

Research Article

LNNLS-KH: A Feature Selection Method for Network Intrusion Detection

Xin Li ¹, Peng Yi ¹, Wei Wei,² Yiming Jiang,¹ and Le Tian ¹

¹Information Technology Institute, PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China

²Center for Energy Environment & Economy Research, Zhengzhou University, Zhengzhou 450001, China

Correspondence should be addressed to Peng Yi; yipengndsc@163.com

Received 15 September 2020; Revised 27 November 2020; Accepted 17 December 2020; Published 6 January 2021

Academic Editor: Jesús Díaz-Verdejo

Copyright © 2021 Xin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As an important part of intrusion detection, feature selection plays a significant role in improving the performance of intrusion detection. Krill herd (KH) algorithm is an efficient swarm intelligence algorithm with excellent performance in data mining. To solve the problem of low efficiency and high false positive rate in intrusion detection caused by increasing high-dimensional data, an improved krill swarm algorithm based on linear nearest neighbor lasso step (LNNLS-KH) is proposed for feature selection of network intrusion detection. The number of selected features and classification accuracy are introduced into fitness evaluation function of LNNLS-KH algorithm, and the physical diffusion motion of the krill individuals is transformed by a nonlinear method. Meanwhile, the linear nearest neighbor lasso step optimization is performed on the updated krill herd position in order to derive the global optimal solution. Experiments show that the LNNLS-KH algorithm retains 7 features in NSL-KDD dataset and 10.2 features in CICIDS2017 dataset on average, which effectively eliminates redundant features while ensuring high detection accuracy. Compared with the CMPSO, ACO, KH, and IKH algorithms, it reduces features by 44%, 42.86%, 34.88%, and 24.32% in NSL-KDD dataset, and 57.85%, 52.34%, 27.14%, and 25% in CICIDS2017 dataset, respectively. The classification accuracy increased by 10.03% and 5.39%, and the detection rate increased by 8.63% and 5.45%. Time of intrusion detection decreased by 12.41% and 4.03% on average. Furthermore, LNNLS-KH algorithm quickly jumps out of the local optimal solution and shows good performance in the optimal fitness iteration curve, convergence speed, and false positive rate of detection.

1. Introduction

With the advent of the era of big data, the dimension of information has increased exponentially. In many fields such as machine learning, data analysis, and text mining [1], it is increasingly difficult to handle large amounts of high dimension data. Irrelevant and redundant features increase the complexity of the dimension and interfere with the accurate classification results, resulting in poor performance of the algorithm. Intrusion detection system (IDS) [2] relies on a large amount of network data, which carries out real-time monitoring of network transmission and identifies and processes malicious use of computers and network resources. The “curve of dimensionality” (COD) caused by massive data of IDS leads to low detection rate, poor effect, and high false positive rate, which seriously affect the efficiency of intrusion detection. How to improve the efficiency

of intrusion detection while ensuring the detection accuracy has become an urgent problem to be solved.

As a common method of data dimensionality reduction, feature selection has attracted more and more attention. It reduces the complexity of data by deleting unnecessary features, which is of great significance to IDS. Feature selection algorithms filter out redundant data to reduce the dimensions of network data. In addition, the computing payload of IDS is decreased and the detection speed is improved. Consequently, feature selection is one of the critical links of data preprocessing in IDS, which has a significant impact on detection accuracy and model generalization ability. Generally, the feature selection framework is composed of four parts: search module, evaluation criterion, judgment condition and verification, and output. The search module includes search starting point and search strategy. After the original feature set is processed by the

search module, the corresponding feature subset is generated. Appropriate evaluation criteria are constructed to evaluate the feature subsets. When the termination condition of the feature selection process is reached, the final selected feature subset is output. Meanwhile, it is verified to evaluate the quality of feature selection algorithm. The framework of feature selection is shown in Figure 1.

The swarm intelligence optimization method is a kind of group-oriented random search technology, which provides new ideas for solving the feature selection problem. Krill herd (KH) algorithm is a new type of swarm intelligence optimization method that studies the foraging rule and clustering behavior of krill herd in nature. By simulating the movement induced by other krill individuals, foraging activity, and physical diffusion motion of krill herd, the position of individuals is constantly updated. While looking for food and the highest krill herd density, they will move towards the best solution and finally get the global optimal solution. KH algorithm has been widely concerned by many scholars and engineers for its excellent optimization performance and is considered to be one of the fastest developing natural heuristic algorithms in solving practical optimization problems [3]. It integrates the local robust search method with the population-based method and has a good performance in high-dimensional data processing. It is widely used in network path optimization [4], text clustering analysis [5], neural network training [6], multiple continuous optimization [7–9], combinatorial optimization [10, 11], constraint optimization [12–14], and other scenarios [3]. KH algorithm has good exploitation ability, but the exploration ability is not satisfactory, which means that the algorithm is easy to fall into local optimal solution when solving practical problems. Although there are existing optimization algorithms for KH algorithm, the research on the optimization algorithm that can provide high convergence rate and global optimal solution is continuing. Therefore, the improvement of KH algorithm to balance the global exploration and local exploitation abilities is of great significance for improving the solution accuracy and optimization efficiency.

In this paper, an optimized LNNLS-KH algorithm for feature selection is proposed to address the problem of large number and high dimension of intrusion detection datasets. It filters out the redundant features of IDS data so that the efficiency of intrusion detection is significantly improved and the time cost is enormously reduced.

The main contributions of this paper are listed as follows:

- (i) The number of dimensions and detection accuracy of feature selection were introduced into the fitness function, which improved the ability of feature selection.
- (ii) To accelerate the convergence speed of the algorithm, we modified the physical diffusion motion of krill individuals by the nonlinear method.
- (iii) The LNNLS-KH algorithm was proposed for feature selection of intrusion detection data, which

effectively enhanced the local exploitation ability and global exploration ability of the algorithm.

- (iv) The proposed algorithm was comprehensively evaluated by conducting a large number of experiments on NSL-KDD and CICIDS2017 datasets dataset. The experimental results show that the LNNLS-KH algorithm exhibited good competitive performance in the evaluation indicators for intrusion detection.

The remaining sections of this paper are organized as follows. Section 2 presents the related works about feature selection methods and the variants of KH algorithm. Section 3 provides a detailed description of the proposed LNNLS-KH algorithm. Section 4 provides shows the experimental results and discussion. Section 5 is concluded with future research.

2. Related Works

In this section, we show three feature selection methods based on evaluation criteria and feature selection algorithms in IDS. Meanwhile, we summarize swarm intelligence algorithms, especially KH algorithm and its variants.

