. In IoT, operations are performed among a large number of devices, and significant amounts of data are generated exponentially. The cloud is critical in IoT as it allows end users to use services offered by the Internet. However, owing to its centralized architecture, it fails to handle IoT devices while performing costly computations. Furthermore, a high detection time is incurred owing to the long distance from an IoT device to the centralized anomaly detection system. The anomaly detection in IoT is different from the existing mechanisms as the centralized cloud environment can handle the service requirements of IoT [10]. Therefore, an emerging distributed intelligence approach called "fog computing" is used to bridge the gap.

The fog communicates by processing data close to the data sources, i.e., IoT devices. Security mechanisms can be deployed at the fog layer, in which fog nodes are used for distributed processing [11]. Therefore, expensive computations and storage from IoT devices may be offloaded to deploy distributed security mechanisms [12]. This motivated us to propose an anomaly detection framework for IoT traffic in a fog environment.

The following are the main contributions of this study:

1. Anomaly detection framework in a fog environment for IoT traffic,
2. Training of vector convolutional deep learning (VCDL) approach using fog nodes in a distributed manner,
3. Anomaly detection algorithm to detect anomalies in IoT traffic using VCDL networks in parallel, and
4. Depiction of scalable anomaly detection framework as learning occurs in a distributed fog environment.

The remaining sections are organized as follows: Section 2 discusses the related studies pertaining to IoT-based cyber attacks, anomaly detection methods for IoT, DL-based anomaly detection, DL in big data, and DL in fog. Section 3 describes the proposed anomaly detection framework for the IoT. Section 4 analyzes the performance evaluation of the proposed framework. Conclusions and further research directions are presented in Section 5.

2. Related studies

2.1. IoT-based cyber attacks

IoT networks include workstations, laptops, routers, and IoT devices [13]. The ubiquitous usage of and reliance on IoT devices result in a significant amount of data, which pose vulnerabilities in IoT networks, resulting in cyber attacks [1]. Attackers typically use easily and freely available botnets to launch a number of cyber attacks. Botnets are a network of infected computers called bots that have command and control infrastructure [14,15]. They are used to perform malicious activities, such as DDoS, keylogging, identity threats, and proliferation of other bot malwares [16].

2.2. Anomaly detection methods for IoT

Existing learning techniques for IoT security are reviewed in [17,18]. The challenges addressed include misuse of learning algorithms by hackers, privacy and security of the learning methods, and architecture of the learning algorithms. Machine learning techniques, such as the Bayesian network, support vector machine (SVM), artificial neural network, k-nearest neighbor, logistic regression, genetic algorithm, decision tree, DL, and extreme learning machine (ELM) have been used for anomaly detection [19,20]. These learning techniques can provide suitable solutions for anomaly detection as they achieved good results [21]. It is observed that not all learning techniques can solve all the problems. Each technique has its advantages and disadvantages [22]. Among the learning techniques above, DL is adaptable to dynamic environments which motivated us to construct a DL-based anomaly detection framework for IoT traffic.

2.3. DL-based anomaly detection

Conventional learning algorithms are dependent on hand-designed features for learning. DL approaches possess inherent capability of overcoming the drawbacks of traditional algorithms [23]. DL techniques for anomaly detection include the CNN [24], deep AE [25], RNN [26], and LSTM [4]. These techniques can be used for either supervised or unsupervised learning depending on the problem to be solved [27], such as health care, smart grids, vehicular communications, and transportation. Furthermore, these approaches can be applied in IoT-security-related applications, such as finger print recognition, malicious code identification, and DDoS attacks detection [28].

A deep AE can be used to reduce dimensionality, and it performs unsupervised learning [29]. Approaches such as RNN and LSTM perform the same operation for every input, and the output depends on past computations [30]. These operations are performed repeatedly as they are recurrent in nature. The CNN performs the unsupervised mode of pre-learning to compress and extract features with multiple levels of abstraction and uses the supervised mode to perform feature learning. These special characteristics of the CNN motivated us to propose a VCDL approach for the anomaly detection of IoT traffic.

2.4. DL in big data

In the current digital world, significant volumes of data are generated from smart devices, which birthed the concept of big data. Despite the many opportunities provided by big data in business analytics, industrial control applications, and smart healthcare, data and information processing still poses issues owing to the characteristics of big data, i.e., large volume, velocity, variety, and veracity [8]. DL is critical in big data for automatically learning large volumes of data to extract features and recognize patterns from complex data [31]. DL can be widely used to address big data problems in an efficient manner, which is impossible using conventional machine learning algorithms as high human involvement is required for designing the algorithms [32]. Fine-grained patterns have been uncovered by these algorithms to yield timely and accurate detections; however, major challenges exist in terms of scalability and distributed computing [33]. Therefore, fog computing is applied for learning big data.

2.5. DL in fog

A survey regarding learning in fog is provided in [34]. The goal of fog computing is to mitigate the network burdens at the cloud by moving costly computations and storage closer to the end users. To address the issue of scalability for processing traffic generated by IoT devices, fog nodes were used as a proxy for computations [35]. DL models can be used to illustrate the feasibility of deploying data analytics in the fog [10]. Further, these models can reduce the learning time by distributing and processing the traffic in parallel. This motivated us to propose an anomaly detection framework for IoT traffic in a fog environment.

3. Proposed anomaly detection framework for IoT in fog

The IoT network comprises numerous smart devices and can be placed in different locations. Therefore, the anomaly detection system must be capable of handling the traffic generated by these devices to provide a fast response in minimal time. In this case, the centralized anomaly detection system provides poor performance in terms of accuracy and detection time. The proposed anomaly detection framework is designed to handle the traffic generated by IoT devices by distributing the load to fog nodes.

Fig. 1 depicts the proposed anomaly detection framework. The proposed framework comprises of the following layers: device, fog, and cloud layers. The device layer comprises IoT devices connected to each other. It has restricted processing capabilities and limited bandwidth; hence it cannot handle evolving events. The fog layer comprises of network tools and devices with network connection. This layer is responsible for minimizing the computational load on resource-constrained IoT devices. The cloud layer validates the information obtained from the fog. It provides guidelines to the fog layer for enhancing the quality of applications provided by the fog devices.

The proposed anomaly detection framework for IoT traffic performs training in a distributed manner using fog nodes. The VCDL approach [24] is used to train the proposed framework, as this approach trains the IoT traffic in different abstraction levels. The master fog node shares the model with the worker fog nodes. The trained model is used to classify the IoT traffic as either normal or attack and then passed to the cloud layer for further processing of the traffic data.

