Design of Cognitive Fog Computing for Intrusion Detection in Internet of Things

S.Prabavathy, K.Sundarakantham, and S.Mercy Shalinie

Abstract: **Internet of things (IoT) is penetrating into every aspect of our lives including our body, our home and our living environment along with numerous security challenges. With rapidly growing number of connected devices in IoT, the scope for cyber-attack also increases exponentially. Therefore an effective intrusion detection system (IDS) is needed to efficiently detect the attack at faster rate in highly scalable and dynamic IoT environment. In this paper, a novel intrusion detection technique is proposed based on fog computing using Online Sequential Extreme Learning Machine (OS-ELM) which can intelligently interpret the attacks from the IoT traffic. In the proposed system, the existing centralized cloud intelligence in detecting the attack is distributed to local fog nodes to detect the attack at faster rate for IoT application. The distributed architecture of fog computing enables distributed intrusion detection mechanism with scalability, flexibility and interoperability. The analysis of the proposed system proves to be efficient in terms of response time and detection accuracy.**

Index Terms: **Extreme learning machine, fog computing, Internet of things (IoT), intrusion detection system.**

I. INTRODUCTION

INTERNET of things (IoT) is regarded as a virtual network that interacts with real world, taking wireless sensor network and Internet as its core technology [1]. The vision of IoT is to deploy intelligent sensors and actuators to provide pervasive environment and ubiquitous experience. The main challenge in deploying IoT is providing transparent and wide range of seamless services with security [2]. The deployment of IoT devices in both managed and unmanaged environment increases the complexity of existing computing and communication systems, which in turn increases the vulnerabilities of IoT. The wireless transmission of data makes IoT application to meet security requirements of Internet and device network along with its own security requirements to ensure safe and reliable operations [3]. From a security and privacy perspective, the foreseen pervasive introduction of devices and sensors into intimate spaces such as home, car and wearable devices injects immense privacy and security threats in IoT application. As the physical objects continuously detect and share personal data of our daily life, it is essential to ensure security and privacy in IoT applications. The impact of attack on these devices may lead to disaster or loss of life [4]. Most of the existing security approaches for IoT are cloud based centralized mechanism. Centralizing the massive data from millions of devices in IoT has communication limitations including bandwidth constraints [5], latency [6] and battery power [7]. Similarly, IoT involving opportunistic network [8], [9] suffers from significant security concerns due to its decentralized architecture. Therefore, the distributed nature of IoT requires, a distributed security mechanism which supports interoperability, flexibility and scalability with unified security management among its heterogeneous devices [10]. To meet this requirement, the security mechanism can be implemented using fog computing which has processing nodes closer to the physical system and provides processing and storage capabilities at the edge to detect threats at faster rate [11]. Fog nodes connects the end system to cloud computing resources to provide fast, actionable decisions to be made based on vast amount data generated from the IoT [12].

A variety of cutting-edge technologies such as cloud computing, software defined networking, big data analysis, intelligent sensors, etc. have been developed to utilize the complete power of IoT. However, most of these technologies for IoT are in the developing stage and subjected to increased technical implications in implementing these technologies for IoT [13]. Hence these new technologies of IoT have new challenges in ensuring security and privacy. Albeit, fog computing mainly provides distributed service for computational offloading [14], an intelligent security technique is required for detecting the attacks generated from voluminous number of IoT devices in large-scale applications. The intrusion detection techniques are broadly classified into signature based technique and anomaly based technique. In signature based technique, attacks are detected based on the predefined attack signatures [15]. The traffic data is matched against these signatures to detect the attack. This technique proves to be efficient in detection accuracy but it cannot detect novel attack that has no predefined signatures. In anomaly based technique the attack is detected based on the deviation from regular normal behavior. The advantage of this technique is it can detect previously unknown attack but it suffers from high false alarm rates. IoT lead to the edge of data explosion from millions of connected devices due to the expanded associations of people with technology and data [16]. Learning from these massive data is essential to identify the security events and their associations by correlating the internal and external information to predict the unknown threats and to view a wider picture of threats in IoT applications. Traditional learning algorithms are slower which is not suitable for highly dynamic IoT applications [17]. The efficiency of extreme learning machine (ELM) in faster learning can be used for detecting the attacks in highly dynamic environment like IoT. In ELM algorithm, the input layer weights and hidden layer bias values will be randomly chosen and the output weights values will be computed analytically based on Moore-

Manuscript received November 30, 2017.

The authors are with the Department of Computer Engineering, Thiagarajar College of Engineering, email: s.praba.pranauv@gmail.com, {kskcse, shalinie}@tce.edu.

S.Prabavathy is the corresponding author.

