Hindawi
Journal of Sensors
Volume 2019, Article ID 5451263, 20 pages
https://doi.org/10.1155/2019/5451263

Research Article

Intrusion Detection in Wireless Sensor Networks with an Improved NSA Based on Space Division

Ruirui Zhang[1] **and Xin Xiao**[2]

[1]*School of Business, Sichuan Agricultural University, Ya'an 625014, China*
[2]*School of Computer Science, Southwest Minzu University, Chengdu 610000, China*

Correspondence should be addressed to Ruirui Zhang; zhangruiruisw@gmail.com

Received 3 August 2018; Revised 4 January 2019; Accepted 21 February 2019; Published 24 April 2019

Academic Editor: Antonio Martinez-Olmos

Inspired by the biological immune system, many researchers apply artificial immune principles to intrusion detection in wireless sensor networks, such as negative selection algorithms, danger theory, and dendritic cell algorithms. When applying the negative selection algorithm to wireless sensor networks, the characteristics of wireless sensor networks, such as frequent changes in network topology and limited resources, are not considered too much, which makes the detection effect to need improvement. In this paper, a negative selection algorithm based on spatial partition is proposed and applied to hierarchical wireless sensor networks. The algorithm first analyzes the distribution of self-set in the real-valued space then divides the real-valued space, and several subspaces are obtained. Selves are filled into different subspaces. We implement the negative selection algorithm in the subspace. The randomly generated candidate detector only needs to be tolerated with selves in the subspace where the detector is located, not all the selves. This operation reduces the time cost of distance calculation. In the detection process of detectors, the antigen which is to be detected only needs to match the mature detectors in the subspace where the antigen is located, rather than all the detectors. This operation speeds up the antigen detection process. Theoretical analysis and experimental results show that the algorithm has better time efficiency and quality of detectors, saves sensor node resources and reduces the energy consumption, and is an effective algorithm for wireless sensor network intrusion detection.

1. Introduction

With the progress and development of wireless communication, the microcomputer electrical system, microelectronics, signal processing, computer network and other technologies, and wireless sensor networks (WSN) with intelligent characteristics emerge [1]. At present, wireless sensor networks have been widely used in aerospace, defense military, environmental monitoring, medical and health care, industrial facilities, and other fields [2–5]. Due to the small size, monitoring distribution, resource-constrained sensor nodes, dynamically changeable routing, and no gateways or switches monitoring the information flow, how to effectively solve the problem of the safety of sensor networks is the key of wireless sensor network applications. Now, researches on sensor network security mainly focus on the following aspects: (1) various attack models and defense strategies, (2) encryption algorithms, (3) key managements, and (4) network security architecture and intrusion detection and response models [3]. Different security defense and detection methods can be used in different network layers, and they complement each other. This paper mainly studies the intrusion detection system (IDS) in wireless sensor networks.

The main challenges of wireless sensor network intrusion detection are as follows.

(1) Attack form is varied, and the means and characteristics of attacks in wireless sensor networks have a bigger difference with traditional computer networks, such as most of attacks in the link layer and network layer which are peculiar to wireless sensor networks [2]. Traditional computer network

resources such as networks, files, system logs, and processes cannot be used in wireless sensor networks, and we need to consider the feature information which can be applied to the wireless sensor network intrusion detection

(2) There are many new attacks in wireless sensor networks, which are different from traditional networks [3, 6]. How to improve the ability of the intrusion detection system to detect unknown attacks and select appropriate algorithms is a problem to be solved. Some algorithms are suitable for detecting known attacks, while others are suitable for detecting unknown attacks. Some algorithms are suitable for a flat surface network structure, and some algorithms are suitable for a hierarchical network structure. We should select or design the appropriate algorithm according to requirements of the network

(3) Wireless sensor networks have limited resources, including storage space, computing power, bandwidth, and energy [2–5]. Limited storage space means that it is impossible to store large amounts of system logs on sensor nodes. The intrusion detection system based on knowledge is required to store large amounts of defined intrusion patterns. The system detects intrusion by means of pattern matching, and invasion behavior characteristics need to be stored in libraries. With the increase in invasion types, the scale of feature library will also increase. Limited computing power means that the node is not suitable for running the intrusion detection algorithm which requires a lot of computation. The current wireless sensor network adopts the communication technology of low speed and low power consumption, and the node has the characteristic of having limited energy. It is requested that intrusion detection systems cannot bring too much communication overhead, which is less considered in the traditional computer network

Inspired by a negative selection algorithm in the biological immune system, this paper proposes a wireless sensor network intrusion detection model based on the spatial division negative selection algorithm (SD-RNSA). The main contributions of this model are as follows. (1) The algorithm analyzes the distribution of self-set in the real-valued space and divides the real-valued space. (2) The negative selection algorithm is implemented in the subspace, which reduces the tolerance range of the candidate detector and saves resources of sensor nodes. (3) In the detection process of detectors, the antigen to be detected only needs to match with mature detectors in the subspace where the antigen is located, thus accelerating the detection process. In this paper, the performance of the model is analyzed in theory; experimental results show that the model has better time efficiency and detector quality, saves sensor node resources, reduces the energy consumption, and is an effective algorithm for wireless sensor network intrusion detection.

The remainder of this paper is organized as follows. Section 2 introduces the current research of intrusion detection in wireless sensor networks. Section 3 discusses preliminaries and shortcomings of NSA. Section 4 describes the algorithm, including the algorithm flow, space division approach, detector coverage of non-self-space, and complexity analysis. Experiments are performed in Section 5, including Iris data set and applications in WSN. In Section 6, some conclusions are added.

2. Related Work

In wireless sensor networks, there are several methods which are in the following for the research of intrusion detection [2, 3].

2.1. Research on Specification-Based Detection Technology. The technology first defines several specifications and then monitors activities in the network according to those specifications. Any deviation from the normal behavior is abnormal. These normal behaviors are defined manually, not by machine learning or training data. Therefore, the definition and update of specifications will be a time-consuming task. Bhuse et al. [7] proposed an intrusion detection system for detecting witch attacks. Two techniques are used in the testing process. One is mutual protection. When the sensor node receives the packet, the source ID of the packet is checked. The second is SRP, which verifies the number of packets received and sent by a node. Simulation results show that the mutual protection technology is expensive and cannot be applied to the case where the communication range of attackers is small. da Silva et al. [8] proposed a distributed IDS based on predefined rules. The system is divided into three stages: data acquisition, rule application, and intrusion detection. Some monitoring nodes are set up in the system. They are in hybrid mode, collecting packets, and performing intrusion detection tasks by detecting the characteristics of known attacks. The distribution of monitoring nodes and the selection of rules affect the performance of the algorithm, and resource consumption of monitoring nodes is large. Singh et al. [9] proposed a new method for wireless sensor network intrusion detection based on clusters. In the proposed method, an effective MAC address-based intruder tracking system is developed for early intrusion detection and prevention. Jadidoleslamy [10] proposed a complete intrusion detection architecture with combinatorial hierarchy. The architecture designs the intrusion detection system based on sensors and deploys it on sensor nodes, designs the intrusion detection system based on cluster heads and deploys it on cluster heads, and designs the intrusion detection system of the network level and deploys it on a central server. All three types of intrusion detection systems use a specification-based approach to distinguish exceptions.

2.2. Research on Misuse Detection Technology. The misuse detection system detects known attacks based on the intrusion mode. According to the propagation characteristics of wireless communication and the assumption that nodes are densely distributed, gatekeeper technology can be used in

wireless sensor networks. Roman et al. designed a framework for applying IDS to wireless sensor networks [11]. This architecture consists of local and global agents. The local agent exists on each node and checks the local data flow. The global agent exists on some nodes of the network and checks the data flow of the neighbor nodes. In hierarchical network topology, cluster heads can be global agents. In planar topologies, the architecture uses a technique called spontaneous watchdog to select global agents to ensure that the minimum number of global agents cover the entire network. There are two drawbacks to this approach. One is that, due to the randomness of global proxy selection, it cannot guarantee that all nodes are monitored. The second is that packet collisions are not considered.

2.3. Research on Abnormal Detection Technology. Abnormal detection determines whether an intrusion occurs by learning the normal behavior of sensor nodes. Abnormal detection is easier to be used than misuse detection and specification-based detection. Therefore, most researchers use this method as the main means of intrusion detection, and the techniques are in the following.

(1) Methods Based on Statistical Models. Onat and Miri [12] proposed an intrusion detection system for resource depletion attacks. Each node monitors the average receiving rate and average arrival rate of packets from neighboring nodes to build a statistical model. Only the nearest n packets of neighbor nodes are used for statistical analysis. When the next packet conforms to the statistical model of this neighbor node, it is considered as normal. The statistical model is relatively simple; cannot detect selecting forward attack, worm hole attack, etc.; and does not analyze computational cost and resource consumption.

(2) Methods Based on Clustering Algorithm. Loo et al. [13] proposed an intrusion detection model for routing attacks. The model uses a fixed-width clustering algorithm to model normal behaviors. In the training stage, the clustering algorithm is used to generate a set of several clusters in the feature space. When the number of samples in the cluster is less than a specific threshold, it is considered as an exception. During the detection phase, each sample is compared to the cluster in the collection to determine if it is abnormal. The intrusion detection model is deployed on each sensor node, which consumes a lot of computing resources. The model proposed in literature [14] is similar to [13]. The difference lies in the input of the clustering algorithm. The model also has the problem of resource consumption. When the sensor network size is large, it is not feasible to collect and train data on each sensor node.

(3) Isolation Table. Chen et al. [15] proposed an intrusion detection model based on isolation table for three-layer WSN (base station-main cluster head-secondary cluster head). In this model, the isolation table records the exception information and is used by the detection agent to segregate suspect nodes. All cluster heads can generate isolation tables. The secondary cluster head is responsible for monitoring the main cluster head and sensor nodes, and the main cluster head is responsible for monitoring the secondary cluster head. The isolation table is eventually passed to the base station, which can distribute the isolation table to any node. Simulation results show that when there are more sensor nodes, the energy consumption of the model is also larger. In addition, the model does not consider the case of node failure or node capture. In literature [16, 17], the authors expanded their work and proposed a lightweight intrusion detection model based on ontology, but still failed to solve the problem of excessive energy consumption.

(4) Methods Based on Machine-Learning. Misra et al. [18] used the learning automata (LA) method for intrusion detection. This method is used to sample data packets in the network and determine whether the nodes are malicious based on whether the feedback from the environment is favorable or not. In literature [19], Misra et al. extended their approach and proposed a simple low-complexity energy-sensing protocol. This protocol combines the concept of random learning automata and sampling mechanism to implement an energy-sensing intrusion detection system. Rajasegarar et al. [20] used a class support vector machine (SVM) to detect network exceptions. This paper proposes two methods based on support vector machines, namely, centered hyperellipsoidal support vector machine (CESVM) and quarter-sphere support vector machine (QSSVM). CESVM has advantages in parameter selection flexibility and computational complexity. However, in distributed WSNs, it faces certain limitations because it adopts a centralized approach. QSSVM works well in a distributed environment. Gunasekaran and Periakaruppan [21] proposed an intrusion detection model based on genetic algorithms to solve sleep denial attacks in wireless sensor networks. The model implements a modified RSA (MRSA) algorithm in a base station to generate and distribute key pairs between sensor nodes. Before sending/receiving packets, sensor nodes determine the optimal routing through ad hoc on-demand distance vector routing (AODV) and then use fitness calculation to ensure the reliability of relay nodes. Cross and mutation operations detect and analyze the methods which are used by attackers. When determining the attacker nodes, base station broadcasts blocked messages to the network. Shi et al. [22] proposed a state transition model based on the continuous time Markov chain (CTMC), studied the sensor's behavior under internal attacks, and positioned attacks. The model linked the detection model for internal attack with the epidemiological model and considered the current state, viability, availability, and energy consumption of WSN to balance network features and security. The above four factors need to be quantified.

