Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7472095, 10 pages
https://doi.org/10.1155/2018/7472095

Research Article

Sample Selected Extreme Learning Machine Based Intrusion Detection in Fog Computing and MEC

Xingshuo An,[1] Xianwei Zhou,[1] Xing Lü,[1] Fuhong Lin,[1] and Lei Yang[2]

[1]*School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China*
[2]*Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV, USA*

Correspondence should be addressed to Fuhong Lin; fhlin@ustb.edu.cn

Received 5 September 2017; Accepted 16 November 2017; Published 14 January 2018

Academic Editor: Shangguang Wang

Fog computing, as a new paradigm, has many characteristics that are different from cloud computing. Due to the resources being limited, fog nodes/MEC hosts are vulnerable to cyberattacks. Lightweight intrusion detection system (IDS) is a key technique to solve the problem. Because extreme learning machine (ELM) has the characteristics of fast training speed and good generalization ability, we present a new lightweight IDS called sample selected extreme learning machine (SS-ELM). The reason why we propose "sample selected extreme learning machine" is that fog nodes/MEC hosts do not have the ability to store extremely large amounts of training data sets. Accordingly, they are stored, computed, and sampled by the cloud servers. Then, the selected sample is given to the fog nodes/MEC hosts for training. This design can bring down the training time and increase the detection accuracy. Experimental simulation verifies that SS-ELM performs well in intrusion detection in terms of accuracy, training time, and the receiver operating characteristic (ROC) value.

1. Introduction

With the rapid development of smart devices, we are embracing an era of the Internet of Things (IoT). The IoT applications require mobility support, geodistribution, location awareness, and low latency. However, it is difficult for cloud computing to meet these requirements. Edge paradigms such as fog computing (FC) and mobile edge computing (MEC) are proposed to overcome the above challenging issues [1–3]. The nodes, which could perform computing tasks, are called fog nodes/MEC hosts in FC and MEC hosts in MEC, which can provide low-latency service. FC and MEC are somewhat different. For example, MEC hosts are typically deployed by mobile service providers [4], and fog nodes are made up of edge servers or devices with communication and computing power. However, their network model and many features are similar [5]. They both extend cloud computing to the edge. In this work, we study a generalized network model that can be applied in FC and MEC.

Generally, the infrastructure of FC or MEC consists of three layers: the cloud server, fog node/MEC host layer, and user device layer, where fog nodes/MEC hosts are computing nodes unique to FC and closer to the user [6]. The purpose of using fog nodes/MEC hosts is to provide service with lower latency, more flexible access, and more secure network communication to network users [7].

As a new network paradigm, FC/MEC presents several challenges in network stability and performance. In FC/MEC, most of the terminal devices are resource-constrained; for example, the terminal connected to a fog node/MEC host can be a smart home appliance, a smart phone, an unmanned aerial vehicle (UAV), or a VR device [8]. Threats may come from various aspects for a network with such characteristics, such as denial of service (DoS), man in the middle (MIM), rogue gateway, privacy leakage, and service manipulation [5].

Since there are many attacks against the FC network, the benefits of FC are diminished by the damage caused by malicious attacks if no proper security and privacy protection mechanisms are available. To effectively handle security threats in FC infrastructure and to minimize the associated damage, we focus on the security precautions. One way turns to the intrusion detection system (IDS). An

IDS is an important security barrier that can quickly detect intrusion and security risks in the network [9, 10]. The detection algorithm is one of the most important parts in IDS. An intrusion detection algorithm suitable for critical infrastructures can accurately and quickly detect intrusions [11]. Therefore, to adapt to the new paradigm of FC, the present study focuses on the intrusion detection scheme to ensure the security and meet new challenges.

This paper will first analyze the security problems in FC/MEC. According to fog nodes/MEC hosts' resource-constrained characteristics, an intrusion detection scheme with a lightweight algorithm is proposed. This design takes full advantage of the computing resources of fog nodes/MEC hosts. It deploys classifiers for intrusion detection on fog nodes/MEC hosts and stores training data sets in the cloud server. The contribution of this work is as follows:

(1) According to the network characteristics of fog, this paper proposes a general scheme of intrusion detection system in FC/MEC environment.

(2) Based on the traditional ELM, this paper innovatively adds the sample selection process in the training phase. This design is used to improve the classifier algorithm so that it can become more lightweight when fog nodes/MEC hosts perform tasks.

(3) We compare the performance of BP, SVM, ELM, and SS-ELM as intrusion detection classifiers and verify the superiority of SS-ELM.

The paper is organized as follows. In Section 2, we will review related works on FC/MEC security and extreme learning machine- (ELM-) based IDSs. In Section 3, we will discuss the requirements and schemes of an IDS in an FC/MEC environment. In Section 4, we will introduce an ELM-based intrusion detection algorithm for FC/MEC. Section 5 will describe the simulation to verify the algorithm. The paper is concluded in Section 6.

2. Related Works

Existing research on FC/MEC security mainly focuses on analyzing security threats in FC/MEC and the corresponding countermeasures [7, 12–14].

Dsouza et al. [8] first described the advantages of FC as a new paradigm and elaborated the security problems in several different scenarios of FC, including intelligent instrument authentication, MIM attacks, and privacy issues in FC/MEC. Yi et al. [7] suggested that FC security is a problem worthy of studying, particularly the aspects of authentication, access control, intrusion detection, and user privacy issues. FC/MEC's possible contribution to network security has been discussed from the perspective of computer forensics [13, 14]. In particular, Wang et al. compared cloud computing and FC/MEC in terms of security [14] and noted that FC/MEC and honeypot technology can be integrated into cloud computing forensics to protect the security of cloud servers.

There are some studies related to intrusion detection in FC/MEC [5, 15, 16]. The security of edge computing has previously been reviewed [5], and the author suggested that FC/MEC is a paradigm of edge computing. In that paper, various security issues encountered in edge computing and security mechanisms were discussed in detail. In addition, the author noted that a fog IDS should meet the defense needs of both local fog nodes/MEC hosts and the entire network, be able to detect lasting threats, and have an intrusion prevention mechanism that can operate autonomously. It has been noted that an IDS can be deployed on the fog node/MEC host side (host-side detection) or fog network/MEC network side (network-side detection) to monitor host-side intrusions or network-side attacks [15]. A new cloudlet mesh security architecture was proposed based on cloudlets [16], and this architecture is used to protect the mobile cloud network. If cloudlet servers are treated as fog nodes/MEC hosts, then such architecture constitutes a host-side IDS.