Digital Object Identifier: 10.1109/JCN.2018.000041

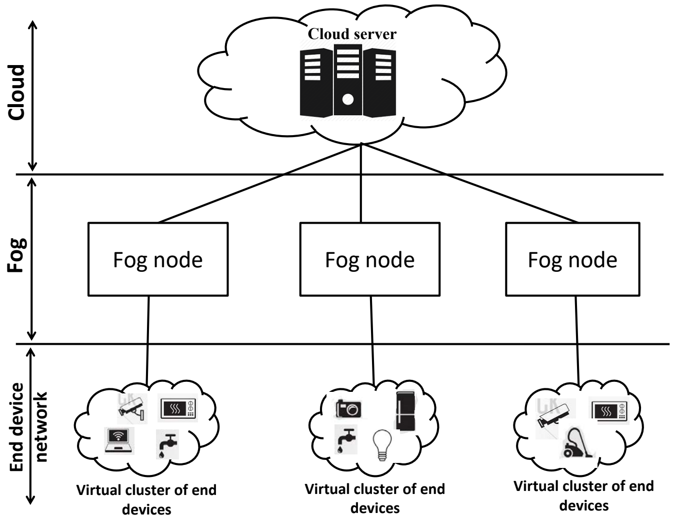
Cloud server
Cloud
Fog
Fog node
Fog node
Fog node
End device network
Virtual cluster of end devices
Virtual cluster of end devices
Virtual cluster of end devices

Fig. 1. Fog computing based IoT architecture.

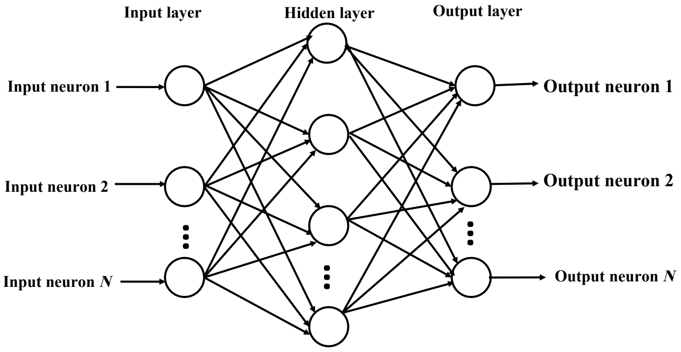

Fig. 2. Single hidden layer feed forward neural network structure.

Penrose generalized inverse. Hence, the common problems of traditional gradient based algorithms such as parameter tuning, stopping criteria decision and local minima will get avoided in ELM [18]. Its online version online sequential extreme learning machine (OS-ELM) is used for real-time classification problem can be adopted to detect the cyber-attacks in IoT environment which generate data at high velocity.

ELM proves to be more powerful in terms convergence speed and generalization power when compared to popular MLP neural networks. Since IoT is a dynamic network with heterogeneous devices, the security mechanism should be a faster learning approach which learns the environment quickly and adapt to changes for detecting the security threats at faster rate. The streaming nature of IoT application traffic favors the need for OS-ELM. The training phase is performed in online manner from streaming data sequentially delivered from IoT devices. OS-ELM provides a fast learning model which can adapt to new data from IoT devices quickly along with a good generalization power. The goal of the proposed work is designing a novel intrusion detection system at distributed fog nodes to enable the detection of attacks in IoT application.

The major contribution of this paper is

1. Distributed detection of cyber-attacks in IoT applications using fog computing.
2. Intrusion detection system is implemented by using OS-ELM algorithm at distributed fog nodes.

The remainder of this paper is organized as follows. Section II gives an overview of the technologies used in the proposed intrusion detection technique. Section III provides existing research performed so far in this domain. In Section IV, the detailed architecture and functionality of proposed intrusion detection mechanism is presented. Section V presents the implementation as proof-of-concept to evaluate the proposed work. Section VI provides the results and discussion. Finally, we concluded the paper in Section VII by providing possible extensions in the proposed intrusion detection system.

II. BACKGROUND

This section gives a brief introduction to the fog computing paradigm and Extreme learning machine algorithm used in the proposed system.

A. Fog Computing

Fog computing is a paradigm developed by CISCO that shifts the data and services to the edge of the network from cloud. It is based on distributed computing mainly developed to handle the voluminous data from rapidly growing number of IoT devices with minimum latency [19]. In addition to reduced latency and lower bandwidth consumption, fog computing provides more support to IoT applications. Fog nodes can be traced to provide location awareness for end devices in IoT. It can be geographically distributed to support high availability and scalability for large-scale IoT applications. Fog computing has protocols to support mobility of IoT devices. It also handles the heterogeneity issues of IoT by supporting interoperability and flexibility in IoT applications using uniform programmable interface by virtualization. Fog computing based IoT architecture consist of 3 tiers: Cloud tier, fog tier and end device tier [11] as depicted in Fig. 1. The main functionality of end device tier is to collect data from its environment and send it to the fog tier. The fog tier performs processing, storage and services at the edge devices such as access point, gateway and routers. The cloud tier performs the global management of IoT application by providing final presentation based on the requirement of application.

B. Extreme Learning Machine (ELM)

ELM is a fast learning algorithm for single hidden layer feed forward neural network (SLFN) shown in Fig. 2. The traditional gradient based learning is slow due to much iterations involved in parameter tuning which is not suitable for real-time applications. ELM solves this problem by randomly choosing the input weights and biases to analytically determine the output weights using simple matrix computations.