(5) Approaches Based on Game Theory. Reddy [23] proposed the zero-sum game method to detect malicious nodes in the positive data path. In this paper, the game model of the energy probability required for packet transmission is studied, and the model of using the confirmation probability of the message source to detect malicious nodes is given.

Cheng [24] proposed a differential game model between an intrusion detection system and an attacker for wireless sensor networks. The model calculates the Nash equilibrium and implements a solution that balances security mechanism overhead and risk, providing the best defense. However, the application of the model in a wireless sensor network, the energy consumption, and the communication cost are not given in the simulation experiments. When the scale of the sensor network increases, the performance of the model will decrease.

(6) Approaches Based on Artificial Immune. Drozda et al. proposed an intrusion detection model based on artificial immunity [25]. The model simulates the negative selection algorithm (NSA) and detects the intrusion by deploying detectors on sensor nodes. Sensor node resources are limited, and it is impossible to store a large number of detectors. At the same time, this model is in the hybrid mode, which can provide global knowledge, but this mode prevents the nodes from entering the sleep state and consumes a lot of energy. Liu and Yu proposed a wireless sensor network intrusion detection system based on the immune principle [26]. The system also deploys the intrusion detection module on each sensor node, simulating the principle of negation selection and cloning selection in the biological immune system. Experiments show that the system has high detection rate, but also a high false alarm rate, which reaches 92.3% in the jamming attacks. Although the false alarm rate can be reduced through artificial collaborative stimulation, it is unattended and manual collaborative stimulation is not convenient most of the time for wireless sensor networks. Fu et al. [27] proposed a wireless sensor network intrusion detection model based on danger theory. The model adopted the distributed mechanism. It is not necessary to run a complete intrusion detection system on each node. The general node just perceives danger, and the center node maintains an antibody set and receptor library for intrusion detection. The model computes statistical specific deviation on the characteristics of the data link layer and network layer, considers them as risk perception information for local nodes, and does not think of the link between the heterogeneous network and multiple features, making the risk information not comprehensive enough. Salmon et al. [28] proposed a wireless sensor network intrusion detection model inspired by danger theory. The model introduces danger theory and a customized dendritic cell (DC) algorithm. The model includes the following modules: monitoring module, intrusion detection management module, environment module, decision module, parameter library module, rule library module, and operation library module. These modules are classified as four subsystems: monitoring environment subsystem (environment module), intrusion detection subsystem (monitoring module, intrusion detection system management module, and decision module), storage subsystem (parameter database, rule database), and response subsystem (operation database). Experiments show that the model has good detection rate and low energy consumption. However, this model uses a time window to calculate the MCAV index of antigens on a regular basis. The signal setting and parameter setting are complex, and the real-time performance of intrusion detection needs to be improved. Xiao and Zhang [29] proposed a real-time distributed intrusion detection model based on the differentiation mechanism of DC cells. The model abstracts the information fusion process of DC cells, defines the meaning and function of external signals applied to wireless sensor networks, and defines the mathematical evolution model of DC cells. Finally, the performance analysis is carried out, including scalability analysis, robustness analysis, and complexity analysis. Similarly, the signal and parameter settings of this model are complex. Guo et al. [30] proposed a negative selection algorithm based on the differential evolution constraint multiobjective optimization problem and applied this algorithm to the intrusion detection system of wireless sensor networks. The algorithm first combines constraint processing and multiobjective optimization techniques to generate detector sets with maximum non-self-space coverage and minimize overlaps between detectors. Then, black holes in the non-self-space are reduced by differential evolution. In the experiments, the effectiveness of the algorithm is verified through network simulation. The communication cost, computational cost, and energy consumption of the system are not analyzed.

Through the analysis of the existing intrusion detection schemes of wireless sensor networks, it can be seen that the research on the intrusion detection technology of wireless sensor networks is not mature. A lot of detection systems are based on the traditional network transplantation of intrusion detection technologies, which consider the characteristics of wireless sensor networks not enough and to need to make improvements according to the characteristics of wireless sensor networks. In addition, although the simulation experiments were carried out, the intrusion detection system was not applied to the real wireless sensor networks. Analysis and simulation experiments are important, but the practical application is very important to verify the availability of the intrusion detection system.

3. Details of Negative Selection Algorithm

This section introduces the definitions of the negative selection algorithm and shortcomings of the negative selection algorithm.

3.1. Definitions of NSA. The negative selection algorithm first proposed by American scholar Forrest [31] is the most important anomaly detection algorithm in the field of artificial immunity. The idea of the negative selection algorithm is derived from the negative selection behavior of T lymphocytes in thymus during immune tolerance. An immune explanation for this behavior is that in the thymus tolerance issue, T lymphocytes which can identify self-antigen will be of apoptosis or inactivation, and T lymphocytes which cannot identify self-antigen after a period of tolerance will be mature and exercise their immune function in peripheral lymphoid tissue. The development of the negative selection

algorithm has greatly promoted the research and application of the artificial immune system in the field of abnormal detection. Specifically, the idea of the negative selection algorithm is often used in fault detection, virus detection, network intrusion detection, machine learning, and other directions [32]. The following is a brief introduction to the concepts of the negative selection algorithm.

The state of the system can be defined as the eigenvector $x = (x_1, x_2, \dots, x_n)$, and n is the system dimension. Convenient for processing, each feature of the vector is normalized to the real-valued interval [0, 1], and the state space of the entire system can be expressed as $\Omega = [0, 1]^n$. The system state space can be further divided into self-space *Self* and non-self-space *Nonself*. In abnormal detection, self-space *Self* is composed of the states when the system is normal, and non-self-space *Nonself* is composed of the states when the system is abnormal.

In the artificial immune system, antigens represent the whole states of the system, the self-set represents the system's own space, and the non-self-set represents the system's non-self-space, as defined below.

Definition 1. Antigen. $Ag = \{ag | ag = <x, r_s> = <x_1, x_2, \dots, x_n, r_s>, x_i \in [0, 1], 1 \le i \le n, r_s \in [0, 1]\}$ is the entire samples of the problem space. ag is an antigen in the set, including two parts, x and r_s. x represents the position of sample ag in the real-valued space, and r_s is the radius of ag, representing the changing threshold of ag. Then, ag is a hypersphere in space.

Definition 2. Self-set. $Self \subset Ag$ represents all the normal samples in the antigen set, $Self = \{ag \mid ag \subset Ag \cap ag \text{ is normal}\}$.

Definition 3. Non-self-set. $Nonself \subset Ag$ represents all the abnormal samples in the antigen set, $Nonself = \{ag \mid ag \subset Ag \cap ag \text{ is abnormal}\}$. Self and non-self have different meanings in different fields. For network intrusion detection, non-self represents network abnormality, and self represents normal network activity. For virus detection, non-self represents virus signature code, and self represents legal code.

$$Self \cap Nonself = \varnothing, Self \cup Nonself = Ag. \quad (1)$$

Definition 4. Training set. $Train \subset Self$ is a subset of selves and is priori knowledge of detection.

Definition 5. Detector set. $D = \{d \mid d = <y, r_d> = <y_1, y_2, \dots, y_n, r_d>, y_j \in [0, 1], 1 \le j \le n, r_d \in [0, 1]\}$. d is a detector in the set. The detector has the same structure as the antigen and consists of two parts, y and r_d. y represents the position of detector d, and r_d is the radius of detector d.

Definition 6. Match rules. $f(ag, d)$ denotes the affinity between antigen ag and detector d, i.e., the matching degree between data structures. In real-valued space, the affinity can be measured by calculating the Euclidean distance between two eigenvectors in the state space.

$$f(ag, d) = \sqrt{\sum_{i=1}^{n} (ag.x_i - d.y_i)^2}. \quad (2)$$

During the generation of detectors, if $f(ag, d) \le r_s + r_d$, detector d causes immune self-reaction and fails to become a mature detector. Table 1 describes the process of the negative selection algorithm.

In the detection process of detectors, if $f(ag, d) \le r_d$, antigen ag is denoted as non-self by detector d. When the detection system is working, TP is set as the correct affirmation, indicating the number of non-selves correctly identified by detectors. TN is the correct negation, indicating the number of selves correctly identified by detectors. Two kinds of errors can occur. False-positive FP occurs when a sample that was originally self is identified as a non-self. False negation FN occurs when an original non-self-sample is identified as a self. Table 2 describes the detection process.

Definition 7. Detection rate. DR is the proportion of the number of non-self-samples correctly identified by detectors to all the non-self-samples, as shown below.

$$\text{DR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (3)$$

Definition 8. False alarm rate. FAR is the proportion of the number of self-samples wrongly identified by detectors to all the self-samples, as shown below.

$$\text{FAR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \quad (4)$$

3.2. Shortcomings about NSA. The negative selection algorithm proposed by Forrest et al. [31] is based on binary representation. However, binary representation is insufficient in dealing with numerical data and multidimensional space problems. Gonzalez and Dasgupta et al. [33] proposed the real-valued negation selection algorithm (RNSA) that the detector position can evolve. Ji [34] and Ji and Dasgupta [35] proposed a real-valued negative selection algorithm with variable detector radius (V-Detector). This algorithm dynamically determines the radius of a mature detector by calculating the closest distance between the center of the candidate detector and all selves. Figure 1 shows the comparison between RNSA and V-Detector. Self-set is the first 30 elements in the Iris data set which are classified as "setosa" [36]. In order to display conveniently in 2D space, feature "sepalL" and "sepalW" of elements were taken as attributes of selves. In the figure, circles filled with blue are the self-elements, circles filled with cyan are the mature detectors, and the unfilled part is the vulnerability area.

It can be seen that compared to RNSA, in V-Detector, detectors with a large radius cover most of the non-self-space, and detectors with a small radius cover "holes," which reduces the number of detectors and the number of "holes."

Table 1: The process of the negative selection algorithm.

Input: training set *Train*, The default number of required detectors *maxNum* Output: detector set *D* Step 1. Initialize the self-training set *Train* Step 2. A candidate detector d_{new} is generated at random, and the Euclidean distance between d_{new} and all selves in the training set *Train* is calculated. If $f(ag, d_{\text{new}}) \le r_s + r_d$ exists, execute step 2. Otherwise, perform step 3. Step 3. Add d_{new} to the detector set. Step 4. If the detector set size $N_d > maxNum$, return *D* and the procedure exit. Otherwise, skip to step 2.

Table 2: The detection procedure.

Input: detector set *D*, antigen set to be detected Ag' Output: TP, FN, FP, TN Step 1. $TP = 0$, $FN = 0$, $FP = 0$, $TN = 0$. Step 2. An antigen ag is sequentially extracted from the antigen set to be detected Ag'. Step 3. Calculate the Euclidean distance between ag and all detectors in detector set *D*. When $f(ag, d) \le r_d$ exists, if the antigen ag is non-self, TP++; if ag is self, FP++. When $f(ag, d) \le r_d$ does not exist, if the antigen ag is self, TN++; if the antigen is non-self, FN++. Step 4. If all the antigens in Ag' have been detected, the program ends, output TP, FN, FP, TN. Otherwise, go to step 2.

But there are still some problems to be solved. The first is the vulnerability problem, that is, there is always some non-self-space that is not covered by detectors. The characteristics of intrusion behavior always develop towards the direction of vulnerability, which affects the detection rate. Moreover, it is very difficult to effectively cover the vulnerability in the boundary area between self and non-self, so it is difficult to determine whether it belongs to self or not. Secondly, too many candidate detectors are required to pass the tolerance to become mature. Assuming that N_s is the size of the self-training set, P' is the matching probability between any antigen and antibody, and P_f is the failure rate (i.e., the probability that any non-self cannot be matched by antibodies). The number of candidate detectors $N_c = -\ln(P_f)/(P'(1-P')^{N_s})$, and the time complexity of the algorithm is $O(N_c \bullet N_s)$ [31]. It can be seen that with the increase of the self-training set, the number of candidate detectors increases exponentially, so the storage cost and calculation cost of the algorithm are very high. The third point is that there is a large amount of redundant coverage between mature detectors, which leads to a large time cost of detection. Because of these problems, it is not realistic to apply the negative selection algorithm directly to the wireless sensor networks with limited resources.

