J Inf Process Syst, Vol.15, No.1, pp.137~150, February 2019
https://doi.org/10.3745/JIPS.03.0108

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

# An Intrusion Detection Method Based on Changes of Antibody Concentration in Immune Response

Ruirui Zhang* and Xin Xiao**

**Abstract**

Although the research of immune-based anomaly detection technology has made some progress, there are still some defects which have not been solved, such as the loophole problem which leads to low detection rate and high false alarm rate, the exponential relationship between training cost of mature detectors and size of self-antigens. This paper proposed an intrusion detection method based on changes of antibody concentration in immune response to improve and solve existing problems of immune based anomaly detection technology. The method introduces blood relative and blood family to classify antibodies and antigens and simulate correlations between antibodies and antigens. Then, the method establishes dynamic evolution models of antigens and antibodies in intrusion detection. In addition, the method determines concentration changes of antibodies in the immune system drawing the experience of cloud model, and divides the risk levels to guide immune responses. Experimental results show that the method has better detection performance and adaptability than traditional methods.

**Keywords**

Antibody Concentration, Artificial Immune, Cloud Model, Evolutionary Algorithms, Intrusion Detection

## 1. Introduction

With the rapid development of network technology and the extreme dependence of people on network applications, security problems of network systems are becoming more and more serious. On the basis of the natural defense mechanism of biological immune system which can identify self and non-self, artificial immune systems can be used to solve practical problems including network security, intelligent optimization, pattern recognition, control theory, etc. [1,2].

Forrest et al. [3] from the University of New Mexico were the earliest scientists to introduce biological immune mechanism into anomaly detection field, and they thought that the anomaly detection problem can be regarded as the problem of distinguishing between self and non-self. Self refers to the legitimate, users or protected data, and non self refers to unauthorized users or virus-distorted data. Accordingly, she proposed the negative selection algorithm drawing the experience of immune tolerance of lymphocytes, and applied to file virus detection. After that, Hofmeyr and Forrest [4,5] further proposed

Manuscript received June 29, 2017; first revision August 14, 2017; accepted September 7, 2017.

**Corresponding Author:** Ruirui Zhang (zhangruiruisw@gmail.com)
* School of Business, Sichuan Agricultural University, Yaan, China (zhangruiruisw@gmail.com)
** School of Computer Science, Southwest Minzu University, Chengdu, China (xiaoxin618@gmail.com)

the artificial immune system named ARTIS, and applied it to the network intrusion detection system. The contributions of Hofmeyr and Forrest [4,5] lay the foundation for the research of immune-based anomaly detection technology.

Since then, a large number of researchers began sustained and in-depth study on the theory and application of immune-based anomaly detection technology. Dasgupta and Forrest [6] for the first time put forward applying the negative selection algorithm to the field of fault detection and achieved very good results. Kim and Bentley [7] and Kim et al. [8] proposed an immune based anomaly detection algorithm which can adapt for dynamic changes of selves, named DynamiCS. Williams et al. [9] developed a distributed computer immune system named CDIS. Harmer et al. [10] extended CDIS to the field of virus detection. Aickelin et al. [11] proposed an intrusion detection model based on the immune danger theory. Greensmith et al. [12] proposed an anomaly detection model based on dendritic cells of the biological immune system.

The research of immune-based anomaly detection technology has made some progress, researchers have proposed a variety of algorithms which are suitable for static and dynamic environments of selves, and these immune-based algorithms have been applied in many fields such as computer virus detection, network intrusion detection, spam detection, fault detection. But these anomaly detection algorithms based on immune still have some defects which have not been effectively resolved, and the main defects are in the following.

Firstly, under the static-self condition, because there are holes in the non-self space coverage of detectors, detection rate is low in the anomaly detection. In the dynamic-self environment, due to the lack of adaptive capacity for detectors, there are problems of low detection rate and high false alarm rate. Secondly, there is a problem of the exponential relationship between training cost of mature detectors and size of self-antigens. If the self set is great, generating mature detectors will cost too much time. If the self set is small, false alarm rate of the system will be high [3].

In order to improve and solve existing problems of immune based anomaly detection technology, this paper proposed an intrusion detection method based on changes of antibody concentration in immune response, named AC-Id. The main contributions of the method are as follows. Firstly, the method introduces blood relative and blood family to classify antibodies and antigens and simulate correlations between antibodies and antigens. Secondly, the method establishes dynamic evolution models of antigens and antibodies in intrusion detection. Thirdly, the method determines concentration changes of antibodies in the immune system through the experience of cloud model, and divides risk levels to guide immune responses.

The remainder of this paper is organized as follows. The theories of the model including expressions of antigens and antibodies, expressions of affinities, definitions of blood relative and blood family, description of concentration computation of blood family, and implementation mechanism of cloud model are described in Section 2. The architectures of the model including processes of normal data modeling and intrusion detection, and evolution models of antigens and antibodies are described in Section 3. The effectiveness of AC-Id is verified in Section 4. Experimental results show that the method has better detection performance and adaptability than traditional methods. Finally, the conclusions are given in the last section.