3.1. IoT traffic representation

An IoT traffic record is denoted by $T_r = [tf_1, tf_2, \ldots, tf_k, C_l]$, where tf_i is the traffic features and C_l is the class label of the traffic record. The IoT traffic dataset is represented as follows:

$$T_D = \begin{bmatrix} tf_1^1 & tf_2^1 & tf_3^1 & \cdots & tf_n^1 \\ tf_1^2 & tf_2^2 & tf_3^2 & \cdots & tf_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ tf_1^k & tf_2^k & tf_3^k & \cdots & tf_n^k \end{bmatrix} \quad (1)$$

The traffic features are normalized using min–max normalization in the range [0,1] using (2).

$$X_n^k = \frac{tf_n^k - min_{fn}}{max_{fn} - min_{fn}} \quad (2)$$

where tf_n^k is the raw data, min_{fn} is the minimum value of the feature, and max_{fn} is the maximum value of the feature. After normalization, the bias free IoT traffic dataset is represented as follows:

$$NT_D = \begin{bmatrix} X_1^1 & X_2^1 & X_3^1 & \cdots & X_n^1 \\ X_1^2 & X_2^2 & X_3^2 & \cdots & X_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_1^k & X_2^k & X_3^k & \cdots & X_n^k \end{bmatrix} \quad (3)$$

The data in (3) are passed to the proposed VCDL model for training and anomaly detection.

3.2. VCDL model

The VCDL model is based on the CNN with two levels, i.e., a convolutional layer (CL) and pooling layer (PL). Fig. 2 depicts the structure of the VCDL model. It comprises two modules: vector convolutional network (VCN) and fully connected network (FCN). The VCN extracts the features, and the FCN learns the extracted features to detect IoT traffic class. The hyper-parameters used to construct the VCDL structure include the numbers of CLs and PLs, hidden layer (HL) in the FCN, and nodes in the HLs. These parameters and values are tabulated in Table 1.

Table 1
Hyper-parameters of the VCDL approach.

Parameters	Values
Convolutional layer	2
Pooling layer	2
Hidden layer	2
Nodes in First Hidden layer	9
Nodes in Second Hidden layer	7

3.2.1. VCN

The VCN uses the convolution operation with the input features and the kernel to obtain a new feature vector; it requires zero padding to perform a convolution operation. The proposed network functions in a vector; therefore, zeros are padded at the left and right of the input vector. The VCN in the learning process overcomes the over-fitting of IoT traffic data [24]. Let $\{X_1, X_2, \ldots, X_n\}$ be the features of the IoT traffic. The convolution of the input vector X_i^l and kernel G can be computed as follows:

$$CL_i^l = X_i^l \times G \quad (4)$$

The VCN is constructed by placing a PL between two CLs to reduce the size of the feature vectors. Each pattern in the PL is connected to the pattern of its corresponding previous CL. The PL performs maxpooling to compress the feature vector. The maxpooling size is 2 and it is computed as follows:

$$PL(CL_{ij}^l, CL_{ik}^l) = max(CL_{ij}^l, CL_{ik}^l) \quad (5)$$

Eqs. (4) and (5) are used for the next level of the CL and PL, respectively. The output of the VCN is the extracted feature vector that is then passed to the FCN module.

3.2.2. FCN

The FCN trains the pre-trained feature vector. The structure of the FCN is $k - 9 - 7 - c$, where k is the number of nodes in the input layer, which is similar to the number of output nodes of the VCN; and c is the output layer nodes. The value of c is 2 for binary classification and 5 for multiclass classification. The activation function used in the FCN is the rectified linear unit (ReLU) function. The ReLU activation function was used because it could improve the performance of the anomaly detection framework in less learning time. The input layer of the FCN passes the extracted feature vector to the HL and no computation is performed in this layer. The computations in the HL are performed by computing the sum of the products of the feature vector and the weights. The computation in the first HL, H_{1i}, is performed as follows:

$$H_{1i} = \sum_{i=1}^{p} In_{ij} \times W_{ij}^{H1} \quad (6)$$

where In_{ij} is the feature vector obtained from the VCN. The ReLU activation function is applied to compute the output of the HL and is computed as follows:

$$f_{ReLU}\left(H_{1i}\right) = max\left(H_{1i}, 0\right) \quad (7)$$

The ReLU performs a nonlinear transformation and exhibits faster learning compared with other activation functions. Eqs. (6) and (7) are used in the next HL computation. The output, I_j^O, is computed by multiplying the output of the last HL with the weights

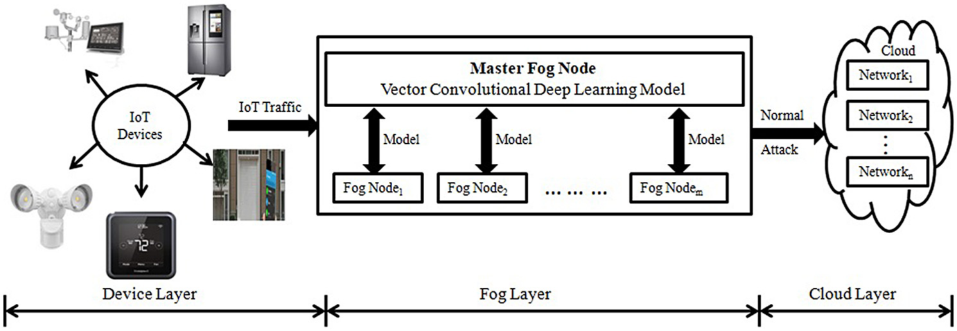


Fig. 1. Block schematic of the proposed anomaly detection framework.

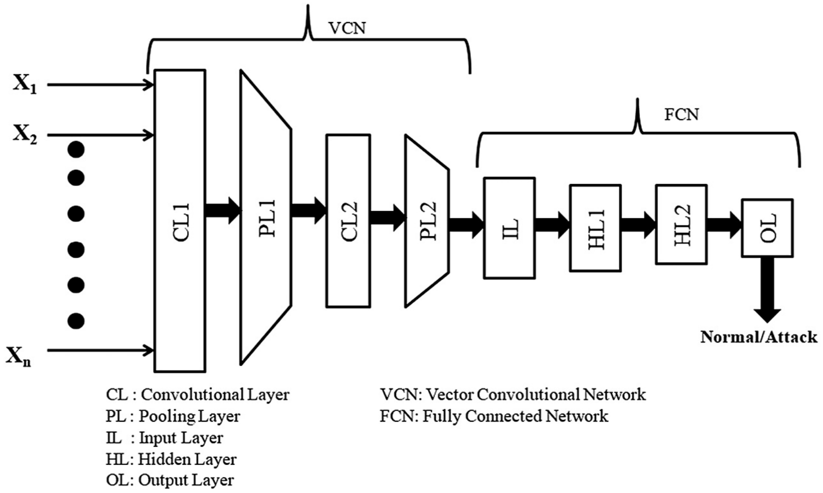


Fig. 2. Block schematic of VCDL approach.

and a bias is summed to push the traffic record to the appropriate class, which is computed as follows:

$$I_j^O = \sum_{i=1}^{q} H_{2i} \times W_{ij}^{Oi} + Bias \qquad (8)$$

The softmax activation function is applied in the output layer, which produces results in a vector with probabilities between 0 and 1 that add to 1; the computation is as follows:

$$O_i = \frac{Exp\left(I_{j_i}^{O}\right)}{\sum_{i=0}^{k} Exp\left(I_{j_i}^{O}\right)} \qquad (9)$$

The loss incurred in the learned network is evaluated using the cross entropy, *CE*, whose output is a probability between 0 and 1. Note that the cross-entropy loss increases as the predicted value deviates from the actual class label. It is computed as follows:

$$CE(T_i, O_i) = -\sum_{i=1}^{k} T_i log\left(O_i\right) \qquad (10)$$

where T_i is the actual class label value. The learning continues until the cross-entropy loss reaches the defined threshold, and the learned model is constructed by the master fog node for the anomaly detection of IoT traffic.