FC/MEC is an extension computing paradigm of cloud computing. Therefore, we refer to the virtual machine (VM) IDS in cloud computing in the present study. For example, an intrusion detection scheme deployed in a cloud VM was introduced, which was based on the Smith-Waterman algorithm to collect and detect the anomalous behavior of VMs [17]. This algorithm improved the system's detection efficiency and thus reduced the detection time to some extent. In a previous report [18], attacking threats in federated cloud environments were analyzed, and a detection system for misuse cataloging was proposed. An approach named VAED which is deployed in VMs was given [19]. Its results seem to be promising for detecting evasion-based malware attacks.

Many studies focused on intrusion detection. However, since the scenarios of these studies are in the cloud, there is no focus on IDS in resource-constrained environments, especially FC/MEC. That is to say, lightweight IDS is worth studying in FC/MEC.

Intrusion detection techniques include statistics-based detection, data-mining-based detection, expert-system-based detection, support vector machine- (SVM-) based detection, genetic-algorithm-based detection, and neural-network-based detection. Considering the heterogeneity and dynamic characteristics, rich network resources, and complex attack behaviors in FC/MEC, we will use the ELM-based algorithm to design and implement an intrusion detection scheme in FC/MEC in this paper. The ELM is a neural network method. At the end of this section, we will review the application of an ELM in intrusion detection.

The ELM was first proposed by Huang et al. [20]. This algorithm uses a random mechanism to reduce the number of parameters to set and select and thus is a simple and fast learning algorithm. ELMs have been widely used in many fields since their proposal, including intrusion detection. An ELM was used for intrusion detection and showed greater accuracy in intrusion detection than that of an SVM [21]. Ye and Yu proposed an intrusion detection method in which each class was combined into an ensemble classifier using a one-to-all strategy [22]. A weighted ELM was also proposed for intrusion detection [23]. Cai et al. proposed a new fusion method which combined with a ball vector machine (BVM), ELM, and a back propagation (BP) neural network for intrusion detection. This method has shown strong performance in detection accuracy and false positive rate [24].

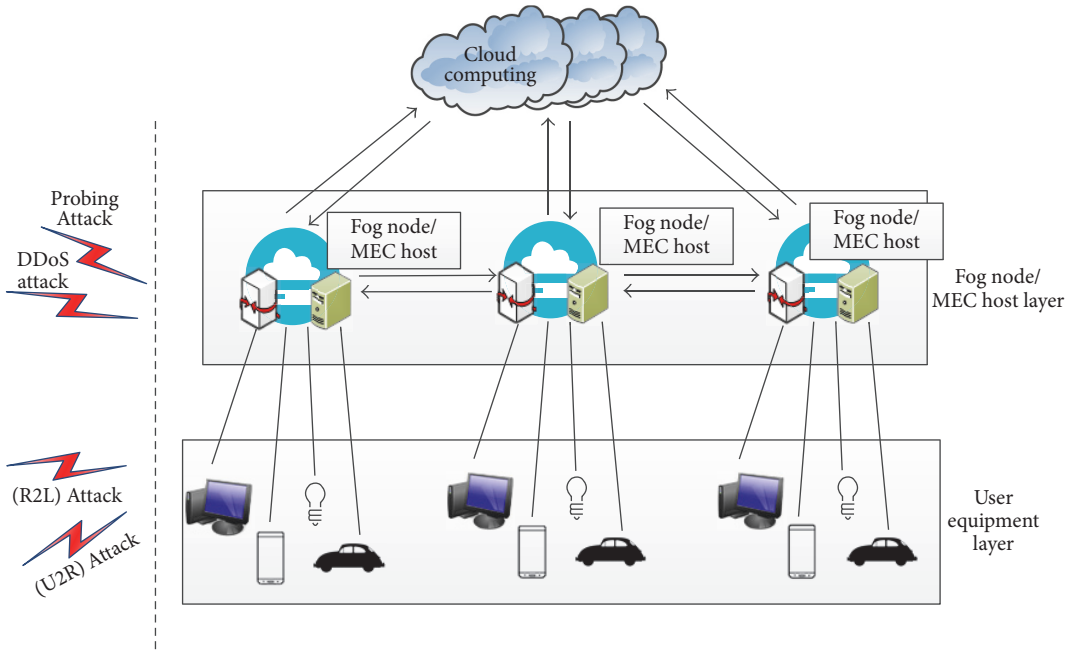

Figure 1: FC/MEC network security threats.

3. Intrusion Detection Scheme in FC/MEC

3.1. FC/MEC Network Architecture and Its Security Threats. Fog networks/MEC networks consist of user devices, fog nodes/MEC hosts, and cloud computing centers. The typical differences from the traditional cloud computing paradigm lie in the following. (1) A cloud computing center manages and controls multiple fog nodes/MEC hosts. (2) Fog nodes/MEC hosts located at the edge of the network between the network center and the user have a certain computing power. (3) Fog nodes/MEC hosts can handle computing tasks and can provide network services to user devices directly.

User devices are heterogeneous and include smart sensors, smart phones, UAVs, and other networkable terminals. We hypothesize that fog nodes/MEC hosts and terminal devices in a fog network/MEC network are all likely to be attacked and are thus not reliable. Considering the fog network/MEC network structure, we present the intrusion diagram of a network attack in FC/MEC as follows.

As shown in Figure 1, fog nodes/MEC hosts/MEC hosts in the FC/MEC layer provide a network connection service to user devices and also exchange data with the cloud computing server. Security threats come from direct or indirect attacks on fog nodes/MEC hosts/MEC hosts and user devices. In FC/MEC, network nodes are widely distributed and lack effective physical protection. They are vulnerable to invasion by malicious attackers. The detection ability and response speed of traditional IDS have been unable to meet the needs of detection of fog environment. Therefore, the establishment of an efficient intrusion detection system in FC/MEC environment has become an important research direction of FC/MEC data security.

In a fog network/MEC network, the detection of intrusion is the process in which the IDS determines whether the host state or network connection data are legal through a security audit. In addition, the fog network/MEC network IDS deployment is related to the security of the entire system. Both the fog nodes/MEC hosts/MEC hosts and the cloud computing center can compute and store data. In order to distribute the computational load of IDS reasonably, we decided to deploy IDS in the fog network/MEC network using cloud servers' storage capability and fog nodes/MEC hosts' limited computing power. There are some reasons for the deployment scheme. (1) From the perspective of detection efficiency, if an IDS is deployed in the cloud computing center only, then the network communication cost will increase because all data must be sent to and processed by the cloud server. Furthermore, the computation load of the cloud server will also increase. (2) From the perspective of security, the cloud center as the central node of the entire network will easily become the target of attacks. Once the central node is attacked, data obtained from fog node/MEC host detectors become unreliable. (3) Intrusion detection data sets are needed for training if a neural network is used for intrusion detection. Training data sets are usually large and thus will greatly increase the training time if the training is conducted only by the cloud computing center, compromising the training efficiency. (4) Due to the heterogeneity of user devices, different fog nodes/MEC hosts can result in greatly different network environments in FC/MEC. For example, certain fog nodes/MEC hosts mostly provide networking services to cars and thus communicate with vehicle sensors or car control systems, while other fog nodes/MEC hosts serve primarily smartphones. Moreover,

the customer population of a fog node/MEC host is also dynamically changing: a user device may join or exit a fog node/MEC host at any time.