2.1. Feature Selection Methods Based on the Evaluation Criteria. There are three types of feature selection methods based on the evaluation criteria: the filter method, the wrapper method, and the embedded method [15]. The filter method assigns weights to the features of each dimension, filters the features in the order of weight, and uses the feature subsets to train the classification algorithm. Therefore, the process of feature selection is independent of the classification algorithm. Although the filter method occupies fewer computing resources and saves more time for feature selection, the selected feature subset lacks the adjustment of the classification algorithm, resulting in low classification accuracy. The wrapper method takes into account the effect of the performance of the classification algorithm on the feature subsets, so it derives a high classification accuracy, but the computation and time are consumed enormously. The embedded method integrates the feature selection process and the classification algorithm and simultaneously performs feature selection during the classification training. Its computation cost and classification accuracy are between the filter method and the wrapper method. The feature selection of intrusion detection data requires high accuracy, and the training time of offline data is not concerned. Therefore, the wrapper method is adopted as the feature selection method in this paper. The frameworks of the three types of feature selection methods based on the evaluation criteria are shown in Figure 2.

2.2. Feature Selection Algorithms in IDS. Feature selection is one of the most important parts of data preprocessing in intrusion detection, which is of great significance to IDS. The

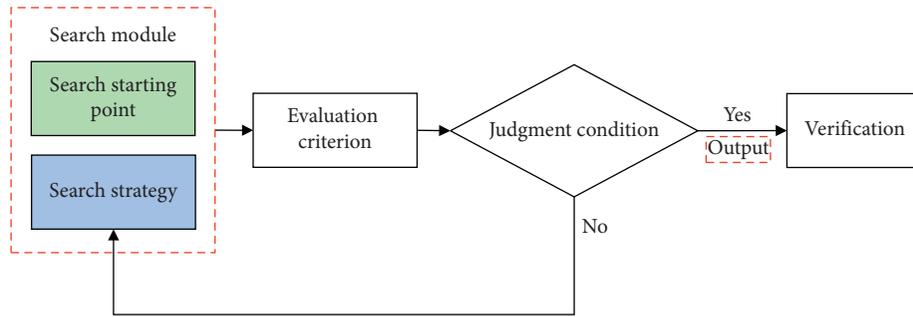


FIGURE 1: Framework of feature selection.

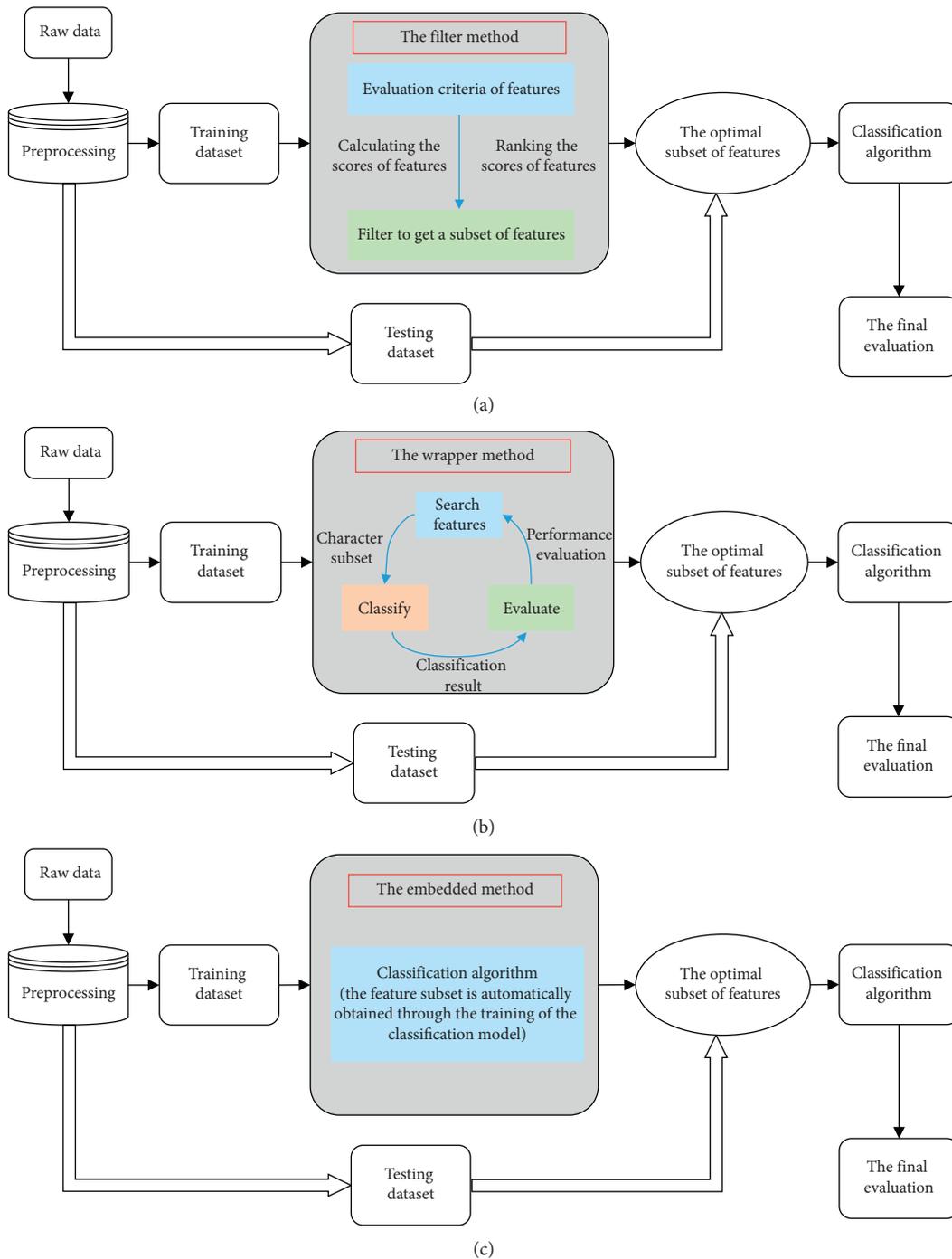


FIGURE 2: Frameworks of the three types of feature selection methods: (a) framework of the filter method for feature selection, (b) framework of the wrapper method for feature selection, and (c) framework of the embedded method for feature selection.

characteristics of network intrusion detection data are multiple features and large scale. Features of different categories have different attribute values, including redundant features that interfere with the classification results. A large number of redundant features reduce the efficiency of detection algorithms and increase the false positive rate of intrusion detection. However, a feature selection algorithm with good performance decreases the dimensionality of network data and improves the accuracy and detection speed of IDS.

