Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.045282

ARTICLE

Artificial Immune Detection for Network Intrusion Data Based on Quantitative Matching Method

Cai Ming Liu[1,2,3], Yan Zhang[1,2,*], Zhihui Hu[1,2] and Chunming Xie[1]

[1]School of Electronic Information and Artificial Intelligence, Leshan Normal University, Leshan, 614000, China

[2]Intelligent Network Security Detection and Evaluation Laboratory, Leshan Normal University, Leshan, 614000, China

[3]Internet Natural Language Intelligent Processing Key Laboratory of Education Department of Sichuan Province, Leshan Normal University, Leshan, 614000, China

*Corresponding Author: Yan Zhang. Email: zhangyan@lsnu.edu.cn, yan_zhang67@outlook.com

Received: 22 August 2023 Accepted: 11 December 2023 Published: 27 February 2024

ABSTRACT

Artificial immune detection can be used to detect network intrusions in an adaptive approach and proper matching methods can improve the accuracy of immune detection methods. This paper proposes an artificial immune detection model for network intrusion data based on a quantitative matching method. The proposed model defines the detection process by using network data and decimal values to express features and artificial immune mechanisms are simulated to define immune elements. Then, to improve the accuracy of similarity calculation, a quantitative matching method is proposed. The model uses mathematical methods to train and evolve immune elements, increasing the diversity of immune recognition and allowing for the successful detection of unknown intrusions. The proposed model's objective is to accurately identify known intrusions and expand the identification of unknown intrusions through signature detection and immune detection, overcoming the disadvantages of traditional methods. The experiment results show that the proposed model can detect intrusions effectively. It has a detection rate of more than 99.6% on average and a false alarm rate of 0.0264%. It outperforms existing immune intrusion detection methods in terms of comprehensive detection performance.

KEYWORDS

Immune detection; network intrusion; network data; signature detection; quantitative matching method

1 Introduction

Network intrusion data is one of the threats to network security. The early concept of intrusion detection (ID) came from the security monitoring system (SMS) proposed by Anderson in a technical report [1] in 1980. As an important supplement to network security defence technology, it can find unauthorized access events inside and outside the network and timely warn network managers. It is an active network security protection technology, and also an invisible defense line to protect network security. It has made positive contributions to the maintenance of network security. According to the different data sources, intrusion detection can be classified into network-based intrusion detection and host-based intrusion detection. Network-based intrusion detection takes network data streams as

detection objects to detect whether there are intrusions in network data streams. Host-based intrusion detection takes the terminal data on the host as the detection object, such as system logs, system activities, file information, etc., to detect whether the host system has been invaded. NIDS is concerned with the whole network system, whereas HIDS is only tied to one system and, as its name implies, is only interested in risks relating to the host system or computer. NIDS evaluates the actions and traffic of all the systems in the network.

In 1987, Denning published the paper "An Intrusion-Detection Model" [2]. Based on statistical analysis and the rule-matching method, this paper proposed a general intrusion detection technology framework. It is a model of early intrusion detection research [3]. The authors also used this technology as the basis to develop a general-purpose intrusion detection expert system (IDES). IDES is a stand-alone system that collects information on user activity on one or more monitored computer systems. IDES monitors individual users, groups, remote hosts, and whole systems for suspected security infractions committed by both insiders and outsiders [4]. IDES learns users' behaviour patterns adaptively over time and identifies deviations from these patterns [5]. IDES additionally has a rule-based component for encoding information about known system vulnerabilities and intrusion scenarios. Since the publication of Denning's achievements, researchers have carried out extensive research on intrusion detection [6]. To improve the comprehensive performance of intrusion detection, machine learning and intelligent methods such as association rules, clustering, Bayesian, neural network, artificial immune and imbalanced generative adversarial networks are applied to intrusion detection to enhance the detection ability of mutated and unknown intrusions [7,8]. Immune algorithms, detect network patterns and respond with specific antibodies. They adapt to mutated or unidentified threats by generating more general antibodies and improving their specificity over time. These algorithms remember past intrusions, distinguish normal from abnormal behaviour, and dynamically adjust to evolving threats. Collaboration among systems enhances their effectiveness, making them part of a layered defence strategy. However, although the above methods can identify dynamic patterns of intrusion data and promote the rise of detection rate, many methods are incomplete and often mistakenly identify normal data as intrusions, resulting in a substantial increase in the False Alarm Rate of intrusion detection. This kind of large-area false alarm distracts the network managers' attention, making it difficult for them to judge whether intrusions are true or false, which makes it difficult for them to apply it to actual intrusion detection.

The main contribution of the proposed model is discussed as follows: This paper proposes an artificial immune detection model for network intrusion data using a quantitative matching method. The model uses network data and decimal values to define features and simulate artificial immune mechanisms. It trains and evolves immune elements using mathematical methods, increasing the diversity of immune recognition and enabling successful detection of unknown intrusions. The model aims to accurately identify known intrusions and expand the identification of unknown ones, outperforming traditional methods.

In the following, Section 2 introduces the research background of signature-based intrusion detection methods and immune-based intrusion detection methods, for network intrusion data. Section 3 introduces the proposed artificial immune detection model, which is the key part of this paper. Section 4 introduces the experiment and analysis. And Section 5 concludes this paper.

2 Research Background

Important intrusion detection methods for network data include signature-based intrusion detection methods, immune-based intrusion detection methods, etc. Signature-based intrusion detection

methods play a practical role in the application of intrusion detection. Immune-based intrusion detection methods play a promoting role in the research of intrusion detection theory. They have their advantages in improving the performance of intrusion detection. In the following, this paper introduces the research progress of signature-based intrusion detection methods and immune-based intrusion detection methods.

2.1 Signature-Based Intrusion Detection Methods

Signature-based intrusion detection method uses the classic intrusion feature information to construct the intrusion feature knowledge library. During detecting intrusions, it extracts the feature information of the data to be detected and uses a certain matching algorithm to compare it with the information in the intrusion feature knowledge library, to judge whether the data to be detected is an intrusion.

A rule-based intrusion detection method is proposed based on the principle of state transition analysis [9]. This method analyzes a series of state changes in the intrusion process from the initial security state to the successful state and constructs the state transition diagram of the intrusion process. It used the state transition graph to construct the detection rules. Based on these rules, an expert system that is called the state transition analysis tool (USTAT) in UNIX is designed and implemented.

A new method was proposed to describe the intrusion features [10]. This method uses regular expressions and logical operators to analyze intrusion features and uses high-level syntactic structure with general feature information to represent intrusion features.

An automatic attack feature extraction method based on multi-sequence combination, which uses a clustering method to classify network data stream sequences [11]. They presented two algorithms which are Continuous-Matches Encouraging Needleman-Wunsch and Hierarchical Multi-Sequence Alignment to match the network data stream sequences, to automatically extract attack features and generate corresponding intrusion detection rules.

Snort, a typical signature-based network intrusion detection system (NIDS) was developed by [11,12]. The software uses rules as the characteristics of network intrusion, which are composed of protocol type, direction operator, IP address, subnet mask, port number, IP address, keywords of package content, etc. The software can alarm, record, ignore, discard and link the network data packets that meet the corresponding rules. The rules defined by Snort have clear structure and clear meaning. After professionals analyze the intrusion characteristics, they can quickly and easily construct rules to detect known network intrusions. Snort is an open-source software, which is favored by many intrusion detection technology developers and researchers. Many people carry out secondary development and experimental verification based on Snort.

2.2 Immune-Based Intrusion Detection Methods

In recent years, many researchers have improved artificial immune algorithms, or combined immune algorithms with other learning algorithms, continuously improving the performance of immune-based intrusion detection [13–15]. It can be seen from the above literature that the immune-based intrusion detection method is an effective intelligent method to improve intrusion detection performance and ability [16].

A general framework of intrusion detection system is proposed based on an immune agent, which designs immune agents that can flow among network nodes [17]. Each agent can identify each other's activities and take appropriate actions according to potential security policies. The agent

designed in this paper can dynamically learn and adapt to the network environment and can detect known and unknown intrusions. On this basis, the author developed a multi-agent intrusion detection system based on immunity, which can monitor the computer activities of the user layer, system layer, processing layer and packet layer.

An artificial immune model is proposed to solve the problems of traditional network intrusion detection systems [18]. The model includes three evolutionary stages: gene library evolution, negative selection, and clonal selection. It adopts a distributed strategy and uses an independent detector set in the local intrusion detection system. The model has self-organization characteristics and can perform three stages of evolution in each local intrusion detection system.