3.2. Intrusion Detection Scheme in FC/MEC. As described in Section 3.1, the relationship between a fog node/MEC and user devices is of highly dynamic variability. To adapt to the dynamic fog network/MEC network and to protect the security and high efficiency of fog IDS, we propose a general architecture for FC/MEC intrusion detection systems. This scheme takes advantage of the computing capability and storage capacity of fog nodes/MEC hosts and cloud servers. The architecture includes data processing, detection, and knowledge mining in IDS. The scheme is divided into 6 layers according to the data flow of fog network/MEC network.

User Equipment Layer. FC/MEC user equipment is heterogeneous, including personal computers (PC), intelligent terminals, vehicles in Internet of Vehicles (IOV), and sensors. These devices may access different fog nodes/MEC hosts through different protocols.

Network Layer. It provides link services for different fog network/MEC network protocols. It is responsible for receiving data transmitted from the network and user equipment layer and packing and transmitting data.

Data Processing Layer. The primary role is to deal with network intrusion data from user equipment, including packet capture, data cleaning, data filtering, and data preprocessing. The task of this layer is done on fog nodes/MEC hosts.

Detection Layer. After the intrusion data is pretreated, the detection layer is sent to the classifier to detect the attacks. (1) It uses the classifier to analyze the intrusion data to determine what kind of attack it belongs to. (2) This layer is equipped with a security monitoring system to monitor the host state of fog nodes/MEC hosts. (3) At the same time, it manages and records network protocols and logs for fog network/MEC network packets. After collecting and storing a certain amount of data, the fog node/MEC host sends the test results and the relevant logs to the cloud server. In the IDS of FC/MEC, the detection layer is the core layer, and the detection task is completed in fog node/MEC host.

Analysis Layer. This layer is deployed in the cloud computing center. Its main function is to analyze the results and related logs reported by fog nodes/MEC hosts. It can integrate information into knowledge and generate some application services. For example, this layer can generate security status reports for a fog node/MEC host and store them on the cloud server.

Management Layer. Management in the cloud server is mainly responsible for (1) uniform monitoring and management of the safety status of fog nodes/MEC hosts, (2) the decision and response of the intrusion detection system, and (3) the data and logs of the intrusion fog nodes/MEC hosts that can be stored so as to facilitate intrusion forensics.

As shown in Figure 2, the intrusion detection classifier of the detection layer is the most important part of the system for intrusion data processing. It is related to the subsequent cloud server intrusion response to ensure the security of the system. Therefore, our work focuses on the detectors of intrusion attacks. Deploying detectors on fog nodes/MEC hosts can make full use of the computation capability and storage capacity of fog nodes/MEC hosts for intrusion detection. However, large-scale data cannot be processed or stored due to the limited computation and storage capability of fog nodes/MEC hosts. In addition, as described in Section 3.1, simply deploying the detector on fog nodes/MEC hosts cannot meet the requirements of a dynamic network. The cloud server has a larger storage space than fog nodes/MEC hosts; thus, the cloud server could store a large number of training sets and training set selecting rules. It can distribute training sets to fog nodes/MEC hosts dynamically for training so that different fog nodes/MEC hosts become specific and dynamic. Figure 3 shows the workflow of this scheme's steps as follows:

(1) The terminal accesses the fog node/MEC host and establishes a connection with the fog node/MEC host. The network environment of each fog node/MEC host is different.

(2) Fog node/MEC host cluster is controlled by the cloud server, so the total training set is managed by the cloud server. Store the total training set in the cloud server, and select a sample according to rules. The premise of selecting samples is that the cloud server has a perception of the network environment of fog nodes/MEC hosts.

(3) The cloud server sends the selected training set to the fog node/MEC host.

(4) Complete the training process on the fog node/MEC host.

(5) The communication between the fog node/MEC host and the terminal generates a data stream. Intrusion detection is performed on fog nodes/MEC hosts.

At the fog node/MEC host layer, intrusion detection and real-time response are required to ensure the safety and reliability of fog nodes/MEC hosts. For example, when external intruders attack a fog node/MEC host, the fog node/MEC host should be able to detect the type of intrusion and issue an alarm to prevent intrusion. Because of its computing and storage capabilities, a fog node/MEC host can complete the computation task locally to ensure short latency. We can deploy lightweight and energy-efficient algorithms on fog nodes/MEC hosts for intrusion detection.

4. SS-ELM for FC/MEC Intrusion Detection

In FC/MEC, fog nodes/MEC hosts are responsible for providing network services for massive heterogeneous intelligent devices, as the members of smart devices that provide services by a fog node/MEC host are dynamically changing. In other words, a fog node/MEC host may release a device's service or establish a connection with the device at any time. This creates a dynamic network security threat in FC/MEC. For the intrusion detection system of the fog node/MEC host, it is necessary to train the classifier with the new targeted