B.1 Basic ELM

Given training set with N arbitrary samples containing n attributes and m classes such that (x_i, t_i) where input vector $x_i = (x_{i1}, \cdots, x_{in})^T \in R^n$ and expected output vectors $t_i = (t_{i1}, \cdots, t_{im})^T \in R^m$. The SLFN with $\tilde{N}$ hidden neurons and $g(x)$ activation function can be mathematically formulated as

$$f_N(x_j) = \sum_{i=1}^{N} \beta_i g(w_i.x_j + b_i) = o_j, j = 1, \cdots, N, \quad (1)$$

where w_i is the n-dimensional weight vector connecting ith hidden neuron with input neurons, β_i is the n-dimensional weight vector connecting ith hidden neuron with output neurons and b_i is the n-dimensional threshold of ith hidden neuron. $w_i.\ x_j$ is the inner product of w_i and x_j. Approximating N samples with zero error , then there exist β_i, w_i and b_i such that

$$f_N(x_j) = \sum_{i=1}^{N} \beta_i g(w_i.x_j + b_i) = t_j, j = 1, \cdots, N \quad (2)$$

Equation (2) can be rewritten compactly as

$$H\beta = T, \quad (3)$$

where H is the hidden layer output matrix as in (4), β is the output weight matrix as in (5) and T is target output matrix as in (6).

$$H = \begin{bmatrix} g(w_1.\ x_1 + b_1) & \ldots & g(w_{\tilde{N}}.\ x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1.\ x_N + b_1) & \ldots & g(w_{\tilde{N}}.\ x_N + b_{\tilde{N}}) \end{bmatrix}_{NX\tilde{N}} \quad (4)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix}_{\tilde{N}Xm} \quad (5)$$

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{NXm} \quad (6)$$

It is impossible to train real applications with zero error, since the number of hidden nodes will always be less than the number of training samples. To train SLFN with randomly fixed w_i and b_i is equivalent to obtaining the least square solution of (3) as

$$\beta = H^{\dagger}T, \quad (7)$$

where $H^{\dagger}$is the Moore-Penrose generalized inverse of H [20]. Given the training data, activation function $g(x)$ and hidden layer neuron number N, the ELM algorithm can be summarized as: Assign the input weights w_i and the bias b_i for $i = 1, \cdots, N$, next calculate the hidden layer output matrix H and finally calculate the output weight using (7).

B.2 Online Sequential ELM (OS-ELM)

OS-ELM is the online variant of basic ELM for handling online applications [21]. The basic ELM is modified in matrix H and $rank(H) = \tilde{N}$ is considered as the rank of hidden neurons. The pseudo inverse of H is derived as

$$H^{\dagger} = (H^T H)^{-1} H^T \quad (8)$$

using the fact $H^{\dagger}H = I_{\tilde{N}}$. The estimation is given as

$$\hat{\beta} = (H^T H)^{-1} H^T T, \quad (9)$$

which is the least-squares solution to $H\beta = T$. The sequential implementation of the least-square of (9) is referred as recursive least square algorithm and it is the solution of OS-ELM. OS-ELM consists of two main phases: initialization phase and sequential learning phase. The initialization phase is similar to training in basic ELM but with reduced training data. Given small initial training set $N_0 = (x_i, t_i)_{i=1}^{N_0}$, randomly assign weights w_i and bias b_i. Calculate the initial hidden layer matrix

$$H_0 = \left[h_1 \cdots h_N\right]^T. \quad (10)$$

Using the hidden layer matrix values calculate the initial output weights as

$$\beta_0 = (H_0^T H_0)^{-1} H_0^T T_0. \quad (11)$$

The initial training data are replaced with chunk by chunk online data $N_1 = (x_i, t_i)_{i=N_0+1}^{N_0+N_1}$ for online learning with continuous training. In sequential learning phase the latest hidden layer output matrix is calculated as

$$H_{k+1} = \left[h_1, \cdots, h_N\right]^T. \quad (12)$$

The output weights are calculated for new training data using recursive least square algorithm as

$$M_{k+1} = M_k - \frac{M_k h_{k+1} h_{k+1}^T M_k}{1 + h_{k+1}^T M_k h_{k+1}}, \quad (13)$$

where $M_k = (H_k^T H_k)^{-1}$. Therefore the generalized output weights are calculated as

$$\beta_{k+1} = \beta_k + M_{k+1} h_{k+1} (t_i^T - h_{k+1}^T \beta_k). \quad (14)$$

III. RELATED WORK

Various studies have been performed on the privacy and security requirements in IoT [22], [23]. The results of the study states that perimeter defenses are not sufficient for IoT environment and the security approach for IoT must analyze and interpret the massive structured and unstructured data from IoT devices to build security instincts and to provide effective defense response to threats as they evolve. Although vast research have been performed in intrusion detection systems but still it is infancy for IoT applications. At present, only few studies in the area of intrusion detection in IoT. Signature based intrusion detection system (IDS) [24] were proposed for resource constrained sensor network connected to IP. In this system signature matching for detection of attack is performed at sensor nodes which reduce the lifetime of low power sensors.