A dynamic intrusion detection model is proposed based on immunity and constructed immune sub-models such as dynamic self, antigen evolution, dynamic tolerance, and dynamic immune memory [19]. In this model, the self is defined naturally and dynamically, which solves the problem of dynamic description of self and non-self in the computer immune system. The life cycle of mature immune cells is simulated. The problem that the cost of training mature cells increases exponentially with the number of self is overcome. The generation efficiency of mature cells is improved. The simulation results show that the proposed immune-based dynamic intrusion detection model has better adaptability than the traditional immune-based intrusion detection model. Intelligent intrusion detection technology includes biological immune principles, providing a unique perspective on investigating intrusion detection systems. This study describes an artificial immune-based network intrusion detection system that has been developed using an optimization technique. The model was evaluated quantitatively and qualitatively using the KDD CUP99 Data Set [20].

To enhance immune-based anomaly detection technology, an intrusion detection approach based on variations in antibody concentration in the immune response was proposed [21]. It categorizes antibodies and antigens, simulates correlations, develops dynamic evolution models, calculates antibody concentration changes, divides risk levels, and coordinates immune responses. The experimental findings demonstrate that conventional approaches perform better in terms of detection and flexibility.

A quantitative risk assessment model was proposed for network security based on body temperature (QREM-BT), which addresses drawbacks in conventional approaches [22]. The model improves detection quality and reduces false detection rates by using the biological immune system's imbalance and fluctuations in body temperature. The model also emphasizes the process of increasing antibody concentration, which accurately represents network danger induced by diverse assaults. The simulation findings suggest that the model is more effective in real time. A Multiple-Detector Set Artificial Immune System (mAIS) is proposed for identifying mobile malware based on Android app information flows. Unlike traditional AISs, mAISs develop simultaneously through negative selection. The mAIS employs feature selection and the split detector method (SDM). It was compared to traditional AISs and mAISs [23].

2.3 Comparison of Two Methods

At present, the main method used in the actual intrusion detection systems is the signature-based intrusion detection method. According to the known intrusion feature information, this method can accurately identify the classic intrusions, and almost will not mistakenly identify the normal data features as intrusions. However, due to the extreme complexity of the Internet security situation, the intrusion methods are more changeable, resulting in feature information of intrusions is not immutable. The signature-based intrusion detection method is very sensitive to the characteristics of the intrusion information. For the current frequently changing intrusions, its detection ability is

greatly limited, resulting in the detection rate of this method cannot be greatly improved. Immune-based intrusion detection method adopts immune mechanisms, which can recognize the mutated or unknown attacks that cannot be recognized by feature recognition. It can make up for the defects of the signature-based intrusion detection method and has an important supplementary role in improving the detection rate of intrusion detection and reducing the false alarm rate. The fusion application of the above two intrusion detection methods, taking their respective advantages, has practical significance for improving the comprehensive performance of intrusion detection. The comparison of existing methods is represented in Table 1.

Table 1: Comparison of existing methods

Reference	Methodology used	Results	Limitations
[9]	This paper proposes the design and implementation of a UNIX-specific prototype of this expert system, called USTAT.	The functional capabilities and conceptual soundness of the approach to state state transition analysis are validated.	The proposed STAT does not concentrate more on adapting multiple hosts and other platforms.
[12]	Snort was designed in this paper. It was designed to fulfill the needs of prototypical lightweight network intrusion detection system.	It effectively replaces expensive commercial intrusion detection systems in cost-prohibitive locations.	The proposed snort run on the specific platform and it is very difficult to implement this design in any other platforms.
[15]	This paper introduces a novel and sophisticated hybrid approach that incorporates two defensive lines by employing the machine learning and intrusion detection mechanisms.	As additional execution cycles transpire, the effectiveness of the suggested schema in detecting intrusions appears to increase steadily.	The results of the two training stages in delineating self and non-self boundaries, similar to maneuvers, may result in false alarms if not determined accurately.
[17]	An artificial immune system (AIS) is proposed in the paper based on network intrusion detection scheme.	The complexity issues are addressed in the paper. Further, the proposed model outperforms existing models in terms of detection accuracy.	The proposed model is not cost-efficient.

(Continued)

Table 1 (continued)

Reference	Methodology used	Results	Limitations
[20]	In this paper, a network intrusion detection system is designed based on artificial immune principle of the new model.	The detector generation algorithm achieves linear time complexity, minimizing redundancy, maximizing coverage of non-self space, and demonstrating validated high efficiency in experiments.	Enhance research on communication protocol in intrusion detection models based on artificial immunity is ineffective.
[22]	Quantitative risk evaluation model for network security based on body temperature (QREM-BT) is proposed in this paper.	Assessing network security risk in real time can be more effectively and intuitively accomplished by correlating body temperature values with corresponding colors.	The results provided by the proposed model in real time is not accurate.

3 Proposed Artificial Immune Detection Model

An artificial immune detection model for network intrusion data based on the quantitative matching method (AID-QMM) is proposed in this paper. We adopt artificial immune mechanisms to detect network intrusion data and use a quantitative matching method to improve the traditional immune detection process. The key points of the proposed model include architecture, digital expression of network data, simulation of immune mechanisms, quantitative matching method, computing method of matching result, generating immature detector, training immature detector, evolution of mature detector, signature detection, and immune detection.

3.1 Architecture

The architecture of the proposed model is shown in Fig. 1, where the solid arrows represent the flow direction of network data, and the dotted arrows represent the flow direction of the immune detector. In the proposed model, the network data to be detected is preprocessed, and then the signature detection and immune detection methods are used to identify whether the data to be detected are intrusions. The ability to recognize specific patterns or characteristics for identifying threats or abnormalities is shared by immune detection and signature detection. They both rely on pattern recognition, specifically in targeting known threats, initiate responses when a match is found, and can adapt to changing patterns over time. Immune detection is a biological defense mechanism in living organisms, whereas signature detection is a technology-driven approach in cybersecurity for identifying and responding to digital threats. The known intrusions are detected by the signature detection algorithm, and the mutated and unknown intrusions are detected by the immune algorithm. After the network data to be detected is preprocessed, the network data features are formed, which are the data basis of immune detection in the proposed model. The network data features are firstly used to train the mature detectors in the immune model, and then they are sent to the signature detection for detection. The initial utilization of network data features involves a series of essential steps to prepare and use data effectively for training machine learning models. These steps include

data preprocessing, feature extraction, scaling, normalization, feature selection, data augmentation, encoding categorical data, and dimensionality reduction. After these preparations, the data features are used as input to train the machine learning or neural network model. The signature detection detects the known intrusions according to the existing intrusion signature library, and all the network data features that are not identified as intrusions are sent to the memory detectors for immune detection. In memory detection, immune system-inspired algorithms are used to develop robust and adaptive security systems. They assist in detecting abnormal memory activities, distinguishing between legitimate and malicious behaviour, and effectively responding to threats.

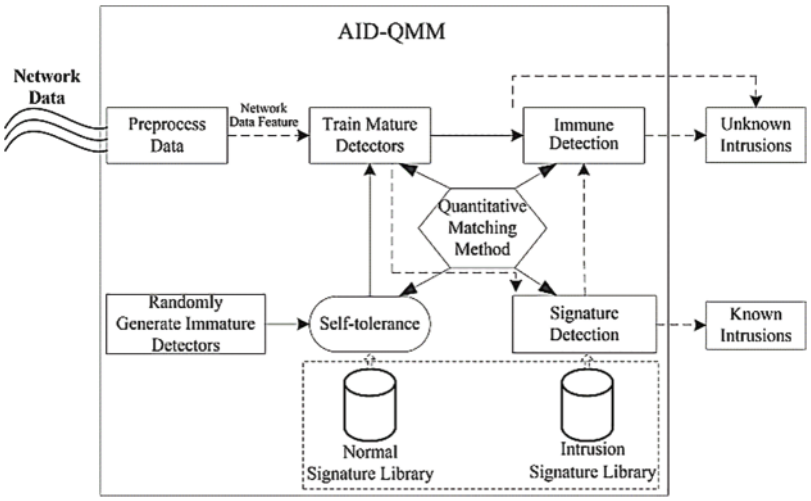

Figure 1: The architecture of the proposed model

Before immune detection, the proposed model uses the immune algorithm to generate a memory detector. Memory detectors evolve from immature detectors and mature detectors in turn. In the proposed model, immature detectors are generated randomly, and the self-tolerance algorithm is used to judge whether they can evolve into mature detectors. Self-tolerance is the immune system's capacity to detect self-produced antigens as non-threats while correctly responding to external harmful substances. This balance of immune defense and self-tolerance is essential for good physiological function and general health. The self-tolerance algorithm matches the immature detectors with signatures in the normal signature library. The matching process between immature detectors and signatures from a normal signature library requires training the immature detectors to recognize typical system behavior. This training data is used to generate normal signatures, which represent typical system operations. Detectors continuously monitor the system during the detection phase for deviations from these patterns, comparing observed behavior to stored normal signatures. If a significant mismatch is detected, an alert is generated, indicating a potential anomaly or security threat. Only unmatched immature detectors can evolve into mature detectors. Mature detectors are trained by network data features. If one of them recognizes a network data feature, its affinity will increase. The immune system's continuous generation of immature detectors is influenced by genetic rearrangements, cellular diversity, and self-tolerance mechanisms. These detectors are diverse and maintain a broad spectrum of specificities. Exposure to antigens during development or in the environment can shape the maturation process. Also, the immune responses expand the selection of detectors, and some B cells can undergo receptor editing to enhance specificity. Immature detectors can mature when exposed to antigens, generating a feedback loop. This continuous generation is crucial

for a highly adaptable and diverse immune system. Only when it reaches enough affinity within a set period, it can evolve into a memory detector. Memory detectors have the characteristic information of recognizing intrusions. Memory detectors can carry out immune detection for the current input network data features.