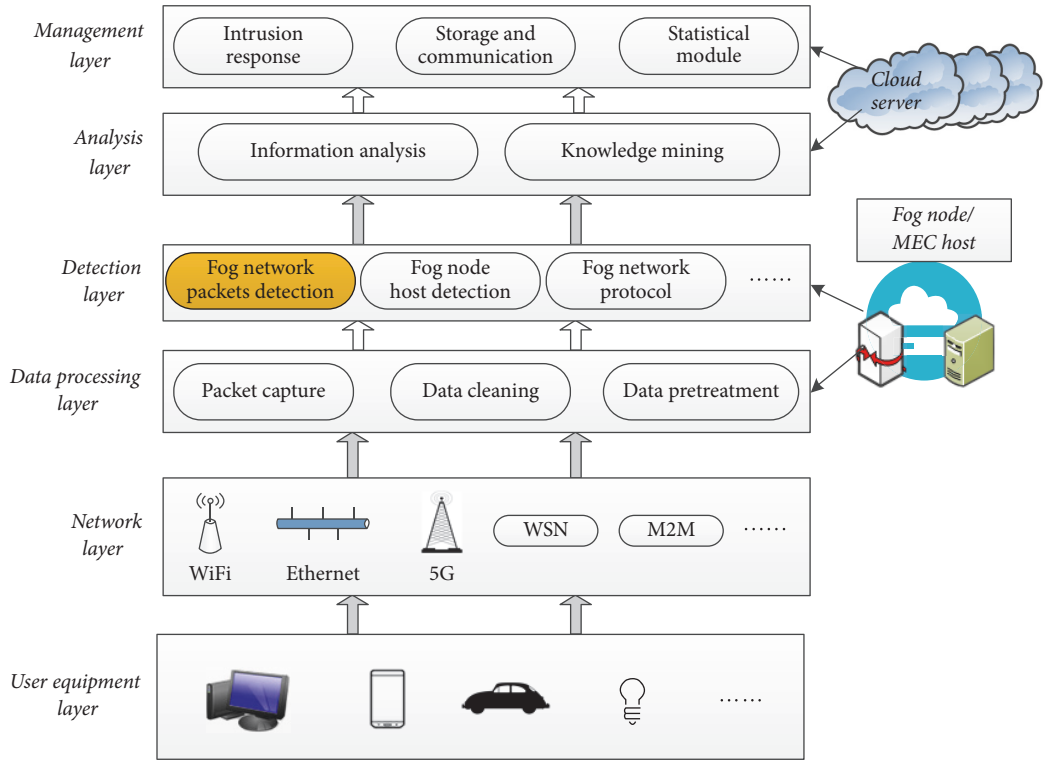

Figure 2: General architecture for FC/MEC intrusion detection systems.

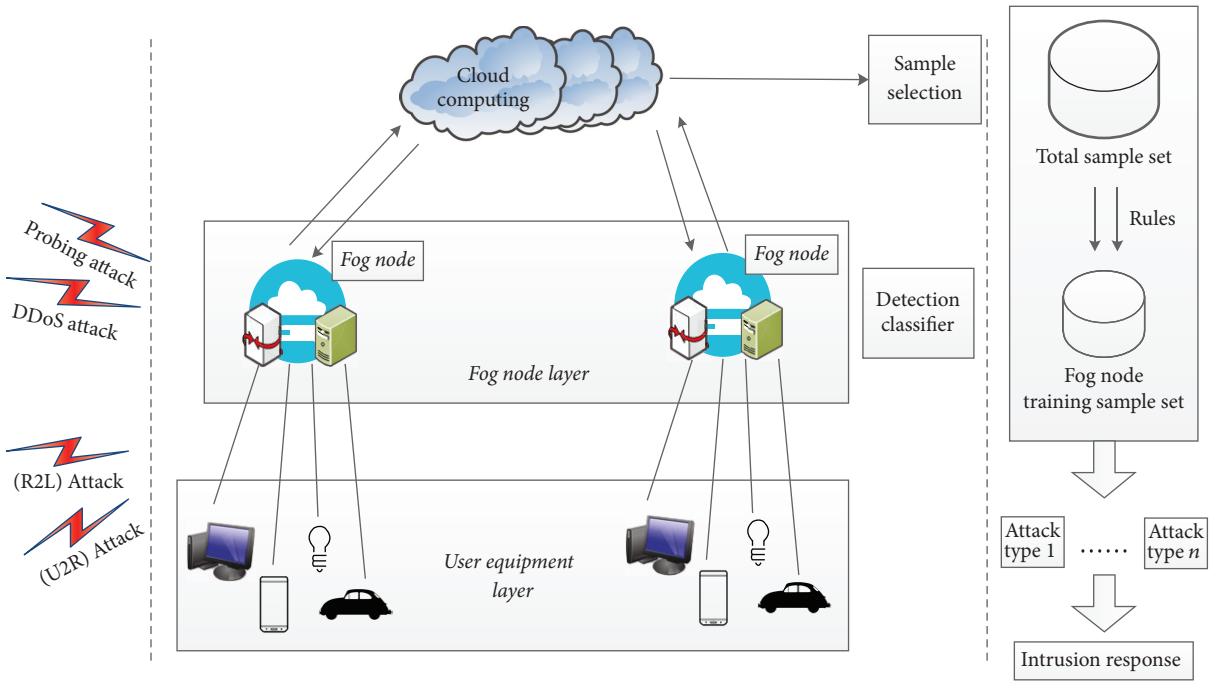

Figure 3: Fog network/MEC network intrusion detection scheme.

training set in real time. Many of the previous intrusion detection classifier algorithms have long training time [19]. Due to the limited computing power and storage capacity of fog nodes/MEC hosts, these intrusion detection schemes are not suitable for FC/MEC paradigm. Therefore, we need to study the application of lightweight intrusion detection algorithm on fog nodes/MEC hosts.

A well-suited classifier algorithm is the most important factor for a fog node/MEC host IDS. The solution for the ELM is straightforward and can be found by finding the minimum norm of a least square problem, which can ultimately be transformed into a Moore-Penrose generalized inverse problem involving a matrix. Therefore, this algorithm is characterized by a small number of training parameters, fast training speed, and satisfactory generalization ability, and thus it is suitable to be used as a classifier algorithm and deployed on fog nodes/MEC hosts. To adapt to the dynamic process of fog nodes/MEC hosts and to reduce the training time of fog nodes/MEC hosts, we select training samples according to the network characteristics and training characteristics of each fog node/MEC host. This chapter describes an algorithm for an SS-ELM for FC/MEC. We describe the principles and processes of traditional ELM algorithms in Section 4.1; furthermore, Section 4.2 describes our proposed SS-ELM algorithm. As described in Chapter Three, the algorithm is characterized by adding sample selection in the cloud server and using the deployed ELM classifier to detect network attacks on fog nodes/MEC hosts.

4.1. ELM Algorithm. ELM was first proposed by Huang of Nanyang Technology University [20]. In this subsection, the basic principles of ELM will be introduced as follows.

The ELM has a more simple and valid study mode relative to the traditional BP algorithm. Therefore, the learning speed of ELM is much faster than that of BP. In addition, traditional BP usually has some problems such as having a local minimum, being inappropriate, and having an overfitting learning rate. It, therefore, usually adopts some special methods to avoid these problems. The ELM does not need to consider these tiny problems and can get the solution of the problem directly. It is simpler than the algorithm of feedforward neural networks. Hence, ELM is more convenient and practical than the traditional ANN model. The calculation construction of ELM algorithm in this investigation is shown in Figure 4.

