Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 8980876, 23 pages
https://doi.org/10.1155/2023/8980876

Research Article

A Modified Gray Wolf Optimizer-Based Negative Selection Algorithm for Network Anomaly Detection

Geying Yang,[1,2] **Lina Wang**,[1,2] **Rongwei Yu**,[1,2] **Junjiang He**,[3] **Bo Zeng**,[1,2] **and Tian Wu**[1,2]

[1]*Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, Wuhan, China*
[2]*School of Cyber Science and Engineering, Wuhan University, Wuhan 43007, China*
[3]*College of Cybersecurity, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Lina Wang; lnwang@whu.edu.cn and Tian Wu; wutian@whu.edu.cn

Received 2 September 2022; Accepted 7 October 2022; Published 24 February 2023

Academic Editor: Yu-An Tan

Intrusion detection systems are crucial in fighting against various network attacks. By monitoring the network behavior in real time, possible attack attempts can be detected and acted upon. However, with the development of openness and flexibility of networks, artificial immunity-based network anomaly detection methods lack continuous adaptability and hence have poor detection performance. Thus, a novel framework for network anomaly detection with adaptive regulation is built in this paper. First, a heuristic dimensionality reduction algorithm based on unsupervised clustering is proposed. This algorithm uses the correlation between features to select the best subset. Then, a hybrid partitioning strategy is introduced in the negative selection algorithm (NSA), which divides the feature space into a grid based on the sample distribution density and generates specific candidate detectors in the boundary grid to effectively mitigate the holes caused by boundary diversity. Finally, the NSA is improved by self-set clustering and a novel gray wolf optimizer to achieve adaptive adjustment of the detector radius and position. The results show that the proposed NSA algorithm based on mixed hierarchical division and gray wolf optimization (MDGWO-NSA) achieves a higher detection rate, lower false alarm rate, and better generation quality than other network anomaly detection algorithms.

1. Introduction

In recent years, security threats and attacks on network infrastructures have become the leading causes of major losses of massive sensitive data. Anomaly detection is widely used in intrusion detection systems due to its characteristics of ensuring data integrity and confidentiality. In brief, it can be regarded as a normal/anomaly classification problem. Several modern technologies have been proposed in recent studies to solve this problem: neural networks [1], support vector machines (SVMs) [2], decision trees [3], and genetic algorithms [4]. Due to the lack of autonomy and self-evolution ability, existing network security technologies are powerless to deal with unknown network threats. By simulating the human immune mechanism, the artificial immune system (AIS) can support unknown attack detection, situational awareness, and other cybersecurity solutions. As a crucial algorithm in artificial immune theory, the negative selection algorithm (NSA) was first introduced by [5], which generates a maturation detector to detect abnormalities by simulating the maturation process of T cells in thymocytes. Therefore, NSA is extensively developed in AIS, and it has been utilized in network anomaly detection, data mining, multiobjective optimization, and other fields [6–8].

Traditionally, the detector is generated by a randomization method and then compared with self-trained samples to achieve tolerance in the NSA training process. This has resulted in a time complexity that is exponentially related to the self-set and the unstable detection rate. This dramatically limits the broad application of the NSA. Therefore, two urgent issues of the NSA algorithm need to be solved: (1) Improper positioning of randomly generated detectors leads to the generation of abundant holes and redundant detectors, which reduces the detection rate. (2) The randomly

generated detectors ignore the diversity of boundary samples, and all detectors share a common generation strategy, which greatly reduces the efficiency of the algorithm.

In addition, overcoming many attack types and network traffic attributes for network anomaly detection remains challenging. Expanding the search space results in higher computational complexity. Notably, feature selection has been proven to be a great solution for IDS. It can detect highly correlated features and eliminate useless features when the performance is slightly reduced, thereby reducing the error rate of detection. At present, the most popular feature selection strategy focuses on selecting the best-fitting function, which depends on the measurement of the dataset and lacks the performance of the classifier. To reduce the computational complexity, a related study [9] optimized dimensionality reduction. Low average detection performance is achieved because the dataset patterns are restricted by the inherent limitations of the dimensionality reduction technology and the randomness of the detector generation process. Moreover, the correlation changes between features over time have not been considered in either PCA or correlation-based feature selection techniques. They have only focused on the features that are more descriptive of the dataset or most correlated with the class label.

Therefore, we propose a new approach for network anomaly detection. The main contributions of this paper are as follows:

(i) A novel anomaly detection architecture is proposed comprising of two stages: (1) feature selection based on interrelationships and (2) anomaly detection based on MDGWO-NSA.

(ii) An interrelationship-based feature selection method is proposed. Through feature clustering, the features are divided by similarity, and useless features are removed. In addition, both interclass correlation and intraclass redundancies are considered, which effectively reduce the features and improve the accuracy of the model.

(iii) A hybrid partitioning-based detector generation strategy is proposed. Then, specific candidate detectors are generated by the boundaries, which effectively solves the problem of low boundary detection rate due to the diversity of sample boundaries.

(iv) An adaptive unbounded detector generation method based on self-sample clustering with GWO optimization is proposed. The global optimal position and the most suitable radius of the detector are adaptively adjusted by the fitness function proposed in this paper. This method effectively reduces the generation time of the detector and improves the detection rates.

2. Related Work

Network anomaly detection based on the MDGWO-NSA can be divided into two parts: feature selection and NSA-based network anomaly detection. In terms of NSA, recent research has focused more on the process of candidate detector generation.

2.1. Feature Selection in Intrusion Detection. Network intrusion detection has multiple and large-scale features. In principle, more features will allow more fine-grained analysis. However, expanding related features will lead to a longer training time, and not all features are valuable in describing data traffic. Not only will the detection efficiency be reduced, but it will also introduce bias in the feature classification process. Therefore, feature selection is also a crucial preprocessing step in network anomaly detection.

However, due to the variety of traffic types, it is difficult to manage the feature space and directly apply it to network traffic analysis. To reduce the features based on the data dimensionality reduction, Hadri et al. [10] used PCA and fuzzy PCA to remove redundant features, and Benaddi et al. [11] added the KNN method in fuzzy PCA, which can effectively reduce the original features of all connected records stored in the dataset. Based on the ensemble method, Khammassi and Krichen [12] combined a genetic algorithm-based wrapper method and a logistic regression algorithm for feature selection, which increased the classification accuracy. Based on the correlation between features and layer configuration, Nazir and Khan [13] introduced the taboo search-random forest (TS-RF) to reduce the time complexity of the model. The study in [14] applied discrete differential evolution (DDE) and the C4.5 decision tree algorithm to find the optimal feature subset. Based on statistical methods, Mohammed et al. [15] proposed an algorithm based on mutual information that can process both linear and nonlinear correlated features. Mishra et al. [16] combined the chi-square function and a recursive feature elimination method to reduce the dimensionality of server traffic data. Wang et al. [17] proposed a correlation-based feature selection algorithm ECOFS to estimate the correlation between class features and reduce redundant features. Similarly, Guerroumi et al. [18] reduced the space through feature selection based on the coefficient of variation, effectively decreasing the false alarm rate.

Although most of the above methods improve classification accuracy through optimal selection strategies, the dependency and consistency among features are ignored in the evaluation of feature subsets. Moreover, the correlation between features and the combined effect of different feature subsets is not sufficiently considered. Therefore, we propose a new feature selection model by comprehensively analyzing the correlations between features and feature subsets. It selects features sequentially in a two-level approach. First, similar features are classified by clustering, and then intraclass and interclass distances of features are calculated based on symmetric uncertainty (SU) and maximum correlation coefficient (MIC). Multilevel analysis effectively reduces redundant features and improves classification accuracy.

2.2. Candidate Detector Generation. Typically, the randomness of the detector generation process leads to much-repeated coverage of detectors. This makes many detectors unable to be transformed into mature states (they cannot be used normally) and greatly limits the application of the NSA.

To solve the above problem, several methods have been introduced to improve the randomness of detector generation, including those based on sample distribution, the generation process, and the combination of other algorithms. Based on sample distribution, Xiao et al. [19] employed an immune optimization mechanism in morphological space to generate candidate detectors hierarchically from far to near. Cui et al. [20] introduced the self-set edge suppression strategy and the detector self-suppression strategy. In this approach, the individual self-radius dynamically changes to avoid generating too many invalid detectors. Liu et al. [21] focused on a fixed-boundary negative selection algorithm (OALFB-NSA), which generates a layer of detectors around the self-space and can adapt to various real-time changes in the self-space. Based on the generation process, [22] incorporated further training strategies into the training phase to generate self-detectors covering self-regions. Meanwhile, Zheng et al. [23] added a negative selection when the detector was generated to avoid redundant coverage between mature detectors. In terms of incorporating other algorithms, Aydin et al. [24] applied chaotic mapping for parameter selection, which obtained a better coverage. To achieve the best detection performance, Yang et al. [25] combined a real negative selection algorithm with evolutionary preference (RNSAP). In addition, some works have used swarm intelligence optimization algorithms to improve detector generation strategies, such as particle swarm optimization (PSO) [6] and fruit fly optimization (FFO) [26].

Despite the efforts of the above detector generation methods to try to modify the generation of detectors instead of generating candidate detectors randomly, the quality of the generated detectors is still poor. In contrast, we generate specific detectors at the boundaries by dynamically partitioning the detector generation region and optimizing the detector locations in the nonboundary regions with a swarm intelligence optimization algorithm. This two-layer detector generation approach greatly improves the quality of detector generation.

2.3. Detector Matching Tolerance Mechanism. This mechanism is mainly focused on the improvement of the pretreatment method of the self-set and the detector matching rules [27]. With the continuous development of NSA algorithms, the efficiency problems of traditional detectors in the tolerance phase have greatly limited the application of NSA algorithms.

To alleviate the distance calculation cost, Chen et al. [28] utilized the cluster center to replace the self and the candidate detector for matching. Subsequently, [29] narrowed the comparison range of detectors by using different grid partitioning strategies in the feature space. To reduce the time complexity of the detector calculation, Yang et al. [30] applied the antigen spatial density to calculate the low-dimensional subspace of densely aggregated antigens which generates detectors directly in these subspaces. Fouladvand et al. [31] improved the real-valued negative selection algorithm based on Delaunay triangular dissection (dnyNSA) to generate detectors with more rational locations and sizes. Li et al. [32] employed the known nonself as the candidate detector center to generate the detector and thus effectively improve the detection rate.

Obviously, most of the studies have focused on preprocessing the self-sets to improve the detector generation efficiency. For example, dividing the feature space and clustering self-sets have achieved good results.

2.4. Hole Repair. Holes repair is an unavoidable problem for research in negative selection algorithms. The uncertainty of detector generation leads to holes easily generated during detector generation, which fails to cover the non-self-space adequately and generates redundancy. On the other hand, the probability of generating holes due to the diversity of boundaries is high in practical applications.

The coverage is an important indicator of the efficiency of detector generation. Chen et al. [28] analyzed the probabilistic aspects of the non-self-space coverage when given the conditions for detector stop generation. Li and Chen [33] used the Monte Carlo method to calculate the overlap volume of the hypersphere and proposed a non-self-covering calculation method based on confidence estimation. Fouladvand et al. [31] compared the randomly generated pattern with the self-space GMM and retained the low probability random pattern as a detector. Yang et al. [30] applied "antibody inhibition rate" instead of "expected coverage" as the termination condition. During detector generation, to increase detector coverage and reduce holes. Abid et al. [34] added a training phase that divided the non-self-space into multiple layers, which efficiently generated detectors using normal (self) data. Moreover, to improve the detection rate, Li et al. [35] introduced KNN in a variable-sized detector to classify misclassified instances.

In summary, several methods exist for improving detector performance, but most of them neglect the adaptive generation capability of detectors. It is not feasible to dynamically adjust and optimize the detector based on its overlap, coverage, and holes during the generation process. Furthermore, only a few address the hole problem associated with boundaries that affects the overall effectiveness of detection. The next section presents a novel NSA based on a hybrid partitioning method with GWO optimization. This approach can significantly improve the efficiency of detector generation and the anomaly detection rate in the NSA.

3. The Proposed Method

To detect network anomalies, we propose a novel framework in this section. The motivation is to overcome the increasing number of unknown security threats. Artificial immune systems can provide defenses against internal structures. However, their performance is still significantly influenced by the redundancy of the generated detector. The time

complexity increased exponentially because the candidate detectors needed to be compared with all the self-antigens in the detector tolerance phase. Furthermore, the impact of feature selection on the efficiency of NSA-based anomaly detection is neglected in the existing work. Consequently, a new anomaly detection method is implemented that uses novel feature selection and MDGWO-NSA technology, which is dynamically adaptive and adjusts to enhance detection efficiency.