3.2 Digital Expression of Network Data

In the proposed model, network data is not limited to network data traffic, log data, behaviour data of system terminal, etc., but can be any other forms of data to be detected. Before the detection, as long as the appropriate method is used to preprocess the detection data and form the decimal form of network data features, all forms of data can be detected by the proposed model. Let the network data set be

$$ND = \{< \zeta_1, \zeta_2, \ldots, \zeta_n > | \zeta_i \in \text{decimal value}, n \text{ is a natural number}\},$$

$\zeta_i\,(1 \leq i \leq n)$ is the decimal value of the i^{th} item of the data to be detected, which meets $\zeta_i \in [Min_i, Max_i]$, where, Min_i and Max_i represent the minimum and maximum values of the i^{th} itemized data feature, respectively, n represents the number of itemized features contained in a network data feature.

It is necessary to compare various aspects to identify similarities between the itemized data features of two network datasets. This includes entity attributes, connection details, timestamps, categorical and numerical data, textual information, geospatial data, network structure, event descriptions, and graph metrics. A comparison of this type assists in understanding the similarities and differences between the datasets, which can be useful for a variety of analytical purposes.

After conversion through the appropriate algorithm, multiple itemized data features in decimal form are formed. A group of itemized data features represents a data unit to be detected, and the itemized features can be determined according to the specific form of the data to be detected. Intrusion detection relies on collecting itemized data features from network traffic and system logs. This process involves data collection, preprocessing, feature engineering, data representation, model building, training, testing, and real-time monitoring. The effectiveness of this process depends on the quality of data and the model's ability to distinguish between normal and suspicious activity.

Data sources commonly used in intrusion detection include network packets, network connections, system logs, etc. Each kind of data source has its data format and itemized data characteristics. For the data to be detected in the form of network packets, the itemized features of intrusion detection data can be decimal value (Domain of definition is [0, 255]) composed of four segments of IP address, TTL value, protocol type number value, port number value, packet length value, etc. These itemized data features are decimal data, which can be directly used in the intrusion detection of the proposed model. For the data to be detected in the form of a network connection, the itemized characteristics of intrusion detection data can be the connection duration, the number value of protocol type, the network service type of the target host, the number of bytes of data from the source host to the target host, the number of error segments, the number of urgent packets, etc. For the data to be detected in the form of a system log, the characteristics of intrusion detection data can be occurrence time, event source, event ID, event level, task type, execution user, etc.

For example, network data of network traffic is represented as independent network packets. After capturing network packets, to express the features of network traffic, the key characteristics of the packet header of network packets are extracted to construct the features of network traffic. In the network packet header, there are the following characteristics: IP protocol version number,

header length, service type, packet length, identification, flag bit, offset, TTL, protocol number, header checksum, source IP address, source port number, destination IP address, destination port number, option, priority, etc. To accurately express the characteristics of network traffic, this paper selects the most representative packet header characteristics as the key features of network traffic, which include packet length, offset, TTL, protocol number, source IP address, source port number, destination IP address, destination port number and priority.

The element of network data of network traffic is defined as

$$NT = \{nt \,|nt \in \langle length, offset, ttl, protocol, srcIP, srcPort, desIP, desPort, priority\rangle\},$$

where,

length means the packet length $length \in [0, 65535]$;

offset means the offset of the segment, which indicates the place of the current packet at the original IP message $offset \in [0, 8191]$;

ttl means the life cycle TTL (time to live) of the packet $ttl \in [0, 255]$;

protocol means the number of protocol $protocol \in [0, 255]$;

srcIP means source IP address, *srcIP* = <*srcIPSection1*, *srcIPSection2*, *srcIPSection3*, *srcIPSection4*>,

where,

srcIPSection1, *srcIPSection2*, *srcIPSection3*, *srcIPSection4* $\in [0, 255]$;

srcPort means source port $srcPort \in [0, 65535]$;

desIP means destination address, *desIP* = <*desIPSection1*, *desIPSection2*, *desIPSection3*, *desIPSection4*>, where, *desIPSection1*, *desIPSection2*, *desIPSection3*, *desIPSection4* $\in [0, 255]$;

desPort means destination port $desPort \in [0, 65535]$;

priority means priority level $priority \in [0, 255]$.

In the proposed model, the appropriate data preprocessing method is used to transform the data from various data sources to form multiple decimal data features. Data preprocessing is an important step in preparing data from various sources for use by the model. The specific preprocessing methods used are determined by the nature of the data and the model's objectives. In general, data preprocessing consists of steps such as data cleaning, feature selection/extraction, normalization, categorical data encoding, and feature engineering. Data balancing, dimension reduction, integration, and splitting and balancing of the data are also significant factors. The certain validation of these steps ensures that the model performs optimally without introducing biases or issues. Itemized data features can be defined numerically according to the actual meaning of the data to be detected. A group of itemized data features represents a unit of the data to be detected and constitutes an intrusion detection data feature.

3.3 Simulation of Immune Mechanisms

In the immune system, to realize immune mechanisms, it is necessary to simulate the immune elements of specific detection operations. The immune elements simulated in the proposed model include self, antigen, immature detector, mature detector and memory detector, which are defined below.

Let Self be

$S = \{< \zeta_1, \zeta_2, \ldots, \zeta_n > | \zeta_i \in \text{decimal value}, S \subset ND\}$, which has the same meaning as network data features, and is used to train the immature detectors in the proposed model.

The self-set comes from the data pattern of classic normal network data features, which can be static or dynamic. The static self-set does not change after initialization, and the original network data features are always used. The dynamic self-set changes when the proposed model works. It can add new data patterns and delete data patterns that are no longer suitable for the current intrusion detection environment.

Let Antigen be

$A = \{< \zeta_1, \zeta_2, \ldots, \zeta_n > | \zeta_i \in \text{decimal value}, A \subset ND\}$, which has the same meaning as network data, and it has not been judged as normal data or intrusion data.

On the one hand, antigen is used to train mature detectors, on the other hand, it receives the detection of signature detection and immune detection. Once it is recognized by an intrusion feature or memory detector, it will be judged as an intrusion. The antigen is transformed from the data to be detected, and it can be the feature of intrusion detection data directly. In the specific algorithm implementation, it is necessary to adopt the algorithm suitable for the specific data to be detected, preprocess the detected data, and convert it into the data features in decimal value form on the premise of expressing the original meaning.

Let the immature detector be

$$F_I = \{< f, life > | f \in ND, life \in \text{natural number}, life \leq l_I\},$$

where,

f represents the data feature of the immature detector,

life represents the lifetime of the immature detector, and

l_I represents the lifetime of the immature detector.

In the implementation of the algorithm, the life cycle *life* and the survival lifetime threshold l_I of the immature detector can be the generation number of running cycles in the evolution process, and can also be the amount of physical time to survive. If the immature detector does not match any elements in the self-set Self in one cycle, the life cycle *life* is accumulated by 1. In its survival lifetime threshold l_I, if the immature detector does not match any self elements, it is said that the immature detector has passed the self-tolerance, and it evolves into a mature detector according to the evolution rules of the immune mechanism. If the self-set Self is static and the self elements do not change during the immune evolution, the value of l_I is 1, otherwise, it can be set to a value greater than 1 according to the situation. In the survival lifetime threshold l_I of the immature detector, if it matches any self elements, the immature detector will stop evolving, and it will be deleted in the implementation of the specific algorithm.

Let Mature Detector be

$$F_M = \{< f, life, affinity > | f \in ND, life, affinity \in \text{natural number}, life$$
$$life \leq l_M, affinity < \alpha\},$$

where,

f represents the data feature of the mature detector,

life represents the life cycle of the mature detector,

l_M represents the survival lifetime threshold of the mature detector, and

affinity represents the affinity of the mature detector, that is, the number of antigens matched by the mature detector, and

α represents the promotion threshold of mature detector.