For the training data sample $(\vec{x}, y)$, the output function expression of single-hidden layer ahead with L hidden layer neurons to the neural network is

$$f_L(x_l) = \sum_{i=1}^{L} \beta_i G(a_i, b_i, x), \tag{1}$$

where a_i and b_i are weight vector connecting i_{th} hidden neuron and the input neurons, β_i shows the output weight connecting the ith hidden layer with model output, and $G(a_i, b_i, x)$ shows the i_{th} hidden layer relative to hidden layer node output of the sample $(\vec{x}, y)$; the expression of $G(a_i, b_i, x)$ is

$$G(a_i, b_i, x) = g(a_i \cdot x + b_i). \tag{2}$$

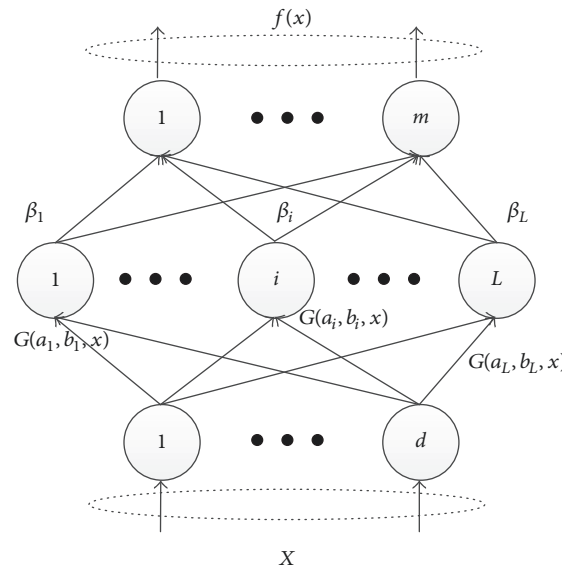

Figure 4: The architecture of the ELM neural network model in this work.

In the expression, $g : R \to R$ is the activation function and $a_i \cdot x$ is the inner product of input weight a_i and sample X in R^n. Consider N inequality data samples $\{(x_i, t_i)\}_{i=1}^{N} \subset R^n \times R^m$, if the single-hidden layer neural network of L hidden layer neurons approaches N inequality data sample with zero error; that is to say, there are a_i, b_i, and β_i, $i = 1, \ldots, L$, which makes formula (2) written as

$$f_L(x_j) = \sum_{i=1}^{L} G(a_i, b_i, x) = y_j, \quad j = 1, \ldots, N. \tag{3}$$

The shorthand formula of (3) is as follows:

$$\mathrm{H}\beta = Y, \tag{4}$$

where H is called the hidden layer output matrix; the corresponding i_{th} column shows the output of i_{th} hidden neural layer corresponding to the input $x_1, x_2, x_3, \ldots, x_n$ and the j_{th} line shows all the hidden layers relative to output amount of inputting x_j.

$$\begin{aligned} &\mathrm{H}_0(a_1, \ldots, a_L, b_1, \ldots, b_L, x_1, \ldots, x_N) \\ &= \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \cdots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix}_{N \times L}, \end{aligned} \tag{5}$$

β is output weight, $\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}$, and $Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N \times m}$.

In most cases, the number of hidden nodes is much smaller than the number of training samples ($L \ll N$), making it challenging for the constructed single-hidden-layer neural network (a total of L neurons in the hidden layer) to infinitely approach these mutually different samples with zero

error, resulting in errors between the network output of the training samples and the actual output. In this case, (4) can be rewritten as

$$\mathrm{H}\beta = Y + E, \tag{6}$$

in which $E = \begin{bmatrix} e_1^T \\ \vdots \\ e_N^T \end{bmatrix}_{N\times m}$. The square loss function can be defined as

$$J = \sum_{j=1}^{N} \left(\beta_i G\left(a_i, b_i, x_j\right) - y_j\right). \tag{7}$$

Equation (7) can be written as follows:

$$J = \left(\mathrm{H}\beta - \mathrm{Y}\right)^T \left(\mathrm{H}\beta - \mathrm{Y}\right). \tag{8}$$

Then, the training of the network parameters is transformed into the problem of minimizing the square loss function, that is, finding the least square solution β so that

$$\left\|\mathrm{H}\widehat{\beta} - Y\right\| = \min_{\beta} \quad \left\|\mathrm{H}\beta - Y\right\|. \tag{9}$$

The following equation can be obtained using the Moore-Penrose generalized inverse:

$$\widehat{\beta} = \arg\min_{\beta} \quad \left\|\mathrm{H}\beta - Y\right\| = \mathrm{H}^{+}Y, \tag{10}$$

in which $\mathrm{H}^{+} = (\mathrm{H}^T\mathrm{H})^{-1}\mathrm{H}^T$. If the hidden layer output matrix is not of full column rank, then the optimal external weight can be obtained using the singular value decomposition (SVD) method.

Shown above is the process of using the traditional ELM algorithm to solve the problem. In the ELM algorithm, the hidden layer node parameters are randomly determined during parameter training (in actual application, the hidden layer node parameter values are often randomly set within the interval $[-1, 1]$ because the experimental samples must be standardized).

4.2. SS-ELM Algorithm. Sample sets in the cloud server (Z_n) can be divided into two parts: sample sets to be distributed to fog nodes/MEC hosts, $Z_{nf}^f = \{z_j^f = (x_j^f, y_j^f)\}_{j=1}^{nf}$, and backup sample sets, $Z_{nc}^c = \{z_j^c = (x_j^c, y_j^c)\}_{j=1}^{nc}$, in which $Z_{nf}^f \cup Z_{nc}^c = Z_n$; $Z_{nf}^f \cap Z_{nc}^c = \varnothing$. The purpose of sample selection is to select Z_{nf}^f from Z_n, so that the network learned from Z_{nf}^f using the ELM-based algorithm can satisfy $J(a_i, b_i, \beta) \le \sigma$, in which $\sigma \in (0, \min J(a_i, b_i, \beta))$ is the predetermined upper limit of the performance index.

Assuming that $U^c = \{u_j^c \mid u_j^c = |y_j^c - f(x_j^c, a_i, b_i, \beta)|\}_{j=1}^{nc}$ is the absolute error between the sample output and network output of Z_{nc}^c, $Z_s^c = (x_m^c, y_m^c)$ is defined to include any elements in Z_{nc}^c that correspond to the maximum element in U^c. After initializing the sample sets (let $Z_{nf}^f = \varnothing$ and $Z_{nc}^c = Z_n$) and the network parameters (let $a_i = \varnothing, b_i = \varnothing, \beta = \varnothing$), the following learning rules are followed.