# 2. Model Theory

## 2.1 Antibodies and Antigens

Antigens in the model include selves and non-selves. Self is normal network connection. Non-self represents abnormal network connection. Antigens and antibodies have similar architectures, and are represented by binary strings in the morphological space. If antibodies detect an antigen as non-self, they will direct the immune response.

Define $B=\{0,1\}^{length}$ as the set of all binary strings, $R$ as the set of real numbers and $N$ as the set of natural numbers.

$Ag$ is short for antigen and is defined as (1).

$$Ag = \{< d, type, lifetime >| d \in B, type, lifetime \in N\} \tag{1}$$

where $d$ is the determinant of $Ag$, and consists of $m$ characteristic gene segments. $d$ can be expressed as $d = (d_1, d_2, \ldots, d_m)$. $d_i$ represents the $i^{th}$ component of $d$, $d_i \in \{0,1\}^{l_i}, i = 1,2, \ldots, m$, $l_i$ is the length of $d_i$. $m$ is the number of gene segments which compose $d$. *type* is the type of antigen, and its values are 0 and 1. 0 represents the intrinsic antigen, and 1 represents the foreign antigen. *lifetime* represents the life span of antigen.

Antibody $Ab$ can be split into immature one $AbI$, mature one $AbT$ and memory one $AbM$. Immature antibody $AbI$ is a newly formed immune cell that has not undergone self-tolerance, and is defined as (2). Mature antibody $AbT$ represents the immune cell who passes tolerance and is not activated by antigens, and is defined as (3). Memory antibody $AbM$ represents the immune cell who matches a certain number of antigens and is activated by antigens, and is defined as (4).

$$AbI = \{< d, age > | d \in B, age \in N\} \tag{2}$$

$$AbT = \{< d, age, consistency, count > | d \in B, age, count \in N, consistency \in R\} \tag{3}$$

$$AbM = \{< d, age, consistency, count > | d \in B, age, count \in N, consistency \in R\} \tag{4}$$

$d$ is the determinant of $Ab$, $age$ is the age of $Ab$, *consistency* is the density of $Ab$, and *count* is the number of antigens that $Ab$ matches.

Both $Ab$ and $Ag$ bases are composed of gene segments. Gene segments are extracted from key components of IP packets. All possible values of each gene segment are collected into the gene pool, and corresponding gene values are randomly selected from each gene segment in the gene pool to form legitimate genes. In this paper, gene segment types include source address (32 bit), service type (8 bit), source port (16 bit), protocol type (8 bit), destination address (32 bit), destination port (16 bit), IP packet length (16 bit), packet partial content (16 bit), and so on.

## 2.2 Affinity Computation

The affinities between antigen and antigen, antigen and antibody, antibody and antibody are defined as the match between their data structures. We adopt the improved r-continuous bits matching rule, and is expressed as (5).

$$f_{affinity}(d1, d2) = \begin{cases} 1, \sum_{i=1}^{m} f_{match}(d1.d_i, d2)/m \ge \theta \\ 0, others \end{cases} \tag{5}$$

$d1 \in B, d2 \in B, m$ is the number of gene segments which compose $d1$ and $d2$, and $\theta$ is the matching threshold. $f_{affinity}$ equals to 1, which means that $d1$ and $d2$ are matched. $f_{match}$ is expressed as (6). $l$ is the length of binary string $y$.

$$f_{match}(x, y) = \begin{cases} 1, \exists i, j, j - i \ge |x|, 0 < i \le j \le l, x_i = y_j, x_{i+1} = y_{j+1}, \dots, x_{|x|} = y_{i+|x|-1}, \\ 0, others \end{cases} \tag{6}$$

## 2.3 Blood Relative and Blood Family

In this paper, we adopt blood relative and blood family to simulate correlations between antibodies. Blood relative is defined as (7), and $\theta$ is the matching threshold.

$$blood\ relative = \{< x, y > |f_{affinity}(x.d, y.d) \ge \theta \cap x, y \epsilon Ab\} \tag{7}$$

For any set $X$ of antibodies, if $\forall x, y \in X, < x, y > \in blood\ relative$ exists, that is, the affinity of any element *x* and *y* is higher than the given threshold, then *X* is named a blood relative. If any element of *ab-X* is not consanguinity, it is said to be the largest sanguinity of $Ab$.

Set that $= \{\varphi_1, \varphi_2, \dots, \varphi_n\}$, Ab$^1$ = Ab, $Ab^i = Ab - \bigcup_{1 \le j < i \le n} \varphi_j$. $\varphi_i$ is a largest blood relative with most elements of Ab$^i$, and $Ab = \bigcup_{1 \le i \le n} \varphi_i$, then $\omega$ is called a blood family. Set $1 \le j < i \le n$, there exists $\varphi_i \cap \varphi_j = \emptyset$. Table 1 illustrates steps of getting a blood family from antibody set X. The main idea is in the following. All the antibodies are regard as vertexes in an undirected graph. Then divide the graph, and figure out all the maximal complete sub-graphs. Each sub-graph is a blood relative, and all the sub-graphs compose blood family.