In recent years, there has been a great deal of research studies on feature selection in intrusion detection. Smith et al. combined Bayesian network and principal component analysis (PCA) to conduct feature selection for intrusion detection data [16]. They used Bayesian networks to adjust the correlation of attributes and PCA to extract the primary features on an institute-wide cloud system. The disadvantage is that the detection accuracy is considered to be further improved as an improvement. Zhao et al. [17] proposed a feature selection method based on Mahalanobis distance and applied it to network intrusion detection to obtain the optimal feature subset. Feature ranking based on Mahalanobis distance was used as the principle selection mechanism and the improved exhaustive search was used to select the optimal ranking features. The experimental results based on the KDD CUP 99 dataset show that the algorithm has good performance on both the support vector machine and the k-nearest neighbor classifier. Singh and Tiwari proposed an efficient approach for intrusion detection in reduced features of KDD CUP 99 dataset in 2015 [18]. Iterative Dichotomiser 3 (ID3) algorithm was used for feature reduction of large datasets, and KNNGA was used as a classifier for intrusion detection. The method performs well on evaluation measures of sensitivity, specificity, and accuracy. However, both Zhao et al. and Singh and Tiwari [17, 18] conduct experiments on the outdated datasets, which are difficult to reflect the new attack features of modern networks. In [19], Ambusaidi et al. proposed a feature selection algorithm based on mutual information to deal with linear and nonlinear related data features. They established an intrusion detection system based on least-squares support vector machine. Experimental results show that the proposed algorithm performs well in accuracy, but poor in false positive rate. Shone et al. proposed an unsupervised feature learning method based on nonsymmetric deep autoencoder (NDAE) and a novel deep learning classification model constructed using stacked NDAEs [20]. The results demonstrated that the approach offers high levels of accuracy, precision, and recall together with reduced training time. Meanwhile, it is worth noting that the stacked NDAE model has 98.81% less training time than the mainstream DBN technology. The limitation is that the model needs to assess and extend the capability to handle zero-day attacks.

In [21], a self-adaptive differential evolution (SaDE) algorithm was proposed to deal with the feature selection problem. It uses adaptive mechanism to select the most appropriate among the four candidate solution generation strategies, which effectively reduced the number of features. The disadvantage is that the experiment uses small sample data and more data is needed to further support the conclusion. Shen et al. adopted principal component analysis

and linear discriminant analysis to decrease the dimensionality of the dataset and combined with Bayesian classification to construct an intrusion detection model [22]. Simulation experiments based on CICIDS2017 dataset show that the proposed algorithm filters out the noise in the data and improves the time performance to a certain extent. However, the algorithm still needs to be optimized to further improve the classification accuracy. In [23], a hybrid network feature selection method based on convolutional neural network (CNN) and long and short-term memory network (LSTM) had been applied to IDS. According to the experimental results, the proposed feature selection algorithm achieves better accuracy compared with the CNN-only model and the LSTM-only model. However, the detection accuracy of Heartbleed and SSHPatator attacks is low. In [24], Farahani proposed a new cross-correlation-based feature selection (CCFS) method to reduce the feature dimension of intrusion detection dataset. Compared with cuttlefish algorithm (CFA) and mutual information-based feature selection (MIFS), the proposed algorithm was demonstrated to have a good performance in the accuracy, precision, and recall rate of classification. However, the author simply replaced the categorical attributes with numeric values when dealing with symbolic data, without considering a more reasonable one-hot encoding method. The summary of feature selection methods in IDS is shown in Table 1.

2.3. Swarm Intelligence Algorithms for Feature Selection.

The core of feature selection is the search strategy for generating feature subsets. Although the exhaustive search strategy can find the globally optimal feature subset, its excessive time complexity consumes huge computing resources, whether exhaustive search or nonexhaustive search. In recent years, swarm intelligence optimization methods inspired by natural phenomena provide a new approach to solve the problem of feature selection [10–17]. Therefore, we propose the LNNLS-KH algorithm with high search efficiency as the search strategy for feature subset. Swarm intelligence optimization methods simulate the evolution of survival of the fittest in nature and are a group-oriented random search technique that can be used to solve complex problems in large-scale data analysis [25]. Common swarm intelligence optimization methods include particle swarm optimization (PSO) [26], ant colony optimization algorithm (ACO) [27], cuckoo algorithm (CA) [28], artificial fish swarm algorithm (AFSA) [29], artificial bee colony algorithm (ABC) [30], fruit fly optimization algorithm (FOA) [31], monkey algorithm (MA) [32], bat algorithm (BA) [33], and salp swarm algorithm (SSA) [34].

Moreover, Ahmed et al. proposed a new chaotic chicken swarm algorithm (CCSO) for feature selection [35]. By combining logical maps and chaotic trend maps, the CSO algorithm acquires a strong spatial search ability. The experimental results show that the classification accuracy of the model is further improved after CCSO feature selection. The disadvantage is the lack of comparison with other chaotic algorithms. Ahmtabakh proposed an unsupervised

TABLE 1: Summary of feature selection methods in IDS.

Method	Author	Year	Ref. no
Bayesian network-based dimensionality reduction and principal component analysis (PCA)	Smith et al.	2010	[16]
Ranking based on Mahalanobis distance and exhaustive search	Zhao et al.	2013	[17]
Iterative Dichotomiser 3 (ID3) algorithm	Singh and tiwari	2015	[18]
Mutual information method	Ambusaidi et al.	2016	[19]
Nonsymmetric deep autoencoder (NDAE)	Shone et al.	2018	[20]
Self-adaptive differential evolution (SaDE)	Xue et al.	2018	[21]
Principal component analysis (PCA) and linear discriminant analysis (LDA)	Shen et al.	2019	[22]
Hybrid network of convolutional neural network (CNN) and long short-term memory network (LSTM)	Sun et al.	2020	[23]
Cross-correlation-based feature selection (CCFS) method	Farahani	2020	[24]

feature selection method based on ant colony optimization (UFSACO) [36], which iteratively filtrates feature through the heuristic and previous stage information of the ant colony. Simultaneously, the similarity between features is quantified to reduce the redundancy of data features. However, the efficiency of feature selection process needs to be improved.

To solve the problem that it is easy to fall into the local optimal solution, Arora and Anand proposed a butterfly optimization algorithm (BOA) based on binary variables [37]. Based on the foraging behavior of butterflies, the algorithm uses each butterfly as a search agent to iteratively optimize the fitness function, which has good convergence ability and avoids the premature problem to a certain extent. Experimental results show that the algorithm reduces the length of feature subset while selecting the optimal feature subset and improves the classification accuracy to a certain extent. However, the time cost is larger than that of genetic algorithm and particle swarm optimization algorithm, and the optimization result of the feature subset for repeated experiments is inaccurate and has poor robustness.