In [25], an IDS proposed is a combination of centralized and distributed architecture for IPv6 over low-power wireless personal area networks (6LoWPAN) running routing protocol for low power and lossy networks (RPL) routing protocol. It detects RPL topology inconsistency, rank attack and Denial of Service attacks (DoS). The centralized module 6Mapper at 6BR builds

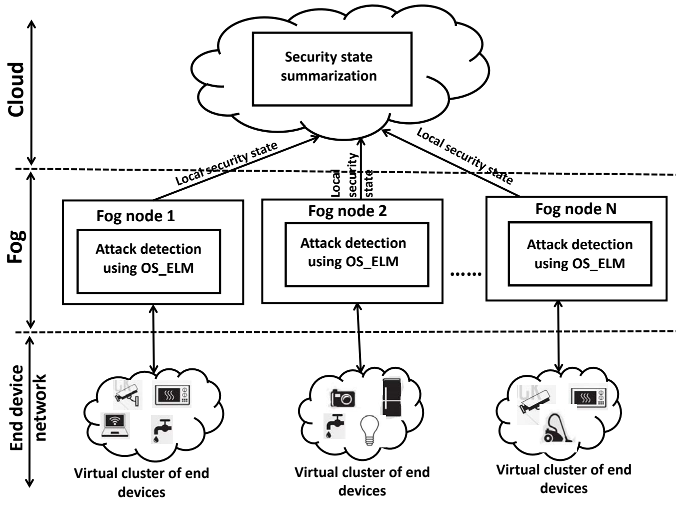
Cloud
Security state summarization
Local security state
Local security state
Local security state
Fog
Fog node 1
Attack detection using OS_ELM
Fog node 2
Attack detection using OS_ELM
......
Fog node N
Attack detection using OS_ELM
End device network
Virtual cluster of end devices
Virtual cluster of end devices
Virtual cluster of end devices

Fig. 3. Proposed Fog computing based IDS for IoT.

the RPL topology and maps it with IDS parameters for intrusion detection. The energy required for communicating the topology information for every two minutes by each sensor node is serious consideration for low power sensor networks. The earliest possible detection time of an intrusion is about four minutes not suitable for critical infrastructure. A Denial of Service attack detection architecture built on top of ebbits network [26] is capable of detecting only DoS attacks and this architecture is specific to ebbits network. The IDS [27] is used to detect wormhole attack of 6LoWPAN occurring at non leaf nodes of RPL routing topology and it does not detect the attack and attacker at leaf nodes. A specification based IDS [28] detects only the RPL topology attacks. All the above mentioned intrusion detection system are specific to 6LoWPAN based IoT which follows RPL routing scheme and this cannot be applied other IoT architectures.

Game theory based intrusion detection method was for IoT which combines the signature and anomaly based techniques [29]. A game model was created for authorized user and intruders, based on the computed Nash Equilibrium values, the intrusion detection system was activated. The drawback of this method is it cannot analyze all kinds of attacks evolving in the highly dynamic IoT environment. In [30] a real-time pattern matching algorithm using complex event processing (CEP) was proposed to identify the intrusions based on the event flow features in IoT devices. Although this approach reduces the false alarm rate, it consumes more resources since, the attack pattern rules for intrusion detection are defined as event processing model (EPM) which stores the events in the rule repository leading to high resource consumption for storing and retrieval. Hence, it is not suitable for resource constrained IoT devices. Fog based anomaly detection was proposed for IoT application [31]. In this hyperellipsoidal clustering algorithm is used to detect anomaly from heterogeneous data sources of IoT application. The results proves that fog empowered detection has higher accuracy than distributed detection but this system does not support continuous learning which is very much needed for dynamic IoT application where new kinds of cyber-attacks evolves day by day. A deep learning approach is used for detecting cyber-attacks in IoT using fog computing [32]. This approach is similar to client server model where the distributed fog nodes trains the model to detect the attack and a coordinating node also trains the global model using the parameters obtained from the distributed fog nodes. The major drawback of this method uses backpropagation stochastic gradient based technique to update the weights which takes longer time to train and update the model which is not suitable for highly dynamic IoT environment.

The proposed method is novel as it allows faster parallel learning at fog nodes from streaming data sequentially delivered in IoT environment with good generalization power. It also detects the attack at faster rate in the distributed fog nodes.

IV. PROPOSED METHOD

The proposed intrusion detection system for IoT application uses fog computing for implementing intrusion detection in distributed fashion. The proposed system consist of two modules: Attack detection at fog nodes and summarization at cloud server. The generic architecture of proposed system is shown in Fig. 3.

A. Attack Detection at Fog Nodes

The intrusion detection system at fog nodes uses OS-ELM to identify the attacks in incoming traffic from IoT virtual clusters. The IoT virtual cluster is a group of IoT devices under a single fog node. The OS-ELM algorithm classifies the incoming packet as normal or attack based on the training data. Given a network traffic training data set $D = \{x_i, y_i\}_{i=1}^{n}$ with input x as IoT traffic parameter and y as output for n observations at each fog node. The OS-ELM algorithm is trained using the training dataset D along with the initial parameters containing number of hidden layer neurons, activation function $g(x)$, random input weights w_{fi} and biases b_{fi} for the hidden nodes. Based on these inputs, the initial hidden layer output matrix is calculated from which initial output weights are calculated. This process is repeated for all consecutive incoming packets from IoT traffic and latest values of hidden layer output matrix and output weight are calculated using recursive least square algorithm. The algorithm for intrusion learning at fog node is given Algorithm 1.