3.1. System Model. The overall architecture and composition of the MDGWO-NSA algorithm for network anomaly detection is shown in Figure 1. The framework divides the detection problem into two main phases: feature selection based on interrelationships and anomaly detection using NSA. (1) In the feature selection phase, an unsupervised feature selection scheme (DP-SUMIC) is used to select the most appropriate subset of features. Weighted distances are added to the k-nearest neighbor density, which calculates the influential cluster center to achieve clustering. To improve the clustering effect, a feature contribution scoring method that incorporates symmetric uncertainty and the maximum information coefficient is proposed, which is built on the principle of minimum intraclass distance and maximum interclass distance. Feature preference is introduced to select features with minimum redundancy and maximum correlation. In the anomaly detection phase, a grid-based partitioning strategy is developed to divide the non-self-space into the boundary and nonboundary regions. The location and radius of a specific boundary detector are calculated based on the boundary samples. The radius and location of the nonboundary detectors are adaptively adjusted according to the self-set clustering and the optimized GWO algorithm. As holes will be inevitably generated during the detector generation process, three hole repair methods are presented in this paper. Finally, the anomalies are identified and classified by the generated detectors. Evidently, the detection generation time is reduced and the detection accuracy is improved in our approach.

3.2. Interrelationship-Based Feature Selection. Feature selection is a two-step process that includes searching and evaluating feature subsets. Our work is motivated by the relationship between features. Compared to feature selection performed by direct application of the maximal information coefficient (MIC) [36] and symmetrical uncertainty (SU) [37], the two directions of feature selection are enhanced in our work. The first step of the algorithm is to employ density-based K-nearest neighbor (KNN) clustering, which improves the ability to remove redundant features. After ranking the features using redundancy control, redundant features are removed from the selected feature set. Second, we use the MIC and SU to select relevant features. The details of the two-stage data processing approach are summarized in Figure 2.

3.2.1. Feature Clustering. In high-dimensional data, similar features can be grouped into the same clusters to reduce the redundancy of features. However, current clustering techniques have inevitable limitations when they are used to cluster features. For example, in most clustering methods, the number of clusters needs to be set in advance, but the feature clusters are generally difficult to determine. In traditional density-based clustering, the cluster centers are determined by drawing a decision diagram of local densities and minimum clusters. Therefore, we propose a new clustering algorithm based on KNN and density peaks. After we select the clustering centers by weighted distance and local density, the labels of the clusters are propagated to the remaining samples using the nearest neighbor label propagation algorithm. More importantly, we introduce central influence, which is determined by computing the weighted distances and the local density of K-nearest neighbors. We combine the local density [38] with the information entropy-based distance. For the local density estimation of sample point i, the density calculation range is reduced from the entire sample set to the K nearest sample i. This better reflects the local information of sample point i. This is also more time-efficient and less complex without searching the sample K nearest neighbors. The detailed calculation steps are as follows:

(1) Standardize the attribute values and construct the attribute weight matrix.

$$att_{ij} = \frac{\max\left(x_{ij}\right) - x_{ij}}{\max\left(x_{ij}\right) - \min\left(x_{ij}\right)}, \tag{1}$$

where att_{ij} is the proportion of the j th dimensional attribute of object x_i.

(2) Calculate the entropy value and weight of the j th dimensional attribute.

$$\text{Entropy value: } H_j = -\frac{\sum_{i=1}^{n} att_{ij} \log_2\left(att_{ij}\right)}{\log_2 n}, \tag{2}$$

$$\text{weight value: } w_j = \frac{1 - H_j}{\sum_{j=1}^{m}\left(1 - H_j\right)}, \tag{3}$$

where $0 \le w_j \le 1$, $\sum_{j=1}^{m} w_j = 1$.

(3) Calculate the weight coefficient between adjacent attributes. Let attribute o be a neighbor of attribute i; then, the formula for calculating the weight coefficient between them is as follows:

$$w_{io} = \sum_{p=1}^{m} w_p \times \frac{x_{ip}}{\sum x_{op}}, \tag{4}$$

where x_{ip} is the attribute value of the p th dimension of object x_i, object x_o denotes the neighboring data object of object x_i, and ω_p is the weight value of the attribute of the p th dimension.

From the previous equation, the weight coefficients of the neighboring objects are determined by their

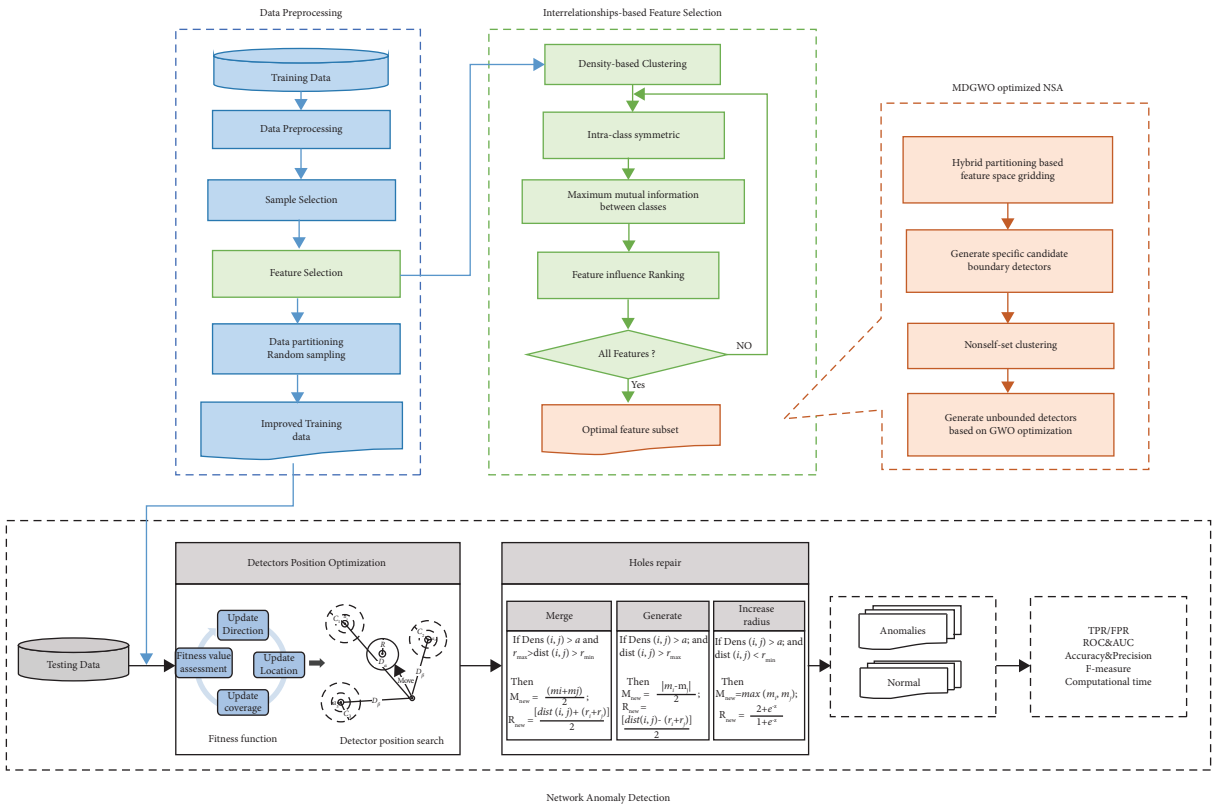

Figure 1: General layout of the proposed framework.

attributes jointly with all neighbors. Furthermore, it extends to take into account the influence of adjacent objects on each other and the dependencies of all their attributes when calculating distances.

(4) Then, calculate the distance based on information entropy. The improved distance calculation method fully considers the influence of attribute values. It calculates the previous difference of each object more accurately, which improves the accuracy of clustering in the actual clustering algorithm. The formula is as follows:

$$d(i,j) = w_{ij} \times \sqrt{\sum_{k=1}^{g} \left(x_{ik} - x_{jk}\right)^2}. \tag{5}$$

(5) If the density of sample point x_i is smaller than the density of other sample points, the distance δ of sample point x_i is the minimum distance between x_i and the sample with higher density. If sample point x_j has the highest density, δ_i is equal to the maximum distance between x_i and the other samples. δ_i is calculated as follows:

$$\delta_i = \begin{cases} \min_j \left(\text{dist}\left(i,j\right)\right), & \rho_j > \rho_i, \\ \max_j \left(\text{dist}\left(i,j\right)\right), & \rho_j \le \rho_i. \end{cases} \tag{6}$$

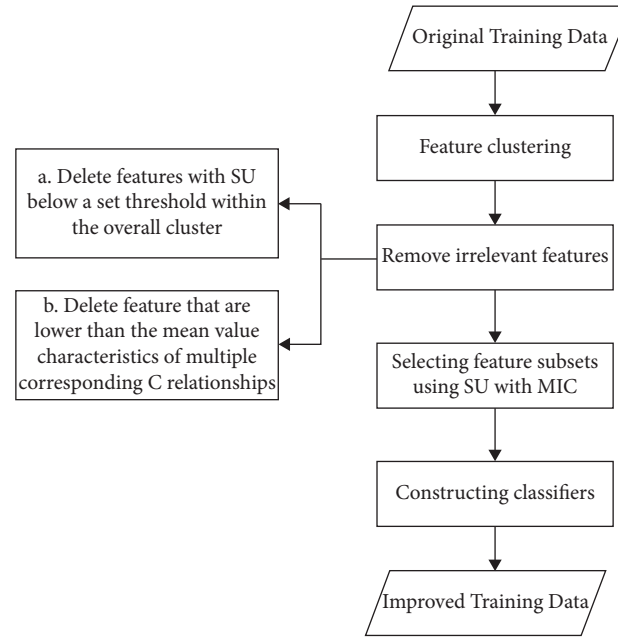

Figure 2: The improved feature selection.

(6) Calculate the local density ρ_i of attribute i. The smaller the distance between sample point i and the K nearest neighbors, the larger the density value [39].

$$\rho_i = \sum_{j \in \mathrm{KNN}(i)} \exp\left(-d_{ij}\right), \tag{7}$$

where KNN (i) is the set of K nearest neighbor samples of sample i and d_{ij} is the Euclidean distance between attributes i and j.

(7) According to the local density and weighted distance based on neighbors, the ability of the current data point as the center point is calculated. The larger the value of D is, the stronger the ability of the point as the cluster center will be. D is defined as follows:

$$D_i = \left(\frac{\rho_i}{\rho_{\max}}\right) * \left(\frac{\delta_i}{\delta_{\max}}\right). \tag{8}$$

The classification is improved after the connection threshold calculation and center influence ranking. To avoid the multidensity peak problems, after selecting a clustering center, all connected sample points are searched. Finally, the next cluster center is chosen from the unclassified sample until the stopping condition is satisfied.

3.2.2. Intraclass Symmetric Uncertainty and Interclass Maximum Correlation Coefficient. The inter-relationships and interactions between variables in biological systems are complex. Among the current nonredundant candidate features, the distinguishability of simply selecting the feature subset composed of the feature most relevant to the class label is not necessarily the best. Consequently, we combine unsupervised clustering with fast filtering for feature selection. As Figure 2 shows, the method uses the maximum mutual information to analyze the correlation between features and their class while exploring the redundancy between the features of different clusters using symmetric uncertainty.

(1) Intraclass Symmetric Uncertainty. The mutual information (MI) of two random variables measures mutual dependence. Intuitively, mutual information measures the degree to which the uncertainty of another variable is reduced when one variable is known. The mutual information $I(x; y)$ between two random variables is calculated as follows:

$$\begin{aligned} &I(X;Y) = H(X) - H(X|Y), \\ &H(X) = -\sum_x p(x)\log p(x), H(X|Y) = H(X,Y) - H(Y), \end{aligned} \tag{9}$$

where $p(x)$ is the probability mass function of X and $H(x, y)$ is the joint entropy of the two random variables X and Y.

Symmetric uncertainty is the standardized mutual information that allows information shared between random variables to be compared with each other. The symmetric uncertainty is calculated by the following formula:

$$SU(X,Y;C) = 2.0 \times \frac{I(X,Y;C)}{H(X,Y) + H(C)}. \tag{10}$$

The value of $SU(X, Y; C)$ varies from 0 to 1. If it is closer to 1, it means that the class label C is more relevant after combining the features of X and Y. Accordingly, it is determined which feature is added to the candidate feature subset by evaluating the cumulative joint symmetric uncertainty between the candidate features and all the already selected feature subsets. First, the sum of the joint symmetric uncertainty of each feature in the cluster and all features in the selected feature subset is calculated. The average value is used as the standard SU of the features in the cluster. Then, the features in the cluster with the largest cumulative SU are added to the selected feature subset in the descending order. Finally, the joint symmetric uncertainty of features f_i and f_j with class label C is assumed to be greater than the sum of their respective symmetric uncertainties with class labels; i.e.,

$$SU(f_1, f_2; C) > SU(f_1, C) + SU(f_2, C). \tag{11}$$

Then, it can be assumed that the features f_i and f_j can be combined to obtain a greater correlation with the class label. Any two features in the set S of related features that satisfy the equation will be added to the initially empty list to obtain the feature pair list (FPL). Each element in the FPL is a feature pair that satisfies the formula. Finally, the FPL is sorted in the descending order according to the joint symmetric uncertainty of the feature pair and the class label C. For the feature pairs containing the same features in the list FPL, the one with the lowest joint symmetric uncertainty with the class label C is deleted.