In [38], Yan et al. proposed a hybrid optimization algorithm (BCROSAT) based on simulated annealing and binary coral reefs, which is used for feature selection in high-dimensional biomedical datasets. The algorithm increases the diversity of the initial population individuals through the league selection strategy and uses the simulated annealing algorithm and binary coding to improve the search ability of the coral reef optimization algorithm. However, the algorithm has high time complexity. In [39], a new chaotic Dragonfly algorithm (CDA) is proposed by Sayed et al., which combines 10 different chaotic maps with the search iteration process of dragonfly algorithm, so as to accelerate the convergence speed of the algorithm and improve the efficiency of feature selection. The algorithm uses the worst fitness value, best fitness value, average fitness value, standard deviation, and average feature length as evaluation criteria. The experimental results show that the adjustment variable of Gauss map significantly improves the performance of dragonfly algorithm in classification performance, stability, number of selected features, and convergence speed. The disadvantage is that the experimental data is small, and the algorithm needs to be verified on large-scale datasets. Zhang et al. [40] mixed genetic algorithm and particle swarm optimization algorithm to conduct taboo search for the produced optimal initial solution, and the result of quadratic feature selection is the global optimal

feature subset. The algorithm not only guarantees the good classification performance but also greatly reduces the false positive rate and false negative rate of classification results. The disadvantage is that the algorithm takes a large calculation cost and a long offline training time.

2.4. Krill Herd (KH) Algorithm and Variants. Krill herd (KH) algorithm is a new swarm intelligence optimization method based on population proposed by Gandomi and Alavi in 2012 [41]. The algorithm studies the foraging rules and clustering behavior of the herding of the krill swarms in nature and simulates the induced movement, foraging activity, and random diffusion movement of KH. Meanwhile, it obtains the optimal solution by continuously updating the position of krill individuals.

Abualigah et al. introduced a multicriteria mixed function based on the global optimal concept in the KH algorithm and applied it to text clustering [5]. By supplementing the advantages of local neighborhood search and global wide area search, the algorithm balances the exploitation and exploration process of krill herd. In [42], the influence of excellent neighbor individuals on the krill herd during evolution is considered and an improved KH algorithm is proposed to enhance the local search ability of the algorithm. In [43], a hybrid data clustering algorithm (IKH-KHM) based on improved KH algorithm and k-harmonic means was proposed to solve the problem of sensitive clustering center of K-means algorithm. This algorithm increases the diversity of KH by alternately using the random walk of Levi flight and the crossover operator in the genetic algorithm. It improves the global search ability of the algorithm and avoids the phenomenon of premature convergence of the algorithm to some degree. The simulation experiments of the 5 datasets in the UCI database show that the IKH-KHM algorithm overcomes the noise sensitivity problem to a certain extent and has a significant effect on the optimization of the objective function. However, its slow recovery speed results in a high time cost of the algorithm. In 2017, Li and Liu adopted a combined update mechanism of selection operator and mutation operator to enhance the global optimization ability of the KH algorithm. They solved the problem of unbalanced local search and global search of the original KH algorithm [44].

For enhancing the global search ability of KH algorithm, a global search operator improved KH algorithm was proposed by Jensi and Jiji [9] and applied to data clustering.

The algorithm continuously searches around the original area to guide the krill herd to the global optimal movement. It defines a new step size formula, which is convenient for krill individuals to fine tune their position in the search space. At the same time, the elite selection strategy is introduced into the krill herd update process, which is helpful for the algorithm to jump out of the local optimal solution. Experimental results show that the improved KH algorithm has higher accuracy and better robustness.

In [45], Wang et al. proposed a stud KH algorithm. The method adopts a new krill herd genetics and reproduction mechanism, replacing the random selection in the standard KH algorithm with columnar selection operator and crossover operator. To balance the exploration and exploitation abilities of the KH algorithm, Li et al. proposed a linear decreasing step KH algorithm [46]. In the algorithm, the step size scaling factor is improved linearly, which makes it decrease with the increase of iteration times, thereby enhancing the search ability of the algorithm.

Although KH algorithm and its enhanced version show better performance than other swarm intelligence algorithms, there are still deficiencies such as unbalanced exploration and exploitation. In this paper, to minimize the number of selected features and achieve high classification accuracy, both parameters are introduced into the fitness evaluation function. The physical diffusion motion of krill individuals is nonlinearly improved to dynamically adjust the random diffusion amplitude to accelerate the convergence rate of the algorithm. At the same time, a linear nearest neighbor lasso step optimization is performed on the basis of updating the position of the krill herd, which effectively enhances the global exploration ability. It helps the algorithm achieve better performance, reduce the data dimension of feature selection, and improve the efficiency of intrusion detection.

3. Algorithm Design

In this section, we first provide a brief description of the KH algorithm; subsequently, we present an improved version of KH, named LNNLS-KH, to address the problem of large number and high dimension in feature selection of intrusion detection.

3.1. Standard KH Algorithm. The framework of KH algorithm is shown in Figure 3. It includes three actions of krill individual, crossover operation, and updating position and calculating the fitness function. Krill individuals change their position according to three actions after completing initialization. Then, the crossover operator is executed to complete the position update and the new fitness function is calculated. If the number of iterations does not reach the maximum, krill individuals repeat the process until the iteration is completed.

As a novel biologically inspired algorithm for solving optimization tasks, the KH algorithm expresses the possible solution of the problem with each krill individual. By

simulating the foraging behavior, the krill herd position is continuously updated to obtain the global optimal solution. The motions of krill individuals are mainly affected by the following three aspects:

- (1) Movement induced by other krill individuals
- (2) Foraging activity
- (3) Physical diffusion motion

The KH algorithm adopts the Lagrange model to search in multidimensional space. The position update of krill individuals is shown as follows:

$$\frac{dX_i}{dt} = N_i + F_i + D_i, \quad (1)$$

where $X_i = \{X_{i,1}, X_{i,2}, \dots, X_{i,NN}\}$, N_i is the movement induced by other krill individuals, F_i is the foraging activity of krill individual, and D_i is random physical diffusion based on density region.