**Table 1.** The algorithm for blood family

```
Input: $X = \{ab_1, ab_2, \dots, ab_n\}$
Output: $\omega = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$
Step 1. $\omega = \emptyset$.
Step 2. Calculate affinities between $ab_i$ and $ab_j$ $(1 \le i \le n, 1 \le j \le n)$. If $<ab_i, ab_j> \in$ blood relative, there exists an
        edge $e_{ij} = <ab_i, ab_j>$ between $ab_i$ and $ab_j$. Because the relationship of consanguinity is mutual, we use
        undirected edge to replace the bidirectional edge. So undirected graph $G = <V, E>$ is generated. $G.V = X$ is
        a non-empty finite set, which is vertices of graph $G$. $G.E = \{e_{ij} \mid e_{ij} = (ab_i, ab_j) \in$blood relative$\}$, which is edges
        of graph $G$.
Step 3. Find the entire maximal complete sub-graph $X' = \{X_1, X_2, \dots, X_k\}$. Where, $X_i = < V, E > (1 \le i \le k)$.
Step 4. Select $X_i$ in $X'$. $X_i = \left\{ag \middle| ag \in X_i.V, |X_i.V| = \max_{1 \le j \le k} |X_j.V| \right\}, \varphi = X_i.V, \omega = \omega + \varphi$.
Step 5. Set $X' = X' - \{x | x \in X', \forall ag(ag \in x.V) \notin X_i.V\}$.
Step 6. Go to step 4, until $X' = \emptyset$.
```

## 2.4 Density Computation of Blood Family

On the micro level, density of each blood family consists of density of each antibody in the set. The status of antibody density will directly reflect the security situation of the network. In this paper, rules for the change of antibody density are as follows.

The immature antibody will have an initial density when it is changed into mature antibody.

When an antigen is identified as non-self, it will issue stimulus signals $\eta$ to antibodies. When the antigen matches a memory antibody, the antigen will produce a stimulus signals to the corresponding antibody and the antibody density increases; when the antigen matches a mature antibody, it will produce a stimulus signal to the corresponding antibody as well and the antibody density increases.

Normal death of an antigen causes an inhibiting signal $\zeta$ to the antibody. That is, when the memory antibody does not match the antigen within a certain time, the antibody density will decrease. When the mature antibody does not match the antigen for a certain period of time, the antibody density will also decrease. When the life cycle of a mature antibody reaches a threshold and is not activated, the antibody is deleted. The formula is as follows.

$$Consistency(t) = f_{init}(t) + f_{\eta}(t) - f_{\zeta}(t) = \sum_{i=1}^{n} ab_i(t).consistency \tag{8}$$

$n$ is the count of antibodies in this kind of family, $i$=1,2,... $n$. $f_{init}(t)$ is the initial density at the moment $t$. $f_{\eta}(t)$ is the density function of antibodies which match non-selves at the moment $t$. $f_{\zeta}(t)$ is the influence function of normal death of antigens on the density of antibodies at the moment $t$.

## 2.5 Cloud Model

Cloud model is an uncertain transformation model between a certain qualitative concept represented by linguistic value and its quantitative representation [13,14]. The biggest problem in intrusion detection system based on antibody density is how to judge the danger. In the process of judgment, risk and safety are qualitative concepts with uncertainties, while resources are quantitative data. Therefore, cloud model can be adopted to represent them.

Usually, we can monitor system variables like memory occupancy rate, CPU usage rate, I/O usage conditions, network delay, packet loss rate, network flow etc., and sample their changed values. Then we can construct the normal cloud and abnormal cloud to estimate danger. But there are too many variables which are related to each other. If you model multiple one-dimensional or multi-dimensional clouds, the error will be larger. In the immune system, when the system is invaded, the most direct change is the change of antibody concentration, which can reflect the network security situation. Therefore, the antibody concentration can be modeled to determine the risk.

First, collect data in the safe state. $t_0$ is the starting point of sampling, and $T$ is sampling interval. Antibody densities of different blood family are sampled respectively. Obtain $k$ sample points: $t_0\{A_{10}, A_{20}, \ldots, A_{n0}\}$, $t_1\{A_{11}, A_{21}, \ldots, A_{n1}\}$, …, $t_k\{A_{1k}, A_{2k}, \ldots, A_{nk}\}$. Reduce sample values between 0 and 1. In this way, the spatial distribution of sample points of each blood family forms a cloud. According to reverse cloud generator algorithm which is expressed as Table 2, we can obtain cloud's digital characteristics of every blood family in secure state $\{Ex_{safe1}, En_{safe1}, He_{safe1}\}$, $\{Ex_{safe2}, En_{safe2}, He_{safe2}\}$, …, $\{Ex_{safen}, En_{safen}, He_{safen}\}$

**Table 2.** The algorithm of reverse cloud generator

```
Obtain cloud's digital characteristics according to droplets. (Take density of blood family $A_1$ as an example)
Input: sample points $A_{11}$, ..., $A_{1k}$.
Output: ($Ex_1$, $En_1$, $He_1$)
Step 1. Calculate sample mean $\overline{A_1} = (1/k)\sum_{i=1}^{k} A_{1i}$, sample variance $S^2 = 1/(k-1)\sum_{i=1}^{k}(A_{1i} - \overline{A_1})^2$.
Step 2. $Ex_1 = \overline{A_1}$.
Step 3. $En_1 = \sqrt{\pi/2} \times (1/k)\sum_{i=1}^{k}|A_{1i} - Ex|$
Step 4. $He_1 = \sqrt{S^2 - En^2}$.
```

Known attacks are introduced to collect a number of sample points when the system is in danger, generate the cloud in the dangerous state in a similar way, and obtain the digital characteristics of the cloud in the dangerous state of each blood family $\{Ex_{dangerous1}, En_{sdangerous1}, He_{dangerous1}\}$, $\{Ex_{dangerous2}, En_{dangerous2}, He_{dangerous2}\}, \dots, \{Ex_{dangerousn}, En_{dangerousn}, He_{dangerousn}\}$.

