Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8811035, 15 pages
https://doi.org/10.1155/2020/8811035

Research Article

An Edge IDS Based on Biological Immune Principles for Dynamic Threat Detection

Yajing Zhang[1], **Jia Wei**[2] and **Kai Wang**[3,4]

[1]*School of Computer and Control Engineering, Yantai University, Yantai 264005, China*
[2]*Department of Economics and Management, Weifang University of Science and Technology, Weifang 262700, China*
[3]*School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China*
[4]*Research Institute of Cyberspace Security, Harbin Institute of Technology, Weihai 264209, China*

Correspondence should be addressed to Kai Wang; dr.wangkai@hit.edu.cn

Received 22 April 2020; Revised 19 July 2020; Accepted 3 August 2020; Published 17 August 2020

Academic Editor: Fuhong Lin

Edge computing solves such questions as the massive multisource data and resource consuming computing tasks in edge devices. Some new security problems especially the data security and privacy issues have been introduced into the edge computing scenario. Through analyzing the biological immune principles, a novel idea for the problem of intrusion detection in edge computing is provided. Specifically, an edge intrusion detection system (Edge IDS) with a distributed structure, which has the characteristics of an imprecise model, self-learning, and strong interactivity, is constructed in a systematic way inspired by the biological immune principles. Moreover, a newly proposed gene immune detection algorithm (GIDA) is designed. In order that Edge IDS can deal with the dynamic data problem efficiently, the key functional components such as the remaining gene, niching strategy, and extracting vaccine are embedded into the GIDA algorithm. Furthermore, extensive simulation experiments are conducted, and the results show that the proposed Edge IDS can be adapted to the domain of edge computing with comparative performance advantages.

1. Introduction

Nowadays, the Internet is dramatically growing with the proliferation of a wide variety of network-connected devices everywhere. As increasing numbers of applications are continuously migrated from these devices to the cloud [1], current cloud infrastructure has become too overwhelmed to provide guaranteed services. For instance, the massive multisource data and resource-consuming computing tasks generated from these devices go far beyond the ability of cloud computing technologies [2]. To address these challenges, the edge computing paradigm (e.g., cloud-aware mobile fog computing [3]) is emerging and attracting increasing attentions, which enables the use of networked resources from remote cloud datacenters to the edge closer to the data source [4].

In edge computing, such special features as real-time computation and parallel processing [5] have also brought new security problems which can be summarized as follows [6].

(1) Due to the need for computing at the edge, privacy data needs to be transmitted to third parties (e.g., the edge servers operated by certain commercial companies). In this case, it is easy to lose data, have data leakage, or operate illegal data. Thus, it is difficult to guarantee data confidentiality and integrity [7]

(2) Users' data shows characteristics like multiple trust domains and variety structure, which increase the uncertainty and difficulty of data detection

(3) The edge and cloud services can be accessed by all the authorized parties at any situation as per users' requirements, which increases the risk of malicious exploitation on certain data

(4) It will be relatively easy to lose protection and control of all the privacy information, for example, personal identification or location information, due to some outsourced or external computing issues

As an original barrier, the intrusion detection system (IDS) plays a critical role in the problem of security protection. In this paper, the IDS that is implemented in the traditional cloud servers is classified as Cloud IDS and the one implemented in the edge servers is classified as Edge IDS.

Compared with the Cloud IDS relying on remote servers, the Edge IDS can perform nearly real-time computing tasks exactly at the edge of the network and thus can effectively protect the security of the edge computing scene. However, the concepts and methods of traditional intrusion detection are difficult to be directly applied to the security protection scenarios of edge computing architecture. For example, it is difficult for the Edge IDS to establish a precise detection model as the Cloud IDS, since the amount of data that can be obtained by the edge networks is much smaller than that of the cloud central servers. Furthermore, the variety of data sources and mobile devices existing in the edge also increases the difficulty of edge intrusion detection. In addition, traditional intrusion detection mechanisms work independently in each edge domain without cooperation or lack of interactivity; thus, they cannot be entirely and directly applied as the Edge IDS without modification. Thus, it is necessary to design an Edge IDS to solve the above problems timely and effectively.

Although edge computing reduces the scale of data in cloud computing, it is difficult to solve the problem of multisource data in edge computing in many IDSs. Traditional IDSs often establish a detection mechanism by extracting data characteristics, where an exact model is often required. Fortunately, biological immune principles provide a new way for designing a more dynamic and effective Edge IDS without the help of such exact models. The biological immune system can identify foreign invasions primarily by distinguishing *self* and *nonself*. In addition, the immune response is carried out individually and does not require a central controller.

Obviously, in the edge computing environment, the problem of multisource data can effectively be solved through nonspecific immunity in biological immune principles. The distributed structure makes the Edge IDS not only facilitate the operation and reduce the amount of data but also increase the real-time interaction with the Cloud IDS. At the same time, an immune system also has a self-learning function and can adapt to dynamic detection.

In this paper, the basic idea of building Edge IDS based on biological immune principles is presented. The characteristics of Edge IDS and an artificial immune system (AIS) inspired by biological immune principles are analyzed. And the theory of AIS is used to construct a dynamic Edge IDS from the system's point of view. This system is placed in the distributed system associated with the Cloud IDS. Because of the distributed structure, there is an interaction between Edge IDS and Cloud IDS. The Edge IDS in the control layer transmits the detection information to the Cloud IDS in the monitoring layer. Cloud IDS generates cloud vaccine libraries that give feedback to Edge IDS. Edge IDS inspired by the principle of biological immunity does not need to have an exact model and has excellent self-learning function.

The main contribution of this paper can be concluded as follows:

(1) The characteristics and types of intrusion detection in edge computing are analyzed and described in detail. An Edge IDS is constructed with the idea of AIS

(2) The Edge IDS lies in the second layer of the whole system which has a distributed system structure with three layers. The first one is the monitor layer (e.g., Cloud IDS), the second one is the control layer (e.g., Edge IDS), and the third one is the device layer (e.g., host firewall). It also has dynamic detection and adaptive functions because of the use of a distributed structure. For example, Edge IDS can dynamically determine whether to distribute some tasks to Cloud IDS based on the self-assessed network load. On the other hand, Edge IDS can adaptively improve its security ability by acquiring security knowledge from Cloud IDS with continuous convergence of fragmented security data scattered throughout the network

(3) An algorithm named GIDA is proposed. It proposes the way of the remaining gene, niching strategy, and abstracting vaccine, so as to deal with dynamic data efficiently and own the function of self-learning

The rest of this article is organized as follows. Section 2 introduces the background knowledge of biological immune system and briefly summarizes related works on IDS in edge/fog/cloud computing. Section 3 illustrates the theoretical preliminaries of the immune process. Section 4 presents the design of the proposed Edge IDS as well as the details of the GIDA algorithm. In Section 5, experiments are conducted to verify and evaluate the system. Finally, Section 6 concludes our work.

2. Background and Related Work

2.1. Security Threats for Edge Computing. One of the important problems for edge computing is how to ensure the security of the data transmission process [8]. During message transmission, some attacks may be generated to disable some certain network connections by congesting the network bandwidth resources or could monitor some details of the targeted network data flow. The threats for the edge computing, including Forgery Attack, Tampering Attack, Spam Attack, Sybil Attack, Jamming Attack, Eavesdropping Attack, Denial of Service Attack, and Collusion Attack, are shown in Figure 1 [9–11]. In a network or a system, intrusions mean any unauthorized or unapproved activities. Thus, any attacks are intrusions from the outside.