Similar to the immature detector, in the specific algorithm implementation, the life cycle *life* and survival lifetime threshold l_M of the mature detector can be the generation number of running cycles in the evolution process, and can also be the physical time of survival. In the survival lifetime threshold l_M of the mature detector, *affinity* accumulates to 1 for every antigen matching. When the affinity reaches the upgrading threshold α, the mature detector will be upgraded to a memory detector according to the immune evolution mechanism. If it exceeds the life cycle l_M and its affinity has not reached the upgrading threshold α, the mature detector will be eliminated, and it will be deleted in the specific algorithm implementation.

Let Memory Detector be

$$F_R = \{< f, count > | f \in ND, count \in \text{natural number}\},$$

where,

f is the data feature of the memory detector, and

the *count* is the number of network data features recognized by the memory detector.

A memory detector is a learning result trained according to the immune mechanism, and its data signature is not the same as an intrusion data pattern. In the evolution process of detectors, the detectors have experienced the training of self-elements and a large number of antigens, so that a memory detector can recognize a certain range of intrusions. It can not only recognize a specific intrusion but also recognize a series or a kind of intrusions with similar features. In the specific algorithm implementation, we can determine how to use a memory detector to identify a certain range of intrusion data patterns according to the defined matching algorithm (See Sections 3.4 and 3.5).

3.4 Quantitative Matching Method

Network data features (See Section 3.2 for definition) are internal data patterns expressed by various immune elements. All five immune elements defined in Section 3.3 contain network data features. The core task of recognizing intrusions is to judge the similarity between the features of network data. If the similarity value between them reaches a certain degree, it means that the immune elements match each other.

A quantitative matching method is proposed in this paper, which is shown in Eq. (1). The proposed mathematical matching method expresses the similarity between immune elements by computing the comprehensive similarity and importance of itemized data features between two network data. Integrating artificial immunity mechanisms with a quantitative matching approach improves network intrusion detection by allowing network traffic to be measured and compared to predefined baselines. This technique has various advantages, including better anomaly detection, improved signature-based detection, self-learning capabilities, accurate reaction mechanisms, real-time monitoring, and flexibility to emerging threats.

$$M(nd_1, nd_2) = \sum_{i=1}^{n} \omega_i \frac{|nd_1 \cdot \zeta_i - nd_2 \cdot \zeta_i|}{nd_1 \cdot \zeta_i + nd_2 \cdot \zeta_i}, nd_1, nd_2 \in ND, 1 \le i \le n \tag{1}$$

where,

the data features nd_1 and nd_2 are two network data elements.

ω_i represents the weight value of each network data feature.

n is the number of itemized data features.

The mathematical method of similarity is used in the proposed model to compute the similarity between the corresponding itemized data features of two network data. The similarity between the itemized data features adopts the relative distance, and its range is [0, 1]. When the distance value is 0, it means that the two sub-itemized features are absolutely the same. When the distance value is 1, it means that the features of the two network data are completely different. Eq. (1) summarizes and computes the similarity between all the features of the network data, and the range of the similarity value is [0, n]. This similarity computation method fully considers the uniqueness of each feature of the network data and avoids the ambiguity of the general computation of the whole network data features. It can get the actual similarity value between the two network data features more accurately.

For example, when the network data are represented as network traffic, the above quantitative matching method is described as the following.

Let any two network traffic data be

$$nt_1, nt_2 \in NT.$$

The symbol "." is defined to be used for the parent domain to refer to the child domain. For example, $nt_1.length$ represents the value of the packet length of network traffic data $nt_1 nt_1.srcIP.srcIPSection_1$, and represents the value of the first segment of the source IP address $srcIP$ of network traffic data nt_1.

Let one of the 9 values of the feature similarity be M_i, where $i \in [1, 9]$. The 9 features of each network traffic data have different value ranges. For the feature of IP address type, its digital value also has segmentation characteristics. Therefore, this paper defines an independent matching method for each feature. The segment matching method of these 9 features is as follows.

The matching method of packet length is shown in Eq. (2).

$$M_1 = \frac{|nt_1.length - nt_2.length|}{nt_1.length + nt_2.length} \tag{2}$$

The matching method of offset is shown in Eq. (3).

$$M_2 = \frac{|nt_1.offset - nt_2.offset|}{nt_1.offset + nt_2.offset} \tag{3}$$

The matching method of ttl is shown in Eq. (4).

$$M_3 = \frac{|nt_1.ttl - nt_2.ttl|}{nt_1.ttl + nt_2.ttl} \tag{4}$$

The matching method of the protocol number is shown in Eq. (5).

$$M_4 = \frac{|nt_1.protocol - nt_2.protocol|}{nt_1.protocol + nt_2.protocol} \tag{5}$$

The matching method of the source IP address is shown in Eq. (6).

$$M_5 = \sum_{i=1}^{4} \alpha_i \frac{|nt_1.srcIP.srcIPSection_i - nt_2.srcIP.srcIPSection_i|}{nt_1.srcIP.srcIPSection_i + nt_2.srcIP.srcIPSection_i} \tag{6}$$

where,

$i \in [1,4]\,\alpha_i$ means the weight value of each section of the source IP address, which satisfies $\sum_{i=1}^{4}\alpha_i = 1$. In this paper, the 4 weight values of the source IP address are defined as the following: $\alpha_1 = 0.3,\ \ \alpha_2 = 0.3,\ \ \alpha_3 = 0.25,\ \ \alpha_4 = 0.15$.

The matching method of the source port is shown in Eq. (7).

$$M_6 = \frac{|nt_1.srcPort - nt_2.srcPort|}{nt_1.srcPort + nt_2.srcPort} \tag{7}$$

The matching method of the destination IP address is shown in Eq. (8).

$$M_7 = \sum_{i=1}^{4}\beta_i \frac{|nt_1.desIP.desIPSection_i - nt_2.desIP.desIPSection_i|}{nt_1.desIP.desIPSection_i + nt_2.desIP.desIPSection_i} \tag{8}$$

where, $i \in [1,4]\,\beta_i$ means the weight value of each section of the destination IP address, which satisfies $\sum_{i=1}^{4}\beta_i = 1$, In this paper, the 4 weight values of the destination IP address are defined as the following: $\beta_1 = 0.3,\ \beta_2 = 0.3,\ \beta_3 = 0.25,\ \beta_4 = 0.15$.

The matching method of the destination port is shown in Eq. (9).

$$M_8 = \frac{|nt_1.desPort - nt_2.desPort|}{nt_1.desPort + nt_2.desPort} \tag{9}$$

The matching method of priority is shown in Eq. (10).

$$M_9 = \frac{|nt_1.priority - nt_2.priority|}{nt_1.priority + nt_2.priority} \tag{10}$$

The feature weight of network traffic is defined as ω. Each one of the nine network traffic features selected in this paper has a weight value, and the total weight value of network traffic satisfies Eq. (11).

$$\sum_{i=1}^{9}\omega_i = 1 \tag{11}$$

where, $\omega_i \in [0,1]\,i \in [1,9]\,\omega_1, \ldots, \omega_9$ represent the weight values of the nine network traffic features.

In the expression of actual network traffic, network traffic features which include packet length, offset, TTL, protocol number, source IP address, source port number, destination IP address, destination port number and priority have their importance. To express their practical importance, the values of ω_i ($i \in [1,9]$) in this paper are defined in Table 2.

Table 2: Feature weight values

Feature weight name	Feature weight value	Meaning
ω_1	0.15	Feature weight value of packet length
ω_2	0.05	The feature weight value of the offset
ω_3	0.07	Feature weight value of TTL
ω_4	0.1	Feature weight value of protocol number

(Continued)

Table 2 (continued)

Feature weight name	Feature weight value	Meaning
ω_5	0.2	Feature weight value of source IP address
ω_6	0.1	Feature weight value of source port
ω_7	0.2	Feature weight value of destination IP address
ω_8	0.1	Feature weight value of destination port
ω_9	0.03	Feature weight value of priority level

3.5 Computing Method of Matching Result

The computing method of matching the result is shown in Eq. (12).

$$f\left(nd_1, nd_2, \tau\right) = \begin{cases} true, & if \quad m\left(nd_1, nd_2\right) \leq \tau \\ false, & if \quad m\left(nd_1, nd_2\right) > \tau \end{cases} \tag{12}$$

where $0 \leq \tau \leq n$ the meaning of n is the same as Eq. (1). The values of *true* and *false* indicate that the two network data features match and do not match, respectively.