According to the "rule-3En" of clouds, features of the less safe cloud and the less dangerous cloud can be estimated by (9) and (10).

$$Ex_{lesssafe} = Ex_{safe} + 3En_{lesssafe} = Ex_{safe} + 3 * 0.618En_{safe} \tag{9}$$

$$Ex_{lessdangerous} = Ex_{dangerous} - 3En_{lessdangerous} = Ex_{dangerous} - 3 * 0.618En_{dangerous} \tag{10}$$

# 3. Model Architecture

## 3.1 Overall Process

The structure of the system is divided into two parts: normal data modeling and intrusion detection.

The purpose of normal data modeling is to establish the cloud model of the density of each blood family. Firstly, land attack, Smurf attack, death of ping attack and more than a dozen other attacks were used to initialize the system to generate the initial antigen set and the initial memory set. In addition, according to the blood relationship between mature antibodies and memory antibodies, the initial blood family were divided. This is $t_0$. Then, under the normal state of the system, sampling is conducted at intervals of $T$ for $k$ times. At each sampling point, the density of each blood family was calculated to obtain $k$ cloud droplets. According to the algorithm of reverse cloud generator, the digital characteristics of each blood family in a safe state could be calculated. Similarly, at the time of $t_0$, known attacks are introduced respectively. Under the condition that the system is only in the state of one attack, samples are taken at intervals of $T$ for $k$ times respectively. According to the sample values, the digital characteristics of clouds of each blood family in the state of danger are calculated. Then, according to (9) and (10), the numerical characteristics of less safe cloud and less dangerous cloud are calculated. At this point, the data modeling is completed, the one-dimensional variable cloud of each risk level can be obtained.

The purpose of intrusion detection is to judge if the system is under abnormal attack. The process is shown in Fig. 1.

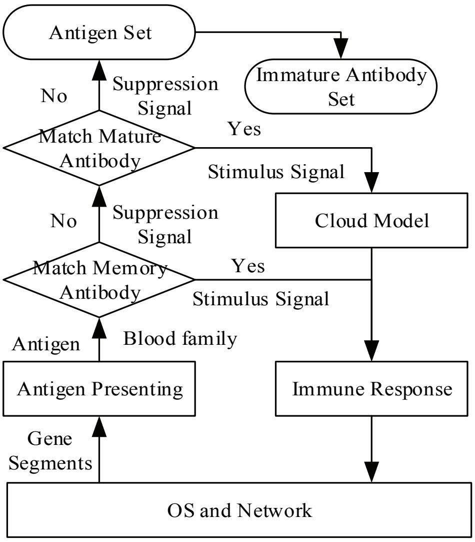

**Fig. 1.** The process of intrusion detection.

After receiving the IP packet, the system extracts the gene segment from the antigen extraction module and encodes it into the epitope, adds the antigen collection, and classifies the antigens by blood families. Then antigen and memory antibody conduct affinity matching. When the affinity is greater than a certain threshold, the antigen is considered to be non-self, and the following operations are performed, that is, remove antigen from the antigen collection, increase the density of matching antibody, and trigger a secondary response. If no secondary response is triggered, the antigen will continue to match with mature antibodies. When the affinity is greater than a certain threshold, the density of the matching antibody will be increased. At this time, the density of the antigens' blood family will be calculated and the membership degree of the risk level of the density will be obtained. When the density of system causes secondary response or triggers a primary response, the antibody varies based on the clonal selection algorithm, and produce new antigens with higher affinity between itself and antibodies to identify danger faster, and produce some original antigens with low affinity into immature collection to guarantee the diversity of the system.

## 3.2 Antigen Evolution Model

In the actual situation, a network activity is considered to be normal at a moment, and is likely to be considered illegal the next moment. In the time $t_0$, administrator opened ftp services in the server, and at this time the network's 21 port connection was normal. In the time $t_1$, administrator closed the FTP service, and the network connection of 21 port was illegal. Therefore, the antigen set in the immune system is evolving. Antigens are divided into inherent ones and foreign ones. The inherent antigen set was unchanged, and the lifetime was always the maximum value $T_L$. The foreign antigen set dynamically changes. The lifetime of newly added antigens is $T_L$, and then decreases gradually until 0. When the lifetime of the external antigen is 0, delete the antigen. The formula is expressed as (11).

$$f_{ag}(t) = \begin{cases} \{< d, type, lifetime > | selves\ when\ t = 0\} & t = 0 \\ f_{ag}(t-1) + f_{agnew}(t) - f_{agdead}(t) & t \geq 1 \end{cases} \tag{11}$$

$f_{agnew}(t) = \{< d, type, lifetime > \mid type = 1, lifetime = \ T_L, d \text{ is newly added determinant at time } t\}$,
$f_{agdead}(t) = \{< d, type, lifetime > \mid type = 1, lifetime = \ 0, d \text{ is identified as non} - \text{self}\}$.