2.2. Overview of Biological Immune System. In a biological immune system, for an organism, antigens are external invasions. When antigens appear, immune function begins to work. The process of the body's identification and exclusion of antigens is called immunity [12].

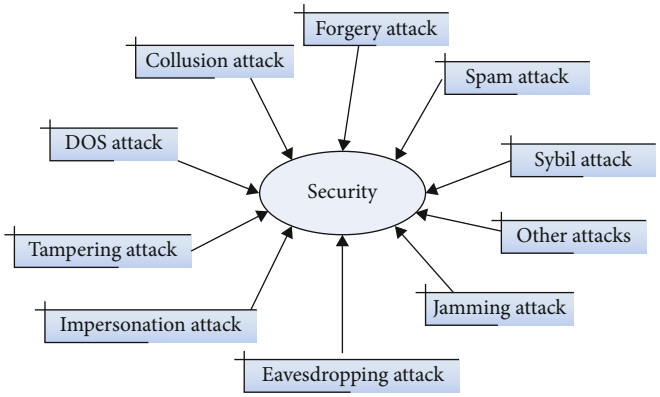


Figure 1: Security threats in edge computing.

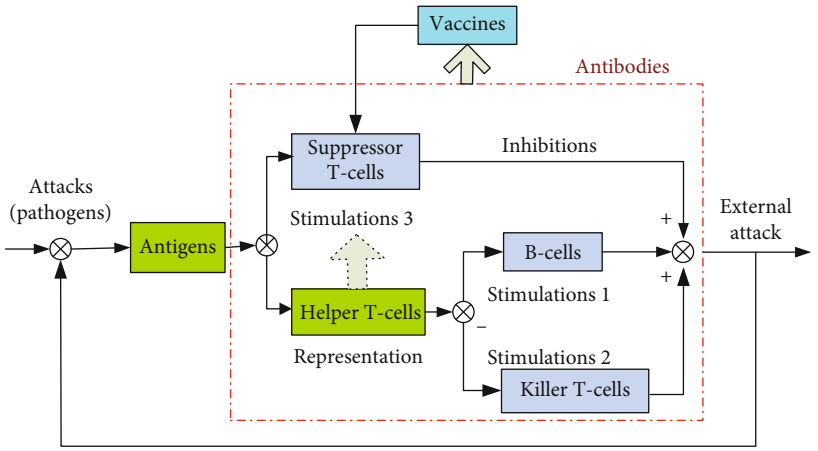


Figure 2: Scheme of the whole immune circle.

Lymphocytes are the main line of defense for the body's immunity. Lymphocytes are divided into B-cells and T-cells. B-cells can be activated by invading pathogens. It has two functions: one is to activate innate immune function and the other is to produce antibodies. B-cells can clone themselves, and their function of negative selection can prevent the resulting antibodies from responding to themselves. On the other hand, positive selection is done through T-cells. After a series of stimulation processes, B-cells, suppressor T-cells, and killer T-cells form antibodies [13]. Helper T-cells are used to help suppressor T-cells for an immune response.

During the process of producing antibodies, the vaccine is extracted and entered into the antibody library. Thus, a closed-loop immune system is produced. The scheme of the whole immune circle is shown in Figure 2.

A biological system is actually a complex information processing and interaction system. An AIS is based on the inspiration of biological principles applied to the field of engineering [14].

During the process of studying the intrusion detection mechanism, it is shown that both the AIS and IDS have high similarity in Table 1. The former protects body recognition and excludes antigens (virus or pathogen), and the latter protects the computer to avoid damage in the case of intrusion. Both of them keep the stability of the system in a continually changing environment.

2.3. Brief Summary of IDS in Edge/Fog/Cloud Computing. In different network environments, there are different requirements for the detection algorithm and application deployment of IDS. Intrusion detection is mainly used to monitor and detect abnormal behaviors on the host side or network side, and many studies in cloud computing environment are also instructive and valuable to the design of the Edge IDS.

In 2010, Zhou et al. [15] suggested a collaborative IDS for cloud environments. Mazzariello et al. [16] present a network-based IDS that determines intrusion behavior by defining a series of intrusion rules with a high detection rate. To handle access to large-scale network data and applications that manage control traffic in the cloud, Gul and Hussain [17] introduced a multithreaded IDS model in 2011. In this model, the Cloud IDS can process and analyze the data packets of high traffic, generate reports effectively, and automatically send security reports to users through third-party IDS detection services.

Internet of Things (IoT), wireless sensing networks (WSN), and other networks are typical applications for edge

Table 1: AIS vs. IDS.

Items	AIS	IDS
Entity in detection process	Antigenic determinant	Behavior mode of the detected strings
	Receivers—the chromosome or organ that needs to be checked	Normal-mode strings
	Single clone lymphocyte	A detector placed at the security monitoring boundary
	Antigen	Nonself string
Behavior representation of detection	Binding—the response of an antibody to an antigen	Match between normal-mode strings and nonself strings
	Behavioral confirmation	Negative selection
	Clone of lymphocytes	Duplication of detectors
	Detection of antigens	Detection of network traffic behavior
	Cleaning antigens	Response action of detectors

computing. IDS for these scenarios also has a reference value for the design of a specific IDS for edge computing. In 2013, Raza et al. [18] proposed an IDS, SVELTE, for the IoT environment, primarily for the detection of routed attacks. In 2014, Shamshirband et al. [19] applied the three-person cooperative strategy game model to the intrusion detection of WSN, focusing on flooding attacks in WSN and verifying that the model's attack detection and defense accuracy as well as the energy consumption are superior to machine learning methods. In 2015, Xiao et al. [20] introduced the concept of friend mechanism and proposed a lightweight IDS model for mobile networks, which has comparative advantages in detection accuracy when detecting black hole attacks in routing. But they are only simulated for attacks at the network routing level.

In 2016, Hosseinpour et al. [21] proposed an IDS with distributed lightweight character based on AIS for fog computing. IDS is distributed across a three-tier IoT structure. However, an in-depth analysis of the detection rate and FP rate (false positive rate) is lacking. In 2017, Wang et al. [22] discussed a privacy shield for a collaboration network based on fog devices, which can improve detection efficiency with the protection of private data and information security of network resources. The study focused on privacy data protection but did not study and analyze the characteristics of limited resources for fog nodes.

In 2018, An et al. [23] introduced a common fog computing IDS framework, provided a classifier model, and proposed an intrusion detection scheme of a cloud and fog hybrid collaboration. Lin et al. [24] discussed the problem of system defense resource allocation under this framework and put forward a single-level advantage and distributed strategy. The performance of IDS is improved by allocating resources.

3. Theoretical Preliminaries and Notation

3.1. Abbreviation Representation. Ab:Antibody,
Ag:The number of antigen,
T_{kill}:Killer T-cell,
T_{sup}:Suppressor T-cell,
T_H:Helper T-cell,
APC:A presenting cell for antigens,
MHC:Major and histocompatibility complex,
TCR:A receptor for T-cell,
S: The overall number of APC conjoining with helper T-cells,
θ_B:The threshold which is the concentration into the blood from the marrow,
B_p: The plasma B-cells,
B:Number of B-cells.

3.2. Modeling of Immune Process. TCR can recognize antigens after they enter into the body of a creature. The function of APC is to transmit the information of the antigen to helper T-cells. By APC, these stimulations will activate helper T-cells. According to the dynamic equations from Qi and Du [25], the following differential equation set is described.

$$\begin{cases} \dfrac{dT_H}{dt} = \alpha + \left(\dfrac{S}{1+T_H} - 1\right) T_H, \\ \dfrac{dS}{dt} = \beta(\text{Ag} - S), \end{cases} \tag{1}$$

where α is a constant that presents the number of macrophages selected as APC and β is a proportional coefficient. α is very small, so we can neglect it in computation.