Various immune elements express different meanings. So, the proposed model uses different matching threshold τ s to judge whether they match each other, and the following matching thresholds are designed in the proposed model.

Matching threshold τ_T: used to train immature detectors.

Matching threshold τ_M: used to train mature detectors.

Matching threshold τ_S: used for signature detection.

Matching threshold τ_R: used for immune detection.

3.6 Generate Immature Detectors

The immature detector is the starting point of the evolution of immune elements. In continuous immune detection, new immature detectors are constantly generated to increase the diversity of immune detection and to identify intrusions with new data patterns. It is the key of the proposed model to identify mutation or even unknown intrusions.

In the proposed model, immature detectors are generated randomly, but the data features of the immature detectors must conform to the definition of network data features. The network data features contain n itemized features, and the value range of the i-th itemized feature is $[Min_i, Max_i]$. Therefore, when generating immature detectors, each itemized feature must be randomly generated independently, and the decimal value of the i-th randomly generated itemized feature must also be in the value range $[Min_i, Max_i]$. According to the above mechanism, the process of generating an immature detector is shown in Eq. (13).

$$F_{I\_new} = \{\langle f, life\rangle \,|life = 0, Min_i \leq f.\pi_i \leq Max_i, 1 \leq i \leq n\} \tag{13}$$

where,

F_{I_new} is the newly generated immature detector set, which meets $F_{I\_new} \subset F_I$.

n is the number of itemized data features of immature detector data features.

The generation principle of immature detectors is shown in Fig. 2. In the range of $[Min_i, Max_i]$ $(1 \leq i \leq n)$, the proposed model generates n itemized feature values independently: $\zeta_1\ \zeta_2\ \zeta_i\ \zeta_n, \ldots$ and then combines these itemized feature values to form the data feature ζ of a new immature detector. At the same time, a life cycle is assigned to 0 to form a completely new immature detector im_j (j is a natural number).

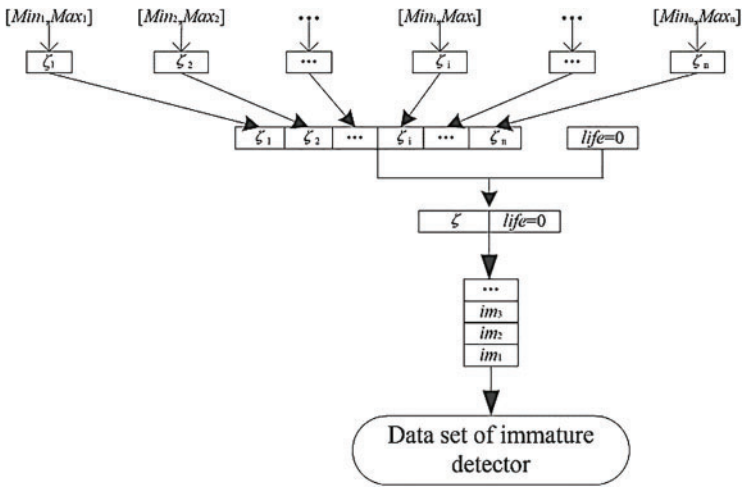

Figure 2: The generation process of immature detectors

3.7 Training Immature Detector

In the immune mechanisms, to prevent the newly generated immune cells from recognizing the normal biological cells, the immune system will carry out self-tolerance training before the immune cells can be used to capture pathogens. Self-tolerance training is a crucial part of the immune system, instructing immune cells to differentiate between the body's cells and foreign pathogens. It prevents autoimmune diseases by preventing immune cells from targeting the body's tissues incorrectly. It also protects healthy tissues and organs from immune cell attacks. Self-tolerance training enhances immune response efficiency by directing resources toward fighting pathogens rather than self-antigens. It minimizes chronic inflammation and tissue damage caused by an overzealous immune response against self-antigens. Thus, self-tolerance training is a vital component of the immune system. If it is found that the newly generated immune cells recognize the self cells, the immune cells cannot be used. In the proposed model, the newly generated immature detectors may recognize the elements in the self-data set Self, that is, they may judge the normal network data features as intrusions. For these immature detectors, the proposed model will stop their evolution.

In the proposed model, the normal data patterns are used to construct the self-set, and the data in the self-set is used to train the immature detectors. If an immature detector does not recognize any elements in the self-set in a certain time range, the immature detector will evolve into a mature detector. If it recognizes any elements in the self-set, it will stop evolving. The training method is shown in Eq. (14).

$$t\,(im) =! f\,(im.f, s, \tau_T), im \in F_I \wedge im.life \leq l_I, \forall s \in S \tag{14}$$

where the symbol "!" represents non-operation. If the immature detector im recognizes any self elements, the function $t()$ returns *false*, otherwise, it returns *true*.

The principle of training immature detectors is shown in Fig. 3. In the survival lifetime threshold, l_I of the immature detector, each immature detector im_j (j is a natural number) needs to match the elements in the self-set. As long as any self element is identified, it indicates that the immature detector has the characteristics of identifying normal data patterns, it can not be used for continuous evolution, to avoid false positives, and the immature detector needs to be deleted. In the actual operation, if the self-set is static, the survival lifetime threshold l_I of the immature detector is 1. Under these circumstances, we only need to match the immature detector with the elements in the self-set once to judge whether the immature detector can evolve into a mature detector. If the self-set is dynamic, that is, the self elements will be updated with the change of evolution generation or time, the immature detector needs to compare with the elements of the self-set many times in all stages of the whole life cycle, and no self elements are identified in the whole life cycle, then the immature detector can evolve into a mature detector.

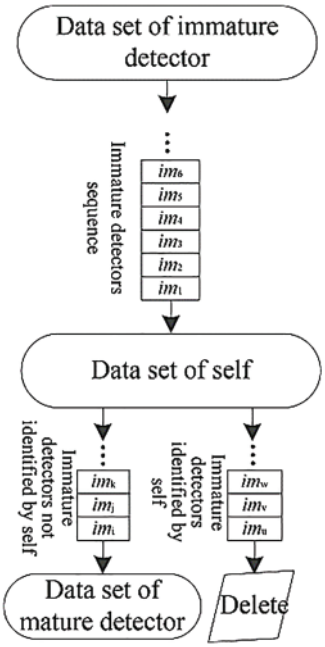

Figure 3: The training process of immature detectors

3.8 Evolution of Mature Detector

Mature detectors are in the middle stage of immune evolution. They evolve from immature detectors which pass self-tolerance.

After immature detectors are trained by all elements in the self-set, qualified immature detectors will be transferred into mature detectors which are shown in Eq. (15).

$$F_D = F_{D0} \cup \{\langle f, life, affinity\rangle \mid f = im.f \wedge t(im) = true \wedge im \in F_I, life = 0, affinity = 0\} \tag{15}$$

where,

$F_{D0} \subset F_D$, it is the original data set of mature detectors.

im is a qualified immature detector, whose data feature is given to the new mature detector.

At the same time, life cycle and affinity are both set as 0, which means the new mature detector is in the initial state.

The qualified mature detectors which recognize enough antigens will evolve into new memory detectors which are shown in Eq. (16).

$$F_R = F_{R0} \cup \{\langle f, count\rangle \,|f = m.f \wedge m.life \leq l_M \wedge m.affinity \geq \alpha \wedge m \in F_D, count = 0\} \tag{16}$$

where,

$F_{R0} \subset F_R$ it is the original data set of memory detesub-R.

m is a qualified immature detector, whose affinity reaches α its life cycle l_M.

A new memory detector inherits the data feature of the qualified immature detector, and its initial *count* is set as 0.

3.9 Signature Detection

The process of recognizing antigens in the proposed model is as follows: mature detectors detect all antigens, the intrusion signature library detects all antigens, and memory detectors detect antigens not recognized by intrusion signatures. Effective immune system detectors become activated and change into memory cells when they come into contact with antigens. These memory cells store the antigen's information, allowing the immune system to respond more quickly and effectively whenever it interacts with the same pathogen. This process is essential for long-term immunity and serves as the foundation for vaccination. To improve the accuracy of intrusion detection, after the antigens to be detected are input into the proposed model, the intrusion signatures in the intrusion signature library are used to identify the antigens. Improving an intrusion detection system's (IDS) accuracy is critical for effective cybersecurity. It improves security by detecting and preventing malicious activity, lowering the risk of data breaches, and minimizing disruptive false alarms. An accurate IDS simplifies resource allocation, allowing for faster responses to security incidents, assists in compliance adherence, and protects sensitive data. It adapts to evolving threats, ensuring business continuity, and can lead to cost savings by optimizing resource allocation and reducing the financial impact of security breaches. This identification has high accuracy and can accurately determine whether an antigen is an intrusion. High accuracy in signature-based intrusion detection is based on factors such as a robust and up-to-date signature database, specific and well-crafted signatures, regular updates, and the use of heuristic and anomaly detection. Improved accuracy is a result of context and correlation, network segmentation, tuning, machine learning, and human analysis. While signature-based systems excel at detecting known threats, they may struggle to detect new or evasive attacks, necessitating a multifaceted security approach. The antigens that are not judged as the type of intrusions by the intrusion signatures are sent to the memory detectors for detection. However, whether an antigen is recognized as an intrusion by the intrusion signatures or not, it needs to be used to train the mature detectors.