In order to solve the above problems, researchers also proposed many improved algorithms. Literature [37] proposed a negative selection algorithm based on hierarchical clustering of self-set. By pretreatment of self-set, the coverage of detectors to non-self-space increased. In literature [38], detectors are divided into self-detector and non-self-detector, covering self-space and non-self-space, respectively. Self-detector is used to replace self-elements, thus reducing calculation costs. Literature [39] proposed a negative selection algorithm based on the genetic principle to solve the problem of spam detection. Literature [40] combines the particle swarm optimization strategy with the negative selection algorithm. Literature [41] introduced the wavelet transform into the negative selection algorithm and made the diagnosis of voltage interference in the power distribution system.

Through the analysis of the improved algorithms, they adopt the following methods. They randomly generate candidate detectors in non-self-space, then determine whether candidate detectors are effective, and finally do optimization to these detectors, which improve the non-self-space coverage. Although the quality of mature detectors is excellent, which can reduce the testing time in the test phase, aforementioned problems still exist in the detector generation phase, such as loopholes and too many candidate detectors. This makes the storage cost and time cost of these algorithms large, and it is overburden for wireless sensor nodes with limited resources.

4. The Algorithm Theory

This section describes the algorithm in detail, including the algorithm flow, space division approach, detector coverage computation approach of non-self-space, and complexity analysis.

4.1. Flow of the Algorithm. In the literature [25, 26], they directly applied the negative selection algorithm to intrusion detection in wireless sensor networks. Although certain achievements have been obtained, they did not consider drawbacks of the negative selection algorithm and communication cost, computation cost, and energy consumption in wireless sensor networks. Intrusion detection systems in wireless sensor networks should have the following characteristics [2, 5].

(1) After the intrusion detection system is introduced, the overall performance of the network should not be decreased

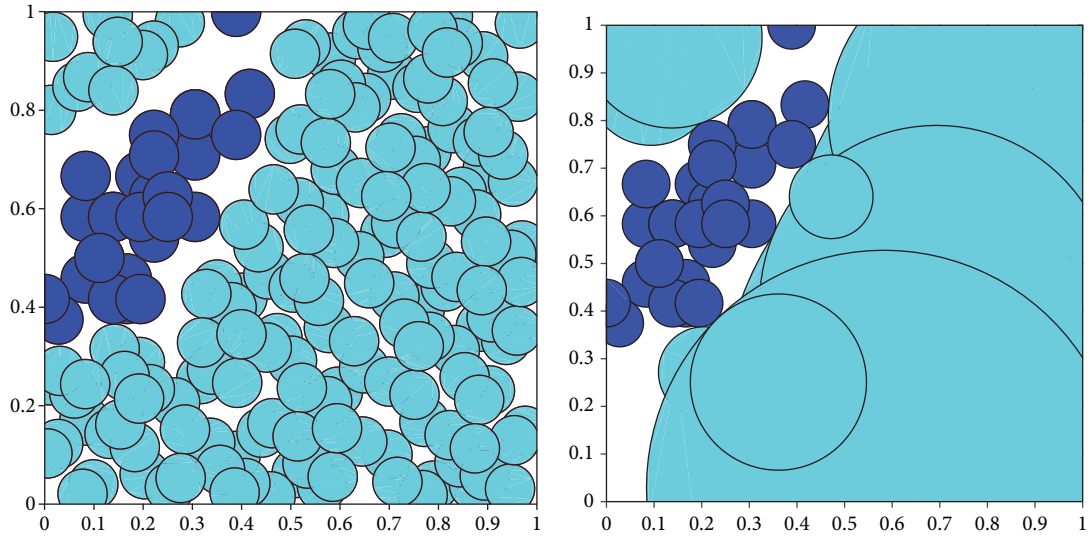
(2) After the introduction of the intrusion detection system, additional weakness should not be introduced

(3) The intrusion detection system should be transparent and sustainable for network

(4) The intrusion detection system should have a higher detection rate and lower false alarm rate

(5) The intrusion detection system should be open, cooperative, easy to deploy, and easy to integrate

The infrastructure of the wireless sensor network is divided into flat structure and hierarchical structure. In this paper, the intrusion detection system is applied to the hierarchical model of wireless sensor networks. That is, the network is divided into clusters; each cluster contains several member nodes, one of which is selected as the cluster head.



(a) RNSA (b) V-Detector

Figure 1: Comparisons of RNSA and V-Detector in generating detectors.

The member node only communicates with the cluster head of its cluster, and only the cluster head needs to communicate with the base station. This structure disperses the energy consumption on all the network nodes and reduces the communication burden. The intrusion detection system can be deployed on the cluster heads to reduce the energy consumption of general nodes and extend the network life by taking advantage of the hierarchical structure. Therefore, the local system of sensor nodes is divided into three layers: application layer, immune layer, and wireless sensor network layer, where the node selected as cluster head starts the immune layer, while the member node does not start the immune layer, as shown in Figure 2. The immune layer of the cluster head is responsible for monitoring the flow changes of member nodes in the cluster and neighboring cluster heads to intrusion detection.

In the immune layer, intrusion detection is realized by the improved negative selection algorithm (SD-RNSA). The algorithm first analyzes the distribution of self-set in the real-valued space, then divides the real-valued space and obtains several subspaces. The selves are then filled into different subspaces. We implement the negative selection algorithm in the subspaces, respectively. The randomly generated candidate detector only needs to be tolerated with selves in the subspace where the detector is located, not all the selves. This operation reduces the time cost of distance calculation. A mature detector is got when a candidate detector passes through tolerance. In the detection process of detectors, the antigen to be detected only needs to match the mature detectors in the subspace where the antigen is located, rather than all the detectors. This operation speeds up the antigen detection process. According to the main ideas of the algorithm, the detector generation process is shown in Figure 3, and the detector detection process is shown in Figure 4.

In the proposed algorithm, the key step is to divide the real-valued space. Only the real-valued space is divided correctly, the cost of distance calculation be saved in the detector generation stage and the detection stage, the resource consumption can be reduced, and the algorithm efficiency can be improved. In Section 3.2, how to partition a real-valued space in detail is introduced. The proposed algorithm uses variable-radius detectors and takes the coverage of detectors to non-self-space as the end condition of the detector generation process. The detailed steps of the detector generation process are shown in Table 3.

The Iris data set is one of the classic machine learning data sets released by the university of California Irvine and widely used in pattern recognition, data mining, exception detection, etc. [36] We choose the type "setosa" as self-set, take feature "sepalL" and "sepalW" of elements as attributes of selves, and select the top 30 elements of "setosa" as the training set. In order to display conveniently in 2D space, we take two features of records, which does not affect the comparisons. Figure 5 shows comparisons between SD-RNSA and the classic negative selection algorithms, RNSA and V-Detector. In the figures, circles filled with blue are the self-elements, circles filled with cyan are the mature detectors, and the unfilled part is the vulnerability area. RNSA generates fixed radius detectors. Figure 5(a) is a schematic diagram of three detectors generated by the three algorithms, and Figure 5(b) is a schematic diagram of detectors generated by the three algorithms for achieving 90% of the expected coverage. V-detector generates a variable radius detector by calculating the closest distance between the center of the candidate detector and the selves. SD-RNSA generates subspaces according to certain rules. Then, the randomly generated candidate detector is only tolerated with the selves in the subspace where the detector is located. The detectors generated by RNSA and V-Detector need to be resistant to all the selves. With the increase in coverage, the redundant coverage between mature detectors and the number of detectors will be excessive. In SD-RNSA, the introduction of

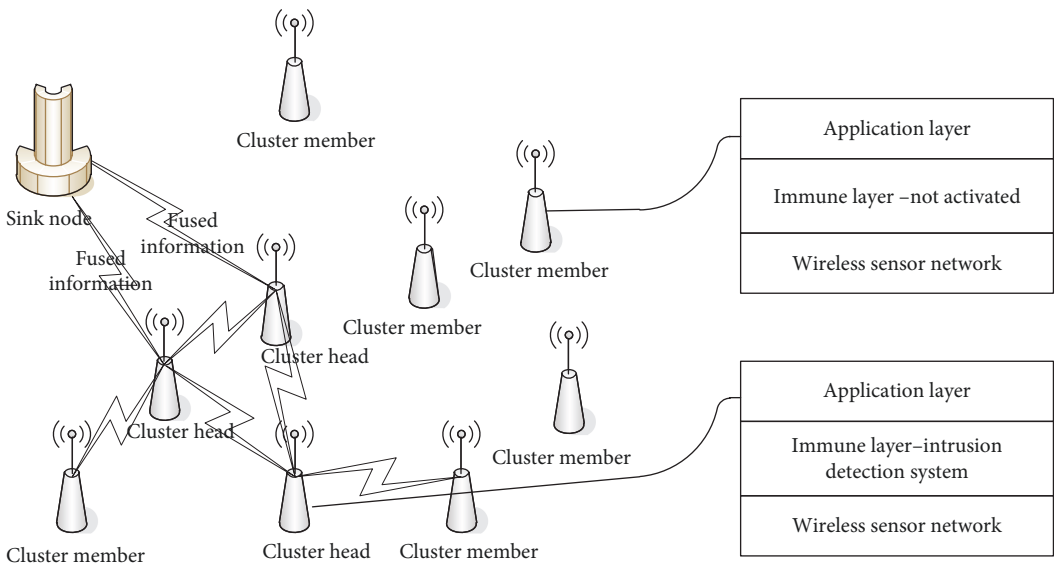


Figure 2: Schematic diagram of wireless sensor networks.

spatial division reduces unnecessary self-tolerance and the number of detectors.

4.2. Space Division. We divide the n-dimension [0, 1] real-valued space. First, determine whether there are selves in space. If there is any self in this space, the space is divided into two subspaces according to the first dimension. Then, we determine whether there are selves in the subspace. If there is any self in the subspace, the subspace continues to be divided into two subspaces according to the second dimension. Then, we determine whether the newly generated subspace has any self. If there is still self in the subspace, the space is further divided. If n dimensions are divided, we continue to divide the space by the first dimension. This process is an iterative process until the diameter of the subspace is minimized or the subspace contains no self. At this point, the spatial partitioning operation is completed.

In the process of space division, selves are filled into the corresponding subspaces. Selves in the subspace *sub* are divided into two categories, containing self and half-contained self.

Definition 9. Subspace set *SubSpaces*. All the subspaces consist of the entire space. $SubSpaces = \{sub \mid sub = <[d_i^l, d_i^h], \text{Half} - contained\ Selves, Contained\ Selves, detectors>, 1 \le i \le n, d_i^l, d_i^h \in [0, 1]\}$, where d_i^l is the lower boundary of the i^{th} dimension in the subspace *sub*, and d_i^h is the upper boundary of the i^{th} dimension in the subspace *sub*.

Definition 10. Selves completely contained in the subspace $[d_i^l, d_i^h]$ *Contained Selves* represent the total selves whose central point x is within the scope of the subspace. *Cont ained Selves* $= \{ag \mid d_i^l \le ag.x_i \le d_i^h, 1 \le i \le n\}$.