Solve the equation set (1). Neglecting the item including α, we can gain

$$\begin{cases} T_H = S + e^{-t} \bullet C_2, \\ S = \text{Ag} + e^{-\beta t} \bullet C_1, \end{cases} \tag{2}$$

where C_1 and C_2 are constants.

In equation set (2), the number of helper T-cells can be considered as a nonline relation linking antigens.

According to the proportion relation between helper T-cells and killer T-cells, the number of killer T-cells $T_{\text{kill}}(t)$ at time t can be calculated. Here, k is a coefficient decided by the immune response.

$$T_{\text{kill}}(t) = k \bullet T_H. \tag{3}$$

Through the function of APCs, helper T-cells will promptly be activated. Then, they will stimulate the resting B-cells and become plasma B-cells (B_p). We use differential equations to represent the speed at which B-cells transform and the collection of B_p. The model is as follows:

Set Hill function $H(\mathrm{Ag}, \theta_B)$, which represents the MHC's effects in the process of B-cell activation.

$$\begin{cases} \dfrac{dB}{dt} = k_1 - k_2 \bullet B \bullet T \bullet H(\mathrm{Ag}, \theta_B) - d_1 \bullet B, \\ \dfrac{dB_p}{dt} = k_2 \bullet B \bullet T \bullet H(\mathrm{Ag}, \theta_B) - d_2 \bullet B_p, \\ H(\mathrm{Ag}, \theta_B) = \dfrac{Ag^2}{{\theta_B}^2 + Ag^2}, \end{cases} \tag{4}$$

where k_1 and k_2 are constants of the duplicating velocity of B-cell and d_1 and d_2 are the usual killed rate of B and B_p.

Through solving, the number of B_p will be obtained. Setting the number of antibodies at t time is $\mathrm{Ab}(t)$.

$$\mathrm{Ab}(t) = k_B \bullet B_p(t). \tag{5}$$

In equation (5), k_B is the proportion coefficient of antibody production by B_p.

The T_{sup} can show the action of all other cells. The relation between T_{sup} and the number of antigens at time t is as follows:

$$T_{\mathrm{sup}}(\mathrm{t}) = k_2 \left\{ \begin{array}{c} [T_{\mathrm{kill}}(t-d) + \mathrm{Ab}(t-d)] \\ -[T_{\mathrm{kill}}(t-d-1) + \mathrm{Ab}(t-d-1)] \end{array} \right\}^2 \bullet \mathrm{Ag}(t), \tag{6}$$

where k_2 is a positive factor, d is the beginning time when it is revealed. In equation (6), the middle term demonstrates the progressive relation between killer T-cells and antigens.

Thus, the overall number of attacking cells at time t is as follows:

$$\mathrm{Attack}(t) = T_{\mathrm{kill}}(t) + \mathrm{Ab}(t) - T_{\mathrm{sup}}(t). \tag{7}$$

According to equations (1)–(7), the generation procedure of antibodies in biological immunity is shown as a schematic module in Figure 3. It illustrates the function of suppressor T-cells and helper T-cells in the process of antibody generation. Killer T-cells and antibodies are taken as detectors defending external invasion in the biological immune system.

4. A System View of the Proposed Edge IDS

4.1. The Basic Idea. The main characteristics in edge computing are high discreteness, real-time, and interaction, which can be considered as a distributed interactive computing system. In Table 2, the basic thought of an intrusion detection mechanism with an uncertain model is suggested, whose features is suitable for edge computing.

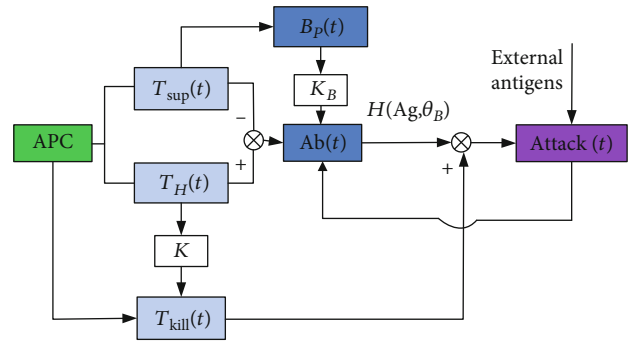


Figure 3: Schematic module of immune system.

Everything with certain functions and multiple elements can be considered as a system. If cloud computing is a great system, edge computing can also be looked at as a tiny system (Figure 4).

Viewing from the holistic perspective, Internet IDS can be regarded as a distributed IDS. The Cloud IDS can be equal to a monitor layer and the Edge IDS corresponds to a control layer. Every edge device has its own data security system, for example, a firewall. The layer of edge devices is considered as the device layer.

In the three-layer distributed structure, IDSs in the device layer protects only the information in their respective devices. They perform the lowest level of security. Edge IDS is located at the control layer. They protect the data generated during edge computation. At the same time, in the network state, the security system in the equipment can be controlled by the communication protocol. Cloud IDS is located at the monitor layer. Edge IDS in the control layer can selectively upload a part of the data to the cloud and is processed by the Cloud IDS. The monitoring layer publishes data processed to the control layer as needed to enable monitoring of the control layer. So each Edge IDS performs its own function and only interacts with Cloud IDS, which is independent of each other.

In edge computing, the amount of data is small, but the data is real time and interactive. This requires that the Edge IDS must have dynamic real-time detection function. In addition, multisourced data caused by the heterogeneous edge devices exists. As a result, it is difficult to model the IDS. Based on the above analysis, there is no exact model required for the system design of the proposed Edge IDS. Fortunately, the AIS inspired by the principles of a biological immune mechanism meets the requirements of the Edge IDS, which provides a feasible way for the design of an effective and efficient Edge IDS. The reasons are as follows:

(1) Nonspecific immunity and specific immunity are divided into the basic principles of immunity. Nonspecific immunity is taken as the general immune effect of organisms, i.e., the use of "self" and "nonself" identification to detect all foreign invasion. It does not require modeling. Specific immunity is the targeted training of antibodies. This results in a vaccine that allows for dynamic detection in advance

Table 2: Analysis of intrusion detection in edge computing.

	Presentation	Characters	Intrusion detection
Data	Data encryption methods which are lightweight	Light data	A distributed and dynamic IDS with uncertain model
	Fine-grained data-sharing systems	Independent	
	Multisource heterogeneous data	Various	
Communication	Independent edge devices	Noninterference	
	Communicating with cloud service	Distributed	
Computing models	Faced with Internet of everything scenario that means various edge services	Uncertain models	

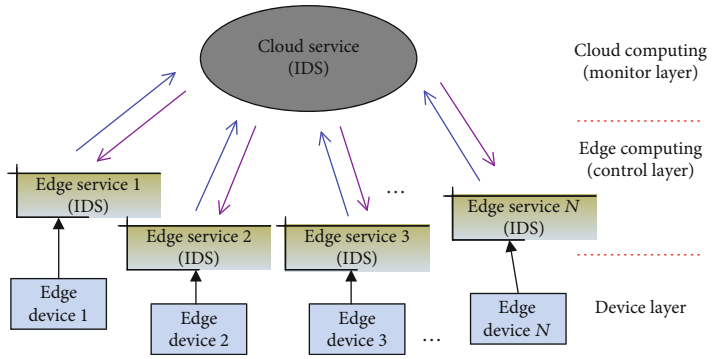


Figure 4: The diagram of distributed IDS.