Rule 1.

$$Z_{nf}^f = Z_{nf}^f \cup \{z_s^c\}. \tag{11}$$

Rule 2.

$$Z_{nc}^c = Z_{nc}^c - \{z_s^c\}. \tag{12}$$

Rule 3.

$$\begin{aligned} a_i' &= a_i + \frac{\{x_s^c\} - \{x_s^c\}_{\min}}{\{x_s^c\}_{\max} - \{x_s^c\}_{\min}}. \\ b_i' &= b_i + \frac{\{x_s^c\} - \{x_s^c\}_{\min}}{\{x_s^c\}_{\max} - \{x_s^c\}_{\min}}. \end{aligned} \tag{13}$$

Rule 4. Use Z_{nf}^f, a_i, and b_i to calculate and update the optimum external weight β (let $J(a_i, b_i, \beta) = \sigma$) with (7); Z_{nf}^f, Z_{nc}^c, a_i, b_i, and β are updated. After several iterations, $J(a_i, b_i, \beta) \le \sigma$.

The procedure of the algorithm is as follows:

(1) Initialize the sample sets (let $Z_{nf}^f = \varnothing$, $Z_{nc}^c = Z_n$) and network structure parameters (let $a_i = \varnothing$, $b_i = \varnothing$, $\beta = \varnothing$) on the cloud server side.

(2) Randomly generate hidden layer node parameters on fog node/MEC host (a_i, b_i), $i = 1, \ldots, L$.

(3) Calculate the hidden layer output matrix H (ensure that H is of full column rank).

(4) Calculate $J(a_i, b_i, \beta)$ and U^c.

(5) If $J(a_i, b_i, \beta) \le \sigma$, turn to step (10); otherwise, continue.

(6) Select Z_s^c from Z_{nc}^c, $Z_{nf}^f = Z_{nf}^f \cup \{z_s^c\}$; $Z_{nc}^c = Z_{nc}^c - \{z_s^c\}$.(7) Update a_i and b_i. $a_i' = a_i + (\{x_s^c\} - \{x_s^c\}_{\min})/(\{x_s^c\}_{\max} - \{x_s^c\}_{\min})$; $b_i' = b_i + (\{x_s^c\} - \{x_s^c\}_{\min})/(\{x_s^c\}_{\max} - \{x_s^c\}_{\min})$.

(8) Use Z_{nf}^f, a_i, and b_i to calculate and update the optimum external weight β (let $J(a_i, b_i, \beta) = \sigma$) with (7).

(9) If $Z_{nc}^c \ne \varnothing$, execute step (4); otherwise, continue.

(10) Use Z_n, a_i, and b_i to calculate and update the optimum external weight β with (7), and the algorithm ends here.

In this work, the hidden layer activation function of ELM model adopts a sigmoid transformation function:

$$G\left(a_i, b_i, x\right) = \frac{1}{1 + e^{-(a_i x + b_i)}}. \tag{14}$$

An algorithm analysis shows that most of the time taken for selecting samples for fog nodes/MEC hosts was spent in calculating $J(a_i, b_i, \beta)$. Assuming that $t(J)$ is the average time for calculating $J(a_i, b_i, \beta)$ and that the delay caused by data transmission between the cloud computing and fog nodes/MEC hosts is t' ($t' \ll t(J)$), then the learning time for fog nodes/MEC hosts is $t(N) \approx n_l \cdot t(J)$.

5. Numerical Simulation

At present, there are no training sets available for FC/MEC intrusion detection. Therefore, we used the KDD Cup 99

Table 1: Comparison of algorithms in testing accuracy and training time.

Algorithm	Accuracy (%)	Training time (s)
SS-ELM	99.07 ± 0.11	4.52 ± 0.10
ELM	96.09 ± 0.07	4.15 ± 0.04
BP	84.16 ± 0.18	387.92 ± 0.21
SVM	94.25 ± 0.08	15.02 ± 0.14
CVM-ELM	96.87 ± 0.23	12.47 ± 0.53

data set [25] for the analysis in the simulation experiments. In the data set, 41 fields were collected for each network connection. The abnormal types were divided into four major categories of 39 attack types, of which 22 attack types were presented in the training set and 17 unknown attack types were presented in the test set. In addition, the KDD99 connection records contained both symbolic features and discrete features. Therefore, the symbolic features needed to be converted prior to the experiment. We used the data preprocessing scheme reported in the literature [26]. The data preprocessing primarily includes converting category features into metric features and then normalizing metric features to prevent greater metric features from dominating and outweighing smaller features. For the supervised learning and prediction conducted in the experiment, all features were normalized to be within $[0, 1]$. The cloud server was simulated in Windows 7 operating system (i7-2760QM, 2.4 GHz CPU, 8.00 GB RAM), and the SS-ELM algorithm was implemented using Matlab 2014a.

We compared the performance of various fog IDS classifiers, including the SS-ELM, traditional ELM, BP neural network, SVM, and CVM-ELM [27]. The ELM, BP, and SVM were deployed on fog nodes/MEC hosts. The kernel function of SVM was *rbf*, $gamma = 0.005$, and $C = 10$. The BP learning rate was 0.06, the momentum coefficient was 0.9, the maximum number of epochs (iterations) was 5,000, and the goal was 0.0001. A total of 2,000 and 10,000 pieces of data were selected from the KDD Cup 99 and used as training samples and test samples, respectively. The training time and detection accuracy were compared. The experimental data were the average of 10 runs.

According to the results in Table 1, the SS-ELM has the highest accuracy, which proves that SS-ELM is the most suitable classifier algorithm for fog node/MEC host deployment, while from the training time perspective, the SS-ELM requires a slightly longer training time than the traditional ELM because sample selection is required in the SS-ELM. BP performed poorer than the SS-ELM and ELM in terms of training time and accuracy, and the SVM and CVM-ELM also require a longer training time. Therefore, in terms of accuracy, SS-ELM is more suitable for intrusion detection in FC/MEC.

In experiment 2, we analyzed the training time and accuracy of the SS-ELM algorithm for training sets of different scales. Based on the results of experiment 1, we selected only SS-ELM, ELM, CVM-ELM, and SVM for comparison

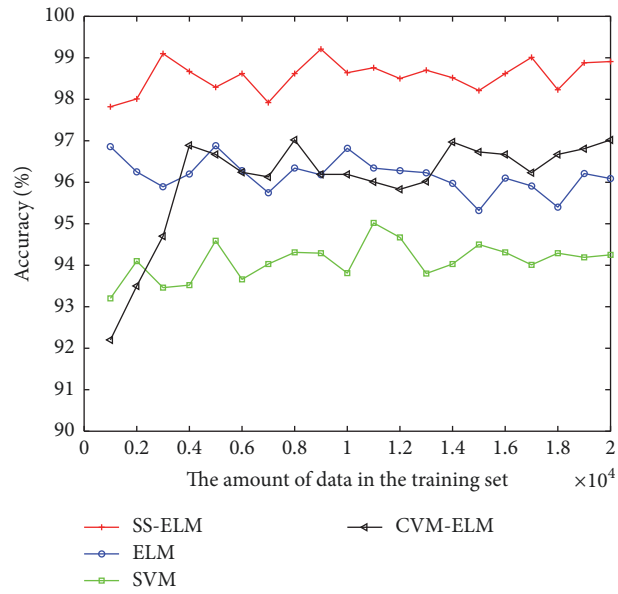

Figure 5: Accuracy rate comparison of SS-ELM, ELM, SVM, and CVM-ELM.