3.3. Fog-based distributed learning

The fog layer comprises of one master fog node for global learning and many worker fog nodes for local learning. Each fog node comprises of application, link, transport, and network layers. The anomalies in the IoT traffic are detected in the transport and in network layers containing IoT devices, such as switches and routers. The fog nodes process the data obtained from these IoT devices connected to the particular fog node.

In the proposed approach, the load is equally shared to the available worker fog nodes. The distributed learning process handles the scalability issues in the centralized approach. Once the learning is completed, the results are stored in the master fog node and used for the anomaly detection of unknown IoT traffic. Furthermore, this process causes the model share the best parameters obtained from learning to avoid overfitting the proposed framework.

3.4. Anomaly detection of IoT traffic

The anomaly in the IoT traffic is detected by the master fog node. The features are automatically extracted using the CL by compressing the IoT traffic features using the PL. The FCN is used for classifying the IoT traffic as either normal or attack. The class of the traffic is detected using the softmax function. The binary classification requires two nodes in the output layer of the FCN, and the multiclass classification requires nodes that are equal to the number of classes in the IoT traffic dataset. Algorithm 1 depicts the proposed anomaly detection framework. This algorithm uses IoT traffic as input and classifies the traffic as either normal or attack. As the fog nodes are used for distributed processing, the algorithm is executed in parallel. The output obtained by executing this algorithm is passed to the cloud layer.

Algorithm 1: Proposed Anomaly Detection Algorithm

```
Input: IoT traffic
Output: Normal/Attack

for each IoT traffic data (compute in parallel) do
    for each levels of CL and PL do
        Compute convolution using (4)
        Compute pooling using (5)
    end for
    for each FCN do
        for each HL do
            Compute sum of products of weights and feature
            vector
            Compute ReLU activation function
        end for
    end for
    Compute softmax using (9)
    return Class of IoT Traffic
end for
```

Table 2
Statistics of UNSW's BoT-IoT dataset in binary class.

IoT Traffic	Training samples	Testing samples
Normal	1018	477
Attack	3,036,915	3,668,045

The normal traffic is forwarded to the network, and the attack traffic is forwarded to the attack mitigation system.

4. Performance evaluation and analysis

This section discusses the performance of the proposed framework through simulation and comparisons with state-of-the-art anomaly detection approaches.

4.1. Experimental setup

The experiments were evaluated on an Intel(R) Core(TM) i7-6700 processor with 16 GB RAM under Ubuntu. The proposed approach was implemented in Apache Spark [36] for a fast computation to achieve parallelism and distribution. DL was implemented in Keras on a Theano package. The centralized experimentation was conducted by executing the DL model on a single node. Experiments were conducted using UNSW's Bot-IoT dataset [37] to test the efficiency of the anomaly detection framework.

4.1.1. Bot-IoT dataset

Table 2 shows the distribution of the Bot-IoT dataset. The Bot-IoT training dataset comprises of 3,037,933 records of IoT traffic, including 1018 records of normal traffic and 3,036,915 records of attack traffic. The testing dataset comprises of 3,668,522 records of IoT traffic, including 477 records of normal traffic and 3,668,045 records of attack traffic. These IoT traffic records were generated using IoT smart home devices, i.e., weather monitoring systems, smart refrigerators, smart-motion-activated lights, smart remotely activated garage doors, and smart thermostats. The following are the four categories of attack records: DDoS, DoS, reconnaissance, and theft. The subcategories of DDoS and DoS include TCP, HTTP, and UDP. The subcategories of reconnaissance are Service_Scan and OS_Fingerprint. The subcategories of theft include Data_Exfiltration and Keylogging. Tables 3 and 4 show the categories and subcategories of IoT traffic, respectively.

Table 3
Statistics of UNSW's BoT-IoT dataset in multiclass.

IoT Traffic	Training samples	Testing samples
Normal	1018	477
DDoS	2,027,166	1,926,624
DoS	894,463	1,650,260
Reconnaissance	115,167	91,082
Theft	119	79

Table 4
Subcategories of Traffic in UNSW's BoT-IoT dataset.

IoT Traffic		Samples	
Category	Subcategory	Training	Testing
Normal	Normal	1018	477
DDoS	TCP	1,471,635	977,380
	HTTP	4576	989
	UDP	550,955	948,255
DoS	TCP	165,412	615,800
	HTTP	7298	1485
	UDP	721,753	1,032,975
Reconnaissance	OS_Fingerprint	25,846	17,914
	Service_Scan	89,321	73,168
Theft	Data_Exfiltration	21	6
	Keylogging	98	73

Table 5
Features of UNSW's BoT-IoT dataset.

Features	Feature Names
Best 10	*seq*, *stddev*, *N_IN_Conn_P_SrcIP*, *min*, *state_number*, *mean*, *N_IN_Conn_P_DstIP*, *drate*, *srate*, *max*
All	*pkSeqID*, *Stime*, *flgs*, *flgs_number*, *Proto*, *proto_number*, *saddr*, *sport*, *daddr*, *dport*, *pkts*, *bytes*, *state*, *state_number*, *ltime*, *seq*, *dur*, *mean*, *stddev*, *sum*, *min*, *max*, *spkts*, *dpkts*, *sbytes*, *dbytes*, *rate*, *srate*, *drate*, *TnBPSrcIp*, *TnBpDstIP*, *TnP_PSrcIP*, *TnP_PDstIP*, *TnP_perProto*, *TnP_Per_Dport*, *AR_P_Proto_P_SrcIp*, *AR_P_Proto_DstIP*, *N_IN_Conn_P_SrcIP*, *N_IN_Conn_P_DstIP*, *AR_P_Proto_P_Sport*, *AR_P_Proto_P_Dport*, *Pkts_P_State_P_Protocol_P_DestIP*, *Pkts_P_State_P_Protocol_P_SrcIP*

4.1.2. Features in Bot-IoT dataset

Table 5 shows the features of the Bot-IoT dataset. To evaluate the performance of the proposed approach, the 10 best features and all features were considered in the experiment [4].