(2) The use of nonspecific immune function can avoid the requirement of precise modeling. The problem of the multisource data does not need to be considered. Dynamic and rapid detection can be achieved with specific immune function. With the help of the process of generating and updating antibodies, the ability of self-learning in the IDS can be achieved

(3) The current disadvantages of the AIS are the algorithm's complexity and storm data. Edge computing effectively avoids them because the amount of data is greatly reduced, so it is feasible to use AIS to construct the Edge IDS.

(4) The interaction between edge computing and cloud computing has formed a distributed IDS structure. Edge IDSs are independent of each other and interact with Cloud IDS for information sharing and data fusion. In this way, conflicts between Edge IDSs are better avoided. At the same time, by constantly learning new knowledge from the Cloud IDS, the Edge IDS can update and upgrade its detection rules as the network security situation changes.

4.2. Details for the Architecture. Based on the characteristics of Edge IDS and the basic principle of AIS, a model of an Edge IDS embedded with AIS advantages is constructed, as shown in Figure 5.

The system model takes a distributed structure. The Edge IDS is the control layer. The main function of the system contains two major parts: a generating detector and detection. The generating detector is a key part with a dynamic process. The structure of the AIS is used in the part of the generating detector. The model is based on negative selection algorithms that identify *self* and *nonself*. The clone selection module and the compensation operator are used as forward channels in the system. The positive selection module is used as a feedforward module. The vaccine module serves as feedback. This forms a feedback system with feedforward control. Feedforward control can quickly generate "genes." Feedback control ensures the diversity and stability of the system. The system can generate multigeneration detectors through feedforward and feedback control, which can detect foreign intrusions in real time. Detection is a process of matching between detectors and external data. In the part of detection, the system can output the result on whether a behavior is malicious or not.

Obviously, the algorithm combines the two outstanding characteristics of nonspecificity and specificity in immunity. Through nonspecificity immunity, an original detector is generated.

Vaccine and gene detection algorithms are included in the specific immunity which are emphasized on the basis of *self* and *nonself* algorithms. Vaccines can be obtained through self-learning and interaction with Cloud IDS. And system updates are made through constant updates of the vaccine. Thus, the system becomes a dynamic system with adaptive capability. The niching strategy is used to optimize the search by dividing the weight coefficient. It guarantees the diversity of detectors, so that the generation of the detector does not become partially optimal.

The distributed system is reflected in the interaction between the Edge IDS and the Cloud IDS. The Edge IDS uploads alarm information to the Cloud IDS. The Cloud IDS collects and compares information about multiple Edge

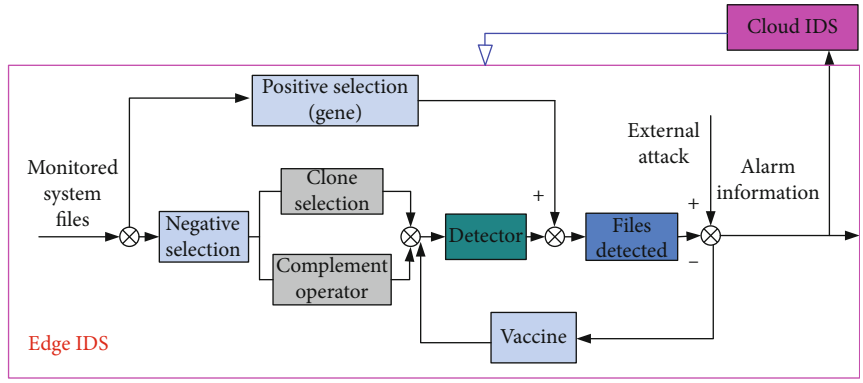


Figure 5: Edge intrusion detection system.

IDSs. Key information is then passed on to the Edge IDS in the form of a vaccine. This ensures the diversity and effectiveness of Edge IDS.

4.2.1. Nonspecific Immunity. Nonspecific immunity is a kind of innate immune function of the immune system, which includes negative selection, cloning selection, and compensating operators.

Forrest et al. [26] first proposed the negative selection algorithm. It is initialized by producing a detector from the protected data. Then, the detector can be used to recognize *self* and *nonself*. Clone selection can effectively detect unknown intrusions by copying the data fragments. A complement operator can help immune response.

(1) Negative Selection. The negative selection algorithm produces plenty of random patterns that are compared with each *self*. If any random pattern does not match the *self*, it becomes a "detector" that is able to monitor profiled patterns of the protected files. In the process, if a detector matches any new pattern, it is then taken as a novel abnormal one named the *nonself*.

The method of negative selection can be described with a set of mathematical models.

(i) U: a whole set which is limited

(ii) S: a subset indicating *self*

(iii) N: a subset indicating *nonself*

Then, $S \cup N = U \cdot S \cap N = \Phi$.

The problem of intrusion detection can be described as follows: in the case that some limited information is given, an element in the whole set U is determined while it is included in the S or N. In the process, two metrics can be used to evaluate the intrusion detection ability of the IDS: true positive (TP) and false positive (FP). TP indicates that a malicious external access behavior is correctly identified by the IDS. On the other hand, FP means that a *self* is considered as a *nonself*, that is, an external legitimate behavior is erroneously classified as abnormal. According to these, TP and FP are available to assess the detection ability of IDS [25].

Table 3: The computing steps of negative selection and clone selection.

Step 1	Define a set S (self) with equal-length strings. A single string generated randomly is divided into equal-sized substrings to generate S.
Step 2	Obtain a set R that does not match the elements in S.
Step 3	Generate random situation in those strings with high weight. And starting from the situation, taking out equal length l, the strings obtained will be duplicated.
Step 4	A random bite is chosen in these cloned string. The bite will be the mutation. Thus, an initial antibody population is obtained.
Step 5	Monitor S by the detectors in R. If any of the detectors matches a string in S, then a normal attack is detected.

The formal definition of the intrusion detection model based on biological immune principles is a classification that is made up of the following two components. IIDS = (f, Ab), where f is a bivariate function of classification and Ab is a detector set that is gained by learning access data constantly in the mechanism of the immune system. IIDS means immune intrusion detection system. IIDS = (f, Ab) shows a process in that a detector Ab detects intrusions.

If u is set to a conventional value, that is, $u \in U$. Then,

$$f(Ab, u) = \begin{cases} u \in N, & \text{when Ab matches,} \\ u \in S, & \text{others.} \end{cases} \quad (8)$$

(2) Clone Selection. Clone selection is used to reproduce valid antibodies [27]. In the process, clone selection can evolve antibodies by random methods.

In Table 3, the computing steps of the negative selection and clone selection are provided.

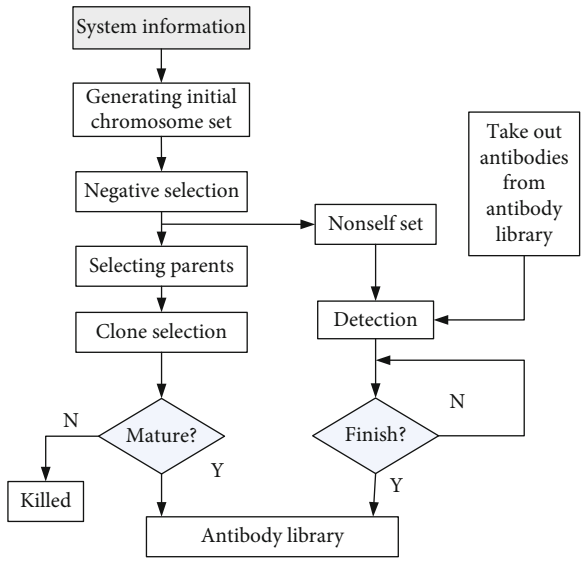


Figure 6: The diagram of vaccine generation.