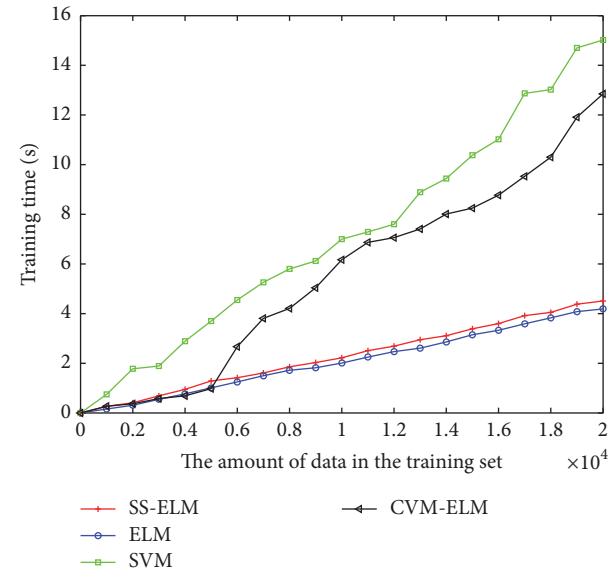

Figure 6: Training time comparison of SS-ELM, ELM, and SVM.

of the training time and accuracy. The training set sizes were selected as 1,000, 2,000,. . ., 20,000. The experimental results are shown in Figures 5 and 6.

As shown in Figure 5, as the size of the training set increases, SS-ELM shows the highest intrusion detection accuracy in FC/MEC relative to that of ELM and SVM. We analyze it from the aspect of algorithm, and we think that the main reason is the following:

(1) Unlike the traditional classic gradient-based learning algorithms which intend to reach minimum training error but do not consider the magnitude of weights, the ELM tends

to reach not only the smallest training error but also the smallest norm of weights.

(2) Unlike the traditional classic gradient-based learning algorithms facing several issues like local minimum, improper learning rate, overfitting, and so forth, the ELM tends to reach the solutions straightforwardly without such trivial issues.

(3) In our improved algorithm, we optimize the square loss function $J = \sum_{j=1}^{N}(\beta_i G(a_i, b_i, x_j) - y_j)$ for each fog node/MEC host in the training phase. This optimization is mainly reflected in the process of sample selection. According to formula (7) to formula (10) in the manuscript, each fog node/MEC host has an optimal external weight β in the network environment which is more suitable for the training.

For the training time, as shown in Figure 6, the simulation result shows that the training time of our proposed SS-ELM is slightly less than of ELM. The main reason is that SS-ELM has more than one sample selection step compared with ELM. Although the SS-ELM performed worse than the ELM, the difference is small and within an acceptable range. Therefore, we conclude that SS-ELM algorithm performs better in intrusion detection of FC/MEC.

As noted in Section 3, fog nodes/MEC hosts in the FC/MEC network are highly dynamic. Thus, we must analyze the robustness of the algorithm designed; specifically, we must analyze the dependency of SS-ELM algorithm on time statistics. In the KDD99 data set, there are 9 features (features 23–31) that are about time-based traffic features of the connection records. These features are based on time statistics. They can present some relationships between the current connection records and the connection records in the previous period. However, in the real FC/MEC network environment, there are a large number of raw data without manual statistical processing. That is to say, it is very difficult for us to obtain data based on time statistics. In order to ensure that the classifier algorithm can work effectively in the network environment with high real time and high dynamics, we removed features 23–31 to carry on experiment 3. In addition, new data sets obtained were used to compare the four algorithms used in experiment 1. The results are shown in Figure 7.

Figure 7 shows that the accuracy of SS-ELM did not decrease significantly after removing the time statistical features, while the ELM, BP, SVM, and CVM-ELM showed that accuracy decreases to various extents, with the decrease of the BP algorithm being the greatest. From the data of experiment 3, we can conclude that the SS-ELM has the least dependency on time attribute and is suitable for network environments that are highly dynamic and feature large changes.

In experiment 4, to analyze the false positive rate of the SS-ELM, we showed the performance of SVM, SS-ELM, and ELM in terms of the receiver operating characteristics (ROCs) [28]: for the ROC curve, the x-axis represents the false positive rate, and the y-axis represents the true positive rate. A classifier algorithm with a larger area under the ROC curve (AUC) performs better. Figure 8 shows that SS-ELM outperformed the other two algorithms in FC/MEC.

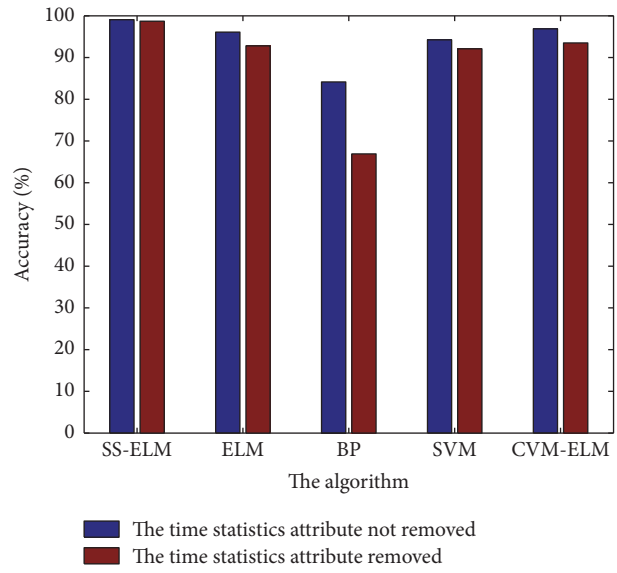

Figure 7: A comparative study on the dependence of time-dependent attributes.