(2) Interclass Maximum Information Coefficient. Existing methods still find it challenging to describe many nonlinear relationships between features. Hence, in 2011, the study in [36] proposed a new information theory-based metric, namely, the maximum information coefficient which measures linear and nonlinear relationships between variables in massive data. Meanwhile, it allows extensive mining of nonfunctional dependencies between functions. Similarly, the relationship between different cluster features is calculated by the maximum information coefficient in our work, because it not only identifies potentially interesting relationships but is also independent of their form.

$$\mathrm{MIC}(x, y) = \max_{|X||Y|<B} \frac{\mathrm{Max}(I(X,Y))}{\log_2(\min(|X|, |Y|))}. \tag{12}$$

To use the maximum information coefficient to evaluate the correlation between features, a feature set $F = \{f_1, f_2, f_3, \ldots f_k\}$ consisting of n samples is given, where k is the number of features.

The correlation between any two classes of features f_i and f_j is recorded as $\mathrm{MIC}(f_i, f_j)$. The larger the value of $\mathrm{MIC}(f_i, f_j)$ is, the stronger the redundancy and substitution between features f_i and f_i will be. Ideally, the value of $\mathrm{MIC}(f_i, f_j)$ is 0, which means that the feature f_i and feature f_i are independent of each other. Therefore, the definition of redundant features is as follows:

Definition 1. (Information redundant features). Feature f_i is redundant, if there exists another feature f_j, s.t. $\mathrm{MIC}(f_j, C) > \mathrm{MIC}(f_i, C)$ and $\mathrm{MIC}(f_j, f_i) > \mathrm{MIC}(f_i, C)$.

```
Input: feature clusters
Output: sorted feature list
(1) Begin
(2) for all (cluster ∈ C) do
(3) for all (feature ∈ cluster) do
(4) Scores = Calculate SU (feature, category)
(5) Scores_su = Sort (scores)
(6) end for
(7) end for
(8) for all (cluster_x ∈ C) do
(9) for (cluster_y ∈ C) do
(10) if cluster_x ≠ cluster_y then
(11) Scores_mic = MIC (cluster_x, cluster_y)
(12) end if
(13) end for
(14) end for
(15) features = Get N features (scores)
(16) for all (f ∈ features) do
(17) list_mic = Scores_mic · get (f)
(18) KV_fmic · add (f, list_mic)
(19) end for
(20) f_res = SUMIC (Scores_fsu, KV_fmic)
```

Algorithm 1: DP-SUMIC feature selection.

Definition 2. (Information dominant criterion). Feature f_j will be kept, if it has the maximum information relevance $\mathrm{MIC}(f_j, C)$ to the target variable C in the candidate feature subset and is not redundant with the features already selected.

To calculate the redundancy of features between classes and to reorder them, we use the best-first search strategy and the maximum information coefficient. First, the maximum information coefficient of the features f_{C_1} in cluster C_1 and the features f_{C_2} in other clusters C_2 are calculated in turn. Then, this value is sorted in ascending order, and the maximum value is taken and saved to the set. A higher value of $\mathrm{MIC}(f_{C_1}, f_{C_2})$ indicates stronger redundancy and substitutability between f_{C_1} and f_{C_2}. Finally, the features with MIC > threshold2 are removed from the feature subset. We set the threshold value to 0.8.

3.2.3. The Details of DP-SUMIC. To filter the optimal feature subset and improve the classification accuracy, we first use a density-based unsupervised clustering method to process the features. For further effective feature correlation analysis and redundancy control, we employ SU and MIC to measure the correlation and joint effect between clustering features. First, a representative subset of features is selected by calculating the SU of intra-class features. Then, the redundancy relationship between interclass feature subsets is calculated based on the MIC. Consequently, we filter out the subset of highly discriminative features without containing redundant information. In summary, our proposed feature selection algorithm is based on unsupervised clustering and can handle both discrete and continuous features. Using SU and MIC, potentially interesting relationships between the attributes of two clusters can be identified. In the density peak clustering algorithm, we only operate on sampled data points, which improves the time and storage efficiency of the algorithm. The algorithm is shown in Algorithm 1.

3.3. The Improved NSA Algorithm Based on Hybrid Partitioning and GWO Optimization. To improve the detection efficiency, we carry out hybrid partitioning of the feature space before detector generation. For the generation method, we propose a boundary detector generation method based on self-sample clustering and a nonboundary detector generation method based on GWO optimization. Furthermore, we introduce a new adaptation function based on the minimum overlap rate and maximum coverage. Our method is capable of adjusting the detector generation position, radius, etc., based on the judgment of the generation state during the detector generation process. This significantly improves the anomaly detection rate. Overall, the MDGWO-NSA-based network anomaly detection method can be divided into three steps: (1) feature space division; (2) identification of boundary grids based on boundary self-samples and generation of specific candidate boundary detections based on boundary grids; and (3) nonboundary detector generation based on self-sample clustering and GWO optimization.

3.3.1. The Division of the Feature Space. The new feature space-partitioning method that focuses on the mixture of planes and dimensions is presented in this section. As shown in Figure 3, the feature space is divided into a grid according to the hierarchical partitioning of the hyperplane, while the detector can be localized based on the grid.

The first step of the algorithm is to set the subgrid in each dimensional range of (r, t). If the antigen density in the

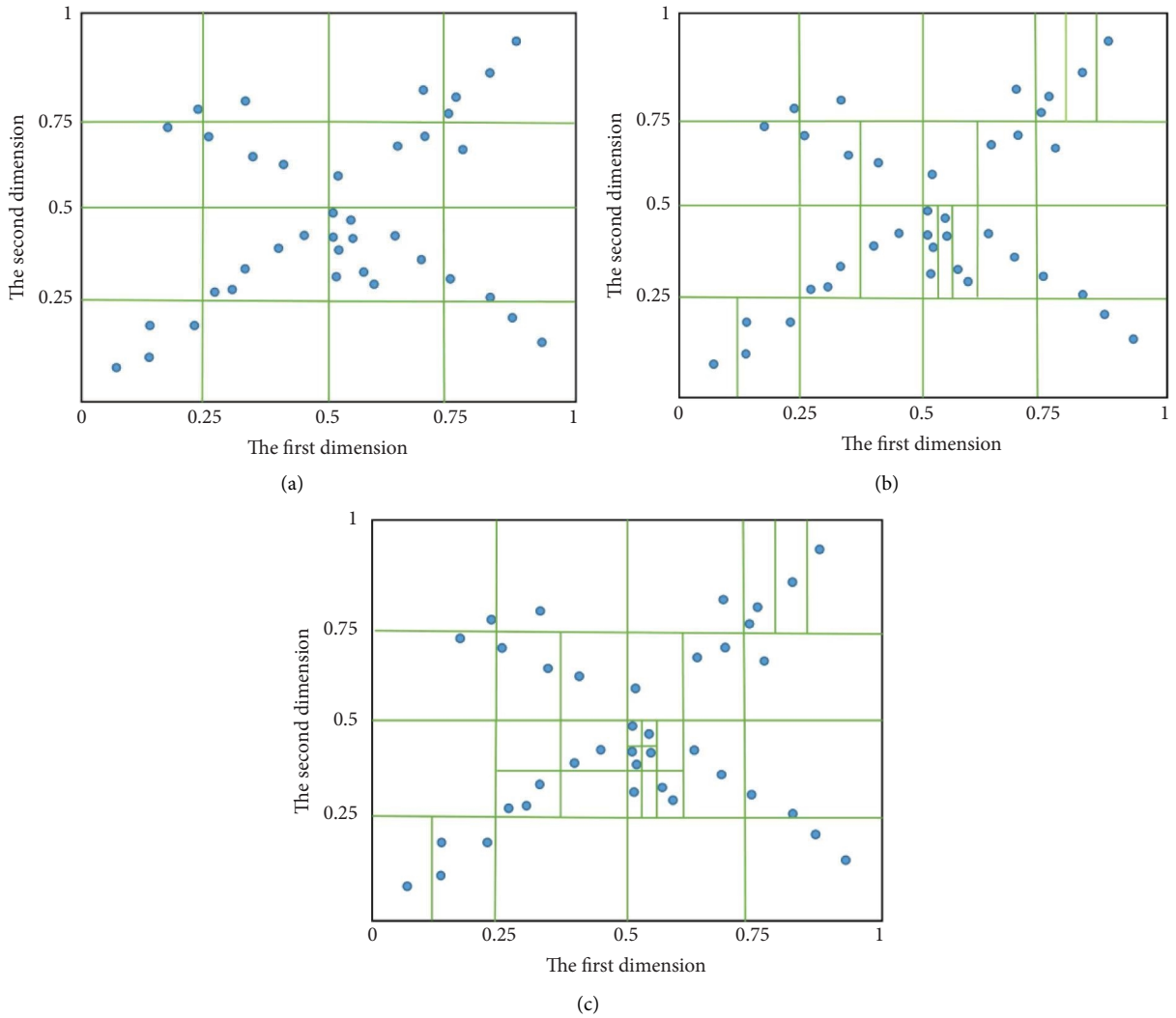

Figure 3: Division of 2-dimensional feature space. Blue points represent non-self-antigens, (a) feature space divided based on plane segmentation strategy, (b) the plane divided by dimension segmentation strategy, and (c) using both the dimension and plane segmentation strategy.

subgrid reaches a threshold, we create a new d-dimensional hyperplane for each dimensional subgrid according to the following equation:

$$L_d = \frac{r + t}{2}, \tag{13}$$

where r and t is the maximum and minimum value of the dimension, respectively.

Thus, we obtain a two-dimensional subgrid where d is the dimension of the data. The partitioning of the subgrid stops, when the density of self-antigens in the subgrid falls below a threshold value. The number of stratifications is calculated based on the density of self-antigens in existing subgrids, and the proportion of the grid containing antigens is calculated using the following equation:

$$P = \frac{D_{\text{sample}}}{D_{\text{grid}}}, \tag{14}$$

where D_{sample} is the number of grids with sample in the n th layer and D_{grid} is the total number of grids in the feature space divided in the n th layer, calculated by the following equation:

$$D_{\text{grid}} = \left(2^d\right)^{n-1}, \tag{15}$$

where d is the number of data dimensions and n is the number of layers divided.

According to (14) and (15), P is negatively correlated with the effect of segmentation layer n. Thus, from equation (15), when the segmentation layer n tends to infinity, the proportion of non-self-antigen subgrids tends to 1. Although the whole non-self-space is covered, many segmentation layers are generated, which seriously affects the efficiency of the algorithm. If $P < 1$, there will be some holes, and we can choose different ratios according to different situations.

When the sample density of the hyperplane is divided n times and is still more significant than the threshold k, the division method is changed. Continuous dimensional segmentation is performed in this hyperplane along the i th dimension, divided by the $(n-1)$ th dimensional hyperplane. All $(n-1)$ th dimensional hyperplanes are composed of n-dimensional grids, each of which has a pointer to its antigen array located in that grid. The division number of each dimension in the i th hyperplane is calculated based on the density of the sample in the dimension. The new hyperplane in equation (16) is generated and the k-dimension is divided.

$$d_k = \frac{m+n}{2}. \tag{16}$$

Sometimes, most samples along one or more dimensions converge to a small range of values. Therefore, to avoid over segmentation along these dimensions, we add the condition that the range should not be further segmented if $m-n$ is less than λ, or if the density within that grid is less than threshold b.

3.3.2. Generation of Specific Candidate Boundary Detectors. The boundary is diversity in real applications. To avoid randomly generated detectors that produce many holes in the boundary region and lead to degraded detection rates, we classify detectors into boundary detectors and nonboundary detectors away from the self-set. We generate specific candidate boundary detectors at the boundary grid by a hierarchical localization method, which can alleviate the Type-2 holes effectively and improve the detection rate. We define the boundary grid and its location in the following.

Definition 3. (Boundary grid). When a grid is empty and at least one of its neighboring grids is nonempty, it is a boundary grid. A nonempty grid in the adjacent grid of a boundary grid is a boundary sample, which is the outermost sample of the self-space.

Definition 4. (Location information of the boundary grid). The location information of the boundary grid is used to record the properties of each neighboring grid. The location information of the boundary grid also determines the location information of the detector generated by the boundary grid, and the center of the detector is the boundary grid center.

In NSA, detector coverage is a prerequisite to obtaining high-quality excellent classification results. The holes that appear during detector generation are challenging to go through. In fact, much work has focused on hole repair during random detector generation while ignoring boundary detectors caused by boundary samples. These detectors that are close to the self-samples are prone to errors and significantly reduce the accuracy. Therefore, this paper addresses the boundary detector hole problem by first dividing the feature space into several grids and then deriving the boundary grids. Finally, it is combined with hierarchical partitioning to generate specific candidate boundary detectors near the self-samples. Our proposed method can improve the detection efficiency to some extent. The hybrid division-based boundary detector (HDBD) generation method is shown in Algorithm 2.