As shown in Fig. 1, the proposed model constructs the intrusion signature library data set Signature, which is used to store data patterns of classic intrusion signatures. According to the description of the network data feature, the intrusion signature library data set Signature is defined, as shown in Eq. (17).

$$Signature = \{< f, type, count > |f \in F, type \in \text{character string}, count \in \text{natural number}\} \tag{17}$$

where,

f is the data feature of an intrusion signature in the intrusion signature library.

It has the same itemized data features as the network data feature. *type* is the type of intrusion, which can be expressed by the name of the intrusion. *count* is the number of intrusion antigens recognized by the intrusion signature.

According to the quantitative matching method proposed in Section 3.4, the intrusion signatures are compared with the network data to accurately identify intrusions. Let the network data set recognized in the signature detection stage be $Intrusion_{bySig}$, which are intrusion antigens. The process of identifying the intrusion antigens by using the intrusion signature library is shown in Eq. (18).

$$Anomaly_{bySig} = \{a \,|a \in A, \exists sig \in Signature, f(sig.f, a, \tau_S) = true\} \tag{18}$$

where A represents the antigen set input into the proposed immune detection model.

3.10 Immune Detection

After signature detection, antigens which are not recognized by the intrusion signature library are detected by memory detectors. Let this part of antigens be A_{left}, which meets $A_{left} = A - Anomaly_{bySig}$. The network data which are recognized by memory detectors are intrusion antigens, too. Let this part of network data be $Intrusion_{byImmune}$, which is shown in Eq. (19).

$$Anomaly_{by\text{Immune}} = \left\{a \,\middle|\, a \in A_{left}, \exists r \in F_R, f(r.f, a, \tau_R) = true\right\} \tag{19}$$

where, a is one of the remaining antigens, which is recognized by one of the memory detectors.

4 Experiment and Analysis

The prototype software was developed to run the proposed immune detection model for network data intrusion. It adopts the appropriate data structure to simulate the immune elements in the artificial immune algorithm and realizes the algorithms of the quantitative matching method, immune evolution, signature detection, immune detection, etc. Through repeated experiments, the experimental results of the proposed immune detection model were collected to verify the effectiveness of the proposed model. At the same time, the experimental results of using signature detection and immune detection were compared and analyzed, and the intrusion detection abilities were compared with other intrusion detection methods.

4.1 Experimental Environment

The prototype software ran on a computer server, which is a 2U rack server with a dual power supply and dual Gigabit network adapter. The hardware environment of the simulation experiments is: (1) CPU: Intel Xeon processor E5-2609, (2) Memory: 32 GB DDR3, (3) Storage: three pieces of 300G SAS hard disk. The software environment of the simulation experiments is (1) Operating system: Windows Server 2003 Professional Edition, (2) Programming language: Java, (3) Java running environment: JRE 1.8.0 under Windows environment, (4) Java programming and compiling environment: Eclipse Neon.3 (4.6.3).

4.2 Experimental Data Set

To verify the proposed immune detection model, we adopt a public network data set KDD CUP 1999 Data. KDD CUP is an annual competition organized by SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) of ACM (Association for Computing Machinery). KDD Cup 1999 Data is the data set adopted by The Third International Knowledge Discovery and Data Mining Tools Competition, jointly held by KDD-99 [24]. This data set comes from the data

set DARPAR'98 [25]. Sal Stolfo of Columbia University and Wenke Lee of North Carolina State University used data mining and other technologies to analyze the features and preprocess the data of DARPAR'98, and formed about 5 million network connection records composed of pure text data. Each network connection record is composed of 41 features. Every record in the KDD Cup 1999 data set shows a "good" or "bad" network connection. "Good" means that the network connection corresponding to the record is normal, and "bad" means that the network connection corresponding to the record is network intrusion. This data set contains multiple data subsets. Some data subsets of this data set do not label the classification attributes of network connections, in which each record contains 41 features, which is mainly used to test the performance of intrusion detection technology. Some data subsets annotate the classification attributes of network connections. For this kind of data subset, each record contains 42 features, including one feature of network connection classification and 41 features of actual network connection information.

We downloaded the KDD Cup 1999 data set for these experiments. The compressed package file of the data set is provided in the link. The main compressed package files and their corresponding data set files contained in the data set and their meanings are shown in Table 3.

Table 3: Files' information of data set KDD CUP 1999 data

ID	Compressed package name	Name of sub data set	Meaning
1	kddcup.data.gz	kddcup. data.corrected	Revised complete data set
2	kddcup.data_10_percent.gz	kddcup.data_10_percent_ corrected	10% of the revised training set
3	kddcup.newtestdata_10 _percent_unlabeled.gz	kddcup.newtestdata_10_ percent_unlabeled	A subset of 10% of the dataset without network connection types labelled
4	kddcup.testdata.unlabeled.gz	kddcup. test data.unlabeled	Test data set without network connection types
5	kddcup.testdata.unlabeled_10 _percent.gz	kddcup. test data.unlabeled_10_percent	10% of test data set without network connection types
6	corrected.gz	corrected	Test data set marked with network connection types after revision

The simulation experiments used the first file kddcup.data.corrected and the second file kddcup.data_10_percent_corrected. The former was used to test the proposed model, which is called a test data set. The latter was used for signature recognition and training the proposed model, which is called a training data set. The two dataset files are described in detail below.

4.2.1 Sub Data Set kddcup. data.corrected

This sub-data set file contains all the network connection records and is a complete data set. The data in this file was used as the test data set in the proposed model. In this dataset, the statistical information of network connection classifications is shown in Table 3. There are 4,898,431 records in this file. Each record contains the label of network connection classification attribute, including 1 type of normal network connection records and 22 types of network intrusion records. The label name of normal network connection records is normal, and the label name of network connection

records of network intrusion type is the name of the intrusion, as shown in Table 3. The number of normal network connections is 972,781, and that of network intrusion type is 3,925,650. The number of network connections corresponding to each network intrusion is shown in Table 4.

Table 4: Statistical table of general data set in classifications

ID	Class name of network connection	Number of network connections	Intrusion or not
1	normal	972781	No
2	buffer_overflow	30	Yes
3	load-module	9	Yes
4	perl	3	Yes
5	Neptune	1072017	Yes
6	smurf	2807886	Yes
7	guess_passwd	53	Yes
8	pod	264	Yes
9	teardrop	979	Yes
10	port sweep	10413	Yes
11	ipsweep	12481	Yes
12	land	21	Yes
13	ftp_write	8	Yes
14	back	2203	Yes
15	imap	12	Yes
16	satan	15892	Yes
17	phf	4	Yes
18	nmap	2316	Yes
19	multihop	7	Yes
20	warezmaster	20	Yes
21	warezclient	1020	Yes
22	spy	2	Yes
23	rootkit	10	Yes

4.2.2 Sub Data Set kddcup.data_10_percent_corrected

This sub-data set file contains a 10% data subset of all network connection records. The data in the file is used for signature detection and training of immune detection elements. This file contains 494,021 records. Each record contains the label of network connection classification attribute, including 1 type of normal network connection records and 22 types of network intrusion records. The number of normal network connections is 97,278, and the number of network intrusion types is 396,743. The statistical information of network connection classifications is shown in Table 5.

Table 5: Statistical table of training data set in classifications

ID	Class name of network connection	Number of network connections	Intrusion or not
1	normal	97278	No
2	buffer_overflow	30	Yes
3	load-module	9	Yes
4	perl	3	Yes
5	Neptune	107201	Yes
6	smurf	280790	Yes
7	guess_passwd	53	Yes
8	pod	264	Yes
9	teardrop	979	Yes
10	port sweep	1040	Yes
11	ipsweep	1247	Yes
12	land	21	Yes
13	ftp_write	8	Yes
14	back	2203	Yes
15	imap	12	Yes
16	satan	1589	Yes
17	phf	4	Yes
18	nmap	231	Yes
19	multihop	7	Yes
20	warezmaster	20	Yes
21	warezclient	1020	Yes
22	spy	2	Yes
23	rootkit	10	Yes

The network connection records contained in this file are all from the data in the overall dataset file kddcup.data.corrected. If the number of certain types of network connections is not less than 10,000, 10% of them will be selected and put into the training set file. If it is less than 10,000, all the data will be selected and put into the training set file. In the proposed model, the normal signature library is used to train the self elements in immune evolution, and the intrusion signature library is used for signature detection, as shown in Fig. 1. To apply the data in the file kddcup.data_10_percent_corrected to the proposed model, this experiment will use the normal network connection records of the file to simulate the normal signature library and use the network connection records of intrusion types of the file to simulate the intrusion signature library. Normal signature libraries define expected network and system behaviour, whereas intrusion signature libraries identify specific patterns associated with known malicious activities. Normal signatures are used for anomaly detection, whereas intrusion signatures are used for detecting and preventing known threats. Normal libraries change less frequently, whereas intrusion libraries require frequent updates to remain effective against evolving threats.