B. Summarization at Cloud Server

The intrusions detected at the fog nodes are sent to cloud server to generate global view about the security state of the IoT application. The results from the fog nodes are summarized at the cloud server to analyze and visualize the current security state of the IoT application. It can be used to predict the next action of the attacker by using attacker plan recognition approaches. It can also be used to identify the multistage attacks and distributed denial of service attacks based on results from geographically distributed fog nodes. By using the global security state of IoT application, effective intrusion response mechanism can be activated. The overall working of the proposed system is shown in Fig. 4.

Algorithm 1 Intrusion learning in fog nodes.

```
Given:
    Number of hidden layer neurons $\tilde{N}_F$
    Activation function $g(x)$
    Number of classes $C_f$
Initialization Phase:
    Initial training set $N_{F0} = (x_{fi}, t_{fi})_{fi=1}^{N_F 0}$
    Assign random weights $w_{fi}$ and bias $b_{fi}$
    Calculate the initial hidden layer matrix:
        $H_{f0} = \left[h_{f1} \cdots h_{fN}\right]^T$
    Calculate the initial output weights:
        $\beta_{f0} = (H_0^T H_{f0})^{-1} H_{f0}^T T_{f0}$
Sequential Learning Phase:
    For further incoming data $N_{f1} = (x_{fi}, t_{fi})_{i=N_{f0}+1}^{N_{f0}+N_{f1}}$
    Calculate the latest hidden layer matrix
        $H_{fk+1} = \left[h_{f1} \cdots h_{fN}\right]^T$
    Calculate the latest output weight based on
    recursive least square algorithm
        $\beta_{fk+1} = \beta_{fk} + M_{fk+1} h_{fk+1} (t_i^T - h_{fk+1}^T \beta_{fk})$
```

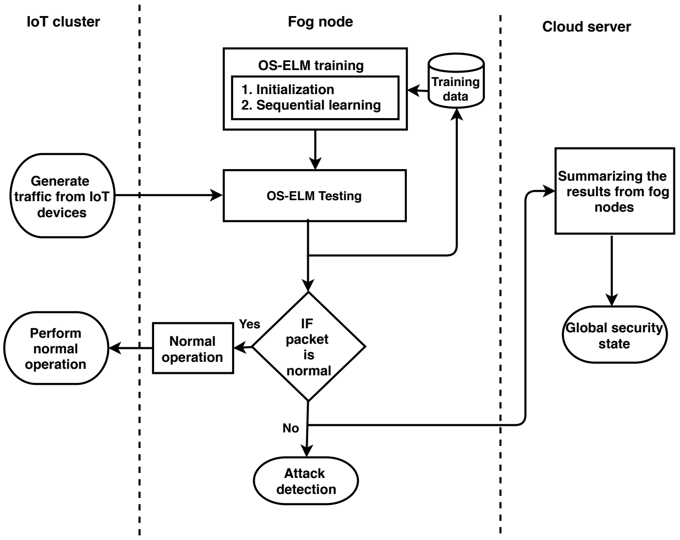

Fig. 4. Flow diagram of the proposed system.

V. EVALUATION

A. Experimental Setup

As a proof-of-concept the proposed system is implemented using computers with the configuration DUALCORE processor, 1 GB RAM, 200 GB HDD as fog nodes. These computers are connected to the Azure cloud service with the computing resource 4 X Dual-Core AMD Opteron 2218 @2.6 GHz, 8 core, 32 GB RAM, 6×146 GB HDD. OS-ELM was implemented using MATLAB (R2013a).

To identify the efficient activation function, the proposed system is implemented using different activation functions such as sigmoid, SIN and RBF. The experiment is repeated with different number of hidden layer neurons and chunk size for each considered activation functions. The sigmoid function is found to be efficient in terms of accuracy and training time when compared SIN and RBF activation function as shown in Fig. 5.

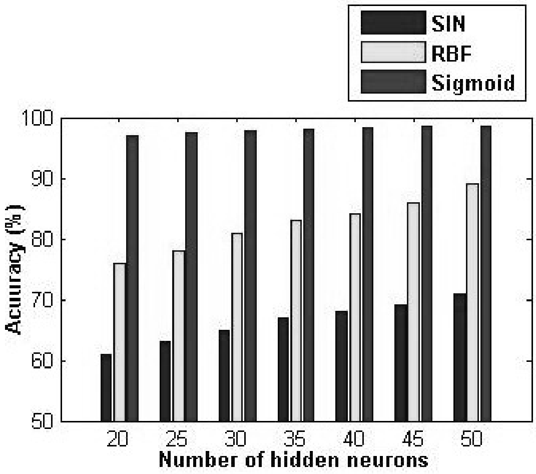

Fig. 5. Accuracy measurement in varying activation functions.