3.3.3. Self-Set Clustering. In the n-dimensional feature space, similar self-data are grouped into the same cluster and cluster centers are used to match candidate detectors representing the cluster members to reduce the number of distance calculations. In the detector generation process away from the self-samples, the radius of the detector has more flexibility relative to the boundary detector and the larger radius can cover more non-self-sets.

The traditional density-based clustering algorithm DBSCAN requires the specification of two basic parameters: the neighborhood radius and the minimum data point threshold. The determination of these parameters has a significant impact on the clustering results, while the effective parameter selection determination methods are inadequate. Therefore, we adaptively select the local neighborhood radius for clustering by determining the peak density points, which simplifies the parameter selection process. Typically, the density peak points have two characteristics: higher local density and a greater distance between nondensity peak points in the same cluster and other clusters. These two characteristics of each data point can be calculated to find the peak density points. The selection criteria of the cluster center are shown in equation (17). The data points are selected based on the minimum number of points in the neighborhood and the local density until no available clustering centers exist.

Definition 5. (Local density of data points). Given the minimum number of points (min Pts), the local density of data point ρ_p is defined as follows:

$$\rho_p = \frac{\min \mathrm{Pts}}{\max_{x \in N_p} \mathrm{dist}(p,x)}, \tag{17}$$

where min Pts is the set of data points that are closest to p, N_p is the neighborhood of p, and $\mathrm{dist}(p,x)$ is the Euclidean distance between p and x. From equation (17), $\max \mathrm{dist}(p,x)$ is the local radius of p. When the local density of $\rho_{x \in Np}$ is larger, the minimum number of points can be satisfied at a smaller local radius. When the local density of p is maximum, p is most likely to be a peak density point.

The clustering radius is an essential parameter for obtaining the radius of the detector in the detector tolerance phase. We find the cluster radius based on the distance from the cluster object to the center, as shown in Equation (18).

Definition 6. (Cluster radius). The radius r_k of the k th cluster is calculated as follows:

$$r_k = \frac{1}{2}\max_{i=1,2\ldots n_i}\{\mathrm{dist}(x_i, C_k)\}, \tag{18}$$

where x_i denotes the i th data object; C_j denotes the j th cluster center; and n_k denotes the number of samples contained in the k th cluster.

Input: S: Self-sets M: max grid number
Output: D_b: Border detectors
(1) Grid = Initialize the grid space
(2) while $i \leq M_{\text{grid}}$ do
(3) All space = Divide (grid)
(4) $\text{Grid}_{\text{self}}$ = calculate self-sets location (S)
(5) for All ($L_{\text{self}} \in \text{Grid}_{\text{self}}$) do
(6) if $L_{\text{self}} \in \text{Space}_{\text{all}}$ then
(7) $\text{Space}_{\text{all}} \cdot \text{delete}(L_{\text{self}})$
(8) end if
(9) end for
(10) Grid = all space
(11) end while
(12) $\text{Grid}_{\text{self}}$ = unique ($\text{Grid}_{\text{self}}$)
(13) for all (border $\in$ Grid) do
(14) Dis_r = Calculate the distance (border, $\text{Grid}_{\text{self}}$)
(15) D = (border, Dis_r)
(16) end for

Algorithm 2: HDBD generation.

3.3.4. Optimization of the NSA using GWO. The non-boundary detectors are generated based on the clustering centers obtained in Section 3.3.3. The GWO optimized detector generation strategy is adopted to obtain the optimal detector position and radius. The fitness function is defined as the sum of the maximum coverage rate and the minimum overlap rate. When the detector is generated far from the boundary, the distance from the cluster center can be directly calculated to find the best detector radius. Therefore, the method proposed in this paper dramatically reduces the comparison time with self-samples in the detector generation process and effectively improves the efficiency of detector generation. The detector generation process is shown in Figure 4.

(1) The Improved GWO. The gray wolf algorithm is a new metaheuristic swarm intelligence algorithm proposed in [40] in 2014 that imitates the predatory behavior of gray wolves. The artificial gray wolves in the population have a social hierarchy and a system of division of tasks. Suppose that the α wolf is the optimal global solution, the β wolf and δ wolf are the global second and third optimal solutions, respectively, and the remaining artificial wolves are ω wolves. Under the guidance of α, β, and δ, the ω wolf follows these three to start hunting behavior (search and optimization). The ω wolf follows the best position of the α, β, and δ wolves and updates its position to gradually approach the prey by

$$X(t+1) = X_p(t) - A \cdot D, \tag{19}$$

$$D = \left| C \cdot X_p(t) - X(t) \right|, \tag{20}$$

where t is the number of iterations and $A = 2a \cdot r_1 - a$ and $C = 2 \cdot r_2$ are vector coefficients. A is the convergence factor, which is used to balance the global search and local search. C is used to simulate the effect of the natural world. The value of a decreases linearly from 2 to 0 as the number of iterations increases, and r_1 and r_2 are random numbers in [0, 1]. When $1 < A$ or $A < -1$, artificial gray wolves are scattered and search for the prey locally at the early stage of hunting. With the depth of the exploration in the superiority search, when $-1 \leq a < 1$, the artificial gray wolves attack the prey. This mode of attack can speed up the late convergence rate.

We find that individuals depend on the optimal solution α wolf to update their positions. From Equations (19) and (20), we can conclude that the gray wolf optimization algorithm can be improved by using an optimal individual preservation-based strategy during the search in the late evolutionary stage. Due to a large number of individuals clustered around α wolves, the diversity of wolf search is lost. Especially when α wolves have difficulty in finding the optimal global solution and fall into the optimal local solution, it will lead to premature convergence and reduce the search accuracy. Inspired by the differential evolutionary algorithm, we design a new weight-based position change strategy to solve the problem by using the difference in information among individuals. It can be expressed as follows:

$$\begin{gathered}\overrightarrow{X}(t+1) = \frac{cr_3\left(\omega_1^* \overrightarrow{x_1}(t) - \overrightarrow{x_i}(t) + \omega_2^* \overrightarrow{x_2}(t) - \omega_3^* \overrightarrow{x_3}(t)\right)}{\left(\omega_1 + \omega_2 + \omega_3\right)}, \\ w_1 = A_1 * C_1,\ w_2 = A_2 * C_2,\ w_3 = A_3 * C_3,\end{gathered} \tag{21}$$

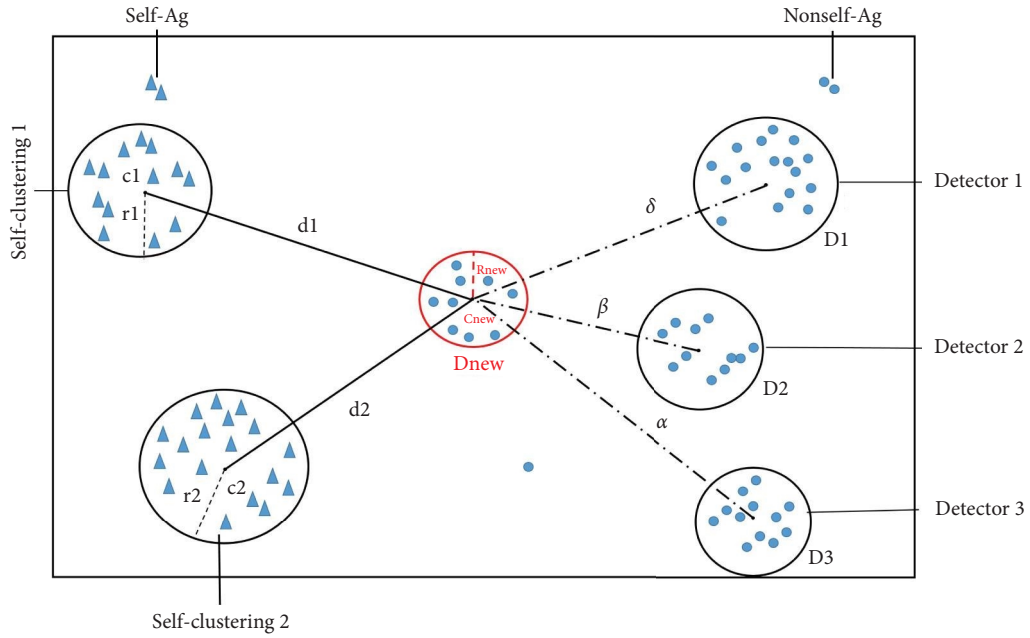

Figure 4: Nonboundary detector generation. The small circle represents self-antigen, and the small triangle represents non-self-antigen. D1–D3 represent the original detector, D_{new} represents the new detector, C_{new} represents the center of a new detector, and R_{new} represents the radius of a new detector. d1 and d2 represent the distance from the new detector to cluster center of the self-sample. r1 and r2 represent cluster radius. c1 and c2 represent the cluster center.

where $c, r_3 \in [0, 1]$, and they are random coefficient.

From the previous equation, it can be seen that the proposed position variation strategy based on weighted distance fully considers the information of individuals in the wolf population and solves the problem of premature convergence in complex multiple calculations. The variability between wolves β and δ is borrowed to improve the diversity of the population search. Meanwhile, by using the change information between the best wolf individual α and the current gray wolf individual, the ability of the algorithm to jump out of the locally optimal solution is enhanced.

It is worth noticing that a specific strategy is needed to maximize success. Therefore, we optimize the hunting mechanism in two directions: global search and local search. The process of finding the optimal detector location is regarded as a global search, and the process of computing the optimal detector radius is a local search. In the global search phase, individuals in the search population exchange information about the domain (under the possible search conditions). In the local search phase, each individual relies on their own search capabilities. Thus, we can compare different locations within the domain and remote locations quickly.

(2) Fitness Function. The fitness function $I(D)$ is calculated by the sum of the maximum coverage rate and the minimum overlap rate of the detectors, as shown in Equations (22)–(24). It searches for normal data to train the model effectively. The radius of each detector should cover the entire grid space as much as possible to ensure maximum coverage.

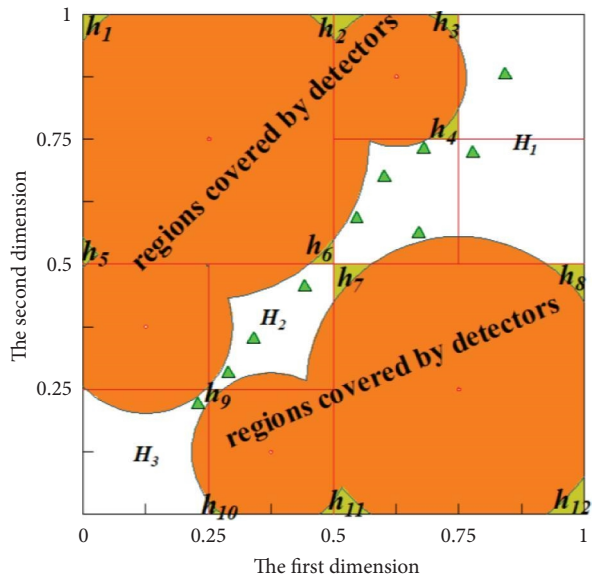

Figure 5: Examples of holes. h1 ~ h12 are Type-1 holes, and H1 ~ H3 are the Type-2 holes.

$$\text{Min Overlap}(D) = \min\left(N_{\text{overlap}}\left(d_i, d_j\right) + S_{\text{coverage}}\left(d_i, s_j\right)\right), \tag{22}$$

$$\text{Max Coverage}(D) = \max\left(N_{\text{coverage}}\left(d_i\right) + N_{\text{coverage}}\left(d_j\right)\right), \tag{23}$$

$$I(D) = \text{Max Coverage}(D) + \text{Min Overlap}(D), \tag{24}$$

where the overlap between the i th detector d_i and the j th detector d_j can be approximately defined as in

$$N_{\text{overlap}}\left(d_i, d_j\right) = \begin{cases} 0, & \text{if } P_1' \geq R_{(d,d)}, \\ \ln\left(1 + \dfrac{r_d^i + r_d^j - \left(\sqrt{\sum_{i,j=1}^{n}\left(P_d^i - P_d^j\right)^2}\right)/n}{r_d^i + r_d^j}\right), & \text{if } P_1' < R_{(d,d)}, \end{cases} \tag{25}$$

where $P_1' = (\sqrt{\sum_{i.j=1}^{n}(P_d^i - P_d^j)^2})/n$ and P_1' represents the average distance between the ith detector and the jth detector center in the n-dimensional plane. $R_{(d,d)} = r_d^i + r_d^j$, $R_{(d,d)}$ represents the sum of the radius of the detectors.