In the immune system, because of limited resources, the situation of the number of antigens increasing with time should be avoided. The number of antigens is assumed to $C_{ag}$. The length of the intrinsic antigen set was $C_{inherent}$, and the length of the foreign antigen set was $C_{foreign}= C_{ag} - C_{inherent}$. The antigen life span should be as large as possible so as to override more self-space and reduce false positive rate. The relationship between $T_L$ and $C_{ag}$ is as follows.

It is assumed that the number of foreign antigens in the time $t_0$ is $N_0$ and the life time of these antigens is $T_L$, then in the time $t_0$ $N_0 \leq C_{foreign}$;

It is assumed that the number of foreign antigens in the time $t_1$ is $N_1$ and the life time of antigens from $t_0$ is $T_{L\text{-}1}$, then in the time $t_1$ $N_0 + N_1 \leq C_{foreign}$;

…

It is assumed that the number of foreign antigens in the time $t_{TL\text{-}1}$ is $N_{TL\text{-}1}$ and the life time of antigens from $t_0$ is 1, then in the time $t_{TL\text{-}1}$ $N_0 + N_1 + \cdots + N_{TL-1} = \sum_{i=0}^{TL-1} N_i \leq C_{foreign}$;

Ideally, the number of foreign antigens at each moment is equal, that is to say, $N=N_0=N_1=\ldots= N_{TL\text{-}1}$, then $T_L \times N \leq C_{ag} - C_{inherent}, T_L \leq (C_{ag} - C_{inherent})/N$.

## 3.3 Antibody Evolution Model

Antibody set includes immature antibody set, mature antibody set and memory antibody set.

The immature antibody set consists of two parts. One part is made up of different gene segments randomly selected from the gene pool, and the other part is generated by the variation of the antibody according to the clonal selection algorithm in the immune response. Newly generated immature antibodies should be compared with the antigen set according to the negative selection algorithm, and these ones which match selves should be deleted. At this time, the age of these newly generated immature antibodies was 0. Then, the immature antibody will pass tolerance and become mature. The size of the antibody set is limited. The above procedure is expressed as follows.

$$f_{abi}(\mathrm{t}) = \begin{cases} \{< d, age > \mid antibodies\ \ when\, t = 0\} & t = 0 \\ f_{abi}(t-1) + f_{abinew}(\mathrm{t}) - f_{abit}(\mathrm{t}) & t \geq 1 \end{cases} \tag{12}$$

$f_{abi}(t) = \{< d, age > \mid 0 < age < T_{tolerance}, d \text{ is immature antibodies in the time } t - 1\}$,
$f_{abinew}(t) = \{< d, age > \mid age =$
$0, d \text{ is antibodies who passes negative selection algorithm in the time } t\}$, $f_{abit}(t) = \{< d, age > \mid age <$
$T_{tolerance}, d \text{ is antibodies which will be add into the mature antibody set in the time } t\}$.

When immature antibody *abi* becomes mature *abt*, *abt.d*=*abi.d*, *abt.age*=0, *abt.consistency*=$\eta_{t0}$, *abt.count*=0. In the life span of $T_{mature}$, when a non-self and a mature antibody match, the match count of this antibody increases by 1, and the density increases by $\eta$; densities of other antibodies decrease by $\zeta$. In the life cycle, the mature antibody will be removed if it matches an antigen known to be self. In addition, if the immune response is not triggered during the life cycle, the mature antibody will be erased. If the immune response is triggered, the mature antibody will develop into memory one on the basis of the clonal selection algorithm. This procedure is formulated as follows.

$$f_{abt}(t) = \begin{cases} \emptyset \ \ t = 0 \\ f_{abt}(t-1) + f_{abtnew}(t) - f_{abtm}(t) - f_{abtdead}(t) \ \ t \geq 1 \end{cases} \tag{13}$$

$f_{abinew}(t) = \{< d, age, consistency, count > \mid age = 0, consistency = \eta_{t0}, count = 0, d \text{ is new mature antibody}\}$, $f_{abtm}(t) = \{< d, age, consistency, count > \mid consistency \in \text{the danger} - \text{state cloud}\}$, $f_{abtdead}(t) = \{< d, age, consistency, count > \mid age \geq T_{mature}, consistency \notin the\ danger - state\ cloud \cup \exists x \in self(f_{affinity}(d,x) \geq \theta)\}$.