Definition 11. Selves not completely contained in the subspace $[d_i^l, d_i^h]$ *Half-contained Selves* represent the total selves

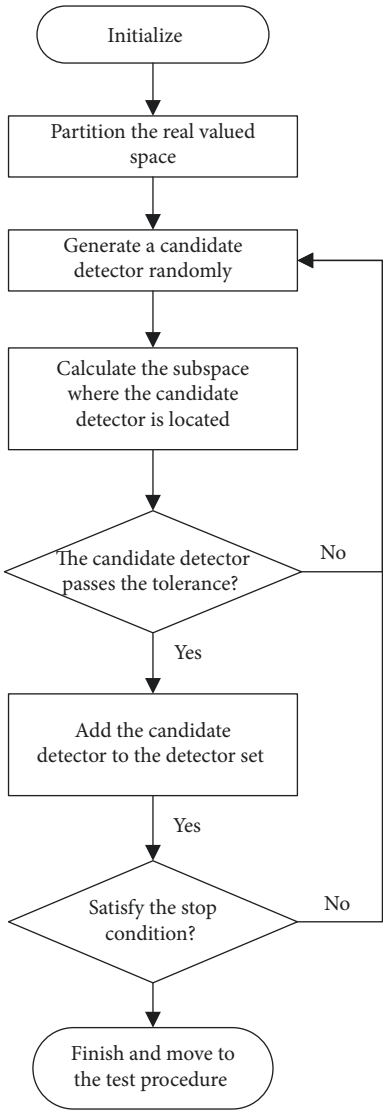


Figure 3: The generation process of detectors.

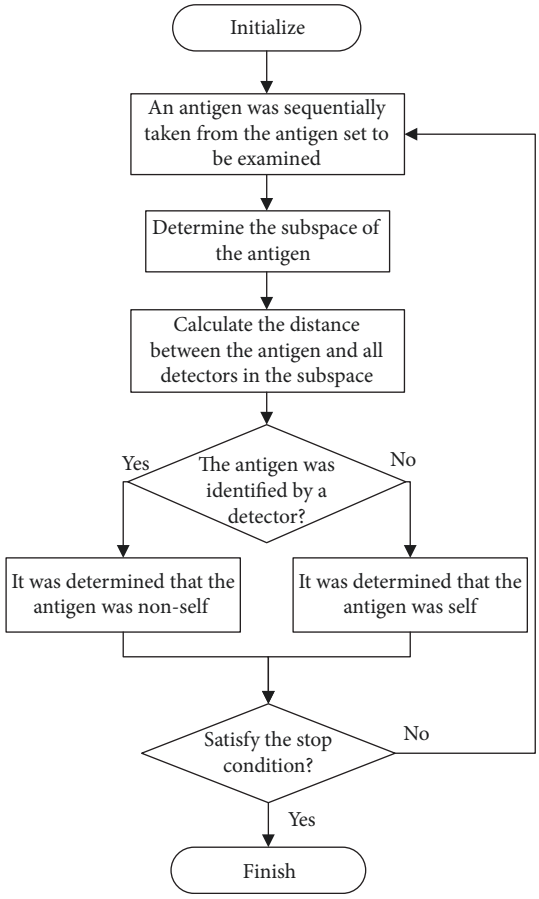


Figure 4: The detection process of detectors.

whose central point x is not within the scope of the subspace, but the changing threshold of x is within the scope. $Half - contained\ Selves = \{ag \mid \neg (d_i^l \le ag.x_i \le d_i^h) \bigwedge \exists i(|ag.x_i - d_i^l| < r_s \bigvee |ag.x_i - d_i^h| < r_s), 1 \le i \le n\}$.

We also take Iris data set as an example. Figure 6 shows the schematic diagram of complete contained selves and half-contained selves in the subspace, where the blue-filled circles are selves, the green-filled rectangle is a subspace, red-filled circles are contained selves, and cyan-filled circles are half-contained selves.

Figure 7 is a schematic diagram of the space division process. For the first time, the first dimension of space is divided, and Figure 7(b) is obtained. If there is self in the subspace and the space diameter is greater than the qualified value, the second dimension of space is divided, and Figure 7(c) is obtained. After all dimensions have been divided, continue to divide the space from the first dimension. Finally, Figure 7(d) is obtained. The resulting subspaces do not contain any self or the diameter satisfies the limit.

Tables 4 and 5 show the detailed steps of spatial partitioning.

In the process of detector generation, a candidate detector only needs to be tolerated with the contained selves and half-contained selves in the subspace where the candidate detector locates and does not need to be tolerated with selves from other subspaces. When calculating the radius of the candidate detector, there are two conditions. Figure 8 shows the radius of the candidate detector, where blue-filled circles are selves, green-filled rectangles are subspaces, red-filled circles are completely contained selves, cyan-filled circles are half-contained selves, and black-filled circles are detectors. If *sub* contains selves, as shown in Figure 8(a), the detector's position is [0.15, 0.22], belonging to the [0, 0.25, 0.25, 0.5] subspace. At this point, only the nearest distance between the center of the candidate detector and the contained selves and half-contained selves in the subspace is calculated, without considering selves in other subspaces. If *sub* does not contain any self, as shown in Figure 8(b), the position of the detector is [0.3, 0.2], belonging to the [0.25, 0.5, 0, 0.5] subspace. At this point, the farthest distance between the center of the candidate detector and the vertex of the subspace is calculated, i.e., the distance from [0.5, 0.5]. The formula is as follows.

$$r_d = \begin{cases} \min\left(f(ag,d)\right) - r_s, ag \in Half - contained\ Selves \vee ag \in Contained\ Selves \\ subspace\ contains\ selves, \\ \max\left(f(ag,d)\right), ag \in \left\{ag \mid ag.x_i = d_i^l \vee ag.x_i = d_i^h, 1 \le i \le n\right\} \\ subspace\ contains\ no\ selves. \end{cases} \tag{5}$$

Every subspace that the algorithm generates is completed. The contained selves and half-contained selves in the subspace represent the self-space, and the area not covered by selves is non-self-space. Mature detectors in the subspace cover most of the non-self-regions within the subspace and may cover areas outside the subspace. But the regions outside the subspace are outside our scope for this time and may belong to another subspace. When considering another subspace, only mature detectors in the other subspace will be considered, and detectors in this subspace will not be considered. So, we do not care if detectors in the subspace cover the area outside the subspace. In intrusion detection, we first determine the subspace where the antigen to be detected locates. For the antigen to be detected, the subspace where it is located is the problem space, including the self-set and the detector set. Conditions of other subspaces are independent of the state of this antigen.

Table 3: The detector generation process.

SD-RNSA(*Train*, $c_{\exp}$, *D*)
Input: training set *Train*, expected coverage $c_{\exp}$
Output: detector set *D*
N_0: sampling times in non-self-space, $N_0 > \max\left(5/c_{\exp}, 5/\left(1 - c_{\exp}\right)\right)$
i: number of non-self-samples
x: number of non-self-samples covered by detectors
CD: candidate detector set CD = $\left\{d \mid d = <y_1, y_2, \ldots, y_n, r_d>, y_j \in [0, 1], 1 \le j \le n, r_d \in [0, 1]\right\}$
SubSpaces: subspace set

Step 1. Initialize the self-training set, $i = 0$, $x = 0$, CD = $\varnothing$, $N_0 = \text{ceiling}\left(\max\left(5/c_{\exp}, 5/\left(1 - c_{\exp}\right)\right)\right)$.

Step 2. Invoke *GenerateSpaces*(*Train*, *SubSpaces*) to divide the space, and several subspaces *SubSpaces* are got.

Step 3. A candidate detector d_{new} is randomly generated, and find the subspace *sub* where d_{new} is located.

Step 4. The Euclidean distance between d_{new} and both contained-selves and half-contained-selves in the subspace *sub* is calculated. If d_{new} can be identified by a self $f(ag, d_{\text{new}}) - r_s \le 0$, discard it and perform step 3. Otherwise, increase *i*.

Step 5. The Euclidean distance between d_{new} and detectors in the subspace *sub* is calculated. If d_{new} is not be identified by any detector $f(d, d_{\text{new}}) - r_d > 0$, add it to the candidate detector set CD. Otherwise, increase *x* and determine whether the algorithm reaches the expected coverage $c_{\exp}$. If it is, return *D* and the procedure ends.

Step 6. Determine whether *i* reaches the sampling times N_0. if $i = N_0$, add the detector in the candidate detector set CD to the collection *D*, and reset *i*, *x*, and CD. Otherwise, perform step 3.

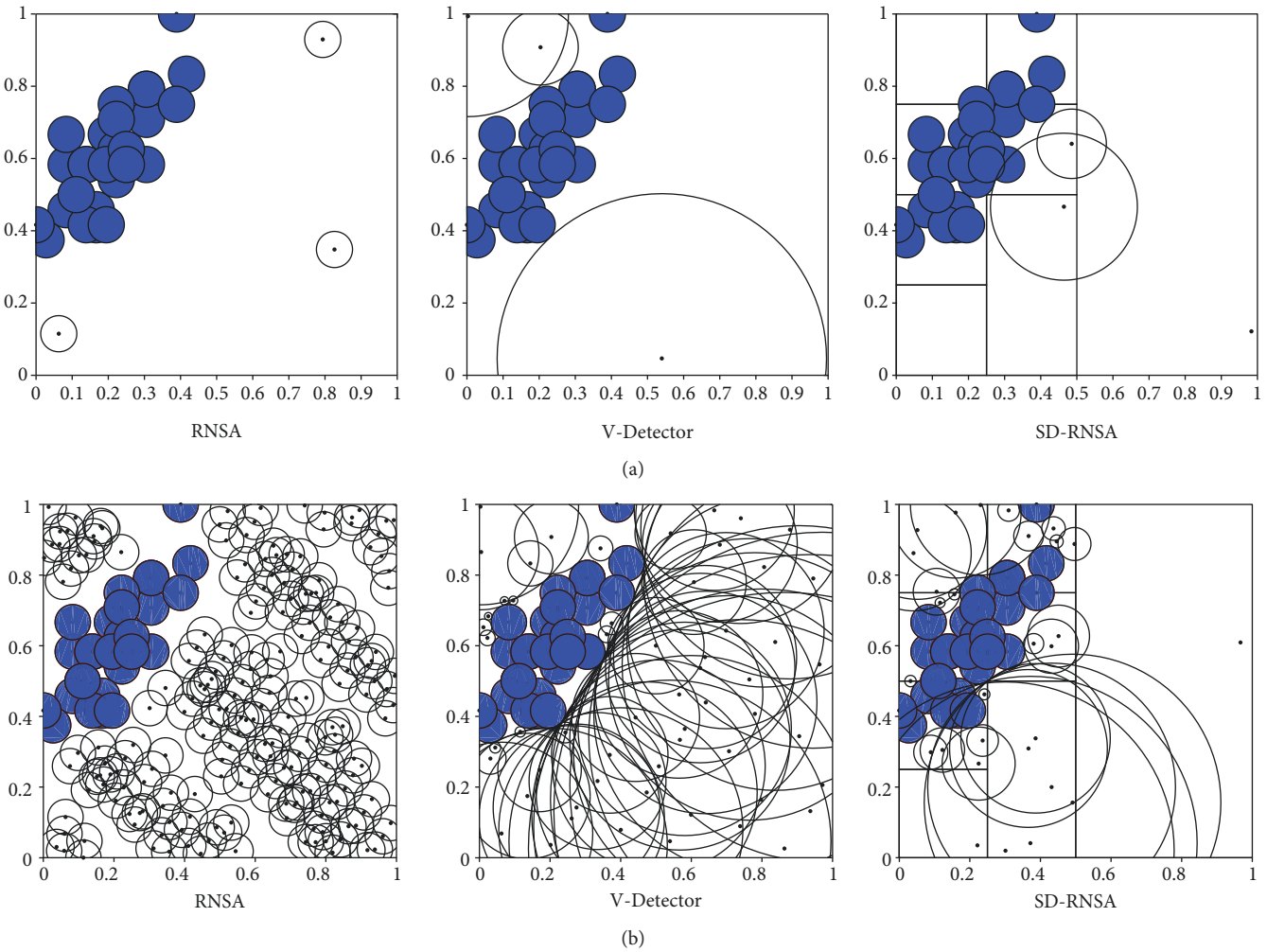
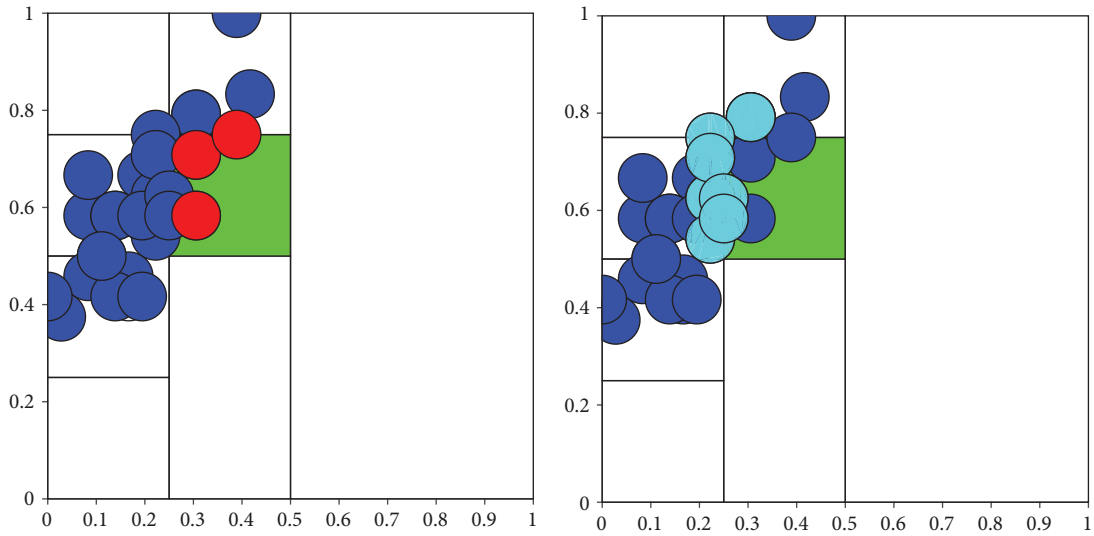


Figure 5: Comparisons between SD-RNSA, RNSA, and V-Detector.