The overlap between the detector d_i and the self-sample s_j is defined as follows [41]:

$$S_{\text{covering}}\left(d_i, s_j\right) = \begin{cases} 0, & \text{if } P_2' \geq R_{(d,s)}, \\ \ln\left(1 + \dfrac{r_d^i + r_s^j - \left(\sqrt{\sum_{i,j=1}^{n}\left(P_d^i - P_s^j\right)^2}\right)/n}{r_i + r_j}\right), & \text{if } P_2' < R_{(d,s)}, \end{cases} \tag{26}$$

where $P_2' = (\sqrt{\sum_{i,j=1}^{n}(P_d^i - P_s^j)^2})/n$ and P_2' represents the average distance between the ith detector and the jth self-sample centroid in the n-dimensional plane. $R_{(d,s)} = r_d^i + r_s^j$, $R_{(d,s)}$ represents the sum of detector radius and self-sets radius.

$$N_{\text{covering}}(D) = \sum_{d_i}\sum_{d_j} N_{\text{covering}}\left(d_i, d_j\right). \tag{27}$$

3.4. Hole Repair. In this section, a novel method for repairing holes in the NSA is proposed. Detectors and antigens are defined in hyperspheres which results in missed detection of non-self-antigens due to holes between hypersphere boundaries. In the MDGWO-NSA, detectors are generated separately in the grid. The recovered area is restricted to the internal space of the grid, thus effectively reducing the redundancy of detectors. As shown in Figure 5, the hole between detectors that wastes resources due to overlapping radius is a Type-1 hole. The hole that exists between detectors and self-region due to insufficient coverage becomes the Type-2 hole. The precondition for the existence of Type-2 holes is that the detector radius is within the grid boundary to avoid covering self-antigens in adjacent grids. Adaptive weights with dynamic detector positions by the method proposed in Section 3.3 effectively mitigate Type-1 holes. Since there is no clear boundary between self-space and non-self-space in NSA, a specific candidate detector based on the boundary grid is proposed in this paper. This alleviates the Type-2 hole issue to an extent. However, it increases the undetected regions in non-self-space simultaneously.

There are three types of hole repair methods as shown in Figure 6. For the existing holes, we first identify the undiscovered gaps in the non-self-space by calculating the distance $\text{Dist}(i, j)$ and density $\text{Dens}(i, j)$ between the current detector and the neighboring detectors in the following equations:

$$\text{Dist}(i, j) = \min\left(\text{dist}\left(m_i, m_j\right) - \left(r_i + r_j\right)\right), \tag{28}$$

$$\text{Dens}(i, j) = \text{Dens}_i\left(\text{grid}_i - \rho_i\right) + \text{Dens}_j\left(\text{grid}_j - \rho_j\right), \tag{29}$$

where m_i and m_j are the detector centers and r_i and r_j are the detector radius. grid_i represents the non-self-set density in the grid, and ρ_i represents the non-self-set density of the detector.

Method 1. If the non-self-sample density of holes between neighboring detectors is greater than the threshold a and the distance between detectors is greater than $r_{\min}$, the neighboring detectors are merged directly.

$$\text{if } \text{Dens}(i, j) > a \text{ and } r_{\max} > \text{dist}(i, j) > r_{\min}, \tag{30}$$

$$M_{\text{new}} = \frac{1}{2}\left[m_i + m_j\right],$$

$$R_{\text{new}} = \frac{1}{2}\left[\text{dist}(i, j) + \left(r_i + r_j\right)\right], \quad \forall i = 1, 2, \ldots, n \text{ and } \forall j_{j \neq i} = 1, 2, \ldots, n. \tag{31}$$

Method 2. If the non-self-sample density is less than the threshold a and the distance between detectors is greater than $r_{\min}$, a new detector is generated to fill the uncovered area.

$$\text{if Dens}(i, j) < a \text{ and dist}(i, j) > r_{\max},$$
$$M_{\text{new}} = \frac{1}{2}\left|m_i - m_j\right|, \tag{32}$$

$$R_{\text{new}} = \frac{1}{2}\left[\text{dist}(i, j) - \left(r_i + r_j\right)\right], \quad \forall i = 1, 2, \ldots, n \text{ and } \forall j_{j \neq i} = 1, 2, \ldots, n. \tag{33}$$

Method 3. If the non-self-sample density is less than the threshold a and the radius between detectors is less than $r_{\min}$, the boundary detector radius is increased.

$$\text{if Dens}(i, j) < a \text{ and dist}(i, j) < r_{\min},$$
$$M_{\text{new}} = \max\left(m_i, m_j\right), \tag{34}$$

$$R = \frac{2 + e^{-z}}{1 + e^{-z}}\, z \in \forall R, \quad \forall i = 1, 2, \ldots, n \text{ and } \forall j_{j \neq i} = 1, 2, \ldots, n, \tag{35}$$

where r_i and r_j are radii of M_i and M_k, respectively.

3.5. The Details of the MDGWO-NSA. In the MDGWO-NSA, the feature space is gridded based on the hybrid method. First, the feature space is divided according to the plane division method based on the sample density. After the plane is split n times, if the antigen density in the grid within the subplane is still more significant than the threshold a, then the dimension division is adopted. Until the sample density in this grid is less than the threshold b and the grid radius is less than c, the feature space stops dividing. Based on the divided grid, the boundary self-sample and the boundary grid generate a specific candidate boundary detector. Then, the generation of nonboundary detectors is optimized based on the improved GWO algorithm dynamically. We selected three boundary detectors as the initial $\alpha, \beta, and \delta$ wolves randomly. According to the randomly selected initial-boundary detector position with the fitness function, the location of the current optimal detector is gradually calculated. According to equations (30), (32), and (34), the fitness function and the weight-based distance obtain the global optimal detector position. Finally, the radius of the current optimal detector is obtained by calculating the Euclidean distance from the nearest cluster center, and a non-boundary detector far away from the self-sample is generated. The detector radius is the difference between the distance from the center of the detector to the nearest cluster center minus the class radius. Moreover, the detector radius is adaptively adjusted by equations (30)–(35) in detector generation and hole repair. The optimized MDGWO-NSA detector generation algorithm is shown in Algorithm 3.

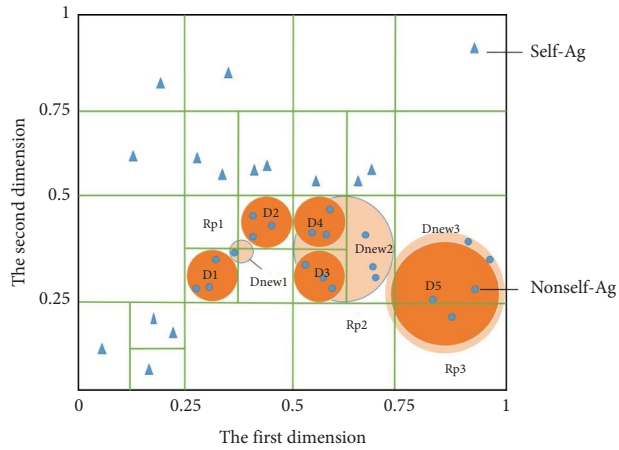

FIGURE 6: Hole repair. Rp1 is method-1 repair, Rp2 is method-2 repair, and Rp3 is method-3 repair. The dark yellow area represents the original detectors D1, D2, D3, D4, and D5, respectively. The light yellow area represents the new detectors Dnew1, Dnew2, and Dnew3, respectively.

3.6. Analysis of Time Complexity

Theorem 1. *The MDGWO-NSA time complexity is* $O((|D| \cdot N_c)/(\rho \cdot P))$, N_c *is the number of self-sample clusters,* $|D|$ *is the number of detectors,* ρ *is the average density of self-samples, and* P *represents the coverage of detectors.*

Proof. In step 1, the complexity of the data normalization process is $T_1 = O(d \cdot N_s)$. In step 2, the time complexity of performing the hybrid feature space partitioning is $T_2 = O(n \cdot d \cdot N_s \cdot \rho)$. n represents the layer division of the feature space, the sample dimension d, the number of samples N_s, and the average density among sample individuals ρ. In step 3, the time complexity of clustering the self-samples and generating the boundary detector is $T_3 = O((d \cdot N_c)/(l \cdot \rho))$, T_3 is mainly determined by the number of cluster classes N_c into which the self-samples are clustered, the minimum radius r_{nd} of the boundary detector, and the average density ρ of the self-samples. In step 4, the time complexity of generating the nonboundary detector is optimized by GWO: $T_4 = O((N_c \cdot D_c)/\rho)$. T_4 is mainly determined by the number of clusters N_c after self-sample clustering, the average radius of clusters D_c, and the average density of self-samples ρ. In step 5, the time complexity of hole repair is $T_5 = O(|D| \cdot N_c/(\rho \cdot P))$, the number of detectors is $|D|$, and the important basis for determining whether a detector needs hole repair is the coverage P of the detector.

$$O\left((d \cdot N_s) + (n \cdot d \cdot N_s \cdot \rho) + \frac{d \cdot N_c}{r_{\mathrm{nd}} \cdot \rho} + \frac{N_c \cdot D_c}{\rho} + \frac{|D| \cdot N_c}{\rho \cdot P}\right) = O\left((d \cdot N_s) + (n \cdot d \cdot N_s \cdot \rho) + \left(\frac{d \cdot N_c + N_c \cdot D_c \cdot l}{r_{\mathrm{nd}} \cdot \rho}\right) + \frac{|D| \cdot N_c}{\rho \cdot P}\right)$$
$$\approx O\left(\frac{|D| \cdot N_c}{\rho \cdot P}\right), \tag{36}$$

where P is the detector coverage and $|D| \cdot N_c/(\rho \cdot P) \gg (d \cdot N_s) + (n \cdot d \cdot N_s \cdot \rho)$.

The time complexity of NNSA, RNSA, and V-Detector is exponentially related to the number of self-antigens N_s. For the GF-RNSA and HD-NSA, $\overline{S}$ and C are less than the full self-sets N_s, which indicates that they are more efficient than the NNSA, RNSA, and V-Detector. For MDGWO-NSA, N_c is the number of self-sample clusters, $|D|$ is the number of detectors, ρ is the average density of self-samples, and P represents the coverage of detectors. Obviously, $(\rho \cdot P)$ is larger than $(1 - P_m)^{N_s}$ and $(1 - \overline{P})^S$ due to the specific candidate boundary detectors generated and the unbounded detectors with adaptive adjustment. Moreover, it has similar time complexity compared to the HD-NSA. However, we can mitigate the boundary diversity problem by dividing the grid to generate specific boundary detectors. Since our algorithm is composed of multiple optimization steps such as GWO, it can cope well with complex non-self-set situations. In the traditional detector generation process, the detector is generated singly and randomly in the face of complex situations, leading to the problem that generation quality cannot be guaranteed. In contrast, our detector in the generation process can give a more appropriate detector generation strategy according to different situations. Secondly, it can adjust the detector generation position size, etc., based on the computational fitness function. Finally, a new hole repair strategy is given after the generation of the detector. Thus, our method guarantees the quality of detector generation in several aspects and has a higher detection rate for complex non-self-sets. In conclusion, the MDGWO-NSA is more efficient than other classical NSAs (see Table1). □

4. Empirical Study and Dataset Analysis

4.1. Datasets and Assessment Metrics. This section verifies the performance of the MDGWO-NSA through experiments. First, we compare the feature selection method proposed in this paper with other methods under the same classifier. Then, we compare the overall performance of classical deep learning and machine learning algorithms for network anomaly detection on the NSL-KDD [45], UNSW-NB15 [46], and CICIDS-2017 [47] datasets. Finally, we validate the effectiveness of the improved NSA algorithm on the UCI dataset [48].

NSL-KDD contains 41 features, which are classified according to different modes. NSL-KDD includes two datasets (KDD Train + .txt, KDDTrain + 20.txt) and two test datasets (KDD Test + .txt, KDD Test − 21.txt). We use this dataset to train and test our proposed model. The number of records contained in the KDD Train + dataset and KDD Test + dataset are 126620 and 22850, respectively. UNSW-NB15 contains 49 features, including host-based flow and packet headers, which cover the comprehensive characteristics of network traffic. The training set has 175341 records. The test set has 82332 records. It is more suitable for intrusion detection system research. Therefore, UNSW-NB15 is used to verify the efficiency of our proposed algorithm, where the normal records in the training set are marked as "self-sets." CICIDS 2017 is the latest dataset in network anomaly detection, which has new attacks. It contains 225,745 packets with more than 80 network traffic features. Moreover, the distribution of data classes is restricted, and most traffic data are malicious. In addition, the types of attacks are unknown, which is advantageous for us to evaluate the detection capability of our algorithm against new unknown network attacks. In this paper, different types of attacks are combined under the "anomaly" category and considered as "non-self," while benign types are considered "self-sets." In addition, the ability of the model to distinguish attacks was not evaluated. Furthermore, the UCI dataset is used to verify the progress achieved by the proposed algorithm. The iris, skin, and abalone datasets are currently the most popular pattern recognition datasets in the UCI database and are widely used in NSA-related research. The following metrics are used in the experiments: Recall, Accuracy, Precision, and F1-score [32] as shown in Table 2.