(3) Complement Operator. After the first match, the function of a complement operator is to exclude those antibodies with high similarity so as to obtain the qualified detectors. These antibodies need to be mutated randomly. The aim is to increase the diversity of antibodies. Data is updated quickly and frequently in edge computing. Compensation operators can increase detection efficiency and reduce detection time by mutation. This is suitable for data processing in edge computing.

4.2.2. Specific Immunity. Specific immunity is the immune function produced against a particular virus. It includes positive selection and vaccine operators. In the process of antibody evolution, the niche strategy is formally adopted as well.

(1) Positive Selection Operator and Gene Detection. Positive selection is a method that a detector is directly extracted from viruses.

The computing process of positive selection operator is also called as the gene detection. In edge computing, genes can be defined as features representing the unique information. Genes cannot be altered under the legal action.

(2) Vaccine Operator. The vaccines can be considered as a kind of memory cell. The immune system can usually generate vaccines after a detector responds the first time. The effect of vaccines is that the system is able to respond rapidly when the antigen is encountered again.

The generating idea of vaccines is as follows: (1) generate the *nonself* set, (2) detect the *nonself*, and (3) select vaccines from the antigens detected and join them into the antibody library.

The rule of extracting vaccines is to select antigens detected with a high match level and affirm that the detector that may be repeated into a vaccine. Vaccines are updated as needed in time. The diagram of vaccine generation is shown in Figure 6, where system information

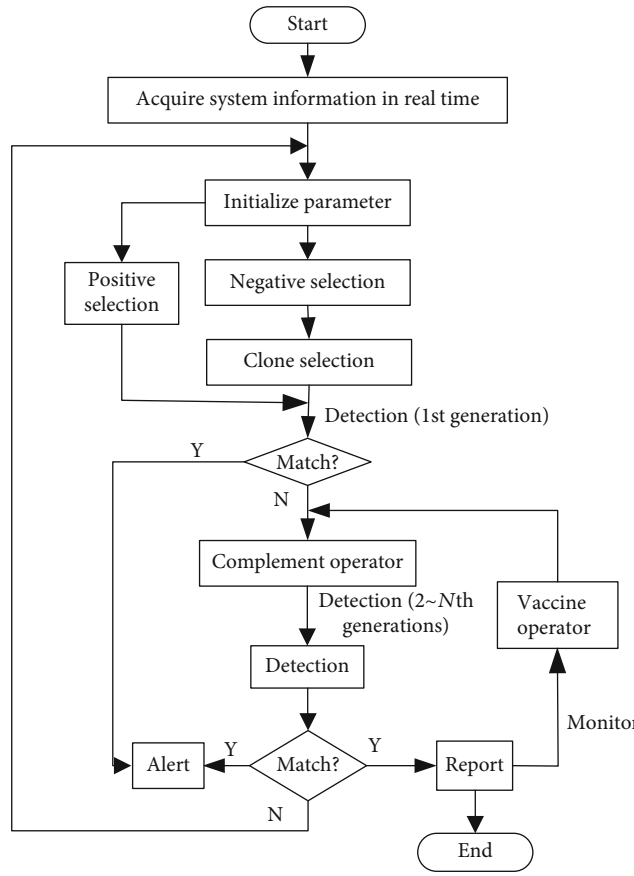


Figure 7: The flow diagram of GIDA.

needed for detection can generate an initial population in a random way. Through negative selection, an original *nonself* set is produced. After executing detection, some antibodies with high weight will be selected as vaccines. In another way, by clone selection, matured antibodies will also become vaccines. These vaccines will join in the antibody library so as to detect external attacks.

(3) Niching Strategy. Niche is mentioned as a certain special circumstance in biology. During the period of antibody generation, strings in a population have different affinities. Depending on the size of the affinity, these strings can be divided into some subpopulation. Then, in every subpopulation, duplication, cross and mutation will be executed individually. Finally, those strings with high affinity are eliminated. Thus, the covering rate of antibodies is added.

Obviously, the niching strategy can avoid the high similarity of antibodies, change the diversity of antibodies, and increase detection efficiency. This improves the convergence of the algorithm.

4.2.3. Step Description of GIDA. The algorithm combining with those operators mentioned above is called GIDA in Edge IDS, and the flow diagram of GIDA is illustrated in Figure 7.

Table 4: The partial samples in 10% KDDCup99 dataset.

Label	Attack type	Attack code	Attack name	Number of exercising set	Number of test set
0	NOM-AL	0	/	97278	60593
1	PRO-BING		/	4107	4166
		1	Ipsw-eep	1247	306
		2	mscan	—	1053
		3	nmap	231	84
		4	Portsw-eep	1040	354
		5	Saint	—	736
		6	Satan	1589	1633
2	DOS		/	391458	229853
		7	Apache2	—	794
		8	Back	2203	1098
		9	Land	21	9
		10	Mailbomb	—	5000
		11	Neptune	107201	58001
		12	Pod	264	87
		13	Processtable	—	759
		14	Smurf	280790	164091
		15	Teardrop	979	12
		16	udpstorm	—	2

In GIDA, the system information which needs to be protected will be acquired in real time. The extracted information is converted to binary code. The process is called initialization. Then, values of some parameters will be given in the algorithm. In the beginning of the algorithm, the positive selection operator will extract the key information as the "gene." They will be a part of the first generation of detectors. The other part of the first generation of detectors can be provided by negative selection and clone selection operator. After the 1st detection is generated, it will be used to detect intrusion. If the match is yes, the system will alert and generate a vaccine operator. If no, a complement operator will be added to the algorithm. Thus, the 2nd generation of detectors appears. As a result, continuous detection has been produced with the N^{th} generation detector. The system will finally approach the performance of real-time detection.

5. Performance Evaluation

As mentioned above, a novel Edge IDS using immunity is set up. In this section, the performance of the proposed Edge IDS as well as the GIDA will be evaluated in detail, with comparative simulation experiments.

5.1. Simulation Background. As a part of a distribution system, the proposed Edge IDS is mainly simulated. In order to establish a trustworthy experimental environment and verify the effect of the system, we completed the experiments on the KDDCup99 dataset. The recall rate, accuracy rate, precision, false negative rate, and F-score were used for assessing the performance of the proposed system. At the same time, for the aim of proving the effect of our algorithm, we compare it with the LISYS [26] and DynamiCS [28].

In the KDDCup99 dataset, there are four usual types of attacks considered, and the ways of attack are described in detail as follows:

$$\begin{cases} (1)\ \text{PROBING,} & \text{port monitor or scanner,} \\ (2)\ \text{DOS,} & \text{denial-of-service attack,} \\ (3)\ \text{R2L,} & \text{remote to user attacks,} \\ (4)\ \text{U2R,} & \text{user to root attacks.} \end{cases} \tag{9}$$

5.2. Establishment and Security Advantages of the Edge IDS. Based on the abovementioned, we build a proposed Edge IDS model based on the principle of immunity.

An experiment is reported in intrusion detection.

The setting parameters are as follows: m is the amount of alphabet symbols, $m = 2$; l is the size of a random string, $l = 32$; r is the threshold of matches, $r = 8$; total populations = 200.

In order to be convenience for the experiment, the KDDCup99 dataset provides 10% exercising set and detection set. Its partial samples are shown in Table 4.