Table 1. Accuracy versus chunk size and hidden layer neurons.

Neurons	Chunk size				
	500	1000	2000	3000	4000
5	55.3	60.4	57.1	58.3	59.5
10	62.7	70.2	63.4	65.8	66.7
15	68.4	75.4	70.5	71.2	73.2
20	72.1	80.1	77.6	78.5	79.7
25	79.5	88.8	80.1	81.8	82.6
30	85.1	85.7	86.3	86.9	87.2
35	88.2	92.9	91.0	92.3	92.5
40	90.1	91.3	91.6	91.1	91.5
45	90.6	90.8	89.9	90.3	91.1
50	91.2	90.6	90.3	90.7	90.8

Table 2. Training time versus chunk size and hidden layer neurons.

Neurons	Chunk size				
	500	1000	2000	3000	4000
5	5.7	8.7	53.1	93.2	163.2
10	6.0	9.5	55.7	97.4	168.3
15	6.2	10.2	58.4	100.8	172.7
20	6.5	10.9	61.2	109.5	179.1
25	6.9	11.2	65.3	120.1	188.3
30	7.5	11.6	70.1	110.5	173.2
35	7.3	12.1	64.1	112.5	206.2
40	8.6	23.1	62.5	115.3	264.1
45	7.9	16.3	66.5	210.2	282.3
50	7.8	16.5	64.3	208.6	289.4

Table 1 presents the accuracy for varying number of hidden layer neurons and chunk sizes. From the results shown in Table 1, it is inferred that the accuracy of the results improves at faster rate with increase in number of hidden layer neurons than the large chunk size. Hence the number of hidden layer neurons has greater influence on improving accuracy than the chunk size. Table 2 presents the training time of the proposed model by varying number of hidden layer neurons and chunk sizes. From the results shown in Table 2, it is known that the training time of the model increases at higher rate for larger chunk size than for more number of hidden layer neurons. Therefore, it is inferred that smaller chunk size trains the model at faster rate.

From the experimental results, it is observed that to model

Table 3. Number of records for Binary classification.

Records	Training	Test
Normal	67343	9711
Attack	8630	12833
Total	125973	22544

Table 4. Number of records for Multi-class classification.

Records	Training	Test
Normal	67343	9711
Dos	45927	7458
Probe	11656	2754
R2L	995	2421
U2R	52	200
Total	125973	22544

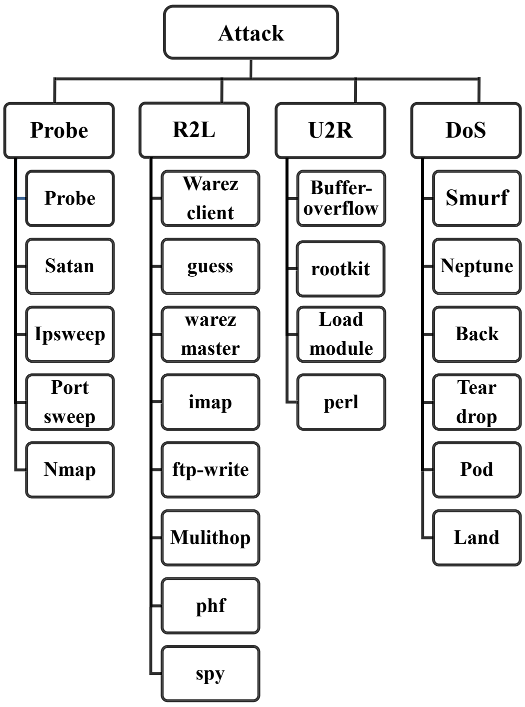

Fig. 6. Attack types NSL-KDD matched with IoT environment.

the OS-ELM efficiently is to use higher number of hidden layer neurons and smaller chunk size. According to the experimental results, 35 hidden layer neurons and 1,000 chunk size is used to model the OS-ELM in the proposed method to yield best result.

B. Dataset

NSL-KDD [33] benchmark dataset for intrusion detection was used to evaluate the proposed system. The dataset contains separate training and test records. The training set contains 125,973 records and the test set contains 22,544 records with 41 features. The dataset can be modeled for binary classification with 2-class and multi-class classification with 4-class as shown in Table 3 and Table 4. The major attacks found in IoT environment is matched with attacks in NSL-KDD dataset [32] as shown in Fig. 6.

VI. RESULTS AND DISCUSSION

In this section, the proposed system is evaluated using the experimental results in terms of accuracy, and response time.

Table 5. Accuracy measurement for binary classification

Overall Accuracy	Detection Rate	False Alarm Rate	TPR	FPR
97.36	96.92	1.53	97.72	0.37

Table 6. Accuracy measurement for Multi-class classification.

Overall Accuracy	Detection Rate	False Alarm Rate
96.54	96.08	1.84

A. Accuracy

The detection accuracy of the proposed system is measured based on accuracy, detection rate and false alarm rate as given in (15)–(17) using the following confusion matrix containing False Positive (FP), False Negative (FN), True Positive (TP) and True Negative (TN) values.