4.3 Experimental Parameters

The parameter information used in the experiments is shown in Table 6, and the parameter values are from the optimized values obtained after repeated experiments.

Table 6: Experiment parameters

ID	Parameter name	Parameter value
1	Number of antigens input in each generation	100
2	Iterations	48985
3	Matching threshold for training immature detectors τ_T	31
4	Matching threshold for training mature detectors τ_M	31
5	Matching threshold of signature-based recognition τ_S	0
6	Matching threshold of immune recognition τ_R	31
7	Number of initial immature detectors	1000
8	The number of immature detectors randomly generated in each generation	50
9	Immature detectors' survival lifetime threshold l_I	1
10	Mature detectors' survival lifetime threshold l_M	15
11	Mature detectors' upgrading threshold α	100

The experiments set two important parameters to verify the effectiveness and performance of intrusion detection, which are Detection Rate and False Alarm rate, where,

$$\text{Detection Rate} = \frac{\text{Number of correctly identified anomalies}}{\text{Number of all anomalies}},$$

$$\text{False Alarm Rate} = \frac{\text{Number of normal network connections indentified as anomalies}}{\text{Number of all normal network connections}}.$$

4.4 Detection Result

In the experiments, the data of the results of signature detection, immune detection and overall recognition are collected. These data not only include the data of detection effectiveness and performance but also include the data of immune elements in the operating process of the proposed model. The data of these results are analyzed respectively in the following.

4.4.1 Results of Signature Detection

The number of intrusions detected in the process of signature detection is shown in Table 7.

It can be seen from Table 7 that the number of intrusion network connections correctly identified by the signature detection algorithm each time is 3,560,817, and the number of normal network connections mistakenly identified each time is 13. The detection results can verify that the signature detection algorithm implemented by the proposed model is stable, and the correct detection rate is high, which can reach 0.90706, while the false alarm rate is only 0.00001336. It also shows that most of the network connections of intrusion type in the test set can be accurately identified by using the

intrusion signatures in the training set, but the percentage of intrusions not identified is still 0.09294, which needs to be detected by immune detection.

Table 7: Detection results of signature detection

Experiment ID	Number of intrusions identified correctly	Number of intrusions misidentified
1	3560817	13
2	3560817	13
3	3560817	13
4	3560817	13
5	3560817	13
6	3560817	13
7	3560817	13
8	3560817	13
9	3560817	13
10	3560817	13

4.4.2 Results of Immune Detection

In the experiments, through immune detection, the immune mechanisms were used to detect the network connections of intrusion type that cannot be identified by signature detection, which greatly improves the accuracy of the overall detection. Intrusion detection systems that use signatures only recognize particular patterns and match them to observable events. However, the approach can identify all known assaults since the patterns are different for detecting new attacks and the process is insufficient. The results of immune detection are shown in Table 8. As can be seen from Table 8, for the overall test data set, the overall immune detection can improve the detection rate by about 0.09, while the false positive rate remains below 0.0003.

Table 8: Detection results of immune detection

Experiment ID	Number of intrusions identified correctly	Detection rate	Number of intrusions misidentified	False positive rate
1	350523	0.08929	241	0.000248
2	350977	0.089406	235	0.000242
3	345546	0.088023	198	0.000204
4	351827	0.089623	279	0.000287
5	349784	0.089102	236	0.000243
6	348079	0.088668	270	0.020198
7	344669	0.087799	285	0.000293
8	346651	0.088304	208	0.000214

(Continued)

Table 8 (continued)

Experiment ID	Number of intrusions identified correctly	Detection rate	Number of intrusions misidentified	False positive rate
9	352221	0.089723	236	0.000243
10	353014	0.089925	249	0.000256

4.4.3 Results of Overall Detection

The overall detection includes signature detection and immune detection. The detection results reflect the real performance indicators of the immune detection model proposed in this paper. The overall detection results are analyzed from two aspects overall detection results and detection results in classification.

1. Overall Detection Results

The overall detection results are shown in Table 9.

Table 9: Overall detection results

Experiment ID	Number of intrusions identified correctly	Detection rate	Number of intrusions misidentified	False positive rate
1	3911594	0.996354744	254	0.000261107
2	3912042	0.996470393	248	0.000254939
3	3906574	0.995086928	211	0.000216904
4	3912936	0.996686918	292	0.00030017
5	3910850	0.996166495	249	0.000255967
6	3909179	0.995732172	283	0.000290919
7	3905784	0.994863526	298	0.000306338
8	3907689	0.99536841	221	0.000227184
9	3913287	0.996787284	249	0.000255967
10	3914093	0.996989288	262	0.000269331

Table 9 shows the number of intrusions identified correctly, the detection rate, the number of intrusions misidentified and the false positive rate in 10 experiments. It can be seen from Table 10 that the detection rate of each experiment is above 0.994, and the experiment with the highest detection rate is No. 10, reaching 0.996989288. Each experiment can detect enough network connections of intrusion type, and its false positive rate is also very low, which has practical application value for overcoming the defects of traditional intrusion detection systems. The average detection rate of 10 experiments is 0.996051, and the average false positive rate is 0.000264, which proves that the overall detection ability of the proposed model is very high, and it can be used in actual intrusion detection applications.

Table 10: Detection results in classifications

ID	Intrusion name	Detection rate of signature detection	Detection rate of immune detection	Overall detection rate
1	buffer_overflow	1	0	1
2	load-module	1	0	1
3	perl	1	0	1
4	Neptune	0.677415	0.319126	0.996541
5	smurf	0.998599	0.0000258	0.998625
6	guess_passwd	1	0	1
7	pod	1	0	1
8	teardrop	1	0	1
9	port sweep	0.657159	0.259781	0.91694
10	ipsweep	0.621585	0.013973	0.635558
11	land	1	0	1
12	ftp_write	1	0	1
13	back	1	0	1
14	imap	1	0	1
15	satan	0.662031	0.254417	0.916448
16	phf	1	0	1
17	nmap	0.387306	0.11671	0.504016
18	multihop	1	0	1
19	warezmaster	1	0	1
20	warezclient	1	0	1
21	spy	1	0	1
22	rootkit	1	0	1

2. Detection Results in Classification

The experiments classified the recognized intrusions according to the intrusion type label in the test data set, and counted the detection rate of various intrusion types, as shown in Table 10. The detection rate in Table 10 is the average of 10 experiments.

It can be seen from Table 10 that 16 intrusion types can be completely detected through signature detection, including buffer_overflow, loadmodule, perl, guess_passwd, pod, teardrop, land, ftp_write, back, imap, phf, multihop, warezmaster, warezclient, spy and rootkit. Most intrusions of smurf types can be recognized. For Neptune, portsweep, ipswep, satan and nmap, the detection rate of signature detection is not high, and immune detection plays a key role, which helps the proposed model greatly improve the overall detection rate.

4.5 Comparative Analysis of Data

To improve the comprehensive performance of intrusion detection theory and algorithm based on artificial immune, researchers introduced other methods to immune mechanisms and proposed a series of intrusion detection models and algorithms based on improved immune methods, to make up for the

shortcomings of traditional immune algorithms. In the aspect of simulation experiments, this paper all use the KDDCUP'99 data set to verify the immune detection performance, and they all have a unified public data set that can be referenced [26,27]. Therefore, it is comparable and scientific to compare the proposed model (AID-QMM) with the above research results. In the following, comparative data between this paper and the above literature are analyzed to verify the advantages of AID-QMM in intrusion detection performance.

The comparative data of AID-QMM and existing immune models and algorithms in intrusion detection performance are shown in Fig. 4 and Table 11.