4.2. Experimental analysis

Performance measures such as true negative (TN), false positive (FP), false negative (FN), and true positive (TP) were used to analyze the performance of the proposed approach. The confusion matrix for binary class classification is tabulated in Table 6. The performance metrics, i.e., precision, recall, F-measure, fall-out, accuracy, and error rate (ER), were computed using Eqs. (11), (12), (13), (14), (15), and (16), respectively, and tabulated in Table 7 for binary class classification. The experimental results show that the best 10 features exhibit significantly better performance compared with all the features.

The confusion matrices for multiclass classification using the 10 best features and all features are tabulated in Tables 8 and 9, respectively. The performance of the proposed approach for multiclass classification is tabulated in Table 10. Promising results are shown except for the normal and theft traffic. These two traffic types performed poorly because their numbers of training records are less compared with those of others. Furthermore, it is

Table 6
Confusion matrix for binary class.

Features	TN	FP	FN	TP
Best 10	432	45	8919	3,659,126
All	419	58	9127	3,658,918

Table 7
Performance evaluation for binary class.

Features	Precision	Recall	F-measure	Fall-out	Accuracy	ER
Best 10	99.9988	99.7569	99.8777	9.4340	99.7557	0.2443
All	99.9984	99.7512	99.8746	12.1593	99.7496	0.2504

Table 8
Confusion matrix for multiclass for best 10 features.

Traffic	Normal	DDoS	DoS	Reconnaissance	Theft
Normal	432	14	26	5	0
DDoS	14	1,921,112	5234	264	0
DoS	26	3216	1,646,946	72	0
Reconnaissance	5	27	42	91,007	1
Theft	0	7	9	2	61

Table 9
Confusion matrix for multiclass for all features.

Traffic	Normal	DDoS	DoS	Reconnaissance	Theft
Normal	419	17	31	9	1
DDoS	17	1,921,008	5176	421	2
DoS	31	2996	1,646,814	418	1
Reconnaissance	9	29	61	90,982	1
Theft	1	9	8	5	56

evident that the experiments conducted with the 10 best features outperformed the experiments with all features.

$$Precision = \frac{TP}{TP + FP} \times 100 \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \times 100 \tag{12}$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

$$Fall - out = \frac{FP}{FP + TN} \times 100 \tag{14}$$

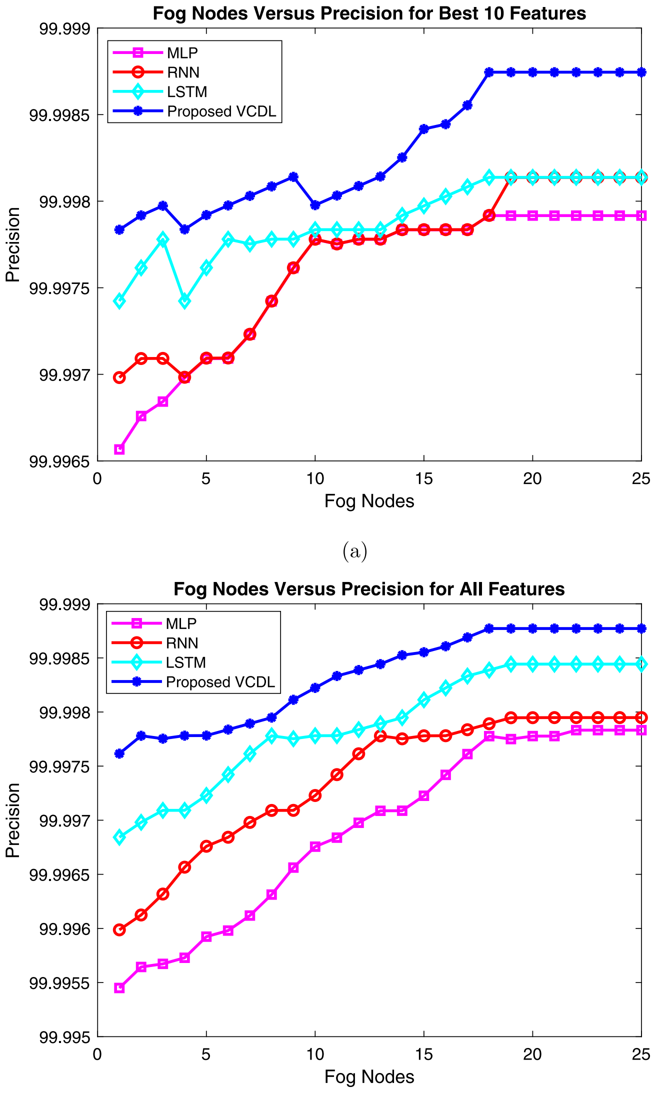
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \tag{15}$$

$$ER = \frac{FN + FP}{TP + FP + TN + FN} \times 100 \tag{16}$$

4.3. Analysis of performance metrics with fog nodes

The performance of the proposed anomaly detection framework is compared with the existing DL models, such as multilayer perceptron (MLP), RNN, and LSTM for varying fog nodes. The MLP is a conventional learning technique that uses a backpropagation algorithm for training. The RNN and LSTM learn the features repeatedly owing to the recurrent structure. These existing approaches are selected for comparison because they are supervised learning approaches, as is the proposed approach.

The IoT traffic was distributed to fog nodes, and the proposed approach was tested by varying the fog nodes from 1 to 25. The parameters of the base DL models used in the experiment are shown in Table 11. The learning, using all features, requires more epochs compared with performing learning using the 10 best features. As shown in Figs. 3–8, the proposed anomaly detection framework exhibits promising results compared with the existing DL models. Part (a) of the figures shows the performances of the 10 best features, and part (b) shows those of all features. It is evident that the performance improved as the number of fog nodes increased. The proposed approach achieved the best performance with 19 fog nodes; beyond that, the performance did not improve.



(a)

(b)

Fig. 3. Fog nodes vs. precision: (a) 10 best features and (b) All features.

4.4. Analysis of detection time

The detection times of the proposed distributed approach for the 10 best and all features were compared with those of existing centralized approach. Fig. 9 shows the detection time comparison of the distributed and centralized approaches for varying percentages of IoT traffic. As shown, the detection time is significantly low for the proposed distributed approach with the 10 best features compared with the centralized and distributed approaches with all features.