		Predicted	
		Attack	Normal
Actual	Normal	TN	FP
	Attack	FN	TP

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

$$DetectionRate = \frac{TP}{TP + FN} \tag{16}$$

$$FalseAlarmRate = \frac{FP}{TN + FP} \tag{17}$$

Table 5 gives the detection accuracy measurement for binary classification and Table 6 gives the detection accuracy measurement for multi-class classification by the proposed method.

The performance of the proposed system is compared with some existing algorithms such as artificial neural network (ANN), Navie Bayes and standard ELM in terms of detection accuracy. Table 7 shows the performance comparison of different algorithms for multi-class classification and Table 8 for binary classification. From the results in Table 7 and Table 8, it is known that the proposed system outperforms the other compared algorithms.

The proposed system achieves 97.36% of accuracy with reduced false alarm rate of 0.37%. The major advantage of the proposed system is that it can incorporate new data online for learning which is not possible with the other compared algorithms.

B. Response Time

To compare the efficiency of the proposed fog computing based intrusion detection system in terms of response time, the attack detection module of proposed system is also implemented in the Azure cloud service as centralized system. The latency of the proposed fog based detection system and existing cloud based detection are measured to show the impact of OS-ELM in fog computing for intrusion detection. The response time for the

Table 7. Performance comparison for multi-class classification.

Algorithm	Normal		Probe		DoS		U2R		R2L	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Navie Bayes	78.37	7.48	69.50	3.21	91.61	2.23	70.6	5.07	80.20	4.79
ANN	97.40	7.48	66.46	0.27	97.13	1.97	51.5	2.87	70.15	0.88
ELM	97.52	6.74	76.07	1.53	94.80	3.46	50.5	7.75	69.25	0.22
Proposed	98.63	4.74	84.2	0.81	96.61	2.32	53.81	0.52	71.87	0.20

Table 8. Performance comparison for binary classification.

Algorithm	Accuracy	TPR	FPR
Navie Bayes	87.29	91.02	13.35
ANN	95.89	96.37	4.32
ELM	95.45	96.73	3.92
Proposed	97.36	97.72	0.37

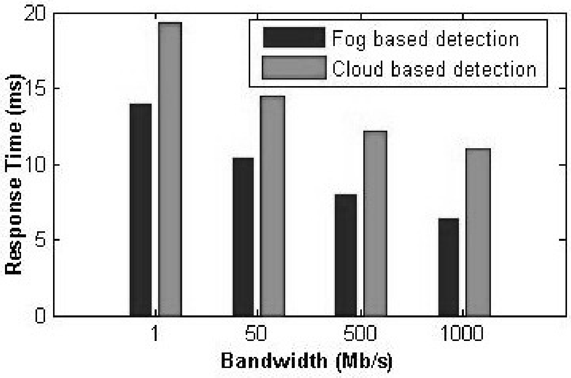

Fig. 7. Comparison between fog based detection and cloud based detection in response time.

proposed system is compared with the cloud based implementation with varying network bandwidths as shown in Fig. 7. The response time to report cyber-attack is nearly 25% less in the proposed fog based approach when compared to cloud based implementation since the fog nodes are closer to the end devices.

Overall, the proposed cognitive fog based intrusion detection approach provides better accuracy in short training time with reduced response time along with the advantage of online learning.

VII. CONCLUSION

In this paper, intrusion detection was performed on the fog nodes for IoT application. The intrusion detection was implemented by providing intelligence to local fog nodes using OS-ELM algorithm. The online learning capability of OS-ELM learns the dynamic environment of IoT application quickly. The local fog nodes detects the attack from the traffic generated from IoT environment and reports to cloud server to summarize the global security state of IoT application. The experimental results show that the fog nodes detect the attack at 25% faster rate than the cloud based implementation with low false alarm rate.

The proposed work provides an initial step for designing intrusion detection system for large-scale IoT application using fog computing. The future work is to predict the next action of the attacker from the result of the proposed system to protect the IoT application proactively.

REFERENCES

[1] H. Cai *et al.,* “IoT-based configurable information service platform for product lifecycle management,” *IEEE Trans. Ind. Inform.,* vol. 10, no. 2, pp. 1558–1567, May 2014.

[2] R. H. Weber, “Internet of Thingsˇ New security and privacy challenges,” in *Comput. Law & Security Rev.,* vol. 26, no. 1, pp. 23–30, Jan. 2010.

[3] X. Sun and C. Wang, “The research of security technology in the Internet of Things,” *Advances in Computer Science, Intelligent System and Environment*, Springer, Berlin, Heidelberg, 2011, pp. 113–119.

[4] C.-W. Ten, G. Manimaran, and C.-C. Liu, “Cybersecurity for critical infrastructures: Attack and defense modeling,” *IEEE Trans. Syst., Man, Cybern.,*, vol. 40, no. 4, pp. 853–865, July 2010.

[5] L. A. Barroso, J. Clidaras, and U. Holzle, “The datacenter as a computer: An introduction to the design of warehouse-scale machines,” *Synthesis Lectures on Comput. Archit.*, vol. 8, no. 3, pp. 1–154, July 2013.