The original data type is not uniform. Binary coding is the only encoding form that the computer can recognize. And a dynamic r-proximity bit matching algorithm based on the weight in the paper is used. For the convenience of the experiment, we need to convert the KDDCup99 dataset into a binary form. Therefore, it is necessary to preprocess the dataset. The standardized method of data is shown in Table 5, where standardization of protocol type data, service type data, and property data are realized. The numbers in parentheses represent the amount of data of various types.

Table 5: Standardized methods of data.

Standardization of protocol type data: protocol type (assignment)						
TCP (1)	UDP (2)	ICMP (3)	Others (4)			
Standardization of service type data: service type (assignment)						
http (1)	smtp (2)	Fingers (3)	Eco_i (4)	ftp_data (5)	ftp (6)	Domain_u (7)
Hostnames (8)	Imap4 (9)	mtp (10)	Netstat (11)	Private (12)	Systat (13)	Telnet (14)
ecr_i (15)	Time (16)	uucp (17)	Login (18)	urp_i (19)	Pop_3 (20)	Auth (21)
Other (21)	URP_I (22)	Others (23)				
Standardization of property data: property type (assignment)						
REJ (1)	S1 (2)	RSTO (3)	RSTR (4)	SO (5)	SH (6)	SF (7)
S2 (8)	RSTRSO (9)	Others (10)				

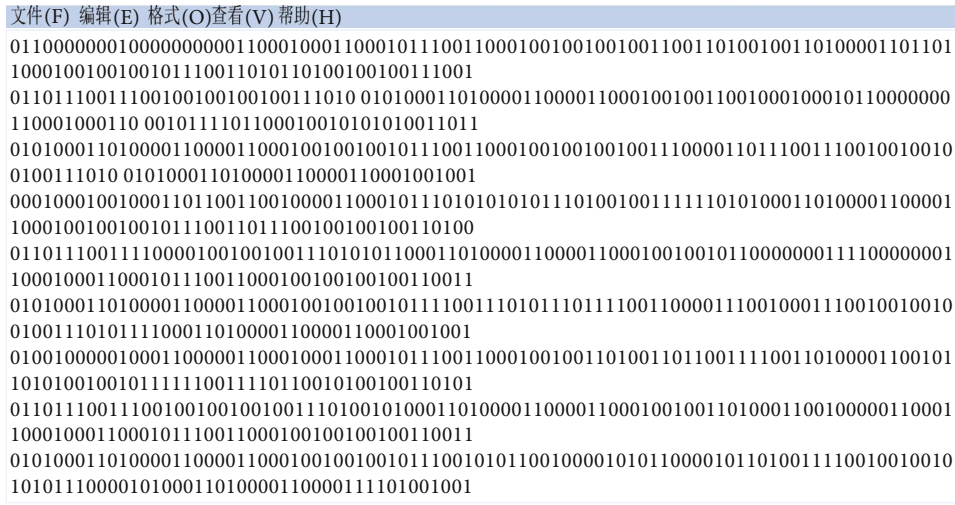

文件(F) 编辑(E) 格式(O)查看(V) 帮助(H)
01100000001000000000011000100011000101110011000100100100100110011010010011010000110110110001001001001001110011010110100100111001001100100100111001
01101110011100100100100100111010 01010001101000011000011000100100110010001000101100000001100010000110 00101111011000100101010100111011
01010001101000011000011000100100100101110011000100100100100111000011011100111001100100100100111001
0100111010 01010001101000011000011000100100100101110011000100100100100111000011011100111001100100100100111001
00010001001000110110011001000011000101110101010101110100100111111101010001101000011000011000100100100100111001101110011001000011000101110100100100111001
10001001001001011100110111001001001001101000
01101110011110000100100100111010101100011010000110000110001001001011000000011110000000110001000110001011100110001001001001001100111
10001001100010111001100010010010010011100111
01010001101000011000011000100100100101111001110101110111100110000111001000111001001000111001000011100100100100111001101110011001000011000101110100100100111001
01001110101111000110100001100001100010010010
01001000001000110000011000100011000101110011000100100110100110110011110011010000110010110101001110011101000011000101110100100100111001101110011001000011000101110100100100111001
10101001001011111100111101100101001001101010
01101110011100100100100100111010010100011010000110000110001001001101000110010000011000110101001110011101000011000101110100100100111001101110011001000011000101110100100100111001
01010001101000011000011000100100100101110011000100100100100111000011011100111001100100100100111001
10101110000101000110100001100001111010010010

Figure 8: Standardized data format.

By classifying the data shown in Table 5, we need to convert these files to binary and take 32 characters as the antibody length. This makes it easy for the program to extract *self* and *nonself*. The process is called the standardization of data formats. Standardized data format is shown in Figure 8.

In order to inhibit the state of the experiment clearly, the item content, step, and result are shown in Table 6. The test is divided into six items. The detection of vulnerable attacks in edge calculations is shown. At the same time, the operating steps and the results that need to be presented accordingly are introduced in Table 6.

The concrete step of the experimental algorithm is shown in Table 7. From these steps, the basic idea of the algorithm is information extraction and detector generation. The detector is updated from generation to generation. In this way, the intrusion detection system dynamically adapts to the data modification in edge computing.

In Table 8, the experimental result is reported. Here, L_S is the size of the source file, L_{Sa} is the size of the file after intrusion, N_R is the number of detectors, Time is the time spent for detection, and P_f is the failure rate of detection.

In Table 8, we set six types of experiments. Firstly, we find that after a file is attacked, its length may change. While a file is short, the whole file needs to be detected. If a file is long enough, it is unnecessary to detect the whole one. The way of sampling the file is adopted in our experiment. We take out samples from the head, tail, and midst of the file. Due to the characteristics of edge computing data files, we use dynamic sampling methods. We selected the same length of data files from the test library to test the two attacks mentioned. Obviously, the system responds to attack quite quickly. The failure rate is also low. From the experimental data, the effect is credible.

Secondly, the detection failure rate in the experiment is within a reasonable range. The number of detectors is related to a positive proportion of detection efficiency, but the results are not clear enough.

In row N_R, the data in every line shows different populations of detectors. For example, in the item of changing password, 67, 90, and 103 represent the number of the 1^{st}, 3^{th}, and 5^{th} generations of detectors, respectively. Every population of detectors is generated dynamically. we find that the value of P_f of the latter is less than that of the former. It illustrates that the system achieves better dynamic adaptability and improves the detection efficiency.

Finally, the performance of Edge IDS against attacks is evaluated. Two types of attacks are tested. Under the various attacks, the effect of the experiment is shown in Table 8. It

Table 6: Detection ways for the intrusion behaviors.