Concentration functions of $\eta$ and $\zeta$ are very important. Curve of $\eta$ is related to attack intensity $\tau$ which increases with the rise of $\tau$. In this way, under continuous attack, this can shorten the duration of immune learning, and the immune system can respond quickly. $\zeta$ should be a function of attack intensity $\tau$ which will gently increase with the reduce of $\tau$. In this way, if an attack occurs again in a relatively short time, the system can keep a high degree of vigilance. $\eta$ and $\zeta$ satisfies the following formulas.

$$\eta(\tau) = (e^{\sqrt{\tau}})^{0.2} - 1 \tag{14}$$

$$\zeta(\tau) = 0.2log(\tau + 1) \tag{15}$$

When the mature antibody *abt* triggers an immune response in the life cycle, it will transform into memory one *abm*, *abm.d*=*abt.d*, *abm.age*=0, *abm.consistency*=*abt.consistency*, *abm.count*=*abt.count*. In the life cycle of memory antibody, when it matches a non-self, its matching count increases by 1, the density increases by $\eta$, and it triggers an immune response; densities of other antibodies decrease by $\zeta$. In the life cycle, the memory antibody will be removed in the case of matching an antigen known to be self. This procedure is expressed as follows.

$$f_{abm}(t) = \begin{cases} \emptyset & t = 0 \\ f_{abm}(t-1) + f_{abmnew}(t) - f_{abmdead}(t) & t \geq 1 \end{cases} \tag{16}$$

$f_{abmnew}(t) = \{< d, age, consistency, count > \mid age = 0, d \text{ is new memory antibody}\}$,
$f_{abmdead}(t) = \{< d, age, consistency, count > \mid \exists x \in self(f_{affinity}(d,x) \geq \theta)\}$.

# 4. Experimental Results and Analysis

## 4.1 Parameter Settings

The experiments were carried out in the laboratory of Sichuan Agricultural University. Ten percent of the compact data set of KDDCUP99 provided by MIT Lincoln Lab was used as experimental data, including a large amount of normal network traffic and various attacks [15].

Limited by the physical properties of the machine, such as memory size, processing speed, we confine the number of antigens and antibodies in the experiments. Set that the size of antigen set is 200, and the number of non-memory antibodies is 300, and the size of memory antibody set is 200. Generally, there is little change about network normal behavior, so the tolerance period of immature antibody is $T_{tolerance} = 1$. In order to leave immune cells enough time to identify non-self, the larger the better without packets loss, so the updating cycle of antigens is $\varepsilon$=50. Fig. 2 shows the influence of the matching threshold $\theta$ on detection rate (TP) and false alarm rate (FP). When $\theta$ is small, the false alarm rate is relatively large because mature antibodies are activated without enough study. So, we select the value of $\theta$ as 0.8. Concentration functions $\eta$ and $\zeta$ are related to the attack intensity. In fact, antigen-matching count is used to approximate the attack intensity of the function $\eta$, and $t$ is used to approximate the attack intensity of the function ζ. Fig. 3 shows the effects of mature antibody life cycle $T_{mature}$ on detection rate and false alarm rate. The small life cycle will lead to lower TP because mature cells do not have enough time to wait for expected non-self. However, a larger life cycle will also lead to a higher FP. So set the life cycle of mature antibody $T_{mature}$=120 according to experimental results.

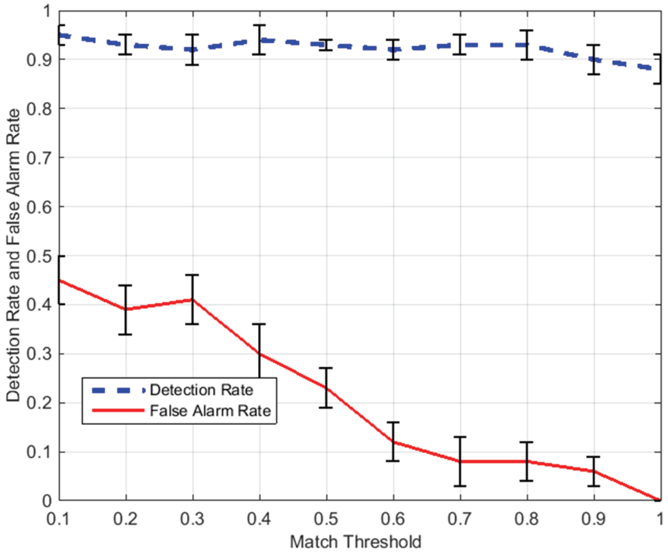

**Fig. 2.** The influence of matching threshold on false alarm rate (FP) and detection rate (TP).

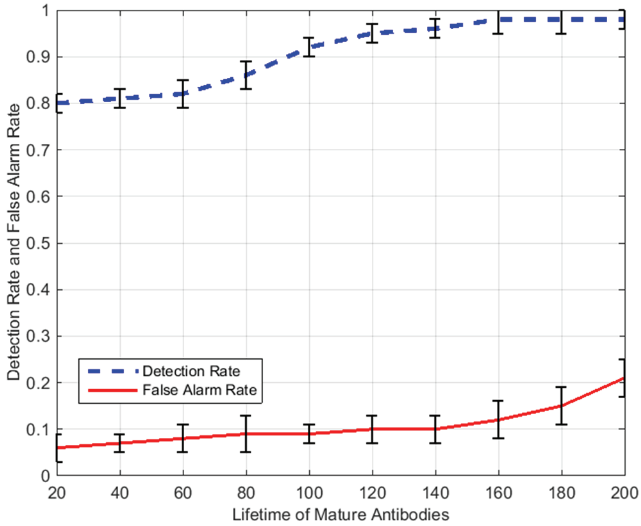

**Fig. 3.** The effects of mature antibody life cycle on false alarm rate (FP) and detection rate (TP).

## 4.2 Comparisons of TP and FP

To test the performance of AC-Id, we conducted targeted comparison experiments. The comparison object was DynamiCS algorithm proposed by Forrest et al. [3] and Kim and Bentley [7], a typical representative of the immune based intrusion detection algorithm, which has an important influence on the later design of intrusion detection systems.