(a) Contained selves (b) Half-contained selves

Figure 6: Contained selves and half-contained selves in the subspace.

4.3. Detector Coverage of Non-Self-Space. The algorithm takes the coverage of detectors to non-self-space as the end condition of the detector generation process. Non-self-space coverage P is equal to the ratio of the space covered by detectors V_{covered} to all non-self-space V_{nonself}, as shown in (6).

$$P = \frac{V_{\text{covered}}}{V_{\text{nonself}}} = \frac{\int_{\text{covered}} dx}{\int_{\text{nonself}} dx}. \tag{6}$$

The total space occupied by non-selves can be calculated by the total space occupied by selves, as shown in (7).

$$P = \frac{V_{\text{covered}}}{V_{\text{space}} - V_{\text{self}}} = \frac{\int_{\text{covered}} dx}{1 - \int_{\text{self}} dx}. \tag{7}$$

Redundant coverage exists between detectors and among selves. We can use the estimation method to calculate [42]. For example, the redundancy between detector i and detector j can be estimated by (8).

$$\text{overlapping}(d_i, d_j) = \begin{cases} 0 \text{ if } \left|\left|d_i.y - d_j.y\right|\right| \geq d_i.r_d + d_j.r_d, \\ \left(\exp\left(\dfrac{d_i.r_d + d_j.r_d - \left|\left|d_i.y - d_j.y\right|\right|}{d_i.r_d + d_j.r_d}\right) - 1\right)^n \text{ if } \left|\left|d_i.y - d_j.y\right|\right| < d_i.r_d + d_j.r_d. \end{cases} \tag{8}$$

So (6) can be written as

$$P = \frac{\sum_{\text{covered}} V_d^i - \sum_1 \sum_j \text{overlapping}(d_i, d_j)}{1 - \left(\sum_{\text{self}} V_{ag}^i - \sum_i \sum_j \text{overlapping}\left(ag_i, ag_j\right)\right)}. \tag{9}$$

For wireless sensor networks, the resource of nodes is limited, and the calculation amount of the above equation is very large, which is not suitable for use. Therefore, we used the method of hypothesis testing to make a statistical estimate of non-self-space coverage P [34, 35]. Assuming that N_{exp} is the theoretical value of required detectors, the maximum coverage $P_{\max}$ can be calculated by

$$P_{\max} = \frac{1 - 5}{N_{\text{exp}}}. \tag{10}$$

In order to satisfy the De Moivre-Laplace theorem, the number of sampling times in non-self-space N_0 is required to satisfy $N_0 > 5/P$ and $N_0 > 5/(1-P)$. Therefore, we chose the sample size as $N_0 = \text{ceiling}(\max(5/P, 5/(1-P)))$. x was set as the number of times of detectors overlaid in N_0 sampling in non-self space, and x should satisfy the following formulas.

$$\frac{x - N_0 P}{\sqrt{N_0 P(1-P)}} = \left(\frac{x}{\sqrt{N_0 P(1-P)}}\right) - \left(\sqrt{\frac{N_0 P}{(1-P)}}\right), \tag{11}$$

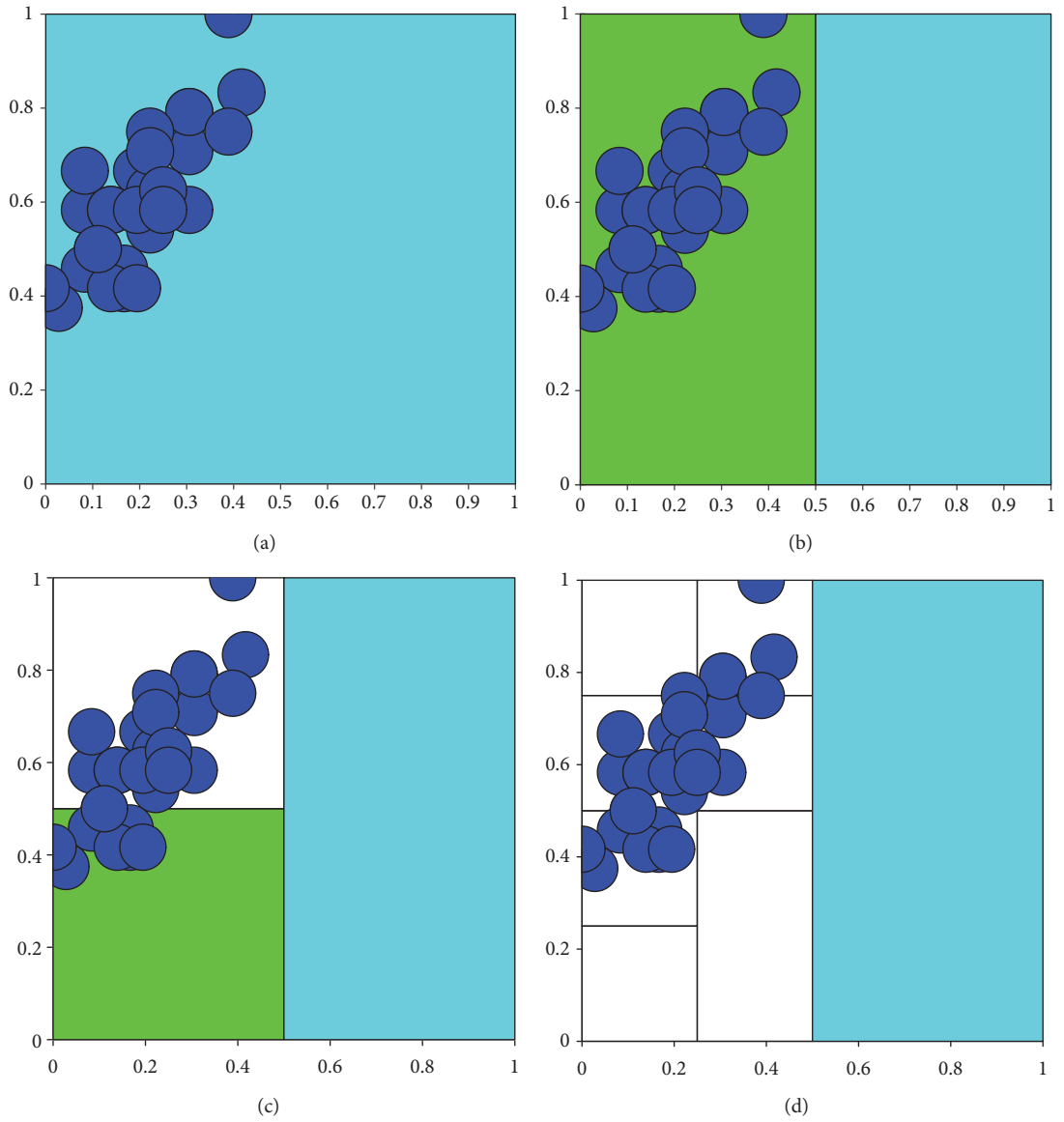


Figure 7: The space division process.

Table 4: Spatial partitioning algorithm.

GenerateSpaces(*Train*, *SubSpaces*)
Input: training set *Train*
Output: subspace set *SubSpaces*

Step 1. Initialize the spatial attributes $sub = <[0, 1]^n, \varnothing, Train, \varnothing>$.
Step 2. Call *DivideSpace*(sub, 1) to divide the real-valued space.

$$x = \sqrt{N_0 P(1-P)}\left(Z_\alpha + \sqrt{\frac{N_0 P}{(1-P)}}\right). \tag{12}$$

α is the significance level, and Z_α is the α quantile of standard normal distribution.

Therefore, we can calculate the coverage of the existing detector set to non-self-space through the following formula. x' is the number of times that detectors continuously overlay the non-self-space during sampling.

$$\text{Coverage}(D) = \frac{x' \cdot P}{x}. \tag{13}$$

Although the algorithm divides $[0, 1]^n$ space into several subspaces, we can sample each subspace separately to determine the non-self-space coverage of each subspace. In this way, the idea is simple and the expected coverage rate of c_{exp} can be guaranteed. However, due to the limited resources of sensor nodes, the algorithm should reduce the resource consumption as much as possible, and sampling in the whole non-self-space and calculating the overall coverage can work. Only when determining whether the detector is overridden should detectors contained in the subspace where

Table 5: Space division iteration process.

DivideSpace(*sub*, *i*)
Input: subspace to be divided *sub*
Input: dimension to be divided *i*

Step 1. Determine whether the subspace *sub* does not contain any self or has a minimum diameter. If it is, the subspace *sub* is added to the *SubSpaces* and return. Otherwise, perform step 2.

Step 2. The i^{th} dimension of the subspace *sub* is divided, and two subspaces are obtained. Compute the contained selves and half-contained selves in the subspaces.

Step 3. $i = mod(i + 1, n)$. if $i = 0$, $i = n$.

Step 4. For every subspace, call *DivideSpace*(*sub*, *i*).

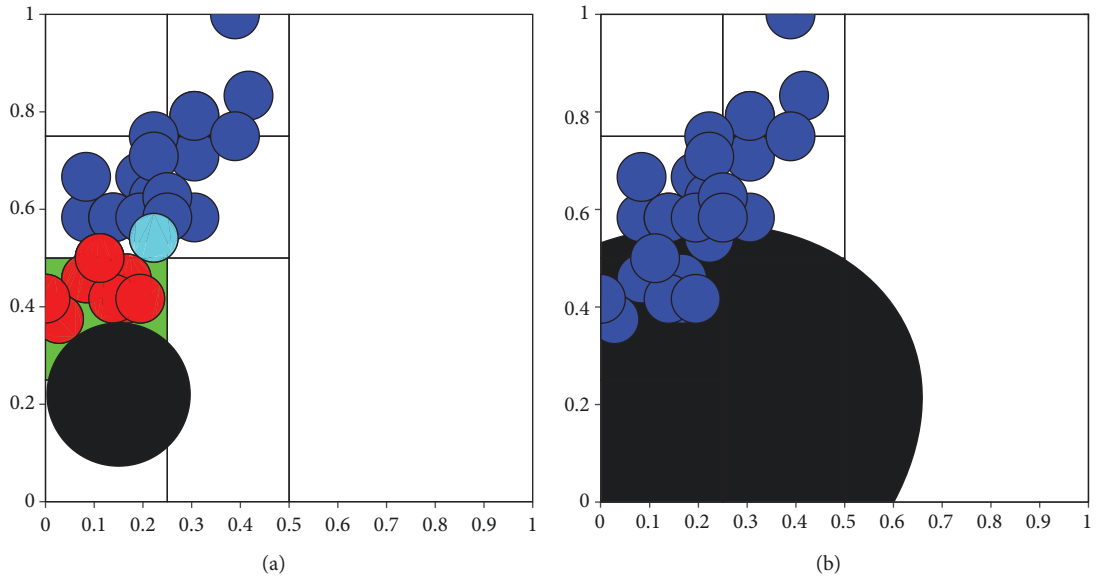


Figure 8: Radius of the candidate detector.

the sampling point is located be selected for judgment, rather than all detectors.