4.5. ROC Analysis

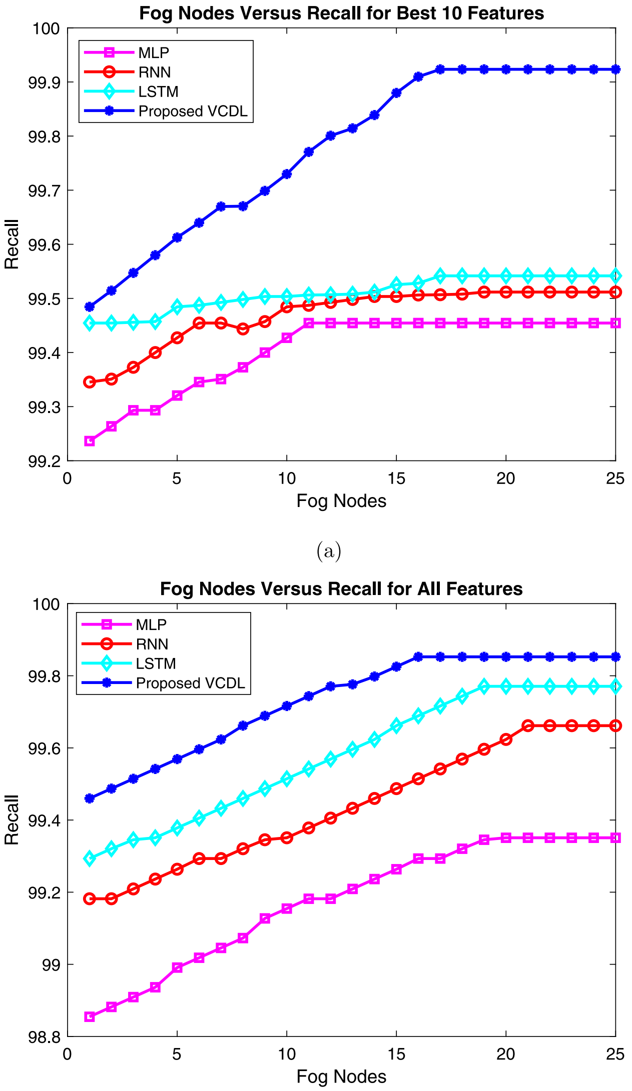
The performance of the anomaly detection framework was analyzed by plotting receiver operating characteristics (ROC) curves

Table 10
Performance evaluation for multiclass.

Features	Traffic	Precision	Recall	F-measure	Fall-out	Accuracy	ER
Best 10	Normal	90.57	90.57	90.57	9.43	90.57	9.43
	DDoS	99.85	99.71	99.78	0.1696	99.71	0.2861
	DoS	99.69	99.80	99.74	0.3214	99.80	0.2008
	Reconnaissance	99.09	99.92	99.50	0.3755	99.92	0.0823
	Theft	98.39	77.22	86.53	1.6129	77.22	22.78
All	Normal	87.84	87.84	87.84	12.16	87.84	12.16
	DDoS	99.84	99.71	99.77	0.1586	99.71	0.2915
	DoS	99.68	99.79	99.73	0.3194	99.79	0.2088
	Reconnaissance	99.07	99.89	99.48	0.9288	99.89	0.1098
	Theft	91.94	70.88	80.05	8.1967	71.25	29.11

Table 11
Parameters of existing DL models.

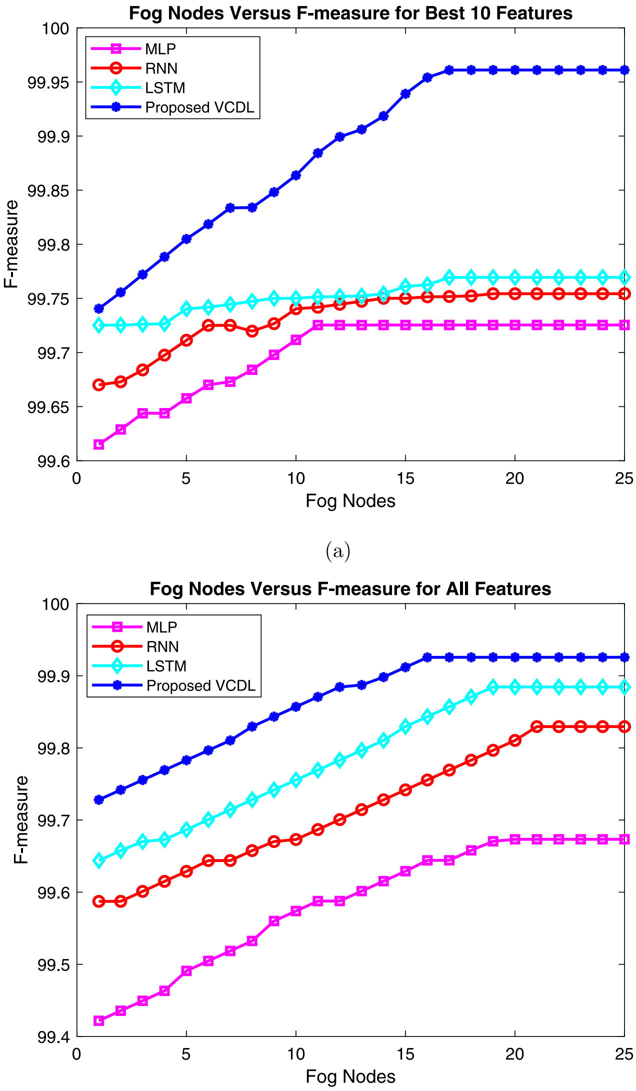
Model	Features Used	Epoch	Structure	Activation function
MLP	Best 10	5	$k-10-8-6-1$	tanh for hidden layers
	All	7	$k-10-8-6-1$	sigmoid for output layer
RNN	Best 10	5	$k-30-20-15-10-1$	ReLU for hidden layers
	All	7	$k-30-23-20-15-10-1$	sigmoid for output layer
LSTM	Best 10	5	$k-20-15-10-1$	tanh for hidden layers
	All	7	$k-25-20-15-10-1$	sigmoid for output layer



(a)

(b)

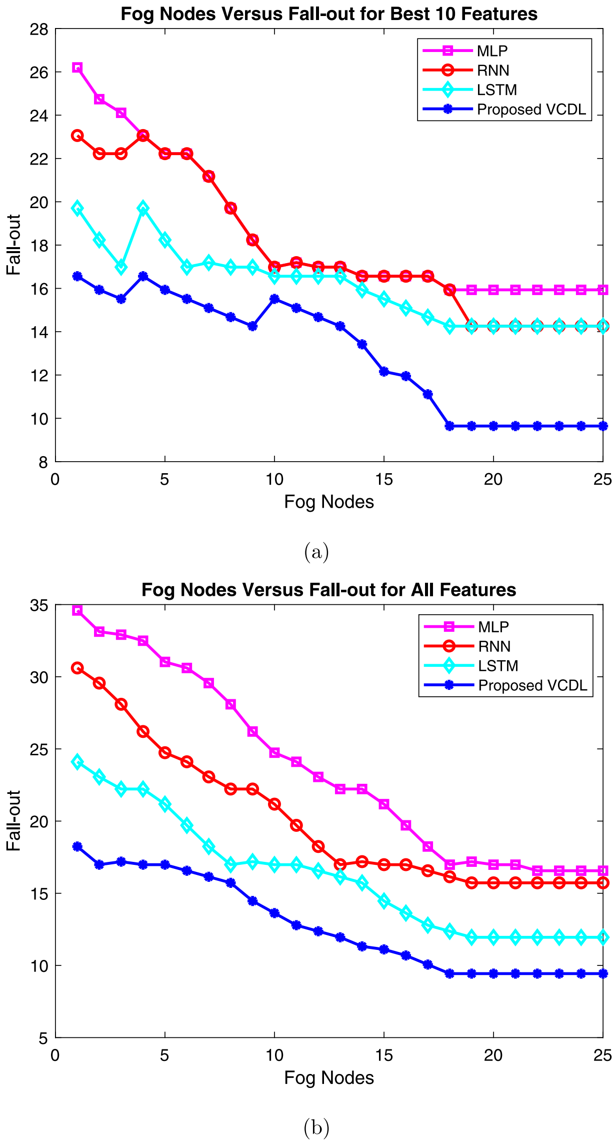
Fig. 4. Fog nodes vs. recall: (a) 10 Best features and (b) All features.