Figs. 4 and 5 show comparisons of detection rates of DynamiCS and AC-Id. In the experiment of Fig. 4, there are 80 non-selves for every 100 packets, and 40 non-selves are just determined, which means that this type of IP packets is considered to be self before and is now considered to be illegal network behavior. For example, shut down 40 ports to stop providing related services in emergency. In the experiment of Fig. 5, we use reduced 10% of KDDCUP99.

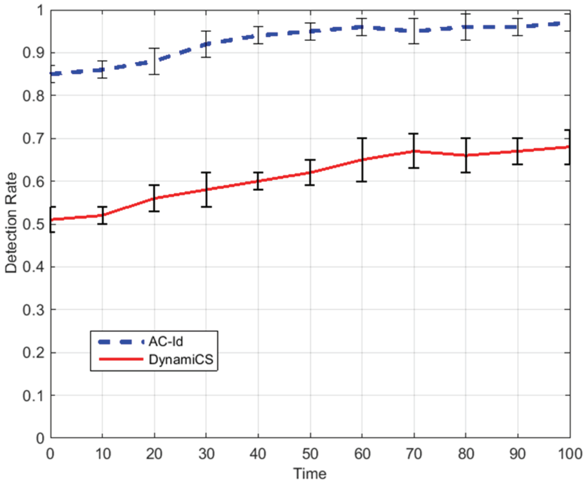

**Fig. 4.** Comparisons of detection rate (TP) of DynamiCS and AC-Id under lab data.

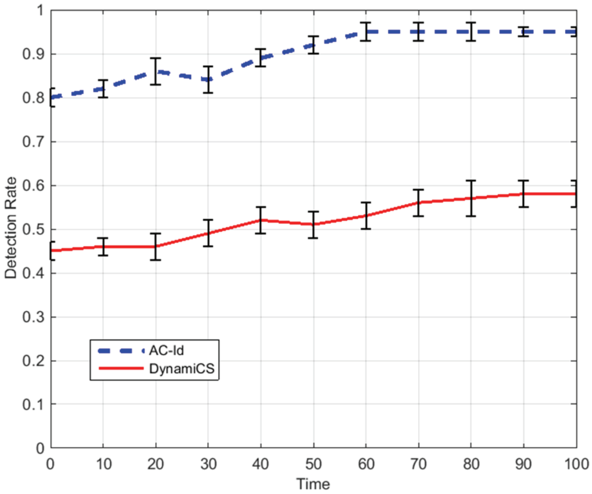

**Fig. 5.** Comparisons of detection rate (TP) of DynamiCS and AC-Id under KDDCUP99 data.

Figs. 6 and 7 show comparisons of false alarm rates of DynamiCS and AC-Id. In the experiment of Fig. 6, there are 40 selves for every 100 packets, and 20 selves are newly defined. For example, 20 network ports have just been opened to offer new services. In the experiment of Fig. 7, we use reduced 10% of KDDCUP99.

The experimental results show that, compared with AC-Id, DynamiCS has lower TP and higher FP. The reason is that, in DynamiCS, definition of selves lacks flexibility, and cannot effectively identify the newly added antigens. On the contrary, immune cells of AC-Id avoid the tolerance of mutated selves through antigen and antibody evolution mechanism, and AC-Id reduces the false negative rate. AC-Id uses cloud to model density of antibodies, and density will increase rapidly with the increase of attack strength, and can accurately reflect the current security situation of the network environment, which improves the detection rate. In the meantime, AC-Id can refrain from the recognition of newly added selves by the mechanism of self-evolution and elimination of memory cells, which reduces the false alarm rate.