4.4. Complexity Analysis. For the sensor nodes running the intrusion detection system, only the subspace set needs to be stored. $SubSpaces = \{sub \mid sub = <[d_i^l, d_i^h], Half - contained\ Selves, Contained\ Selves, detectors>\}$, where each subspace contains the scope of the subspace, the contained self-set, the half-contained self-set, and the included detector set.

In the process of detector generating, time consumption mainly includes determining the subspace where the candidate detector is, calculation distances between the candidate and selves in the subspace, and calculation distances between the candidate and detectors in the subspace.

In the process of generating a mature detector, the time complexity of judging where the candidate detector is located is $O(n \cdot \ln(1/r_{\min})) = O(n)$. Suppose $r_{\min}$ is the minimum radius of the subspace, this complexity is related to the space dimension n. The time complexity of calculating the distance between the candidate detector and all selves in the subspace is no more than $O(|Self\ |)$. $|Self|$ is set as the size of self-set, and the average number of computation times of this operation is $|Self|/|SubSpaces|$, which is smaller than the number |Self| of other algorithms. The time complexity of calculating the distance between the candidate detector and all detectors in the subspace is no more than $O(|D\ |)$. $|D|$ is set as the size of the detector set, and the average number of computation times of this operation is $|D|/|SubSpaces|$, which is much smaller than the number $|D|$ of other algorithms, saving sensor node resources.

The time complexity of generating a mature detector is $O(n) + O(|Self\ |) + O(|D\ |) = O(n+|Self|+|D\ |)$. Let N_c be the number of candidate detectors required to generate detector set D, $N_c \approx |D|/(1 - P)$. Therefore, the time complexity of detector generation is $O(n+|Self|+|D\ |) \cdot N_c = O(|D|\cdot(n+|Self|+|D\ |)/(1 - P))$.

In the detection process of detectors, the main time consumption is to determine the subspace where the antigen to be detected is and to calculate the distance between the antigen and all detectors in the subspace.

The time complexity of determining the subspace where the antigen to be detected is located is $O(n \cdot \ln(1/r_{\min})) = O(n)$. The time complexity of calculating the distance between the antigen and all detectors in the subspace is no more than $O(|D\ |)$. The average number of computation times of this operation is $|D|/|SubSpaces|$, which is much smaller than the number $|D|$ of other algorithms, saving sensor node resources. Therefore, the time complexity of detector detection process is $O(n) + O(|D\ |) = O(n+|D\ |)$.

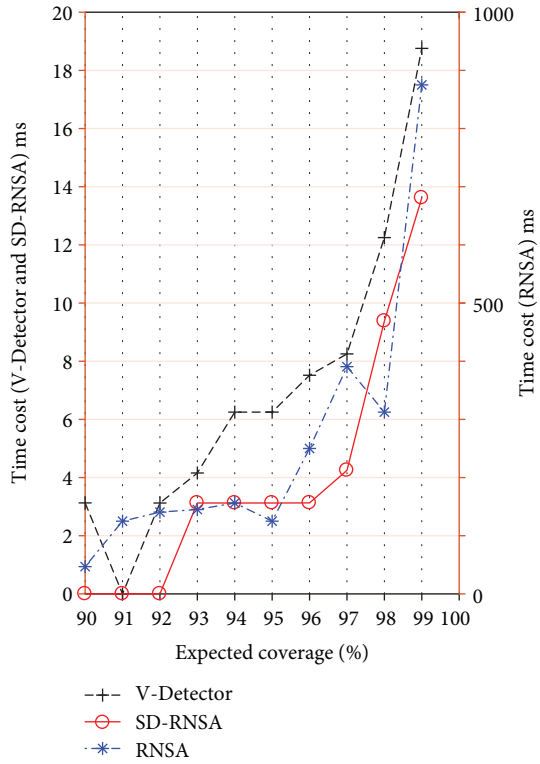


Figure 9: Comparisons of the consumption time in the detector generation stage.

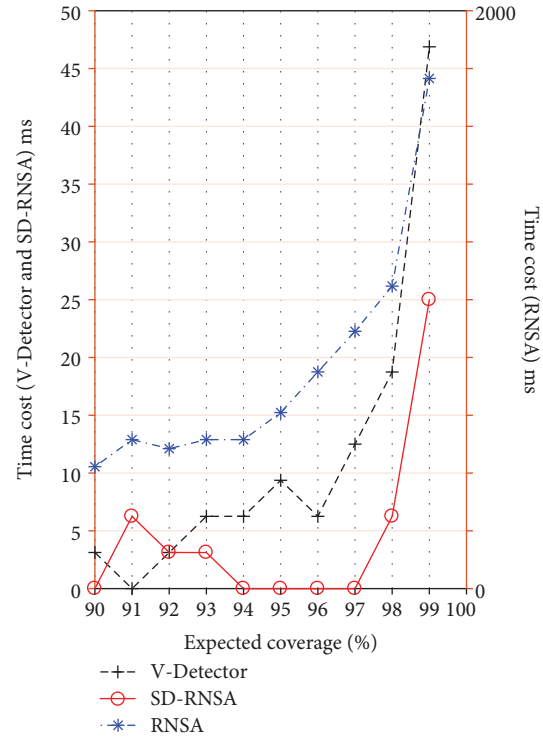


Figure 10: Comparisons of the consumption time in the intrusion detection stage.

5. Experiments

This section verifies the validity of the algorithm through experiments. The Iris data set [36] in UCI and wireless sensor environment are used to analyze the performance of the algorithm.

5.1. Iris Data Set. This data set contains three types of data, which are setosa, versicolor, and virginica [36]. Each type of data has 50 samples. Each sample contains four attributes: sepals width, sepals length, petal length, and petal width. In this paper, setosa samples are considered as self-set, and the top 30 samples are used as training set to generate mature detectors. Samples of versicolor and virginica classes are considered as abnormal data, and together with the last 20 samples of the setosa class, they are used as the data to be checked for anomaly detection.

Figure 9 shows comparisons of the consumption time of RNSA, V-Detector, and SD-RNSA in the detector generation stage. As can be seen from the figure, as the expected coverage rate rises, the time cost of RNSA rises very fast, while the time costs of V-Detector and SD-RNSA rise slowly. When the expected coverage rate is 99%, the time consumption of RNSA is 875.03 seconds, and that of V-Detector is 18.76 seconds, while that of SD-RNSA is 13.62 seconds, decreasing by 98.44% and 27.40%, respectively. Therefore, compared with RNSA and V-Detector, the detector generation efficiency of SD-RNSA has been greatly improved, which is more suitable for sensor networks with limited resources.

Figure 10 shows comparisons of the consumption time of RNSA, V-Detector, and SD-RNSA in the intrusion detection stage. As can be seen from the figure, with the increase in the expected coverage rate, the time costs of RNSA and V-Detector increase very quickly, while that of SD-RNSA increases slowly. This is because less detectors are required by SD-RNSA to achieve the same coverage rate, and samples for inspection do not need to match with all detectors, which can reduce the time cost of the test phase and save the resources of sensor nodes.

Figure 11 shows comparisons of detection rate and false alarm rate of RNSA, V-Detector, and SD-RNSA. As can be seen from the figures, detection rates of the three algorithms are relatively close, among which RNSA is slightly lower. The false alarm rates of the three algorithms is relatively close, among which RNSA is slightly higher. It can be seen that the detection efficiency of SD-RNSA has not been reduced on the basis of saving time cost.

5.2. Applications in WSN. In order to test the efficiency of the algorithm in WSN, TOSSIM was used as the simulator for the simulation test. It is a component-based modular discrete event simulation tool of TinyOS, which is suitable for wireless sensor network simulation. The network deployment area is 1000 m^2. Sensor nodes are randomly distributed in the network with 200 nodes, and the base station locates at (0, 0). The MAC layer protocol is IEEE802.15.4. The routing protocol is LEACH, and 10% of sensor nodes are selected as cluster heads.

Experiments used jamming attack for testing, which is a DoS attack of WSN. The attackers regularly broadcast

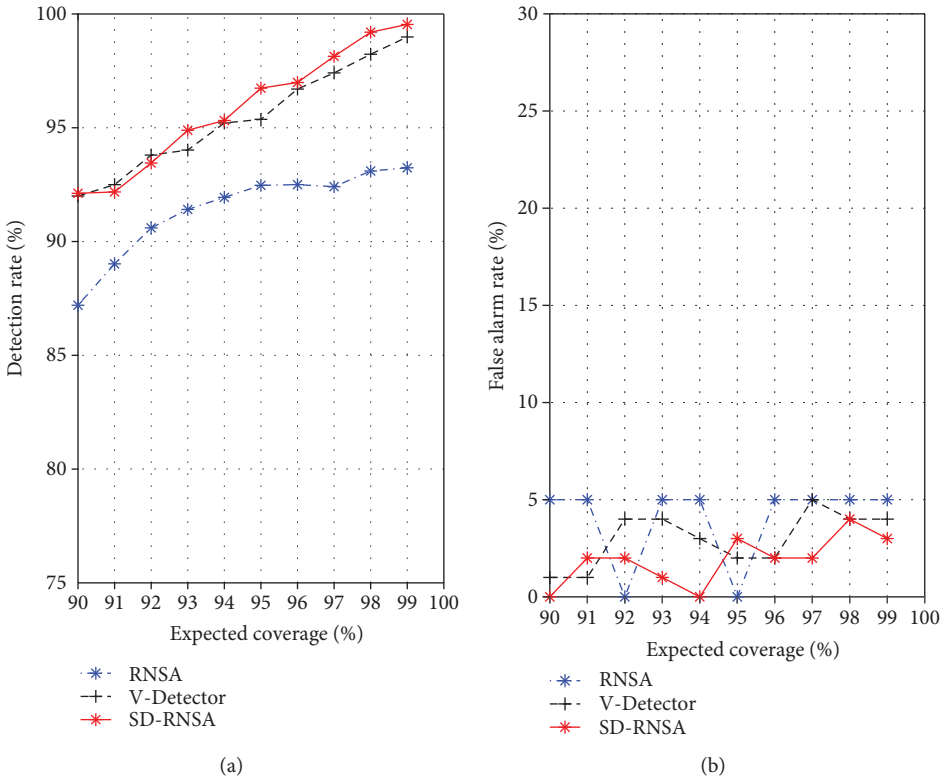


Figure 11: Comparisons of detection rate and false alarm rate.