(a)

(b)

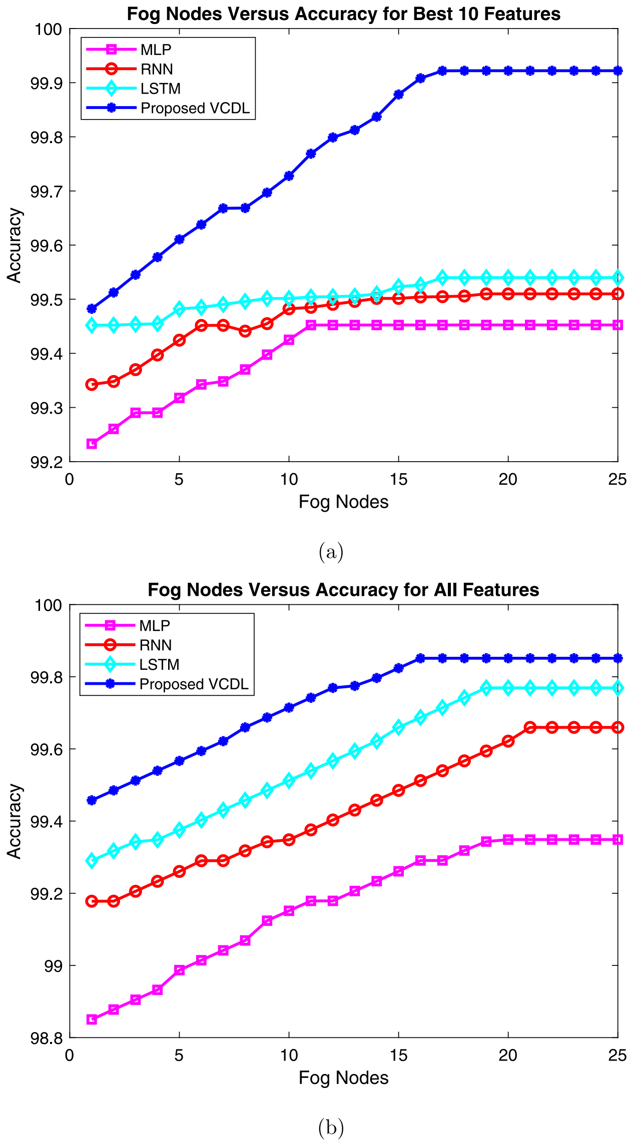
Fig. 5. Fog nodes vs. F-measure: (a) 10 best features and (b) All features.



(a)

(b)

Fig. 6. Fog nodes vs. fall-out: (a) 10 best features and (b) All features.



(a)

(b)

Fig. 7. Fog nodes vs. accuracy: (a) 10 best features and (b) All features.

using the computed FP rate or fall-out and the TP rate or recall. The fall-out and the recall values may vary between 0 and 1. Therefore, the maximum area of the ROC is 1. It is observed that as the area under the ROC curve (AUC) value increases, the accuracy increases. The ROC curves for the 10 best features and all features are depicted in Fig. 10. From the ROC plots, it can be inferred that the AUC for the 10 best features is higher than that for all the features in the dataset.

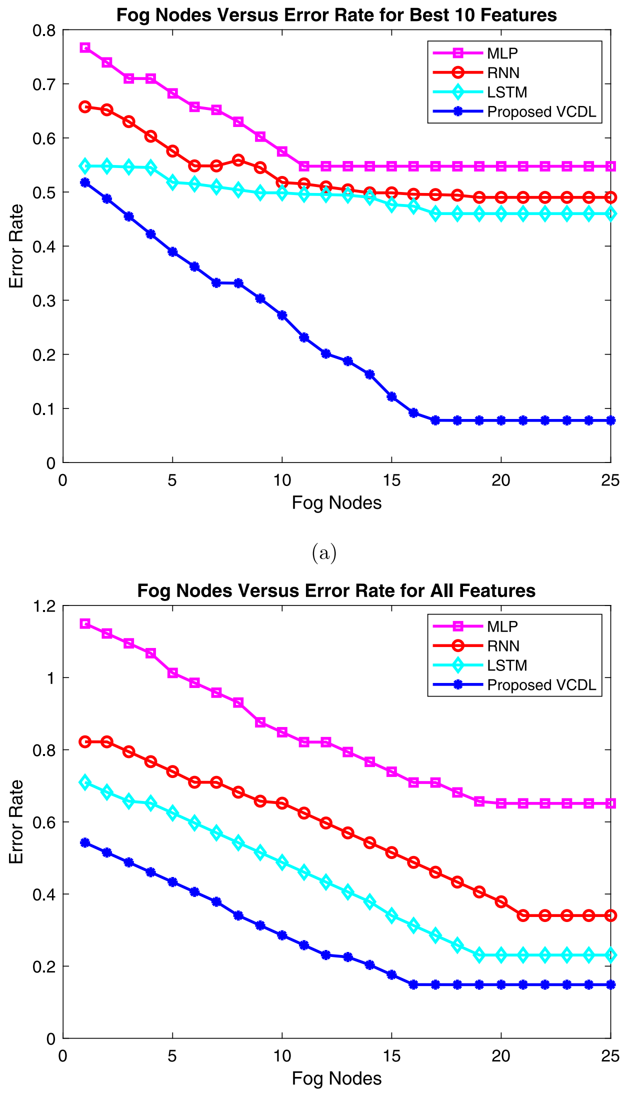
4.6. Comparison with state-of-the-art approaches

The significance of the proposed approach was validated by comparing the proposed approach with the existing IoT-based anomaly detection approaches listed in Table 12. The various aspects considered for comparison are the security architecture, usage of novel or conventional algorithms, and the methodology. It was observed that most of the approaches used a centralized architecture. In [7] and [25], distributed architectures were used but IoT traffic was not considered in experiments.