4.2. Experimental Results and Analysis. To verify the efficiency of the filter-based unsupervised density clustering feature selection method proposed in this paper, we compare it with representative feature selection methods and rank the features according to our proposed method. To validate the overall performance of our proposed method in network anomaly detection, different classifiers and MDGWO-NSA are compared on the same dataset, which includes NSL-KDD, CICIDS-2017, and UNSW-NB15. Moreover, to verify the effectiveness of the improved NSA algorithm in this paper, the MDGWO-NSA is compared with the traditional improved NSA algorithm on the UCI dataset. We have summarized the detailed experimental results, and the best results are displayed in black font. The verification process uses the training and testing sets of each benchmark dataset. We gradually change the size to explore the scalability of the algorithm. In addition, each experimental result described in the manuscript represents an average of 20 runs.

4.2.1. Comparison Feature Selection. To compare the relevance of our selected feature subset with other known algorithms, we apply the information gain (IG), the

Input: border detectors
Output: Detectors
(1) Initialization: Generates detectors at each vertex in space, D_v;
(2) Preliminary population order, S_{dv}; Maximum search times, M_i;
(3) Search agents number, S_n; Optimal scores of the search agents number, O_s;
(4) Self property number, p_num;
(5) wolfα; wolfβ; wolfδ;
(6) Radius ofα, r_a; Radius ofβ, r_b; Radius of δ, r_c
(7) while rate ≤ 0.95 do
(8) $\text{Score}_{\text{best}} = 0$
(9) $\text{Count}_t = 0$
(10) while $l \leq M_i$ do
(11) $a = 2 - l^* ((2)/M_i)$
(12) for $(i \in S_n)$ do
(13) for $(j \in p\_num)$ do
(14) $A1 = 2^* a^* \text{Random} - a$
(15) $A2 = 2^* a^* \text{Random} - a$
(16) $A3 = 2^* a^* \text{Random}^* a$
(17) $X1 = \alpha[j] - A1^* r_a$
(18) $X2 = \beta[j] - A2^* r_b$
(19) $X3 = \delta[j] - A3^* r_c$
(20) $\text{Position}[i, j] = (X1 + X2 + X3)/3$
(21) end for
(22) $\text{Position}[i] = \text{Calculate radius } (\text{Position}[i])$
(23) $\text{Score} = \text{Calculate position score } (\text{Position}[i])$
(24) if $\text{Score} \geq \text{Score}_{\text{best}}$ then
(25) $\text{Position}_{\text{best}} = \text{Position}[i]$
(26) $\text{Score}_{\text{best}} = \text{score}$
(27) **else**
(28) $\text{Count}_t = \text{Count}_t + 1$
(29) end if
(30) end for
(31) if $\text{Count}_t \geq 40$ then
(32) $\text{Detectors.append}(\text{Position}_{\text{best}})$
(33) break
(34) end if
(35) $\text{Rate} = \text{Calculate detection rate } (\text{Detectors})$
(36) end while
(37) end while

Algorithm 3: MDGWO-NSA.

correlation attribute evaluation algorithm (CAE), and a feature selection method based on the coefficient of variation (CVFS). The ranked attributes and the best subset of features selected by our feature selection algorithm DP-SUMIC are reported in Table 3. Similarly, common indicators are used in this comparative study which include training time and classification accuracy. Table 4 summarizes the average performance of our model compared to other feature selection methods.

From the features we selected in Table 3, we find that the number of bytes and packets sent by the source/destination are probably high and often choose the flow duration. That is, the number of source/destination messages is abnormal. The number of source/destination bytes and traffic duration are abnormal, resulting in a high probability of abnormal traffic. Therefore, the results can better guide IDS developers in selecting the features needed for intrusion detection.

Table 1: Time complexity of NSA algorithm.

Algorithm	Time complexity of NSA
NNSA ([5])	$O(-\ln P_f/P_m \cdot (1 - P_m)^{N_s} \cdot N_s)$
RNSA ([42])	$O(\lvert D\rvert \cdot N_s/P_m \cdot (1 - P_m)^{N_s})$
V-detector ([43])	$O(\lvert D\rvert \cdot N_s/P_m \cdot (1 - P_m)^{N_s})$
GF-NSA ([29])	$O(\lvert D\rvert \cdot \overline{S}/(1 - \overline{P})^{\overline{S}})$
HD-NSA ([44])	$O(C \cdot \lvert D\rvert/(1 - P\imath)^{N_s\prime})$
MDGWO-NSA	$O(\lvert D\rvert \cdot N_c/\rho \cdot P)$

We used the feature subset of Table 3 for classification, and the results are shown in Table 4. Compared to the other five feature selection methods on different datasets, DP-SUMIC has the shortest training time on NSL-KDD. DP-SUMIC is suboptimally better than ANOVA-F on the CICIDS-2017 and UNSW-NB15 datasets because ANOVA is a statistical-based method that ranks features by calculating the within-group and between-group variance ratios.

Table 2: Evaluation metrics.

Recall = $TP/(TP + FN)$
Precision = $TP/(TP + FP)$
Accuracy = $(TP + TN)/(TP + TN + FN + FP)$
DR = $TP/(TP + FN)$
FAR = $FP/(TN + FP)$
$F1 - \text{Score} = (2 \times \text{Precision} \times \text{Recall})/(\text{Recall} + \text{Precision})$
D_n: number of detectors
D_{TT}: detector training time
Test: testing time

Table 3: Selected features and rankings.

Dataset	Feature subset
NSL-KDD	src_bytes, flag, same_srv_rate, diff_srv_rate, service, dst_bytes, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_same_srv_rate, Serror_rate, srv_serror_rate, count, dst_host_srv_count, dst_host_diff_srv_rate, logged_in,dst_host_same_src_port_rate, dst_host_count, dst_host_count, protocol_type
UNSW-NB15	Smeansz, id, rate, sload, stcpb, dloss, sttl, dload, sjit, sintpkt, sloss, dintpkt, dpkts, dbytes, dbytes, swin
CICIDS-2017	sloss, dintpkt, dpkts, dbytes, dbytes, swin, pkt_len_va, fw_seg_avg, bw_win_byt, fin_cnt, fw_iat_avg, fw_byt_blk_avg, fl_byt_s, Bw_pkt_l_std, fw_seg_min, subfl_bw_byt, pkt_len_avg, fl_iat_min, pkt_len_std, fw_iat_max, bw_pkt_s, Bw_pkt_l_max, bw_iat_std, Fw_act_pkt

Although the training time of ANOVA-F is the shortest among all the compared algorithms, the classification accuracy is the lowest. In classification accuracy, DP-SUMIC outperforms other FS techniques in all conditions. This is due to the fact that we fully retain more useful feature subsets to maintain accuracy. In particular, in CICIDS-2017 and UNSW-NB15, the training time of DP-SUMIC is slightly longer than that of ANOVA-F, while the classification accuracy is much better. Moreover, in the case of data complexity, there is a lack of significant differences between the class means of the features. ANOVA-F eliminates all these features directly due to the lack of analysis on interfeature dependence and consistency. Considering the dataset's size and the algorithm's randomness, this slight time loss is acceptable, and our model training time can be effectively reduced on large datasets. In terms of classification accuracy, our mechanism achieves the best accuracy on these three datasets. Importantly, our selected features have low redundancy and high relevance, which can significantly improve network anomaly detection.

Overall, our proposed method is the most effective overall. However, the training time is slightly longer than that of ANOVA-F. Nevertheless, the accuracy of ANOVA-F is poor, and we should prefer feature selection methods with high accuracy in network anomaly detection when the training time is approximately equal.

4.2.2. Performance Comparison Experiments on Network Anomaly Detection Datasets. To verify the efficiency of the proposed algorithm in network anomaly detection, the performance of the MDGWO-NSA is evaluated by comparing it with ten baseline algorithms on the same dataset. It includes eight machine learning algorithms widely used for network anomaly detection: naive Bayes (NB), K-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT), random forest (RF), deep neural network (DNN), convolutional neural network (CNN), and long short-term memory (LSTM) [49-53]. In addition, we also include two representative artificial immune algorithms, RNSA and HD-NSA. These results come from the complete NSL-KDD dataset, the CICIDS-2017 dataset, and the UNSW-NB15 complete test set.

As shown in Table 5, MDGWO-NSA algorithms show better performance compared to other algorithms. In fact, under the tests of the NSL-KDD, CICIDS-2017, and UNSW-NB15 datasets, their average recall is 99.23%, 98.46%, and 95.73%, respectively. CNN is suboptimal because it is more suitable for processing image data and it uses a fixed convolutional kernel, which is not good for processing time-series-based network traffic data. The results of NB, DT, RF, KNN, DNN, and LSTM are similar and they are all worse than HD-NSA, RNSA, and MDGWO-NSA-2 (without DP-SUMIC). SVM is the algorithm with the lowest recall. This is due to the different probabilities of training and testing data in NSL-KDD and the uneven distribution of attacks, which makes the traditional machine learning algorithms vulnerable. In contrast, the generation and tolerance phases of the detector in the artificial immunity-based algorithm can easily handle anomaly detection in complex environments. In addition, our improved adaptive generation and tuning capability of the detector makes our algorithm higher than the HD-NSA and RNSA of the artificial immunity-based algorithm. Even when feature selection is removed, the efficiency is still roughly equal to that of the HD-NSA. Thus, the MDGWO-NSA is effective in detecting most attacks.

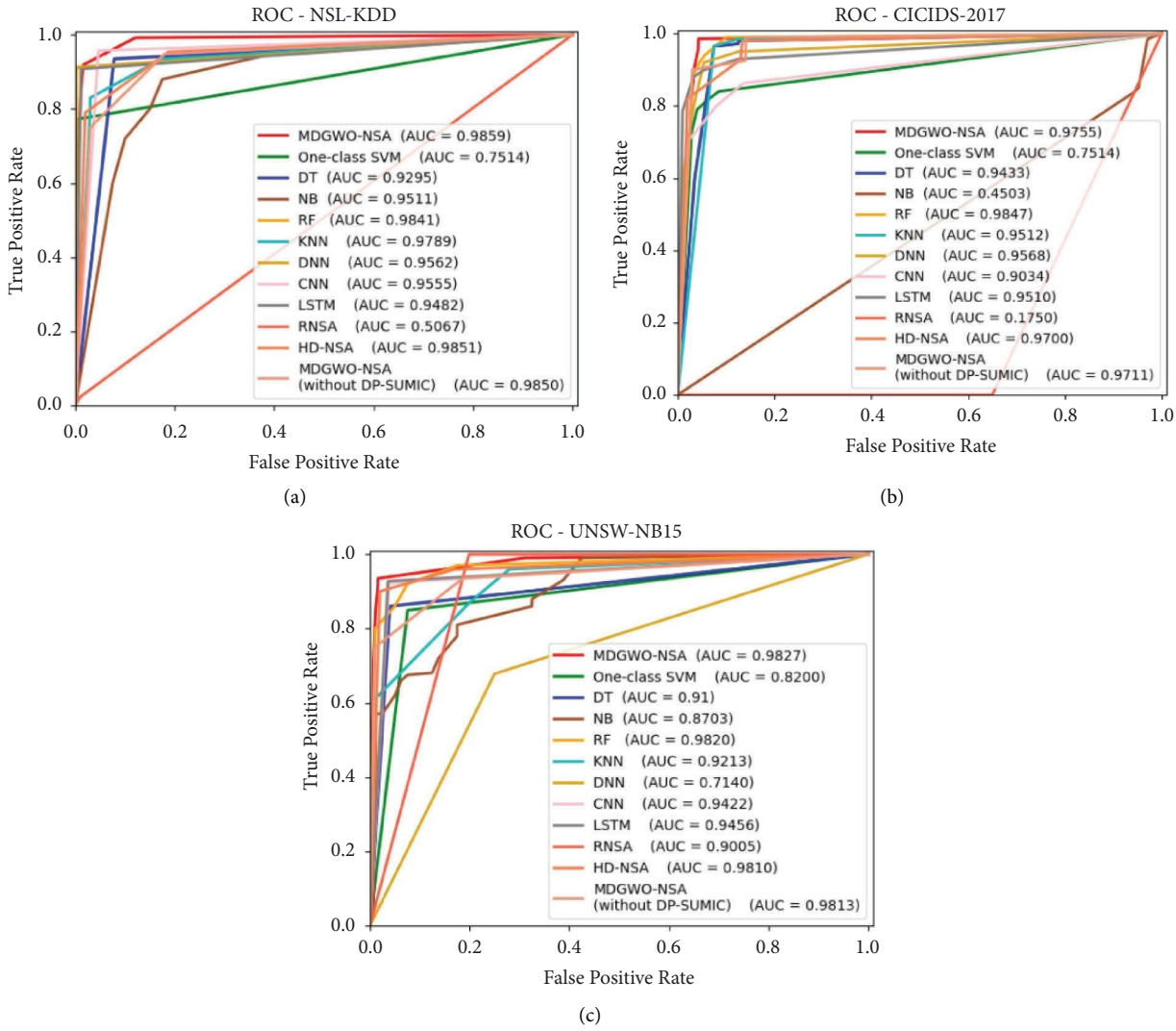

Figure 7: ROC results on (a) NSL-KDD, (b) CICIDS-2017, and (c) UNSW-NB15.