3.1.1. Movement Induced by Other Krill Individuals. The movement induced by other krill individuals is described as follows:

$$N_i^{\text{new}} = N^{\text{max}} \alpha_i + \omega_n N_i^{\text{old}}, \quad (2)$$

$$\alpha_i = \alpha_i^{\text{local}} + \alpha_i^{\text{target}}, \quad (3)$$

where N^{max} is the maximum induction velocity of surrounding krill individuals and it is taken $0.01 \text{ (ms}^{-1}\text{)}$ [5], ω_n represents the inertial weight in the range $[0, 1]$, N_i^{old} is the result of last motion induced by other krill individuals, α_i^{local} is a parameter indicating the direction of guidance, and α_i^{target} is the direction effect of the global optimal krill individual.

α_i^{local} is defined as follows:

$$\alpha_i^{\text{local}} = \sum_{j=1}^{\text{NN}} \widehat{K}_{i,j} \widehat{X}_{i,j}, \widehat{X}_{i,j} = \frac{X_j - X_i}{\|X_j - X_i\| + \varepsilon}, \widehat{K}_{i,j} = \frac{K_i - K_j}{K^{\text{worst}} - K^{\text{best}}}, \quad (4)$$

where K^{best} and K^{worst} are the best and worst fitness value of krill herd, K_i is the fitness value of i th krill individual, K_j represents the fitness value of i th neighbor krill individual ($j = 1, 2, \dots, \text{NN}$), and NN represents the total amount of neighbors. The ε at the denominator position is a small positive number to avoid the singularity caused by zero denominator.

When selecting surrounding krill individuals, the KH algorithm finds the number of nearest neighbors to krill individual i th by defining the ‘‘neighborhood ratio.’’ It is a circular area with krill individual i_{th} as the center and perception distance $d_{s,i}$ as the radius. $d_{s,i}$ is described as follows:

$$d_{s,i} = \frac{1}{5N} \sum_{j=1}^N \|X_i - X_j\|, \quad (5)$$

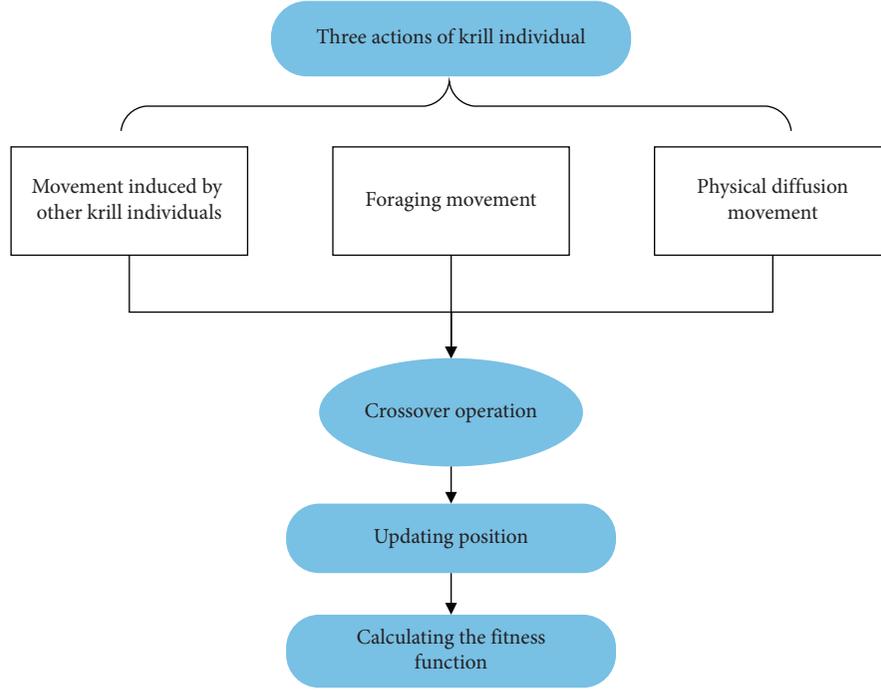


FIGURE 3: The framework of KH algorithm.

where N is the amount of krill individuals and X_i and X_j represent the position of i th and j th krill individuals. α_i^{target} is defined as follows:

$$\alpha_i^{\text{target}} = C^{\text{best}} \widehat{K}_{i,\text{best}} \widehat{X}_{i,\text{best}}, \quad (6)$$

where C^{best} is the effective coefficient between i th and global optimal krill individuals:

$$C^{\text{best}} = 2 \left(\text{rand} + \frac{I}{I_{\text{max}}} \right), \quad (7)$$

where I is the number of iterations, I_{max} is the maximum number of iterations, and rand is a random number between $[0, 1]$, which is used to enhance the exploration ability.

3.1.2. Foraging Activity. Foraging activity is affected by food distance and experience of food location, and it is described as follows:

$$F_i = V_f \beta_i + \omega_f F_i^{\text{old}}, \quad (8)$$

$$\beta_i = \beta_i^{\text{food}} + \beta_i^{\text{best}}, \quad (9)$$

where V_f is foraging speed and it is taken $0.02 \text{ (ms}^{-1}\text{)}$ [41], ω_f is inertia weight in the range $[0, 1]$, and β_i indicates foraging direction and it consists of food induction direction β_i^{food} and the historically optimal krill individual induction direction β_i^{best} . The essence of food is a virtual location, using the concept of ‘‘centroid.’’ It is defined as follows:

$$X^{\text{food}} = \frac{\sum_{i=1}^N (1/K_i) X_i}{\sum_{i=1}^N 1/K_i}. \quad (10)$$

- (1) The induced direction of food to i_{th} krill individual is expressed as follows:

$$\beta_i^{\text{food}} = C^{\text{food}} \widehat{K}_{i,\text{food}} \widehat{X}_{i,\text{food}}, \quad (11)$$

where C^{food} is the food coefficient, and it is determined as follows:

$$C^{\text{food}} = 2 \left(1 - \frac{I}{I_{\text{max}}} \right). \quad (12)$$

- (2) The induced direction of historical best krill individual to i_{th} krill individual is expressed as follows:

$$\beta_i^{\text{best}} = \widehat{K}_{i,\text{best}} \widehat{X}_{i,\text{best}}, \quad (13)$$

where $\widehat{K}_{i,\text{best}}$ represents the historical best individual influence on i th krill individual.

3.1.3. Physical Diffusion Motion. Physical diffusion is a stochastic process. The expression is as follows:

$$D_i = D^{\text{max}} \left(1 - \frac{I}{I_{\text{max}}} \right) \delta, \quad (14)$$

where D^{max} is the maximum diffusion velocity in the range $[0.002, 0.010] \text{ (ms}^{-1}\text{)}$. According to [41], it is taken

$0.005 \text{ (ms}^{-1}\text{)}$. δ represents the random direction vector and the value is taken the random between $[-1, 1]$.