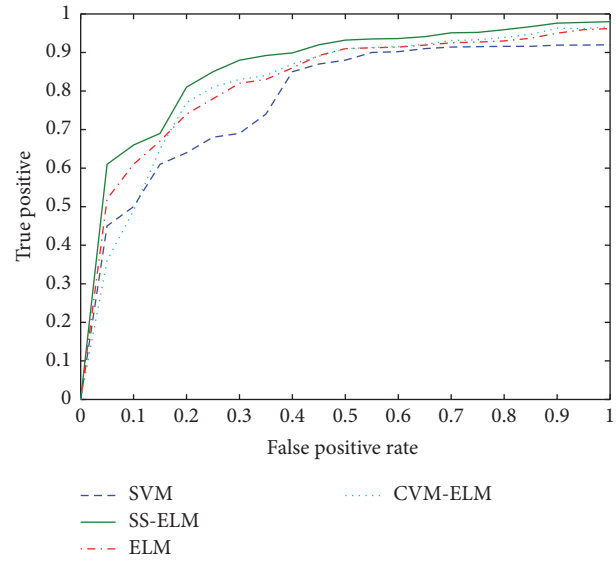

Figure 8: ROC of SVM, SS-ELM, ELM, and CVM-ELM.

6. Conclusions

FC/MEC is a new paradigm. Fog nodes/MEC hosts' resource-constrained features bring about new problems to the security realm, particularly in the area of intrusion detection. For MEC, its characteristics are similar to FC/MEC. In the present study, an FC/MEC IDS scheme was established based on the analysis of the network characteristics and security requirements of FC/MEC. In addition, an SS-ELM algorithm that combines sample selection and ELM is proposed according to the network characteristics of FC/MEC. The training time and detection accuracy of ELM, BP, and SVM are examined experimentally. In the experiment, the SS-ELM

algorithm shows excellent performance in FC/MEC, particularly in accuracy and time dependence. Several experiments have demonstrated the advantages of SS-ELM as a lightweight algorithm from different perspectives and have contributed to solving the problems caused by resource constraints in FC/MEC.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Science and Technology Key Projects (no. 61602034) and the U.S. National Science Foundation under Grants CNS-1559696 and IIA-1301726.

References

[1] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[2] E. Portal, "Mobile-edge computing-introductory technical white paper," Tech. Rep., 2014.

[3] M. Beck, M. Werner, S. Feld, and S. Schimper, "Mobile edge computing: A taxonomy," in *Proceedings of the the 6th International Conference on Advances in Future Internet*, 2014.

[4] S. Wang, Y. Zhao, L. Huang, J. Xu, and C. Hsu, "QoS prediction for service recommendations in mobile edge computing," *Journal of Parallel and Distributed Computing*, In press.

[5] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: a survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[6] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Communications Magazine*, vol. 23, no. 5, pp. 120–128, 2016.

[7] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the Workshop on Mobile Big Data (Mobidata '15)*, pp. 37–42, ACM, Hangzhou, China, June 2015.

[8] C. Dsouza, G.-J. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: preliminary framework and a case study," in *Proceedings of the 15th IEEE International Conference on Information Reuse and Integration (IEEE IRI '14)*, pp. 16–23, Redwood City, Calif, USA, August 2014.

[9] D. E. Denning, "An Intrusion-Detection Model," in *Proceedings of the IEEE Symposium on Security and Privacy*, p. 118, Oakland, Calif, USA, April 1986.

[10] S. Wang, Q. Sun, H. Zou, and F. Yang, "Detecting SYN flooding attacks based on traffic prediction," *Security Communication*, vol. 5, pp. 1131–1140, 2012.

[11] O. Linda, T. Vollmer, and M. Manic, "Neural Network based intrusion detection system for critical infrastructures," in *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2009*, pp. 1827–1834, Atlanta, Ga, USA, June 2009.

[12] I. Stojmenovic and S. Wen, "The fog computing paradigm: scenarios and security issues," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '14)*, pp. 1–8, IEEE, Warsaw, Poland, September 2014.

[13] S. J. Stolfo, M. B. Salem, and A. D. Keromytis, "Fog computing: Mitigating insider data theft attacks in the cloud," in *Proceedings of the 1st IEEE Security and Privacy Workshops, SPW 2012*, pp. 125–128, San Francisco, Calif, USA, May 2012.

[14] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *Proceedings of the 39th IEEE Annual Computer Software and Applications Conference Workshops, COMPSACW 2015*, pp. 53–59, Taichung, Taiwan, July 2015.

[15] S. Yi, Z. Qin, and Q. Li, "Security and Privacy Issues of Fog Computing: A Survey," in *Wireless Algorithms, Systems, and Applications*, vol. 9204 of *Lecture Notes in Computer Science*, pp. 685–695, Springer International Publishing, Cham, 2015.

[16] Y. Shi, S. Abhilash, and K. Hwang, "Cloudlet mesh for securing mobile clouds from intrusions and network attacks," in *Proceedings of the 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2015*, pp. 109–118, San Francisco, Calif, USA, April 2015.

[17] N. Pitropakis, C. Lambrinoudakis, and D. Geneiatakis, "Till all are one: Towards a unified cloud IDS," in *Trust, Privacy and Security in Digital Business*, vol. 9264, pp. 136–149, Springer International Publishing, 2015.

[18] E. C. Oscar, E. B. Fernandez, and M. A. Raúl, "Threat analysis and misuse patterns of federated Inter-Cloud systems," in *Proceedings of the 19th European Conference on Pattern Languages of Programs, EuroPLoP 2014*, Irsee, Germany, July 2014.

[19] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "VAED: VMI-assisted evasion detection approach for infrastructure as a service cloud," *Concurrency Computation Practice Experience*, vol. 29, no. Issue, p. 30, 2017.

[20] 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, 2006.

[21] C. Cheng, W. P. Tay, and G.-B. Huang, "Extreme learning machines for intrusion detection," in *Proceedings of the Annual International Joint Conference on Neural Networks, IJCNN 2012*, Brisbane, Australia, June 2012.

[22] Z. Ye and Y. Yu, "Network intrusion classification based on extreme learning machine," in *Proceedings of the IEEE International Conference on Information and Automation, ICIA 2015*, pp. 1642–1647, Lijiang, China, August 2015.

[23] W. Srimuang and S. Intarasothonchun, "Classification model of network intrusion using Weighted Extreme Learning Machine," in *Proceedings of the 12th International Joint Conference on Computer Science and Software Engineering, JCSSE 2015*, pp. 190–194, Songkhla, Thailand, July 2015.

[24] C. Cai, H. Pan, and G. Cheng, "Fusion of BVM and ELM for anomaly detection in computer networks," in *Proceedings of the International Conference on Computer Science and Service System, CSSS 2012*, pp. 1957–1960, Nanjing, China, August 2012.

[25] KDD CUP 99 data set, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[26] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: a review," *Computers & Security*, vol. 30, no. 6-7, pp. 353–375, 2011.

[27] Y. Zhou, *An incremental ELM algorithm based on instance selection [M.S. thesis]*, Hebei University, Hebei, China, 2015.

[28] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.