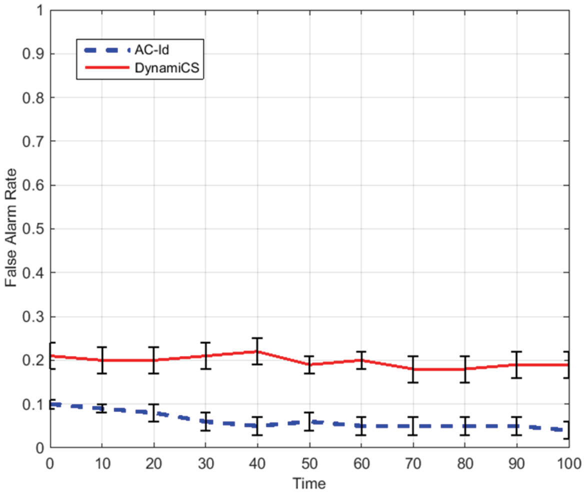

**Fig. 6.** Comparisons of false alarm rate (FP) of DynamiCS and AC-Id under lab data.

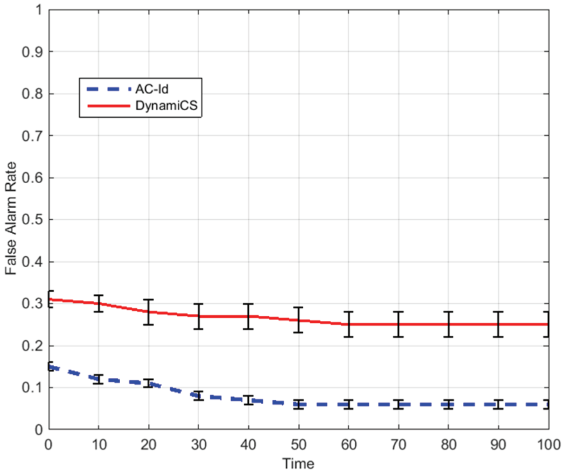

**Fig. 7.** Comparisons of false alarm rate (FP) of DynamiCS and AC-Id under KDDCUP99 data.

## 5. Conclusion

In order to improve and solve existing problems of immune based anomaly detection technology, this paper proposed an intrusion detection method based on changes of antibody concentration in immune response. Firstly, the method introduces blood relative and blood family to classify antibodies and antigens and simulate correlations between antibodies and antigens. Then, the method establishes dynamic evolution models of antigens and antibodies in intrusion detection. In addition, the method determines concentration changes of antibodies in the immune system drawing the experience of cloud model, and divides the risk levels to guide immune responses. This paper verified that this method has better adaptability than traditional methods through simulation experiments.

## Acknowledgement

The authors would like to thank Sichuan Provincial Education Department of China funded project (No. 035Z2258) for providing financial aid.

## References

[1] L. N. de Castro and J. I. Timmis, "Artificial immune systems as a novel soft computing paradigm," *Soft Computing*, vol. 7, no. 8, pp. 526-544, 2003.

[2] J. Shifflet, "A technique independent fusion model for network intrusion detection," in *Proceedings of the Midstates Conference for Undergraduate Research in Computer Science and Mathematics*, Wooster, OH, 2005, pp. 13-19.

[3] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy,* Oakland, CA,1994, pp. 202-212.

[4] S. Hofmeyr and S. Forrest, "Immunity by design: an artificial immune system," in *Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, FL, 1999, pp. 1289-1296.

[5] S. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary Computation*, vol. 8, no. 4, pp. 443-473, 2000.

[6] D. Dasgupta and S. Forrest, "Artificial immune systems in industrial applications," in *Proceedings of the 2nd International Conference on Intelligent Processing and Manufacturing of Materials*, Honolulu, HI, 1999, pp. 257-267.

[7] J. Kim and P. J. Bentley, "Towards an artificial immune system for network intrusion detection: an investigation of dynamic clonal selection," in *Proceedings of the Congress on Evolutionary Computation*, Honolulu, HI, 2002, pp. 1015-1020.

[8] J. Kim, P. J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, "Immune system approaches to intrusion detection: a review," *Natural Computing*, vol. 6, no. 4, pp. 413-466, 2007.

[9] P. D. Williams, K. P. Anchor, J. L. Bebo, G. H. Gunsch, and G. D. Lamont, "CDIS: towards a computer immune system for detecting network intrusions," in *International Workshop on Recent Advances in Intrusion Detection*. Heidelberg: Springer, 2001, pp. 117-133.

[10] P. K. Harmer, P. D. Williams, G. H. Gunsch, and G. B. Lamont, "An artificial immune system architecture for computer security applications," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 252-280, 2002.

[11] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod, "Danger theory: the link between AIS and IDS?," in *International Conference on Artificial Immune Systems*. Heidelberg: Springer, 2003, pp. 147-155.

[12] J. Greensmith, U. Aickelin, and G. Tedesco, "Information fusion for anomaly detection with the dendritic cell

algorithm," *Information Fusion*, vol. 11, no. 1, pp. 21-34, 2010.

[13] D. Y. Li, C. Y. Liu, and L. Y. Liu, "Study on the universality of the normal cloud model," *Engineering Science*, vol. 6, no. 8, pp. 28-34, 2004.

[14] D. Y. Li, H. J. Meng, and X. M. Shi, "Membership clouds and membership cloud generators," *Computer Research and Development*, vol. 32, no. 6, pp. 15-20, 1995.

[15] S. Rathore, A. Saxena, and M. Manoria, "Intrusion detection system on KDDCup99 dataset: a survey," *International Journal of Computer Science and Information Technologies*, vol. 6, no. 4, pp. 3345-334, 2015.

**Ruirui Zhang** https://orcid.org/0000-0003-1898-1487

She received B.S., M.S., and Ph.D. degrees in School of Computer Science from Sichuan University in 2004, 2007, and 2012, respectively. She is a lecturer at the School of Business, Sichuan Agricultural University, China. Her current research interests include network security, wireless sensor networks, intrusion detection and artificial immune systems.

**Xin Xiao** https://orcid.org/0000-0001-8703-4243

She received B.S., M.S., and Ph.D. degrees in School of Computer Science from Sichuan University in 2004, 2009, and 2015, respectively. She is a lecturer at the School of Computer Science, Southwest Minzu University, China. Her current research interests include network security, intrusion detection and artificial immune systems.