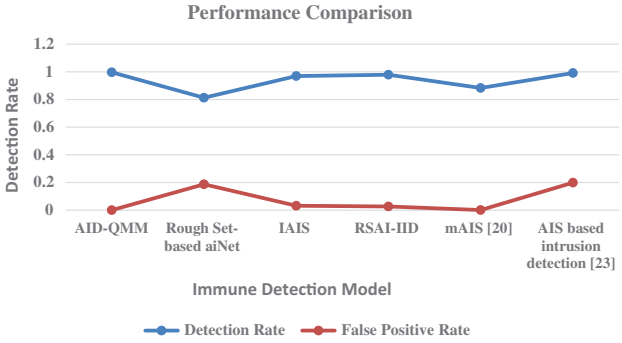

Figure 4: Performance comparison between AID-QMM and existing models

Table 11: Performance comparison between AID-QMM and existing models

Experiment ID	Immune detection model	Detection rate	False positive rate
1	AID-QMM	0.996051	0.000264
2	Rough Set-based aiNet	0.8127	0.1873
3	IAIS	0.9691	0.0321
4	RSAI-IID	0.9786	0.0268
5	mAIS [20]	0.8833	0.00123
6	AIS-based intrusion detection [23]	0.991	0.1987

Reference also uses a rough set to improve the performance of the immune algorithm, the detection rate and false positive rate are 0.8127 and 0.1873, respectively. Reference proposed an immune system IAIS for intrusion detection. The system uses a dendritic cell algorithm and a negative selection algorithm to improve the performance of the immune algorithm in intrusion detection. The dendritic cell algorithm (DCA) is a population-based algorithm inspired by natural DCs in the innate immune system for anomaly identification. These DCs aggregate molecular information and interpret it for T-cells in the adaptive immune system, activating appropriate immunological responses. DCs operate as detectors for numerous body locations and mediators for initiating diverse immunological responses, serving as the body's first line of defence against intruders. The detection rate of the system is 0.9691 and the false positive rate is 0.0321. Rough set method and integrating misuse detection and anomaly detection, an integrated intrusion detection (RSAI-IID) model proposed in reference is an integrated intrusion detection model based on rough set and artificial immune. It uses a rough set to improve the

training method of the vaccine in the immune algorithm, to quickly obtain an effective vaccine. The detection rate of the model is 0.9786, and the false positive rate is 0.0268.

Although the comprehensive performance of the above three kinds of literature is very good, AID-QMM proposed in this paper is superior to them in terms of detection rate and false positive rate. The average detection rate of AID-QMM is above 0.996, and the average false positive rate is below 0.000265. The reason why the proposed model can achieve such high performance of intrusion detection is the following. On the one hand, this paper uses digital expression to represent the data to be tested numerically and uses the quantitative matching method to carry out the numerical matching based on itemized features, which improves the accuracy of feature recognition. On the other hand, this paper uses an immune detection algorithm to detect intrusions, which expands the scope of identifying mutated and unknown intrusions.

5 Conclusions

This paper presents a network intrusion detection model that utilizes the quantitative matching method to optimize the immune detection method.

The proposed quantitative matching method computes the comprehensive similarity and importance of itemized data features between two network data. It can improve the accuracy of similarity calculation between immune elements to advance the ability of artificial immune detection. Signature detection can recognize known intrusions and immune detection can detect unknown intrusions. The traditional signature detection method and the immune detection method both play a vital role in advancing the detection rate. The experiment results verify the detection performance of the proposed model for network data intrusion.

Acknowledgement: The author would like to express their heartfelt gratitude to the techniques that have made valuable contributions to this research.

Funding Statement: This research was funded by the Scientific Research Project of Leshan Normal University (No. 2022SSDX002); and the Scientific Plan Project of Leshan (No. 22NZD012).

Author Contributions: All authors contributed to the design and methodology of this study, the assessment of the outcomes and the writing of the manuscript.

Availability of Data and Materials: No datasets were generated or analyzed during the current study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. P. Anderson, *Computer Security Threat Monitoring and Surveillance (Technical Report)*. Fort Washington, PA, USA: James P. Anderson Company, 1980.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. 13, pp. 222–232, 1987.

[3] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," in *Proc. of 7th Usenix Security Symp.*, San Antonio, TX, USA, 1998.

[4] P. Envelope, H. Mu and G. Envelope, "DESC-IDS: Towards an efficient real-time automotive intrusion detection system based on deep evolving stream clustering," *Future Generation Computer Systems*, vol. 140, pp. 266–281, 2023.
[5] R. Makani and B. V. R. Reddy, "Trust-based-tuning of bayesian-watchdog intrusion detection for fast and improved detection of black hole attacks in mobile ad hoc networks," *International Journal of Advanced Intelligence Paradigms*, vol. 21, no. 1–2, pp. 53–71, 2022.
[6] M. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system," *Processes*, vol. 9, no. 5, pp. 834, 2021.
[7] S. P. Sithungu and E. M. Ehlers, "GAAINet: A generative adversarial artificial immune network model for intrusion detection in industrial IoT systems," *Journal of Advances in Information Technology*, vol. 5, pp. 13, 2022.
[8] Y. N. Rao and K. Suresh Babu, "An imbalanced generative adversarial network-based approach for network intrusion detection in an imbalanced dataset," *Sensors*, vol. 23, no. 1, pp. 550, 2023.
[9] K. Ilgun and R. A. Kemmerer, "State transition analysis: A rule-based intrusion detection approach," *IEEE Transactions on Software Engineering*, vol. 21, pp. 181–199, 1995.
[10] S. D. Shanklin, T. E. Bernhard and G. S. Lathem, "Intrusion detection signature analysis using regular expressions and logical operators: USA," *U.S. Patent No. 6,792,546*. Washington DC: U.S. Patent and Trademark Office, 2004.
[11] Y. Tang, X. C. Lu and H. P. Hu, "Automatic generation of attack signatures based on multi-sequence alignment," *Chinese Journal of Computers*, vol. 29, pp. 1533–1541, 2006.
[12] M. Roesch, "Snort-lightweight intrusion detection for networks," in *Proc. of LISA'99: 13th Systems Administration Conf.*, Seattle, Washington, USA, pp. 229–239, 1999.
[13] Snort. [Online]. Available: https://www.snort.org/ (accessed on 21/04/2023).
[14] Q. Yan, Y. Jiang and J. P. Wu, "Antibody generation and antigen detection component in immune-based network intrusion detection system," *Chinese Journal of Computers*, vol. 28, pp. 1061–1067, 2005.
[15] E. Farzadnia, H. Shirazi and A. Nowroozi, "A novel sophisticated hybrid method for intrusion detection using the artificial immune system," *Journal of Information Security and Applications*, vol. 58, pp. 102721, 2021.
[16] M. A. Rassam, "Maar of artificial immune network clustering approach for intrusion detection," *Journal of Advances in Information Technology*, vol. 3, pp. 147–154, 2012.
[17] D. Dasgupta, "Immunity-based intrusion detection system: A general framework," in *Proc. of the 22nd NISSC (National Information Systems Security Conf.)*, vol. 1, pp. 147–160, 1999.
[18] J. Kim and P. Bentley, "An artificial immune model for network intrusion detection," *Computer Applications in Engineering Education*, vol. 38, pp. 133–135, 2001.
[19] T. Li, "An immune-based dynamic intrusion detection model," *Chinese Science Bulletin*, vol. 50, pp. 2650–2657, 2005.
[20] Z. Yanbin, "Network intrusion detection system model based on artificial immune," *International Journal of Security and Its Applications*, vol. 9, no. 9, pp. 359–370, 2015.
[21] R. Zhang and X. Xiao, "An intrusion detection method based on changes of antibody concentration in immune response," *Journal of Information Processing Systems*, vol. 15, no. 1, pp. 137–150, 2019.
[22] Y. P. Jiang, C. C. Cao, X. Mei and H. Guo, "A quantitative risk evaluation model for network security based on body temperature," *Journal of Computer Networks and Communications*, vol. 2016, pp. 1–10, 2016.
[23] J. Brown, M. Anwar and G. Dozier, "An artificial immunity approach to malware detection in a mobile platform," *EURASIP Journal on Information Security*, vol. 2017, no. 1, pp. 1–10, 2017.
[24] KDD Cup 1999 Data. [Online]. Available: https://www.kdd.org/kdd-cup/view/kdd-cup-1999/ (accessed on 08/11/2017).
[25] MIT Lincoln Laboratory. "DARPA Intrusion Detection Evaluation Dataset," 1998. [Online]. Available: http://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset (accessed on 21/04/2023).

[26] Y. B. Chen, C. Feng and Q. Zhang, "Integrated artificial immune system for intrusion detection," *Journal on Communications*, vol. 33, pp. 125–131, 2012.
[27] L. Zhang, Z. Y. Bai and S. S. Luo, "Integrated intrusion detection model based on rough set and artificial immune," *Journal on Communications*, vol. 34, pp. 166–176, 2013.