Accuracy is the probability of correctly detecting more anomalies than the number of anomalies in the original data. If accuracy is an evaluation of the correctness of a classifier as a whole, precision is an evaluation of its prediction for a particular category. On the other hand, a classifier is effective if it is accurate and precise. The accuracy of MDGWO-NSA is shown in Table 5. From the traditional model, MDGWO-NSA achieves the highest accuracy on NSL-KDD and UNSW-NB15, which are 97.61%, 95.76%, and 98.60%, respectively. MDGWO-NSA is 0.9% lower than the CNN model on CICIDS-2017, but our recall and precision are higher than CNN's at 9.43% and 18.54%. For precision, our model is 2.37% and 1.85% lower than LSTM and HD-NSA on NSL-KDD and UNSW-NB15 datasets. MDGWO-NSA is optimal for the complex dataset CICIDS-2017 because our detector can be adaptively adjusted based on the sample distribution, which makes the detector more focused on the detection of anomalous samples. Among the traditional machine learning algorithms, SVM is the most ineffective, especially on the UNSW-NB15 dataset, with an accuracy of only 74.8% and a precision of about 70.53%. This is due to the high FPR in performing the test. KNN performs roughly the same as DT and RF. In addition, the CNN algorithm outperforms DNN and LSTM among deep learning methods, especially on the CICIDS-2017 dataset, which is the highest among all algorithms by 99.5%. When comparing the algorithms based on artificial immunity, our model outperforms HD-NAS and RNSA in terms of accuracy, even without the inclusion of the feature selection algorithm DP-SUMIC. Especially, the precision is the highest among all algorithms with 96.42% on the UNSW-NB15 dataset. The precision is lower than HD-NSA and RNSA by 1.85% and 1.14% on the UNSW-NB15 dataset. We conclude that the false alarm rate is affected by the complex self-boundaries because the candidate detectors are more likely to cover complex regions and there are many detection vulnerabilities in the complex self-boundaries. Overall, our MDGWO-NSA algorithm significantly outperforms most algorithms.

The relationship between the detection rate and the false-positive rate can be shown by the ROC curve. The AUC shows the trade-off between TPR and FPR, where higher AUC values

Table 4: Classification accuracy for different datasets with different methods.

Dataset	Algorithm	Training time (s)	Classification accuracy (%)
NSL-KDD	Full dataset	724	77.75
	Chi2	249	76.40
	ANOVA-F	153	73.18
	Mutual info	233	77.68
	Random forest	216	74.61
	RFE	240	75.63
	CBFS	186	78.48
	DP-SUMIC	**140**	**79.12**
CICIDS-2017	Full dataset	123	96.61
	Chi2	47	94.67
	ANOVA-F	**18.6**	90.82
	Mutual info	69.8	97.00
	Random forest	76.9	96.87
	RFE	77.1	96.50
	CBFS	42.1	96.50
	DP-SUMIC	38.5	**97.83**
UNSW-NB15	Full dataset	1250	89.60
	Chi2	701	89.24
	ANOVA-F	**199**	85.59
	Mutual info	486	90.10
	Random forest	332	88.69
	RFE	271	89.48
	CBFS	256	90.15
	DP-SUMIC	208	**91.06**

are associated with higher TPR and lower FPR. This is the goal of the intrusion detection algorithm. From Figure 7, it can be seen that the algorithm with the best performance is MDGWO-NSA, with mean values of 98.59%, 97.55%, and 98.27%. Overall, random forest is suboptimal due to the ability of RF to handle large amounts of data in high-dimensional spaces with computational and spatial control efficiency. In contrast, naive Bayes and SVM classifiers have the worst results, especially on the most challenging dataset, the CICIDS-2017 dataset, and these two algorithms are not well suited to providing efficient responses for scaling to large datasets. The AUCs of the DT, KNN, DNN, CNN, and LSTM algorithms reached 94.33%, 95.12%, 95.68%, 90.3%, and 95.1%, respectively, on the CICIDS-2017 dataset. Obviously, they do not work as well as the artificial immunity-based algorithms HD-NSA and RNSA. This is due to the ability of the artificial immunity-based algorithm to dynamically adjust the tolerance and generation of the detector according to the target in situations where the detection task is quite difficult.

The F1-score is the average of accuracy and recall to reflect the overall performance of the classifier in incorrectly identifying anomalies. Table reftab5 depicts the F1-score results of the ten compared algorithms. The results show that the F1-score values of our proposed algorithm are 97.94%, 96.48%, and 97.45% on NSL-KDD, UNSW-NB15, and CICIDS-2017, which are significantly better than the other algorithms. We believe that the main factor is our algorithm's ability to adjust the recognition strategy for different situations during the anomaly detection process. Furthermore, we have introduced a new hole repair method, which greatly improves the efficiency of the algorithm.

4.2.3. Performance Comparison Experiments on UCI Datasets. In addition, five improved negative selection algorithms are compared on the UCI dataset, which includes RNSA, *V*-detector, BIORV-NSA, GF-NSA, and HD-NSA. The parameters of the comparison experiments are shown in Table 6. To avoid the effect of randomly selected training samples, each experiment is trained 20 times independently. The experimental results are shown in Tables 7 and 8.

(1) Performance Comparison on the Iris Dataset. The iris dataset is the most popular pattern recognition dataset in the UCI database. It contains 5 features. The first four features are the relevant attributes of the flowers, and the fifth feature is the category label of the data. The dataset contains three labels: setosa, versicolor, and virginica. Each label is regarded as self in this experiment and the other labels are regarded as non-self. Therefore, three sets of experiments were carried out.

As shown in Table 7, MDGWO-NSA has a great advantage when experiments are performed in the same dataset using different labels and amounts of training self-antigens. Under certain circumstances, in terms of the detection rate, MDGWO-NSA is higher than RNSA, V-Detector, GF-RNSA, BIORV-NSA, and HD-NSA by 17.2%, 6.9%, 5.5%, 6%, and 1%, respectively, because the detector adaptive optimization strategy is used in the detector generation stage. In terms of the false alarm rate, the MDGWO-NSA algorithm is the lowest among all algorithms. The HD-NSA algorithm has the least number of detectors, but it ignores the specificity of the boundary detectors. In terms of training time, the RNSA algorithm has the longest training time because it takes a long time to generate preset 1000 detectors before the algorithm terminates.

Table 5: Summary of results.

Dataset	Algorithm	Recall (%)	Accuracy (%)	Precision (%)	*F*1 (%)
NSL-KDD	NB	92.30	92.60	98.80	95.44
	KNN	91.30	92.60	99.00	94.01
	SVM	77.20	81.10	99.20	86.30
	DT	91.20	92.80	99.00	94.94
	RF	91.00	92.70	99.00	94.83
	DNN	91.30	92.50	98.90	94.90
	CNN	95.73	95.54	95.36	95.54
	LSTM	90.79	94.26	**99.05**	94.74
	RNSA	95.31	80.97	70.86	81.29
	HD-NSA	95.38	96.49	96.21	95.79
	MDGWO-NSA-(without DP-SUMIC)	94.76	95.31	98.97	96.82
	MDGWO-NSA	**99.23**	**97.61**	96.68	**97.94**
UNSW-NB15	NB	65.00	74.35	96.04	77.53
	KNN	96.00	93.71	94.00	94.99
	SVM	83.71	74.80	70.53	76.56
	DT	98.00	94.20	93.00	95.43
	RF	97.00	95.43	96.00	96.50
	DNN	67.80	75.03	90.30	77.45
	CNN	92.28	82.13	96.16	93.68
	LSTM	92.76	82.40	96.37	94.53
	RNSA	80.25	93.85	95.71	94.77
	HD-NSA	95.91	93.76	**96.42**	96.16
	MDGWO-NSA-(without DP-SUMIC)	93.57	94.46	95.18	94.37
	MDGWO-NSA	**98.46**	**95.76**	94.57	**96.48**
CICIDS-2017	NB	80.00	82.00	82.00	80.99
	KNN	94.56	92.33	94.55	94.55
	SVM	84.00	84.00	81.50	82.73
	DT	95.00	92.70	95.00	95.00
	RF	94.00	92.60	94.70	94.35
	DNN	95.03	92.97	95.03	95.03
	CNN	86.30	**99.50**	80.80	83.46
	LSTM	92.95	96.83	98.31	95.41
	RNSA	87.29	95.33	82.57	84.86
	HD-NSA	92.79	97.34	97.11	94.90
	MDGWO-NSA- (without DP-SUMIC)	92.53	94.38	96.75	94.59
	MDGWO-NSA	**95.73**	98.60	**99.24**	**97.45**

Table 6: Experimental parameters of each comparison algorithm.

Algorithm	Parameter	Value
GF-RNSA, V-detector, RNSA, HD-NSA, BIORV-NSA, MDGWO-NSA	Self-radius	0.01
GF-RNSA, V-detector, RNSA	Expected coverage	99%
RNSA	Fixed detector radius	0.1
BIORV-NSA	Self-edge tolerance	0.8
BIORV-NSA	Antibody tolerance	1.2
GF-RNSA	Density threshold	0.1
GF-RNSA	Minimum threshold	0.0625
HD-NSA	Division threshold	0.5
MDGWO-NSA	Grid radius	0.05
MDGWO-NSA	Density threshold	0.2

Table 7: Experimental results of MDGWO-NSA algorithm compared with other algorithms on the iris dataset.

Training data	Algorithm	DR (%)		FAR (%)		Dn		DTT (s)		Test (s)	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Setosa (50%)	RNSA	96.5	2.70	**2.0**	1.63	7393.2	688.9	7.620	2.120	7.665	2.120
	V-detector	100.0	0.00	3.2	1.69	3653.1	836.5	1.230	1.050	1.240	1.030
	GF-RNSA	98.0	0.79	8.6	2.90	6100.0	0.00	**0.045**	0.004	0.114	0.012
	BIORV-NSA	**100.0**	0.00	4.0	0.00	500.0	0.0	1.475	0.013	1.740	0.184
	HD-NSA	**100.0**	0.0	3.8	2.90	**140.3**	17.3	0.169	0.003	**0.0183**	0.004
	MDGWO-NSA	**100.0**	0.00	3.2	0.0	161.4	24.4	0.830	0.010	0.110	0.008
Setosa (100%)	RNSA	97.20	1.80	0.0	0.00	7450.3	393.0	7.612	1.262	7.660	1.260
	V-detector	100.0	0.00	0.0	0.00	3273.9	1016.3	1.020	1.120	1.033	1.210
	GF-RNSA	99.30	0.52	1.6	0.51	6150.0	0.0	0.056	0.011	0.134	0.015
	BIORV-NSA	**100.0**	0.00	**0.0**	0.00	500.0	0.0	1.580	0.051	1.840	0.114
	HD-NSA	**100.0**	0.00	**0.0**	0.00	**164.0**	0.0	**0.023**	0.006	**0.025**	0.004
	MDGWO-NSA	**100.0**	0.00	**0.0**	0.00	170.0	0.0	0.036	0.017	0.041	0.020
Versicolor (50%)	RNSA	86.7	4.59	**2.6**	2.99	8584.9	516.0	10.580	1.780	10.640	1.780
	V-detector	96.5	1.70	8.6	3.90	3568.4	1320.9	1.660	1.060	1.680	1.060
	GF-RNSA	97.3	1.20	11.8	4.20	6100.0	0.0	0.011	0.004	**0.039**	0.007
	BIORV-NSA	98.0	0.00	10.0	0.00	500.0	0.0	1.550	0.142	1.950	0.096
	HD-NSA	98.7	5.60	4.0	3.90	287.7	22.1	0.050	0.012	0.053	0.013
	MDGWO-NSA	**99.3**	0.00	3.35	0.00	**263.0**	17.0	**0.042**	0.050	0.081	0.009
Versicolor (100%)	RNSA	82.8	3.85	0.0	0.00	8226.1	777.2	9.550	2.440	9.618	2.450
	V-detector	93.1	1.29	0.0	0.00	3568.0	1578.5	1.290	0.890	1.310	0.900
	GF-RNSA	94.5	1.25	2.1	0.13	6150.0	0.0	0.105	0.006	0.152	0.018
	BIORV-NSA	94.0	0.00	**0.0**	0.00	500.0	0.0	1.690	0.113	2.020	0.201
	HD-NSA	99.0	0.00	**0.0**	0.00	404.0	0.0	0.054	0.008	0.057	0.009
	MDGWO-NSA	**100.0**	0.00	**0.0**	0.00	**341.0**	0.0	**0.049**	0.020	**0.051**	0.007
Virginica (50%)	RNSA	87.0	4.85	7.2	4.00	8630.8	1019.5	10.830	3.210	10.890	3.220
	V-detector	96.4	0.80	10.0	3.80	2822.3	1127.0	0.720	0.460	0.730	0.460
	GF-RNSA	97.6	1.10	18.8	5.30	9900.0	0.0	**0.018**	0.006	**0.047**	0.010
	BIORV-NSA	98.0	0.00	12.0	0.00	500.0	0.0	1.550	0.067	1.820	0.141
	HD-NSA	99.0	0.00	8.2	3.10	**396.8**	14.8	0.058	0.009	0.061	0.010
	MDGWO-NSA	**99.5**	0.00	**7.0**	3.70	406.0	40.0	0.067	0.040	0.072	0.030
Virginica (100%)	RNSA	86.4	2.10	0.0	0.00	9688.5	1587.0	14.270	6.440	14.340	6.440
	V-detector	95.1	0.70	0.0	0.00	2356.5	1042.6	0.500	0.440	0.520	0.450
	GF-RNSA	97.8	1.50	1.8	0.22	9950.0	0.0	**0.025**	0.004	**0.053**	0.009
	BIORV-NSA	96.0	0.00	**0.0**	0.00	500.0	0.0	1.710	0.092	2.010	0.154
	HD-NSA	**100.0**	0.00	**0.0**	0.00	**431.0**	0.0	0.054	0.010	0.057	0.011
	MDGWO-NSA	**100.0**	0.00	**0.0**	0.00	494.0	0.0	0.059	0.026	0.063	0.030

Table 8: Experimental results of the MDGWO-NSA algorithm compared with other algorithms on the skin dataset.