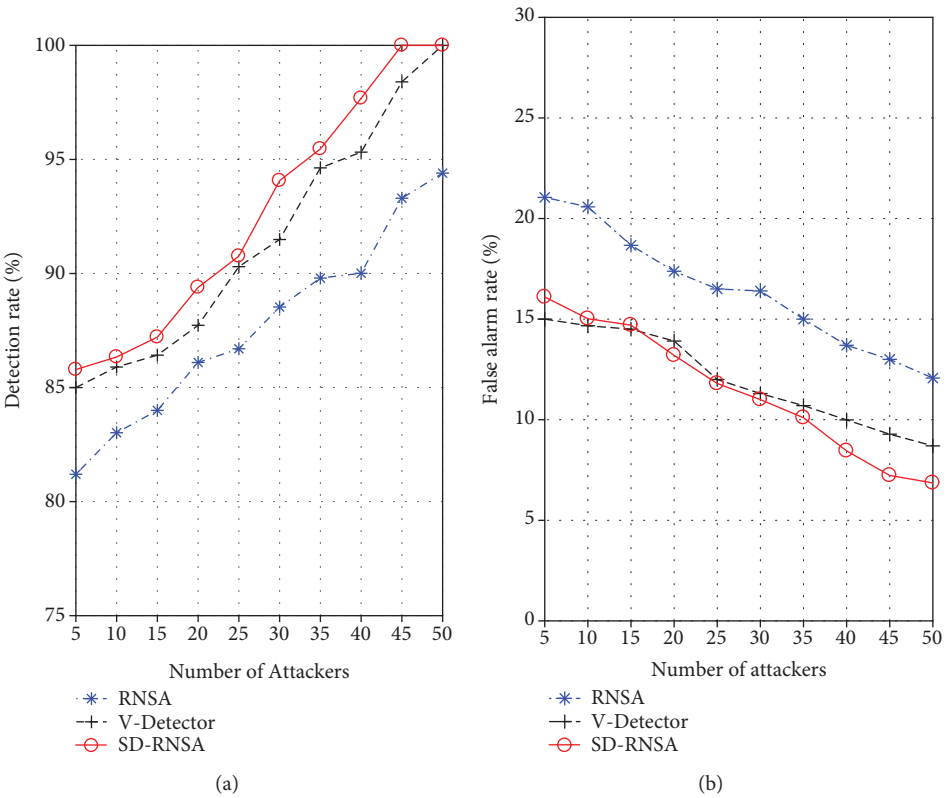


Figure 12: Comparisons of detection rate and false alarm rate in WSN (under jamming attack).

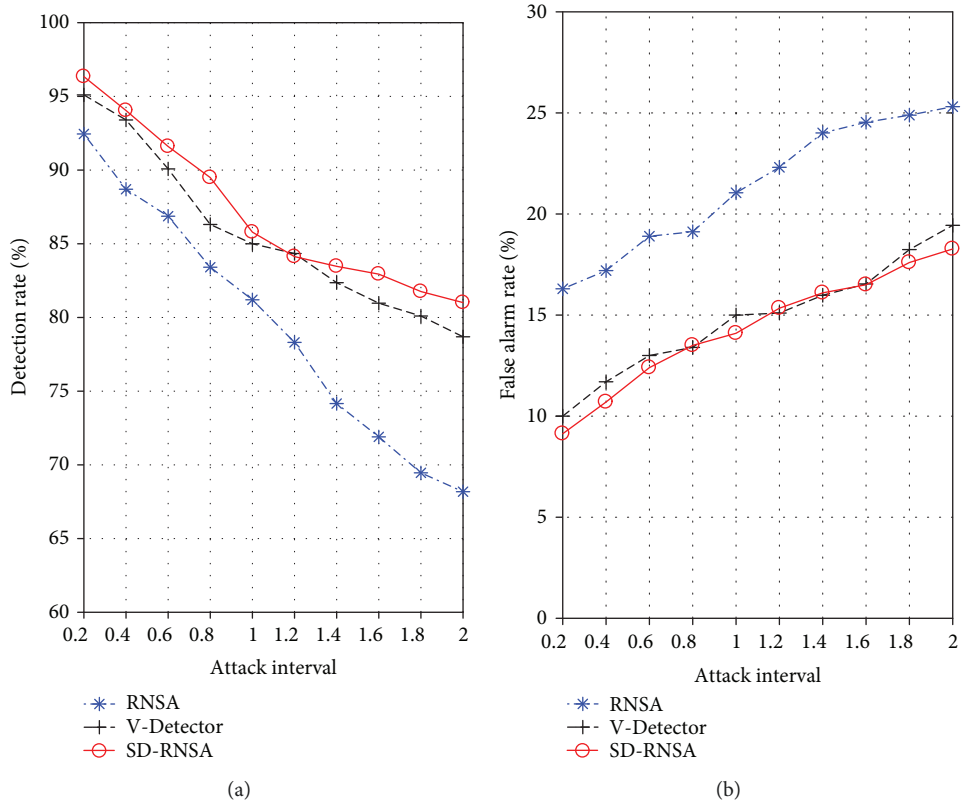


Figure 13: Comparisons of the detection rate and false alarm rate in WSN 2 (under jamming attack).

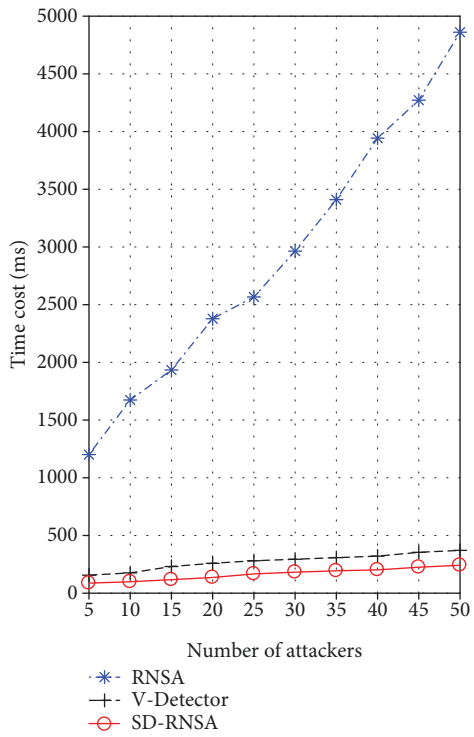


Figure 14: Comparisons of the detection time in WSN (under jamming attack).

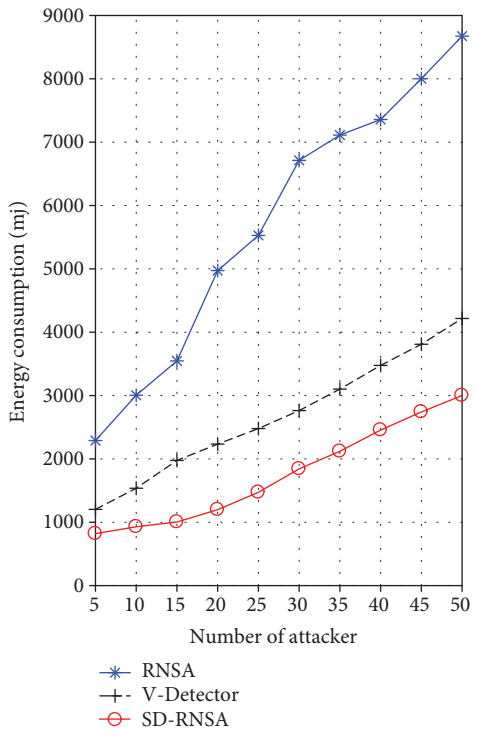


Figure 15: The energy consumption comparisons in WSN (under jamming attack).

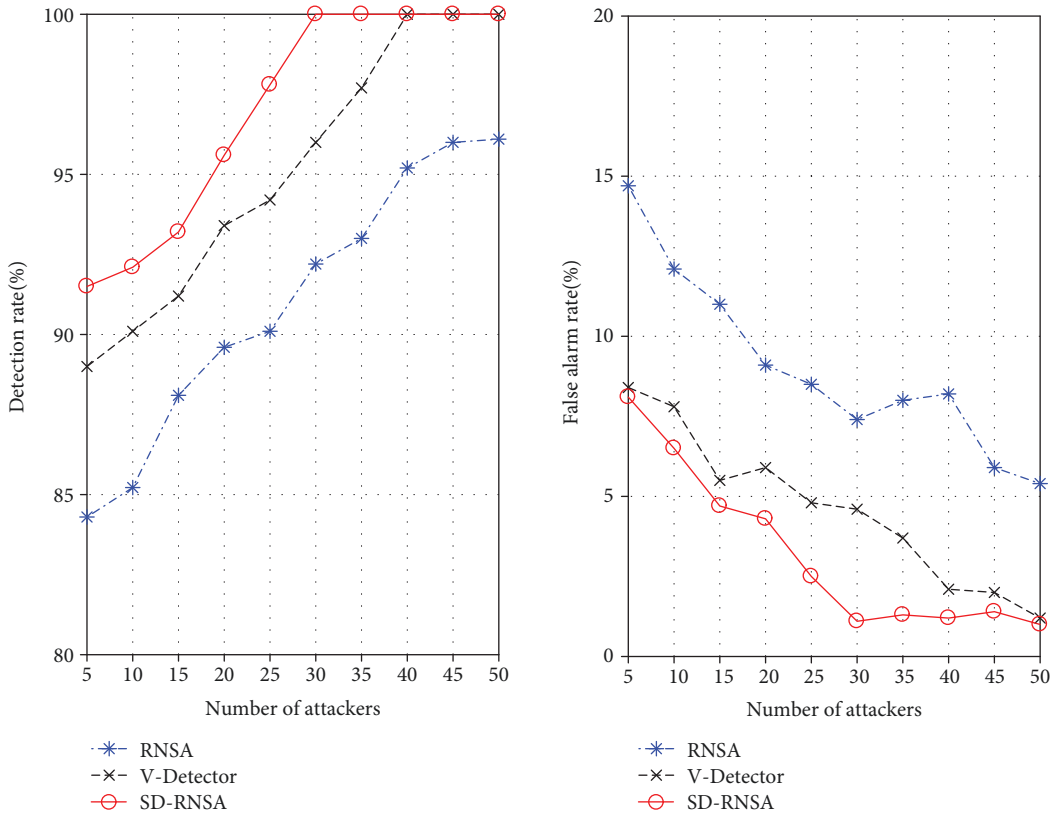


Figure 16: Comparisons of the detection rate and false alarm rate in WSN (under sinkhole attack).

meaningless packets to the network, affecting the communication between nodes and consuming node resources. In the experiment, 10 times of simulation were carried out to take the mean values, and the network changes within 2 hours were collected each time. The first half hour is the training time of algorithms, during which algorithms learn the normal behavior of the network and generate detectors. The remaining time is the intrusion detection of algorithms.

In a wireless sensor network, changes of the node's own attributes reflect the current environment information. When the network is in normal state, attributes of nodes such as throughput, packet loss rate, node reception rate, average node delay, etc., are in a relatively stable range. When the network is under attack, these attributes will change significantly. t can be seen that the algorithm should be more sensitive to the attacks of the network layer. In this paper, the packet sending rate, packet receiving ratio, and node throughput are chosen as the detection characteristics. If the attack has little influence on these attributes, the effect of the algorithm is limited, for example, attacks against the application layer, like location attacks and malicious code.

First, the attack time interval was set to 1 s, and the number of attackers changed from 5 to 50. Figure 12 shows comparisons of the detection rate and false alarm rate of the three algorithms. It can be seen from the figures, with more and more attackers, more and more sensor nodes feel abnormal, and the detection rates of all three algorithms are rising. When there are few attackers, the detection rate of this algorithm is slightly higher than those of the other two algorithms. With more and more attackers, the false alarm rates of the three algorithms are gradually reduced. When there are few attackers, the error rate of this algorithm is slightly lower than those of the other two algorithms. This is because detectors generated by the algorithm in this paper better cover the non-self-space and reduce the vulnerability.

The number of attackers was then set to 10, and the attack time interval changed from 0.2 s to 2 s. Figure 13 shows comparisons of the detection rate and false alarm rate of the three algorithms. When the time interval between attacks is larger and the attack behavior is less and less obvious, it is more difficult for sensor nodes to feel abnormal. The attack behavior is close to normal behavior, which represents the non-self-space closest to self-space. This space is usually vulnerability, which is difficult to be covered by detectors. It can be seen from the figures that the detection rate of all three algorithms is decreasing and the false alarm rate is increasing as the attack time interval is getting larger. This algorithm is still superior to the other two algorithms.