3.1.4. Crossover. Crossover operator is an effective global optimization strategy. An adaptive vectorization crossover scheme is added to the standard KH algorithm to further enhance the global search ability of the algorithm [41]. It is given as follows:

$$X_{i,m} = \begin{cases} X_{i,m} * Cr + X_{r,m} * (1 - Cr), & \text{rand}_{i,m} < Cr, \\ X_{i,m}, & \text{else,} \end{cases}$$

$$Cr = 0.2\hat{K}_{i,\text{best}}, \quad (15)$$

where r is a random number and $r \in [1, 2, \dots, i-1, i+1, \dots, N]$, $X_{i,m}$ represents the m th dimension of the i th krill individual, $X_{r,m}$ represents the m th dimension of the r th krill individual, and Cr is the crossover probability, which decreases as the fitness increases and the globally optimal crossover probability is zero.

3.1.5. Movement Process of KH Algorithm. Affected by the movement induced by other krill individuals, foraging activity, and physical diffusion, the krill herd changed its position towards the direction of optimal fitness. The position vector of $[t, \Delta t]$ krill individual in interval $[t, \Delta t]$ is described as follows:

$$X_i(t + \Delta t) = X_i(t) + \Delta \frac{dX_i}{dt}, \quad (16)$$

where Δt is the scaling factor of the velocity vector. It completely depends on the search space:

$$\Delta t = C_t \sum_{j=1}^{NV} (UB_j - LB_j), \quad (17)$$

where NV represents the dimension of decision variables, LB_j and UB_j the upper and lower bounds of the j variable, $j = 1, 2, \dots, NV$, and C_t is the step scaling factor in the range $[0, 2]$.

3.2. The LNNLS-KH Algorithm. In view of the weakness of the unbalanced exploitation and exploration ability of KH algorithm, we propose the LNNLS-KH algorithm for feature selection to improve the performance and pursue high accuracy rate, high detection rate, and low false positive rate of intrusion detection. The improvement is reflected in the following three aspects.

3.2.1. A New Fitness Evaluation Function. To improve the classification accuracy of feature subset detection, we introduce the feature selection dimension and classification accuracy into fitness evaluation function. The specific expression of fitness is as follows:

$$\text{fitness} = \alpha * \frac{\text{Feature}_{\text{selected}}}{\text{Feature}_{\text{all}}} + (1 - \alpha) * (1 - \text{Accuracy}), \quad (18)$$

where $\alpha \in [0, 1]$, which is a weighting factor used to tune the importance between the number of selected features and classification accuracy. $\text{Feature}_{\text{selected}}$ is the number of selected features, $\text{Feature}_{\text{all}}$ represents the total number of features, and Accuracy indicates the accuracy of classification results. Moreover, k-nearest neighbor (KNN) is used as the classification algorithm and the classification accuracy is defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (19)$$

where TP, TN, FP, and FN are defined in the confusion matrix, as shown in Table 2.

3.2.2. Nonlinear Optimization of Physical Diffusion Motion. The physical diffusion of krill herd is a random diffusion process. The closer the individuals are to the food, the less random the movement is. Due to the strong convergence of the algorithm, the movement of krill individuals presents a nonlinear change from quickness to slowness, and the fitness function gradually decreases with the convergence of the algorithm. According to equations (2) and (9), the movement induced by other krill individuals and foraging activity are nonlinear. In the physical diffusion equation (14), the diffusion velocity D_i of i th krill individual decreases linearly with the increase of iteration times. In order to fit the nonlinear motion of krill herd, we introduce the optimization coefficient λ and the fitness factor μ_{fit} of krill herd into the physical diffusion motion. The optimized physical diffusion motion expression is defined as follows:

$$D_i = D^{\max} \left[1 - \lambda \frac{I}{I_{\max}} - (1 - \lambda) \mu_{\text{fit}} \right] \delta, \quad (20)$$

where λ is in the range of $[0, 1]$ and μ_{fit} is defined as follows:

$$\mu_{\text{fit}} = \frac{K^{\text{best}}}{K_i}, \quad (21)$$

where K^{best} is the fitness value of the current optimal individual and K_i represents the fitness value of i th krill individual. As the number of iterations increases, K_i gradually decreases until approaches K^{best} . Therefore,

μ_{fit} is in the range of $(0, 1]$. Introduce the fitness factor μ_{fit} into equation (20) to get the new physical diffusion motion equation:

$$D_i = D^{\max} \left[1 - \lambda \frac{I}{I_{\max}} - (1 - \lambda) \frac{K^{\text{best}}}{K_i} \right]. \quad (22)$$

According to equation (22), the number of iterations is I , the fitness K_i of krill individual, and the fitness K^{best} of the current optimal krill individual jointly determine the

TABLE 2: Confusion matrix.

Confusion matrix		True condition	
		True condition positive	True condition negative
Predicted condition	Predicted condition positive	True positive (TP)	False positive (FP)
	Predicted condition negative	False negative (FN)	True negative (TN)

physical diffusion motion, so as to further adjust the random diffusion amplitude. In the early stage of the algorithm iteration, the number of iterations is small and the fitness value of the individual is large, so the fitness factor is small, which is conducive to a large random diffusion of the krill herd. As the number of iterations gradually increases, the algorithm converges quickly and the fitness of krill individuals approaches the global optimal solution. At the same time, the fitness factor increases nonlinearly, which makes the random diffusion more consistent with the movement process of krill individual.

To further evaluate the effect of the KH algorithm for nonlinear optimization of physical diffusion motion (NO-KH), we conducted experiments on two classical benchmark functions. $F_1(x)$ is the Ackley function, which is a unimodal benchmark function. $F_2(x)$ is the Schwefel 2.22 function, which is a multimodal benchmark function. The experimental parameters of $F_1(x)$ and $F_2(x)$ are shown in Table 3.

Figure 4 shows the Ackley function and the Schwefel 2.22 function graphs for $n = 2$. We use standard KH algorithm and NO-KH algorithm to find the optimal value on the unimodal benchmark function and multimodal benchmark function, respectively. The number of krill and iterations are set to 25 and 500. Table 4 shows the best value, worst value, mean value, and standard deviation, which are obtained by running the algorithms 20 times. We can see that compared with standard KH algorithm, NO-KH algorithm searches for the smaller optimal solutions on both the unimodal benchmark function and multimodal benchmark function, and its global exploration ability is improved. The smaller standard deviation obtained from repeated experiments shows that NO-KH algorithm has better stability. Therefore, nonlinear optimization of physical diffusion motion of KH algorithm is effective.