[6] K. Heires, “Budgeting for latency,” *Securities Industry News*, vol. 22, no. 1, 2010.

[7] J. B. Predd, S. R. Kulkarni, and H. V. Poor, *Distributed Learning in Wireless Sensor Networks*, John Wiley & Sons: Chichester, UK, 2007.

[8] Y. Liu, Y. Han, Z. Yang, and H. Wu, “Efficient data query in intermittently-connected mobile ad hoc social networks” *IEEE Trans. Parallel and Distrib. Syst.*, vol. 26, no. 5, pp. 1301–1312, May 2015.

[9] Y. Liu *et al.,* “Optimal online data dissemination for resource constrained mobile opportunistic networks” *IEEE Trans. Veh. Technol.,* vol. 66, no. 6, pp. 5301–5315, June 2017.

[10] C. M. Medaglia and A. Serbanati, “An overview of privacy and security issues in the Internet of Things,” in *The Internet of Things*, Springer, pp. 389–395, 2010.

[11] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog computing: A platform for Internet of things and analytics,” in *Big Data and Internet of Things: A Roadmap for Smart Environments,* Springer International Publishing, pp. 169–186, 2014.

[12] C.-W. Stojmenovic, G. Manimaran, and C.-C Liu, “Fog computing: A cloud to the ground support for smart things and machine-to-machine networks,” in *Proc. IEEE ATNAC*, 2014, pp. 117–122.

[13] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[14] Y. Liu *et al.,* “Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach,” *Comput. Netw.*, vol. 129, pp. 399–409, 2017.

[15] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, “Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset,”*IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 184–208, First Quater 2016.

[16] D. E. O’Leary, “‘BIG DATA’, THE ‘INTERNET OF THINGS’ AND THE ‘INTERNET OF SIGNS’,” *Intell. Syst. Account., Finance Manag.*, vol. 20, no. 1, pp. 53–65, Mar. 2013.

[17] T. T. Nguyen and G. Armitage, “A survey of techniques for internet traffic classification using machine learning,” *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, Fourth Quater 2008.

[18] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006.

[19] F. Bonomi, M. Rodolfo, Z. Jiang, and SateeshAddepalli, “Fog computing and its role in the Internet of things,” in *Proc. ACM MCC*, 2012, pp. 13–16.

[20] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *IEEE Trans. Syst., Man, Cybern., Syst. Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[21] G. B. Huang, N. Y. Liang, H. J. Rong, P. Saratchandran, and N. Sundararajan, "On-Line Sequential Extreme Learning Machine," *Comput. Intell.*, vol. 2005, pp. 232–237, 2005.

[22] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of things," *Comp. Netw.* vol. 57, no. 10, pp. 2266–2279, Oct. 2013.

[23] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015.

[24] S. O. Amin, M. S. Siddiqui, C. S. Hong, and S. Lee, "RIDES: Robust intrusion detection system for IP-based ubiquitous sensor networks," *Sensors*, vol. 9, no. 5, pp. 3447–3468, May 2009.

[25] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad hoc Netw.*, vol. 11, no. 8, pp. 2661–2674, Aug. 2013.

[26] P. Kasinathan, C. Pastrone, M. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," in *Proc. IEEE WiMob*, 2013, pp. 600–607.

[27] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in Internet of Things," *Int. J. Comput. Appl.,*, vol. 121, no. 9, pp. 1–9, Sept. 2015.

[28] A. Le, J. Loo, Y. Luo, and A. Lasebae, "Specification-based IDS for securing RPL from topology attacks," in *IFIP WD*, 2011, pp. 1–3.

[29] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology," in *Proc. IEEE ICC*, 2016, pp. 1–6.

[30] C. Jun and C. Chi, "Design of complex event-processing IDS in internet of things," in *Proc. IEEE ICMTMA*, 2014, pp. 226–229.

[31] L. Lyu, J. Jin, S. Rajasegarar, X. He, and M. Palaniswami, "Fog-empowered anomaly detection in Internet of Things using hyperellipsoidal clustering," *IEEE Internet Things J.*, vol. 4, no. 5, Oct. 2017.

[32] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," in *Future Generation Comput. Syst.*, vol. 82, pp. 761–768, May 2018.

[33] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE CISDA*, 2009, pp. 1–6.

S.Prabavathy received the M.E. degrees in Computer Science and Engineering from Anna University, India. She is currently pursuing the Ph.D. degree in the Department of Computer Science and Engineering, Thiagarajar College of Engineering, India. Her research interests include Internet of things and wireless network security.

K.Sundarakantham received her Ph.D. in Information and Communication Engineering from Anna University, India in 2010. She is currently a Associate Professor at Thiagarajar College of Engineering, India. Her research interests include network security, natural language processing.

S.Mercy Shalinie received her Ph.D. degree in Information and Communication Engineering from Madurai Kamaraj University, India in 1999. She is currently a Professor at Thiagarajar College of Engineering. Her research interests include machine learning and information security.