Table 13 tabulates the performance comparison of the proposed approach with the reported results of existing state-of-the-art anomaly detection approaches [4]. The main reason for selecting these approaches for comparison is that the same dataset (UNSW's Bot-IoT dataset) was used for the evaluation; therefore, the comparisons would be more reliable and practical. The performance of the SVM in terms of accuracy and recall is low for the 10 best features, as the SVM is effective in handling higher-dimension data. However, the accuracy of the SVM for all features is better compared with the proposed approach as the SVM uses the kernel trick, which is not used in other approaches. The performances of the RNN, LSTM, and the proposed approach are approximately similar for the 10 best features as these approaches learn the features in different abstraction levels. It is observed that the proposed anomaly detection framework exhibits significantly better results for the 10 best and all features than the existing DL-based anomaly detection systems.

Table 12
Concept Comparison of Proposed and Existing Approaches.

Approach and Year	Security Architecture	Usage of Conventional Algorithm	Methodology
[20] 2014	Centralized	Yes	Naive Bayesian classifier
[19] 2016	Centralized	No	Two-layer dimension reduction and two-tier classification
[22] 2017	Centralized	Yes	Supervised machine learning algorithm
[7] 2018	Distributed	No	ELM-based semi-supervised Fuzzy C-Means
[25] 2018	Distributed	Yes	Deep learning
[4] 2019	Centralized	Yes	SVM, RNN, and LSTM
Proposed Approach	Distributed	No	Vector convolutional deep learning



(a)

(b)

Fig. 8. Fog nodes vs. error rate: (a) 10 best features and (b) All features.

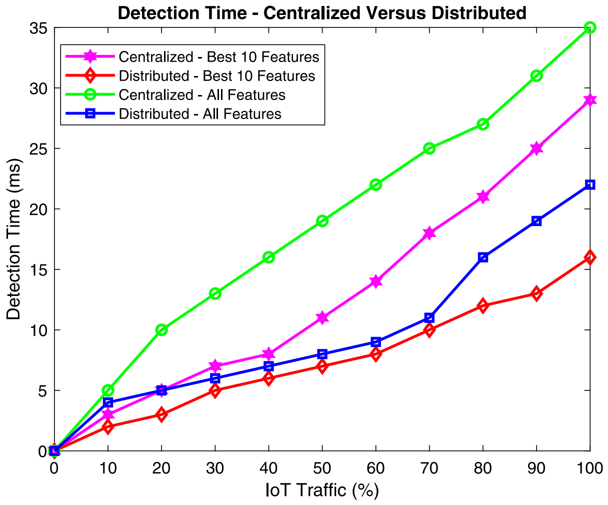


Fig. 9. Detection time for centralized and distributed computations.

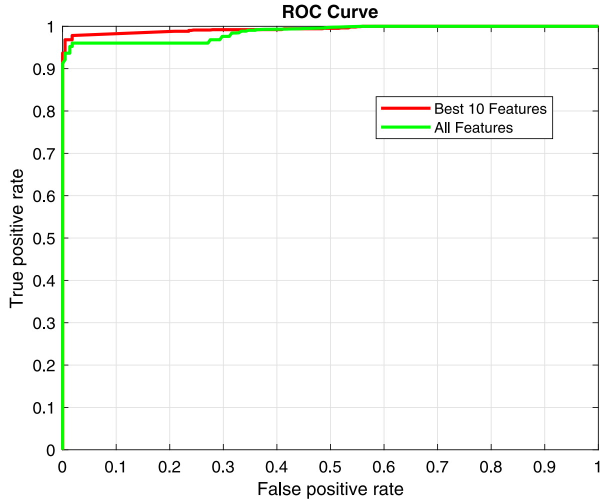


Fig. 10. ROC analysis.

5. Conclusion

In this study, an anomaly detection framework for IoT traffic in a fog environment was proposed using the VCDL approach. The objective was to overcome the nonscalable nature of anomaly detection systems for IoT traffic. For a scalable anomaly detection framework, the traffic records were distributed to several fog nodes for learning the IoT traffic features in parallel. Moreover, the learning was performed using a VCDL network that was trained using the FCN by extracting the features using

Table 13
Comparison with state-of-the-art anomaly detection approaches.

Features	Approaches	Accuracy(%)	Precision(%)	Recall(%)
Best 10	SVM	88.37	100	88.37
	RNN	99.74	99.99	99.75
	LSTM	99.74	99.99	99.75
	Proposed approach	99.7557	99.9988	99.7569
All	SVM	99.99	99.99	100
	RNN	97.91	99.99	97.91
	LSTM	98.06	99.99	98.06
	Proposed approach	99.7496	99.9984	99.7512

the VCN. The learned VCDL model was used by the master fog node to classify known and unknown anomalies in IoT traffic. The proposed anomaly detection framework was evaluated using UNSW's Bot-IoT dataset. The experimental results showed that the proposed framework in a fog environment supported distributed anomaly detection with less detection time than the centralized architecture. From the reported results, it was evident that the proposed approach achieved a significant improvement in performance compared with state-of-the-art anomaly detection systems and conventional DL models. Furthermore, ROC analysis showed that the proposed framework achieved significant performance in terms of accuracy with the selected features compared with all the features. In future studies, the detection accuracy of each class for multiclass classification can be improved by overcoming the class imbalance problem.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I. Makhdoom, M. Abolhasan, J. Lipman, R.P. Liu, W. Ni, Anatomy of threats to the Internet of Things, IEEE Commun. Surv. Tutor. (2018) http://dx.doi.org/10.1109/COMST.2018.2874978.

[2] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, N. Ghani, Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale iot exploitations, IEEE Commun. Surv. Tutor. (2019).

[3] A10 networks security report - IoT and DDoS attacks: A match made in heaven, 2019, https://www.a10networks.com/blog/iot-and-ddos-attacks-a-match-made-in-heaven/, Mar. 01.

[4] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Gener. Comput. Syst. (2019) http://dx.doi.org/10.1016/j.future.2019.05.041.

[5] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, L.T. Yang, Data mining for internet of things: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2013) 77–97, http://dx.doi.org/10.1109/SURV.2013.103013.00206.

[6] N. Moustafa, B. Turnbull, K.-K.R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things, IEEE Internet Things J. (2018) http://dx.doi.org/10.1109/JIOT.2018.2871719.

[7] S. Rathore, J.H. Park, Semi-supervised learning based distributed attack detection framework for IoT, Appl. Soft Comput. 72 (2018) 79–89, http://dx.doi.org/10.1016/j.asoc.2018.05.049.

[8] Q. Zhang, L.T. Yang, Z. Chen, P. Li, A survey on deep learning for big data, Inf. Fusion 42 (2018) 146–157, http://dx.doi.org/10.1016/j.inffus.2017.10.006.

[9] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436, http://dx.doi.org/10.1038/nature14539.

[10] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for Internet of Things, Future Gener. Comput. Syst. 82 (2018) 761–768, http://dx.doi.org/10.1016/j.future.2017.08.043.