The above analysis shows introducing the optimization coefficient λ and the fitness factor μ_{fit} into the physical diffusion motion of the krill herd is conducive to dynamically adjusting the random diffusion amplitude of the krill individuals and accelerating the convergence speed of the algorithm. Meanwhile, it increases the nonlinearity of the physical diffusion motion and the global exploration ability of the algorithm.

3.2.3. Linear Nearest Neighbor Lasso Step Optimization.

When KH algorithm is used to solve the multidimensional complex function optimization problem, the local search ability is weak and the exploitation and exploration are difficult to balance. For enhancing the local exploitation and global exploration abilities of the algorithm, the influence of excellent neighbor individuals on the krill herd during evolution is considered and an improved KH algorithm is

proposed in [42]. The algorithm introduces the nearest neighbor lasso operator to mine the neighborhood of potential excellent individuals to improve the local search ability of krill individuals, but the random parameters introduced in the lasso operator increase the uncertainty of the algorithm. To cope with the problem, we introduce an improved krill herd based on linear nearest neighbor lasso step optimization (LNNLS-KH) to find the nearest neighbor of krill individuals after updating individual position and linearly move a defined step to derive better fitness value. With introducing the method of linearization, the nearest neighbor lasso step of the algorithm changes linearly with iteration times, accordingly balancing the exploitation and exploration ability of the algorithm. In the early iteration, the large linear nearest neighbor lasso step is selected to facilitate the krill individuals to quickly adjust their positions, so as to improve the search efficiency of algorithm. In the later stage of iteration, the nearest neighbor lasso step decreases linearly to obtain the global optimal solution.

In krill herd $X = \{X_1, X_2, \dots, X_n\}$, assuming that j_{th} krill individual is the nearest neighbor of i_{th} krill individual, the Euclidean distance between two krill individuals is defined as follows:

$$\text{distance}_{i,j} = \{X_i, X_j\}, \quad (23)$$

where $\{X_i, X_j\} \subset S$ and $i \neq j$. The equation of linear nearest neighbor lasso step is defined as follows:

$$\text{step} = \begin{cases} \frac{I}{I_{\max}} \times (X_i - X_j), & K_i > K_j, \\ \frac{I}{I_{\max}} \times (X_j - X_i), & K_j > K_i. \end{cases} \quad (24)$$

The fitness function is expressed as equation (18). Therefore, the smaller fitness value means that the number of feature selection is less under the condition of higher accuracy, i.e., the position of krill individual is better. The schematic diagram of LNNLS-KH is shown in Figure 5. The new position Y_k of j_{th} krill individual is expressed as follows:

$$Y_k = \begin{cases} X_j + \frac{I}{I_{\max}} \times (X_i - X_j), & K_i > K_j, \\ X_i + \frac{I}{I_{\max}} \times (X_j - X_i), & K_j > K_i. \end{cases} \quad (25)$$

Considering that the i_{th} and krill j_{th} individuals move to both ends of the food, the new position Y_k will be far from the optimal solution after the linear neighbor lasso step optimization processing, as shown in Figure 6.

TABLE 3: Benchmark functions in the experiment.

Benchmark functions	Dim	Range	f_{\min}
$F_1(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	10	$[-10, 10]$	0
$F_2(x) = -20 \exp(-0.2 \sqrt{(1/2) \sum_{i=1}^n x_i^2}) - ((1/n) \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	10	$[-32, 32]$	0

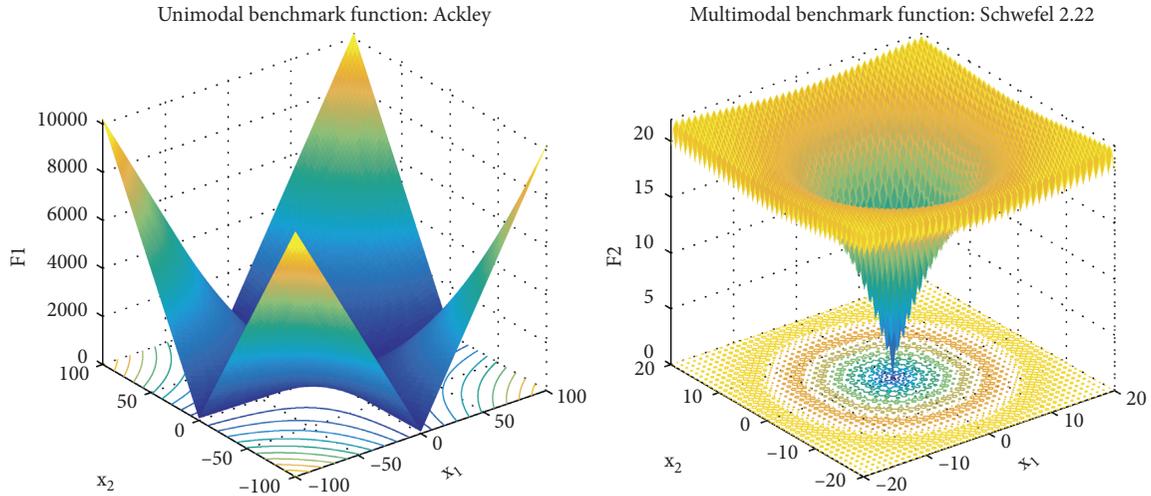
FIGURE 4: Ackley function and Schwefel 2.22 function graphs for $n = 2$. (a) Unimodal benchmark function: Ackley. (b) Multimodal benchmark function: Schwefel 2.22.

TABLE 4: The statistical results of KH and NO-KH algorithms on two benchmark functions.

$f(x)$	Algorithms	Best value	Worst value	Mean value	Standard deviation
F1	KH	1.692E-04	1.099E-02	1.508E-03	3.342E-03
	NO-KH	3.277E-05	9.632E-04	4.221E-04	3.908E-04
F2	KH	5.716E-05	2.168	0.329	0.816
	NO-KH	8.309E-06	1.155	0.116	0.362

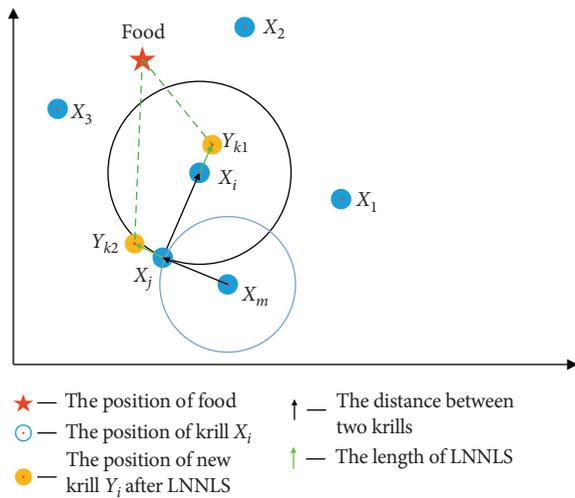


FIGURE 5: Optimization of linear nearest neighbor lasso step for krill individuals at the same end of food.