Algorithm	DR (%)		FAR (%)		Dn		DTT (s)		Test (s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
RNSA	85.80	3.73	**0.93**	0.30	2464.3	822.98	1.110	0.64	20.30	3.29
V-detector	87.20	2.20	3.04	0.69	375.9	106.20	0.093	0.02	2.72	0.91
GF-RNSA	93.25	2.13	10.20	0.59	41950.0	1480.40	2.097	0.21	24.3	0.50
BIORV-NSA	97.32	0.00	5.88	0.00	1000.0	0.00	6.782	0.12	13.52	1.21
HD-NSA	97.90	0.43	4.17	0.54	**240.6**	8.95	**0.072**	0.02	**1.06**	0.06
MDGWO-NSA	**98.30**	0.54	3.22	0.36	251.9	11.40	0.087	0.03	1.38	0.03

(2) Performance Comparison on the Skin Dataset. The skin segmentation dataset contains 245057 instances. Each piece of data is composed of four features; the first three features are color values, and the fourth feature is the label of the sample. The category labels are 1 and 2, where label 1 is self and the label 2 is non-self.

As shown in Table 8, compared with the average detection rates of the typical algorithms RNSA, V-Detector, GF-RNSA, and BIORV-NSA, MDGWO-NSA is increased by 12.5%, 11.1%, 5.05%, 1.02%, and 0.4%, respectively. In terms of detector generation efficiency, MDGWO-NSA is less than RNSA, *V*-Detector, GF-RNSA, and BIORV-NSA by 89.33%, 29.87%,

99.37%, and 73.70%, respectively. The detector generation time of the MDGWO-NSA is less than that of each of the four typical NSAs in proximity to the HD-NSA. Although the number of detectors is more than HD-NSA by 4.5%, the false alarm rate is lower than HD-NSA by 9%. Because we generate finer divisions, the coverage and detection rate of the boundary detectors are improved. It is clear that our method effectively reduces the false alarm rate due to boundary diversity.

5. Conclusions

Various types of attacks and network traffic lead to high time complexity in network anomaly detection. The effectiveness of artificial immune systems used in intrusion detection systems is clearly illustrated by our research. During the anomaly detection phase, the time for candidate detectors to perform self-sets tolerance grows exponentially with the number of self-sets, caused by reduced efficiency and the generation of a large number of redundant detectors. To address these issues, a novel anomaly detection framework is proposed based on negative selection theory. First, the best feature subset is filtered by unsupervised density clustering, which combines maximum mutual information interclass and symmetric uncertainty intraclass. The feature space is also reduced to identify the attributes with the least dispersion. Subsequently, the feature space is gridded according to the density and specific candidate boundary detectors are generated based on the boundary grid. Obviously, the low boundary detection rate and high detector redundancy due to the diversity of boundary samples are effectively mitigated by this method. Moreover, to achieve fast and efficient generation of detectors with better detection performance, the appropriate locations and radius of the detector are generated by incorporating GWO methods that adopt self-sample clustering and parallel optimization in the far self-sample region. Theoretical analysis and experimental results show that the detection rate and detector generation efficiency of the MDGWO-NSA algorithm are significantly better compared with existing network anomaly detection methods and typical NSA algorithms.

The method proposed in this paper improves the negative selection algorithm and has better efficiency in network anomaly detection. However, detector adaptive evolution is still difficult to solve. In future work, the adaptive evolution of detectors in the NSA algorithm will also be investigated to design new NSA algorithms, that implement endogenous secure networks with immunity.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant no. 2020YFB1805400), the National Natural Science Foundation of China (Grant Nos. U1836112 and 61876134), and the National Key Research and Development Program of China (Grant no. 2021YFB3100700).

References

[1] D. Kwon, K. Natarajan, S. Suh, and H. Kim, "An empirical study on network anomaly detection using convolutional neural networks," in *Proceedings of the IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1595–1598, IEEE, Vienna, Austria, July 2018.

[2] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[3] S. S. Sivatha Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Systems with Applications*, vol. 39, no. 1, pp. 129–141, 2012.

[4] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrao, and M. L. Proenca, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Systems with Applications*, vol. 92, pp. 390–402, 2018.

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

[6] 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.

[7] Q. Lin, Y. Ma, J. Chen et al., "An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies," *Information Sciences*, vol. 430-431, pp. 46–64, 2018.

[8] X. Yu, D. Fu, T. Yang, and K. Riha, "The application of negative selection algorithm in multi-angle infrared vehicle images recognition," in *Proceedings of the 38th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 776–780, Prague, Czech Republic, July 2015.

[9] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[10] A. Hadri, K. Chougdali, and R. Touahni, "Intrusion detection system using PCA and Fuzzy PCA techniques," in *Proceedings of the 2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, pp. 1–7, Marrakesh, Morocco, February 2016.

[11] H. Benaddi, K. Ibrahimi, and A. Benslimane, "Improving the intrusion detection system for NSL-KDD dataset based on PCA-fuzzy clustering-KNN," in *Proceedings of the 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1–6, Marrakesh, Morocco, January 2018.

[12] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, 2017.

[13] A. Nazir and R. A. Khan, "A novel combinatorial optimization based feature selection method for network intrusion detection," *Computers & Security*, vol. 102, Article ID 102164, 2021.

[14] E. Popoola and A. Adewumi, "Efficient feature selection technique for network intrusion detection system using

discrete differential evolution and decision," *International Journal on Network Security*, vol. 19, no. 5, pp. 660–669, 2017.

[15] A. A. Mohammad, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[16] P. Mishra, E. Pilli, V. Varadharajan, and U. Tupakula, "Out-vm monitoring for malicious network packet detection in cloud," in *Proceedings of the 2017 ISEA Asia Security and Privacy (ISEASP)*, pp. 1–10, Surat, India, January 2017.

[17] W. Wang, X. Du, and N. Wang, "Building a cloud IDS using an efficient feature selection method and SVM," *IEEE Access*, vol. 7, pp. 1345–1354, 2019.

[18] M. Guerroumi, N. Belhadjaissa, and A. Derhab, "NSNAD: negative selection-based network anomaly detection approach with relevant feature subset," *Neural Computing & Applications*, vol. 32, no. 8, pp. 3475–3501, 2020.

[19] X. Xiao, T. Li, and R. Zhang, "An immune optimization based real-valued negative selection algorithm," *Applied Intelligence*, vol. 42, no. 2, pp. 289–302, 2015.

[20] L. Cui, D. Pi, and C. Chen, "BIORV-NSA: bidirectional inhibition optimization r-variable negative selection algorithm and its application," *Applied Soft Computing*, vol. 32, pp. 544–552, 2015.

[21] D. Li, S. Liu, and H. Zhang, "A boundary-fixed negative selection algorithm with online adaptive learning under small samples for anomaly detection," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 93–105, 2016.

[22] 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.

[23] X. Zheng, Y. Fang, and T. Li, "Dual negative selection algorithm," *China Information*, vol. 43, no. 4, pp. 611–625, 2013.

[24] I. Aydin, M. Karakose, and E. Akin, "Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5285–5294, 2010.

[25] T. Yang, W. Chen, and T. Li, "A real negative selection algorithm with evolutionary preference for anomaly detection," *Open Physics*, vol. 15, no. 1, pp. 121–134, 2017.

[26] F. Zhang, J. Yang, T. Li, and F. Zhu, "DnyNSA: a novel real-value based negative selection algorithm," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1104–1109, Bangalore, India, November 2018.

[27] M. Poggiolini and A. Engelbrecht, "Application of the feature-detection rule to the negative selection algorithm," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3001–3014, 2013.

[28] W. Chen, T. Li, X. Liu, and B. Zhang, "A negative selection algorithm based on hierarchical clustering of self set," *China Information*, vol. 43, no. 5, p. 611, 2013.

[29] T. Yang, C. Wen, D. Xiaoming, and L. Tao, "Negative selection algorithm based on grid file of the feature space," *Knowledge-Based Systems*, vol. 56, pp. 26–35, 2014.

[30] T. Yang, W. Chen, and T. Li, "An antigen space density based real-value negative selection algorithm," *Applied Soft Computing*, vol. 61, pp. 860–874, 2017.

[31] Z. Li and T. Li, "Using known nonself samples to improve negative selection algorithm," *Applied Intelligence*, vol. 52, no. 1, pp. 482–500, 2021.

[32] T. Li and C. Wen, "Parameter analysis of negative selection algorithm," *Information Sciences*, vol. 420, pp. 218–234, 2017.

[33] S. Fouladvand, A. Osareh, B. Shadgar, M. Pavone, and S. Sharafi, "DENSA: an effective negative selection algorithm with flexible boundaries for self-space and dynamic number of detectors," *Engineering Applications of Artificial Intelligence*, vol. 62, pp. 359–372, 2017.

[34] A. Abid, M. T. Khan, and C. W. de Silva, "Layered and real-valued negative selection algorithm for fault detection," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2960–2969, 2018.

[35] Z. Li, T. Li, J. He, Y. Zhu, and Y. Wang, "A hybrid real-valued negative selection algorithm with variable-sized detectors and the *k*-nearest neighbors algorithm," *Knowledge-Based Systems*, vol. 232, Article ID 107477, 2021.

[36] D. N. Reshef, Y. A. Reshef, H. K. Finucane et al., "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.

[37] G. Karakaya, S. Galelli, S. D. Ahipasaoglu, and R. Taormina, "Identifying (quasi) equally informative subsets in feature selection problems for classification: a max-relevance min-redundancy approach," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1424–1437, 2016.

[38] J. Xie, H. Gao, and W. Xie, "*K*-nearest neighbors optimized clustering algorithm by fast search and finding the density peaks of a dataset," *Science China Information Sciences*, vol. 46, no. 2, pp. 258–280, 2016.

[39] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[40] S. M. Mirjalili, S. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[41] H. Wang, X. Gao, X. Huang, and Z. Song, *PSO-Optimized Negative Selection Algorithm for Anomaly Detection*, pp. 13–21, Springer, Berlin, Germany, 2009.

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

[43] Z. Ji and D. Dasgupta, "Real-valued negative selection algorithm with variable-sized detectors," *Genetic and Evolutionary Computation Conference*, pp. 287–298, Springer, Berlin, Gremany, 2004.

[44] J. He, W. Chen, T. Li, B. Li, Y. Zhu, and M. Huang, "HD-NSA: a real-valued negative selection algorithm based on hierarchy division," *Applied Soft Computing*, vol. 112, Article ID 107726, 2021.

[45] L. Dhanabal and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[46] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, Canberra, ACT, Australia, November 2015.

[47] Z. Pelletier and M. Abualkibash, "Evaluating the CIC IDS-2017 dataset using machine learning methods and creating multiple predictive models in the statistical computing language R," *Science*, vol. 5, no. 2, pp. 187–191, 2020.

[48] X. Tao, Q. Li, C. Ren et al., "Real-value negative selection oversampling for imbalanced data set learning," *Expert Systems with Applications*, vol. 129, pp. 118–134, 2019.

[49] P. Wu, H. Guo, and N. Moustafa, "Pelican: a deep residual network for network intrusion detection," in *Proceedings of the 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 55–62, Valencia, Spain, June 2020.

[50] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the

UNSW-NB15 dataset," *Journal of Big Data*, vol. 7, no. 1, pp. 105–120, 2020.

[51] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, Article ID 115524, 2021.

[52] F. Belgrana, N. Benamrane, M. Hamaida, A. Chaabani, and A. Taleb-Ahmed, "Network intrusion detection system using neural network and condensed nearest neighbors with selection of NSL-KDD influencing features," in *Proceedings of the 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, pp. 23–29, Bali, Indonesia, January 2021.

[53] R. Lohiya and A. Thakkar, "Intrusion detection using deep neural network with antirectifier layer," in *Applied Soft Computing and Communication Networks*, vol. 187, pp. 89–105, Springer, Berlin, Germany, 2021.