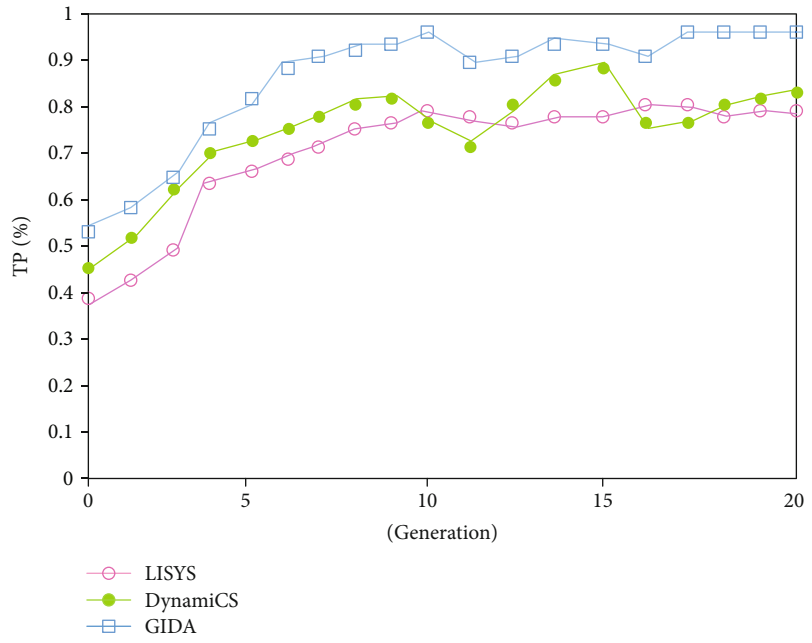
Testing item	Operation step	Anticipation result
Changing password	(1) Obtain authorization of root (2) Changing password file: vi/etc/password (3) Add a backdoor user with the following command lines: newuser:X: 00::/home/newuser:/bin/bash	The number of antigens detected. Computing P_f, FP, and TP
Setting script SUID bit	(1) Obtain authorization of root (2) Setting SUID bit:chmod –perm–4000/bin/tcsh (3) Owner of file is instead of root: chown root tcsh	
Changing the important file self-defined by user	(1) Obtain authorization of root (2) Changing the important file self-defined by user	
Changing host computer's log file	(1) Acquire authorization of root (2) Landing with a new user name, modifying host computer's log file	
Probing attack/insweep/portsweep	(1) Take out the exercising set (2) Generating the detectors by using GIDA (3) Detecting the test set	
DoS attack/back/Neptune	(1) Take out the exercising set (2) Generating the detectors by using GIDA (3) Detecting the test set	

Table 7: The detailed steps of GIDA.

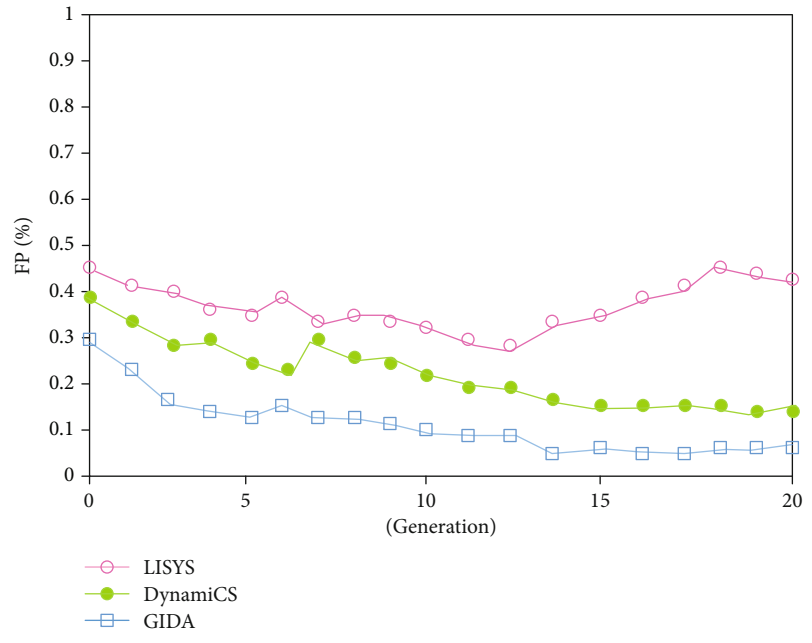
Step 1	Acquire information monitored.
Step 2	Take out genes. Gene detection is done. Once a match is successful, then alert.
Step 3	Generate strings with equal length in random, achieve an initial antibody population. Carry out niching strategy, use immune operators, and get *nonself* set.
Step 4	Evolve population, obtain next generation antibodies from population mentioned in step 3.
Step 5	Consider the *nonself* strings as a detecting aim, select antibodies by niching strategy. Gain the first-generation detector.
Step 6	For several different types of attacks, detecting the system files using the 1st detector.
Step 7	If match is successful, a part of antigens detected becomes vaccines. They can be added into vaccine library. And alert occurs and reports to the monitor layer. Generating vaccines and return step 4, continue.

Table 8: Experimental result.

Mode attacked	GIDA			L_S (bytes)	L_{Sa} (bytes)
	N_R	Time (s)	P_f		
Changing password	67	0.241	0.130	2006	1980
	90	0.301	0.091		
	103	0.409	0.041		
Setting SUID bit of script	110	0.204	0.350	387995	387995
	178	0.569	0.203		
Changing key files signed by user	89	0.017	0.255	14	12
	125	0.046	0.148		
	179	0.087	0.097		
Changing master computer's log files	58	0.104	0.062	2351	1869
	86	0.312	0.010		
Probing attack/insweep/portsweep	115	0.110	0.033	346577	346577
	150	0.087	0.0240		
DOS attack/back/Neptune	108	0.121	0.051	346577	346577
	140	0.101	0.044		



(a) Contrast of TP



(b) Contrast of FP

Figure 9: Comparison of detection ability of three algorithms.

can be observed that our IDS is effective in facing different types of attacks.

At the same time, the execution time is mainly shown. Obviously, the time is not ideal.

5.3. Detection Accuracy of the GIDA. In order to evaluate the accuracy of detection, we compare GIDA, DynamiCS, and LISYS. Though DynamiCS and LISYS are not applied for the area of intrusion detection, as classic immune algorithms, they are the basis of immune computing. The whole accuracy lies in TP and FP. It describes correct extent of an IDS's job. The experimental result is shown in Figure 9. The aim of the contrast is that immune algorithms can be applied for the area firstly. The essence of immunity is that it can resist any attacks from foreign damage, regardless of the attack style or model. Secondly, the data security in edge computing scenario has its special requirements. Therefore, the classic immune algorithms should be improved accordingly. As shown in Figure 9, the proposed system with GIDA has the highest TP rate and the lowest FP rate. For example, after less than 15 generations of iterative upgrades, the GIDA can achieve a TP larger than 0.9 and a FP less than 0.1, which

has obvious performance advantages over other algorithms. In fact, not all the attacks sometimes can cause an abnormal state. So the real-time performance of the Edge IDS should be evaluated. In Figure 9, the 1^{st}-20^{th} generations of detectors are generated. Obviously, the curves of FP and TP are growing. It shows that the system has better dynamic performance against attacks.

6. Conclusion

In this paper, an Edge IDS based on AIS is designed to adaptively detect threats in edge computing scenarios.

The basic idea of constructing an IDS inspired by biological immune principles is that there are such immune functions as resisting viruses and their variations in the biological immune system. A biological immune system can respond to unfamiliar viruses and their variants. It can have a self-adapting immune ability for unknown threats through the immune process. If the immune idea is incorporated into the design of IDS, it is possible to construct an intrusion detection system that adaptively detects novel attacks under the premise of less data requirement. Thus, the security of the edge computing scenario will be improved.

The contribution of this paper is in two aspects:

(1) The design of the system structure

(2) The introduction of an immune algorithm embedded into the Edge IDS

6.1. About the System Structure. Edge IDS is designed under the framework of a distributed system, which is connected with the Cloud IDS and the edge device security system. The Edge IDS mimics the structure of the AIS. It utilizes nonspecific immunity and specific immunity in biological immunity to respond to a variety of unknown and known invasion behaviors.

The Edge IDS is set up at the control layer. It is a feedback control system with feedforward control. Such a system structure has not only the stability and anti-jamming performance of the feedback system but also the timeliness of the feedforward control.

6.2. About the Algorithm. In this system, we have introduced an intrusion detection algorithm—GIDA. Shown in experiments, it is very valid for anomaly detection.