The attack time interval was set to 1 s, and the number of attackers changed from 5 to 50. Figure 14 shows comparisons of the detection time of the three algorithms. The detection time is the time cost for sensor nodes to process antigens to be detected. With more and more attacking nodes and more and more data traffic in the network, there are more and more undetected antigens to be processed by nodes, and

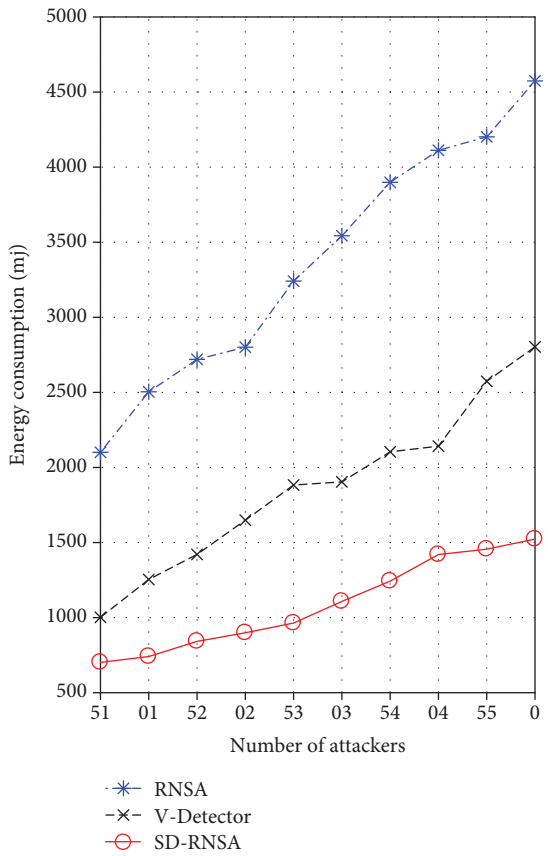


Figure 17: The energy consumption comparisons in WSN (under selective forwarding attack).

the detection time of the three algorithms is longer and longer. Because the time cost of this algorithm in detecting nodes is low, the detection time of this algorithm is obviously smaller than that of the other two algorithms.

The attack time interval was set to 1 s, and the number of attackers changed from 5 to 50. Figure 15 shows the energy consumption comparisons of the three algorithms. As can be seen from the figure, with the increase of the number of attack nodes, the energy consumptions of all three models have increased, but this algorithm has obvious advantages.

Then, experiments adopted the selective forwarding attack and sinkhole attack to test. Selective forwarding attack refers to the process in which an attacker as a routing node chooses to discard or selectively forward data packets in probability. In the sinkhole attack, the target of the attacker is to attract the data flow in a region through the attack node by broadcasting high-quality routing information. Similarly, 10 simulations were carried out to take the mean value. Figure 16 shows contrasts of detection rates and false alarm rates of the three algorithms when the number of attackers changes under sinkhole attack. Figure 17 shows energy consumption comparisons of the three algorithms under selective forwarding attack. As can be seen from the diagrams, the effect of this algorithm is better than the other two algorithms, which is similar to the jamming attack.

6. Conclusions

This paper first analyzes the intrusion detection technology in wireless sensor networks, including research on the specification-based detection, research on misuse detection, and research on anomaly detection. Many detection systems are based on the traditional network intrusion detection technology transplant transformation, which do not sufficiently consider characteristics of wireless sensor networks and are rarely applied to the real wireless sensor networks. Inspired by the negative selection algorithm in the biological immune system, this paper proposes a wireless sensor network intrusion detection model based on the spatial partition negative selection algorithm. In this paper, the model is described comprehensively and its performance is analyzed theoretically. Finally, the article applied the model to the UCI data set and wireless sensor networks. Experimental results show that the model has better time efficiency and detector quality, saves sensor node resources, and reduces the energy consumption. It is an effective algorithm for wireless sensor network intrusion detection.

Data Availability

The data from the hereby described analysis can be made available from the authors' request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank the Sichuan Provincial Education Department of China-funded project (035Z2258) for providing financial aid.

References

[1] J. Zheng and B. X. Zhang, *Wireless Sensor Network Technology*, China Machine Press, 2012.

[2] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 266–282, 2014.

[3] A. Abduvaliyev, A. S. K. Pathan, Jianying Zhou, R. Roman, and Wai-Choong Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.

[4] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[5] A. Ghosal and S. Halder, "A survey on energy efficient intrusion detection in wireless sensor networks," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 2, pp. 239–261, 2017.

[6] F. H. Tseng, H. P. Chiang, and H. C. Chao, "Black hole along with other attacks in MANETs: a survey," *Journal of Information Processing Systems*, vol. 14, no. 1, pp. 56–78, 2018.

[7] V. Bhuse, A. Gupta, and A. Al-Fuqaha, "Detection of masquerade attacks on wireless sensor networks," in *2007 IEEE International Conference on Communications*, pp. 1142–1147, Glasgow, Scotland, UK, June 2007.

[8] A. P. R. da Silva, M. H. T. Martins, and B. P. S. Rocha, "Decentralized intrusion detection in wireless sensor networks," in *1st ACM International Workshop on Quality of service and security in wireless and mobile networks*, Montreal, Quebec, Canada, October 2005.

[9] S. K. Singh, M. P. Singh, and D. K. Singh, "Intrusion detection based security solution for cluster-based wireless sensor networks," *International Journal of Advanced Science and Technology*, vol. 30, 2011.

[10] H. Jadidoleslamy, "A high-level architecture for intrusion detection on heterogeneous wireless sensor networks: hierarchical, scalable and dynamic reconfigurable," *Wireless Sensor Network*, vol. 3, no. 7, pp. 241–261, 2011.

[11] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006*, vol. 1, pp. 640–644, Las Vegas, NV, USA, 2006.

[12] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *(WiMob'2005), IEEE International Conference on Wireless and Mobile computing, Networking and Communications*, vol. 3, pp. 253–259, Montreal, Canada, 2005.

[13] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for routing attacks in sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2, no. 4, 332 pages, 2006.

[14] S. Jian-hua and M. Chuan-Xiang, "Anomaly detection based on data-mining for routing attacks in wireless sensor networks," in *2007 Second International Conference on Communications and Networking in China*, pp. 296–300, Shanghai, China, August 2007.

[15] R.-C. Chen, C.-F. Hsieh, and Y.-F. Huang, "A new method for intrusion detection on hierarchical wireless sensor networks," in *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication - ICUIMC '09*, pp. 238–245, Suwon, Korea, January 2009.

[16] R.-C. Chen, C.-F. Hsieh, and Y.-F. Huang, "An isolation intrusion detection system for hierarchical wireless sensor networks," *Journal of Networks*, vol. 5, no. 3, pp. 335–342, 2010.

[17] C. F. Hsieh, Y. F. Huang, and R. C. Chen, "A light-weight ranger intrusion detection system on wireless sensor networks," in *2011 Fifth International Conference on Genetic and Evolutionary Computing*, pp. 49–52, Kitakyushu, Japan, August 2011.

[18] S. Misra, K. I. Abraham, M. S. Obaidat, and P. V. Krishna, "LAID: a learning automata-based scheme for intrusion detection in wireless sensor networks," *Security and Communication Networks*, vol. 2, no. 2, 115 pages, 2009.

[19] S. Misra, P. V. Krishna, and K. I. Abraham, "A simple learning automata-based solution for intrusion detection in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 3, 441 pages, 2011.

[20] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, "Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 518–533, 2010.

[21] M. Gunasekaran and S. Periakaruppan, "GA-DoSLD: genetic algorithm based denial-of-sleep attack detection in WSN," *Security and Communication Networks*, vol. 2017, Article ID 9863032, 10 pages, 2017.

[22] Q. Shi, L. Qin, L. Song, R. Zhang, and Y. Jia, "A dynamic programming model for internal attack detection in wireless sensor networks," *Discrete Dynamics in Nature and Society*, vol. 2017, Article ID 5743801, 9 pages, 2017.

[23] Y. B. Reddy, "A game theory approach to detect malicious nodes in wireless sensor networks," in *2009 Third International Conference on Sensor Technologies and Applications*, Greece, June 2009.

[24] Z.-m. Cheng, "A Differential game model between intrusion detection system and attackers for wireless sensor networks," *Wireless Personal Communications*, vol. 90, no. 3, pp. 1211–1219, 2016.

[25] M. Drozda, S. Schaust, and H. Szczerbicka, "AIS for misbehavior detection in wireless sensor networks: performance and design principles," in *2007 IEEE Congress on Evolutionary Computation*, pp. 3719–3726, Singapore, September 2007.

[26] Y. Liu and F. Yu, "Immunity-based intrusion detection for wireless sensor networks," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 439–444, Hong Kong, China, June 2008.

[27] R. Fu, K. Zheng, T. Lu, and Y. Yang, "Danger theory inspired intrusion detection model for wireless sensor networks," *Journal of Communications*, vol. 33, no. 9, pp. 31–37, 2012.

[28] H. M. Salmon, C. M. de Farias, P. Loureiro et al., "Intrusion detection system for wireless sensor networks using danger theory immune-inspired techniques," *International Journal of Wireless Information Networks*, vol. 20, no. 1, pp. 39–66, 2013.

[29] X. Xiao and R. Zhang, "Study of immune-based intrusion detection technology in wireless sensor networks," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3159–3174, 2017.

[30] W. Guo, Y. Chen, Y. Cai, T. Wang, and H. Tian, "DIntrusion detection in WSN with an improved NSA based on the DE-CMOP," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 11, 2017.

[31] S. Forrest, L. Allen, A. S. Perelson, and R. Cherukuri, "Selfnonself discrimination in a computer," in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp. 202–212, Oakland, CA, USA, May 1994.

[32] D. Dasgupta, S. Yu, and F. Nino, "Recent advances in artificial immune systems: models and applications," *Applied Soft Computing*, vol. 11, no. 2, pp. 1574–1587, 2011.

[33] F. A. Gonzalez and D. Dasgupta, "Anomaly detection using real valued negative selection," *Genetic Programming and Evolvable Machines*, vol. 4, no. 4, pp. 383–403, 2003.

[34] Z. Ji, "Negative selection algorithms: from the thymus to V-detector [Ph.D. dissertation]," University of Memphis, Memphis, Tenn, USA, 2006.

[35] Z. Ji and D. Dasgupta, "*V-detector*: an efficient negative selection algorithm with "probably adequate" detector coverage," *Information Sciences*, vol. 179, no. 10, pp. 1390–1406, 2009.

[36] "UCI Dataset," http://archive.ics.uci.edu/ml/datasets.

[37] W. Chen, X. J. Liu, T. Li, Y. Q. Shi, X. F. Zheng, and H. Zhao, "A negative selection algorithm based on hierarchical clustering of self set and its application in anomaly detection,"

International Journal of Computational Intelligence Systems, vol. 4, no. 4, pp. 410–419, 2011.

[38] M. Gong, J. Zhang, J. Ma, and L. Jiao, "An efficient negative selection algorithm with further training for anomaly detection," *Knowledge-Based Systems*, vol. 30, pp. 185–191, 2012.

[39] M. R. Abdolahnezhad and T. Banirostam, "Hybrid email spam detection method using negative selection and genetic algorithms," *IJARCCE*, vol. 5, no. 4, pp. 956–960, 2016.

[40] I. Idris, A. Selamat, N. Thanh Nguyen et al., "A combined negative selection algorithm–particle swarm optimization for an email spam detection system," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 33–44, 2015.

[41] F. P. A. Lima, A. D. P. Lotufo, and C. R. Minussi, "Wavelet-artificial immune system algorithm applied to voltage disturbance diagnosis in electrical distribution systems," *IET Generation Transmission and Distribution*, vol. 9, no. 11, pp. 1104–1111, 2015.

[42] H. Wang, X. Z. Gao, X. Huang, and Z. Song, "PSO-optimized negative selection algorithm for anomaly detection," in *Applications of Soft Computing*, E. Avineri, M. Köppen, K. Dahal, Y. Sunitiyoso, and R. Roy, Eds., vol. 52 of Advances in Soft Computing, pp. 13–21, Springer, Berlin, Heidelberg, 2009.