[11] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM, 2012, pp. 13–16.

[12] C. Mouradian, D. Naboulsi, S. Yangui, R.H. Glitho, M.J. Morrow, P.A. Polakos, A comprehensive survey on fog computing: State-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 20 (1) (2017) 416–464, http://dx.doi.org/10.1109/COMST.2017.2771153.

[13] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, DDoS in the IoT: Mirai and other botnets, Computer 50 (7) (2017) 80–84, http://dx.doi.org/10.1109/MC.2017.201.

[14] E. Bertino, N. Islam, Botnets and internet of things security, Computer (2) (2017) 76–79, http://dx.doi.org/10.1109/MC.2017.62.

[15] C.D. McDermott, F. Majdani, A.V. Petrovski, Botnet detection in the internet of things using deep learning approaches, in: 2018 International Joint Conference on Neural Networks, IJCNN, IEEE, 2018, pp. 1–8.

[16] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, J. Lopez, A survey of IoT-enabled cyber attacks: Assessing attack paths to critical infrastructures and services, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3453–3495, http://dx.doi.org/10.1109/COMST.2018.2855563.

[17] M.A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, M. Guizani, A survey of machine and deep learning methods for internet of things (IoT) security, 2018, arXiv preprint arXiv:1807.11023.

[18] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for IoT security based on learning techniques, IEEE Commun. Surv. Tutor. (2019) http://dx.doi.org/10.1109/COMST.2019.2896380.

[19] H.H. Pajouh, R. Javidan, R. Khayami, D. Ali, K.-K.R. Choo, A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks, IEEE Trans. Emerg. Top. Comput. (2016) http://dx.doi.org/10.1109/TETC.2016.2633228.

[20] Q. Chen, S. Abdelwahed, A. Erradi, A model-based validated autonomic approach to self-protect computing systems, IEEE Internet Things J. 1 (5) (2014) 446–460, http://dx.doi.org/10.1109/JIOT.2014.2349899.

[21] I. Cvitić, D. Peraković, M. Periša, M. Botica, Novel approach for detection of iot generated DDoS traffic, Wirel. Netw. (2019) 1–14, http://dx.doi.org/10.1007/s11276-019-02043-1.

[22] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N.O. Tippenhauer, J.D. Guarnizo, Y. Elovici, Detection of unauthorized iot devices using machine learning techniques, 2017, arXiv preprint arXiv:1709.04647.

[23] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26, http://dx.doi.org/10.1016/j.neucom.2016.12.038.

[24] N.G.B. Amma, S. Subramanian, VCDeepFL: Vector convolutional deep feature learning approach for identification of known and unknown denial of service attacks, in: TENCON 2018–2018 IEEE Region 10 Conference, IEEE, 2018, pp. 0640–0645.

[25] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-BaIoT—Network-based detection of IoT Botnet attacks using deep autoencoders, IEEE Pervasive Comput. 17 (3) (2018) 12–22, http://dx.doi.org/10.1109/MPRV.2018.03367731.

[26] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Network traffic classifier with convolutional and recurrent neural networks for Internet of Things, IEEE Access 5 (2017) 18042–18050, http://dx.doi.org/10.1109/ACCESS.2017.2747560.

[27] I.U. Din, M. Guizani, J.J. Rodrigues, S. Hassan, V.V. Korotaev, Machine learning in the Internet of Things: Designed techniques for smart cities, Future Gener. Comput. Syst. (2019) http://dx.doi.org/10.1016/j.future.2019.04.017.

[28] F. Hussain, R. Hussain, S.A. Hassan, E. Hossain, Machine learning in IoT security: Current solutions and future challenges, 2019, arXiv preprint arXiv:1904.05735.

[29] Z.M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, K. Mizutani, State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2432–2455, http://dx.doi.org/10.1109/COMST.2017.2707140.

[30] M. Roopak, G.Y. Tian, J. Chambers, Deep learning models for cyber security in IoT networks, in: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC, IEEE, 2019, pp. 0452–0457.

[31] L. Zhou, S. Pan, J. Wang, A.V. Vasilakos, Machine learning on big data: Opportunities and challenges, Neurocomputing 237 (2017) 350–361, http://dx.doi.org/10.1016/j.neucom.2017.01.026.

[32] B. Jan, H. Farman, M. Khan, M. Imran, I.U. Islam, A. Ahmad, S. Ali, G. Jeon, Deep learning in big data analytics: A comparative study, Comput. Electr. Eng. (2017) http://dx.doi.org/10.1016/j.compeleceng.2017.12.009.

[33] A. L'heureux, K. Grolinger, H.F. Elyamany, M.A. Capretz, Machine learning with big data: Challenges and approaches, IEEE Access 5 (2017) 7776–7797, http://dx.doi.org/10.1109/ACCESS.2017.2696365.

[34] A. Abeshu, N. Chilamkurti, Deep learning: the frontier for distributed attack detection in fog-to-things computing, IEEE Commun. Mag. 56 (2) (2018) 169–175, http://dx.doi.org/10.1109/MCOM.2018.1700332.

[35] A. Alrawais, A. Alhothaily, C. Hu, X. Cheng, Fog computing for the internet of things: Security and privacy issues, IEEE Internet Comput. 21 (2) (2017) 34–42, http://dx.doi.org/10.1109/MIC.2017.37.

[36] M. Zaharia, M. Chowdhury, M.J. Franklin, S. Shenker, I. Stoica, Spark: Cluster computing with working sets, HotCloud 10 (10–10) (2010) 95.

[37] UNSW Bot-iot dataset, 2018, https://www.unsw.adfa.edu.au/unsw/canberra/cyber/cybersecurity/ADFA-NB15-Datasets/bot-iot.php.

Bhuvaneswari Amma N.G. received the M.E. degree in Computer Science and Engineering from College of Engineering, Guindy Campus, Anna University Chennai, India in 2009. She is currently working toward the Ph.D. degree in the Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India. Her areas of interest include computer networks, network security, IoT security, and machine learning.

S. Selvakumar received the Ph.D. degree from the Indian Institute of Technology Madras (IITM), Chennai, India in 1999. He is the Professor in the Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu. He has been the Dean of IIIT, Tiruchirappalli, Tamil Nadu and currently the Director of IIIT, Una, Himachal Pradesh. He has to his credit of publishing 90 research papers. He was the investigator of Rs.100/- lakhs research project, Collaborative Directed Basic Research in Smart and Secure Environment (CDBR-SSE) Project sponsored by NTRO, Government of India. He is the investigator of Rs.106/- lakhs project, Nagarik Rog Pratirakshak: Unified Smart Immunization Coverage Monitoring and Analysis (UniSICMA) Project sponsored by Grand Challenges India - Immunization Data: Innovating for Action (IDIA). His research interests include network security, computer networks, high-speed networks, mobile networks, and wireless sensor networks. He is a member of the IEEE.