The algorithm is characterized by the rational use of genes, vaccines, and compensation operators, so that the effectiveness of the algorithm has been further improved. The application of niching strategy has also improved the algorithm.

(1) Self-learning

Vaccination presents an important advantage, that is, self-learning ability.

Being not similar to the other algorithms, our algorithm is not asked to forecast the number of initial antibodies. The original number may be less. Subsequently, the antibody library would be renewed by self-learning in real time.

However, the key condition of the algorithm implementation is that an antigen can be detected during initial detection at least. Therefore, whether or not the vaccine is produced, the testing effect can be guaranteed through learning experience. In addition, the algorithm will undoubtedly become more complex due to the presence of vaccines.

(2) Detection time

One of the effects of the complement operator is to avoid the shortcomings of the classic immune algorithms. Meanwhile, it is one of the reasons that the detection time can be decreased largely.

The complement operator is able to stimulate antibodies and make the antigens detect death. Therefore, in essence, the operator can help to decrease the number of antibodies and hold the diversity of the detectors.

Niching strategy is another effective way to reduce detection time. This strategy can obtain the diversity of antibodies according to thresholds, thereby increasing the coverage of the detectors, to shorten the detection time.

(3) Convergence of the algorithm

Because of the niching strategy, the similar individual can be limited to duplicate overmuch. Therefore, the variety of detectors is guaranteed.

Vaccines are not produced from all the matched antigens. Vaccines with low similarities are retained. Highly similar vaccines will be discarded. Thus, computational delays due to the large number of vaccines can be avoided.

Vaccines, niching strategy, and complement operator are able to add to the diversity of the population. The diversity will make the algorithm not fall into local optimization. Thus, the convergence of the algorithm is ensured.

6.3. About the Attacks. The ability of self-learning in the Edge IDS can avoid some specific attacks from those attackers who can get IDS for free. Similar to immune function, Edge IDS starts from random strings and it will detect intrusion through dynamic, constant self-learning. Thus, the detectors of Edge IDS will be renewed in time. If some attackers may get them for free, it is impossible to attack some nodes effectively.

There is varying behavior in the Internet, and the data source is limited for immune algorithms although there may be a huge amount of data scattered in the Internet. The malicious behavior of the Internet can be defined as abnormal. Innate immunity can detect any abnormal state. These states will be reported to the Cloud IDS, to perform some necessary security operations in case of malicious behaviors (perhaps not to be attacked).

6.4. About the Edge IDS, Next Steps to Be Done

(1) The convergence of the algorithm needs to be demonstrated via in-depth study

(2) The system stability and anti-jamming performance will be further evaluated

(3) There are many uncertainties in the system model, which require further qualitative analysis

(4) A diverse attack sample is needed to ensure the reliability and practical feasibility of the system

Data Availability

We used the KDDCup99, which is a famous and publicly accessed dataset (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html), for the evaluation of new algorithms in the proposed Edge IDS.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (NSFC) (No. 61702439).

References

[1] S. Li and W. Sun, "Utility maximisation for resource allocation of migrating enterprise applications into the cloud," *Enterprise Information Systems*, pp. 1–33, 2020.

[2] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[3] T. Shuminoski, S. Kitanov, and T. Janevski, "Advanced QoS provisioning and mobile fog computing for 5G," *Wireless Communications and Mobile Computing*, vol. 2018, 13 pages, 2018.

[4] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: state-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[5] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data Security and privacy-preserving in edge computing paradigm: survey and open Issues," *IEEE Access*, vol. 6, no. 4, pp. 18209–18237, 2018.

[6] P. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Commununications Magazine*, vol. 52, no. 11, pp. 65–75, 2014.

[7] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[8] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[9] G. Liu, W. Quan, N. Cheng, K. Wang, and H. Zhang, "Accuracy or delay? A game in detecting interest flooding attacks," *Internet Technology Letters*, vol. 1, no. 2, p. e31, 2018.

[10] K. Wang, D. Guo, and W. Quan, "Analyzing NDN NACK on interest flooding attack via SIS epidemic model," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1862–1873, 2020.

[11] J. Dong, K. Wang, W. Quan, and H. Yin, "InterestFence: simple but efficient way to counter interest flooding attack," *Computers & Security*, vol. 88, p. 101628, 2020.

[12] P. Marrack and J. W. Kappler, "How the immune system recognizes the body," *Scientific American*, vol. 269, no. 3, pp. 80–89, 1993.

[13] W. Chen, J. Zhou, and H. Wei, "Compensatory controller based on artificial immune system," *International Conference on Mechatronics and Automation*, 2006, pp. 1608–1613, Luoyang, China, 2006.

[14] Y. Zhang, Y. Xue, and S. Wu, "A gene immune detection algorithm with a strategy of DNA PRI," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, pp. 476–481, Shanghai, Chin, Aug. 2004.

[15] C. V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *Computers & Security*, vol. 29, no. 1, pp. 124–140, 2010.

[16] C. Mazzariello, R. Bifulco, and R. Canonico, "Integrating a network IDS into an open source cloud computing environment," in *2010 Sixth International Conference on Information Assurance and Security*, pp. 265–270, Miyazaki, Japan, 2010.

[17] I. Gul and M. Hussain, "Distributed cloud intrusion detection model," *International Journal of Advanced Science and Technology*, vol. 34, pp. 71–82, 2011.

[18] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: real-time intrusion detection in the internet of things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[19] S. Shamshirband, A. Patel, N. B. Anuar, M. L. M. Kiah, and A. Abraham, "Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 228–241, 2014.

[20] Y. Xiao, L. Bai, and X. Wang, "Friends mechanism-based routing intrusion detection model for mobile ad hoc network," *Journal on Communications*, vol. 36, no. S1, pp. 203–214, 2015.

[21] F. Hosseinpour, P. V. Amoli, and J. Plosila, "An intrusion detection system for fog computing and IoT based logistic systems using a smart data approach," *International Journal of Digital Content Technology & Its Applications*, vol. 10, no. 5, pp. 203–214, 2016.

[22] Y. Wang, L. Xie, W. Li, W. Meng, and J. Li, "A privacy-preserving framework for collaborative intrusion detection networks through fog computing," in *Cyberspace Safety and Security*, pp. 267–279, Xi'an, China, 2017.

[23] X. An, X. Zhou, X. Lü, F. Lin, and L. Yang, "Sample selected extreme learning machine based intrusion detection in fog computing and MEC," *Wireless Communications & Mobile Computing.*, vol. 2018, no. 9, pp. 1–10, 2018.

[24] F. Lin, Y. Zhou, X. An, I. You, and K.-K. R. Choo, "Fair Resource Allocation in an Intrusion-Detection System for Edge Computing: Ensuring the Security of Internet of Things Devices," *IEEE Consumer Electronics Magazine*, vol. 7, no. 6, pp. 45–50, 2018.

[25] A. S. Qi and C. Y. Du, "Immune system nonlinear module," *Shanghai Science and Technology Press*, pp. 67-68, 1998.

[26] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, pp. 271–281, Los Alamitos, USA, May 1994.

[27] J. Kim and P. J. Bentley, "Towards an artificial immune system for network intrusion detection: an investigation of clonal selection with a negative selection operator," in *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 1244–1252, Seoul, Korea, Aug 2001.

[28] J. Kim and P. Bentley, "Immune memory and gene library evolution in the dynamic Clonal selection algorithm," *Genetic Programming and Evolvable Machines*, vol. 5, no. 4, pp. 361–391, 2004.