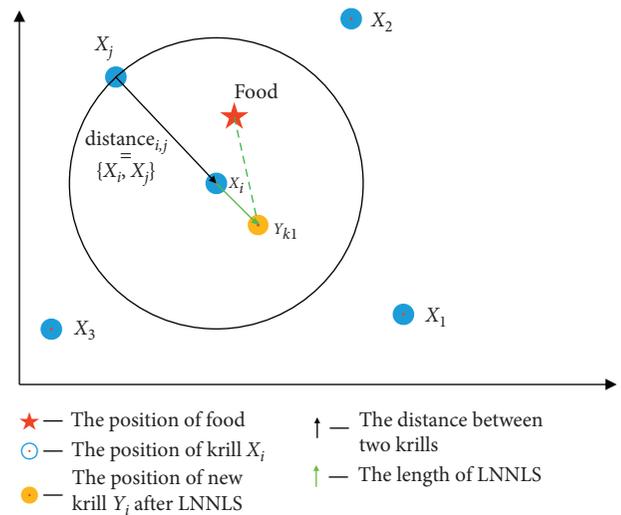


FIGURE 6: Optimization of linear neighboring lasso step for krill individuals at both ends of food.

The pseudocode of LNNLS-KH algorithm is shown in Algorithm 1.

3.3. Analysis of Time Complexity. In KH algorithm, each krill individual updates its position after movement which is induced by other krill individuals, foraging activity, and physical diffusion motion, with the time complexity of $O(N)$. After I_{\max} iterations, the time complexity of the algorithm is $O(I_{\max} \cdot N)$. In LNNLS-KH algorithm, the modified fitness function and the nonlinear optimization of physical diffusion motion hardly perform additional calculations, so the time complexity is not changed. In addition, the linear nearest neighbor lasso step optimization process of the algorithm adds the calculations of equations (24) and (25) after the krill individual completes the position update during iteration, and the time complexity is $O(I_{\max} \cdot N)$. Therefore, the total time complexity of the LNNLS-KM algorithm is $O(2I_{\max} \cdot N)$.

3.4. Description of the LNNLS-KH Algorithm for IDS Feature Selection. IDS is a system to recognize and process malicious usage of computers and network resources. The intrusion detection dataset records normal and abnormal traffic, including network traffic data and types of network attacks, and provides data support for the research and development of intrusion detection technology. IDS is generally composed of data acquisition, data preprocessing, detection units, and response actions, as shown in Figure 7.

The LNNLS-KH algorithm is used to select the high-quality feature subsets of IDS. The features of the intrusion detection dataset are randomly initialized to different real numbers in the range of $[0, 1]$, which constitute the position vectors of the krill herd. By calculating the fitness function and carrying out the LNNLS-KH algorithm, the position vectors of the krill herd are constantly updated. The fitness function is determined by the number of feature selection and the accuracy of classification, so the position vectors of the krill herd move toward the optimal fitness value. According to [47], it is appropriate to set the feature selection threshold to 0.7. When the maximum number of iterations is reached, the position vector of the krill population larger than the threshold is selected. The selected features constitute the feature subset of intrusion detection data. Furthermore, selected feature subset is sent to the detection units. In view of the K-Nearest Neighbor (KNN) algorithm which is relatively mature in theory, the detection units adopt KNN algorithm to construct intrusion detection classifier. Finally, the intrusion detection results are evaluated through test dataset. The process of LNNLS-KH algorithm for IDS feature selection is shown in Figure 8.

4. Results and Discussion

To verify the performance of the LNNLS-KH algorithm in IDS feature selection, we adopt the NSL-KDD network intrusion detection dataset and the CICIDS2017 dataset for experiments.

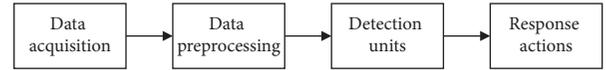


FIGURE 7: The framework of IDS.

4.1. Datasets Analysis. The NSL-KDD dataset is a classic dataset that has been used in the field of anomaly detection. As an improved version of the KDD CUP 99 dataset, it is currently one of the most reliable and influential intrusion detection datasets. Compared with the KDD CUP 99 dataset, the NSL-KDD dataset eliminates duplicate data, so the dataset hardly contains redundant records. Meanwhile, the proportion of each type of record in the NSL-KDD dataset has been adjusted to make the proportion of each type of data reasonable. Each record in the NSL-KDD dataset includes 41-dimensional features and a classification label. KDDTrain+ and KDDTest+ in the NSL-KDD dataset are selected as the training subset and the test subset. The types of attacks are divided into four types: denial of service (DoS), scan and probe (Probe), remote to local (R2L), and user to root (U2R). The detailed attack names and distribution of sample categories are shown in Tables 5 and 6. The features of NSL-KDD dataset are shown in Table 7.

The NSL-KDD dataset includes four types of features, which are the basic features of TCP connections (9 in total), the contents of TCP connections (13 in total), the time-based network traffic statistics (9 in total), and the host-based network traffic statistics (10 in total). Among all the features, “Protocol_type,” “service,” and “flag” are features of character types, which need to be preprocessed and mapped to ordered values. Because the mixed data types of numeric and character are difficult to deal with, the one-hot encoding is used to map different characters to different values. For example, the “Protocol_type” feature includes three types of protocol denoted by $icmp = [1, 0, 0]$, $tcp = [0, 1, 0]$, and $udp = [0, 0, 1]$. Similarly, the 70 attributes in “service” and the 11 attributes in “flag” are also numeralized in the same way. The 41-dimensional feature is expanded to 122-dimensional after one-hot encoding. At the same time, the dataset is normalized to eliminate the influence of features of different orders of magnitude on the calculation results, thus reducing the experimental error. The data preprocessing is helpful to improve the accuracy of classification and ensure the reliability of the results. The values corresponding to each feature are normalized to the interval $[0, 1]$, and the normalization expression is as follows:

$$X^* = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \quad (26)$$

where X^* is the normalized eigenvalue, X is the original eigenvalue, and X_{\max} and X_{\min} represents the maximum and minimum values in the same dimension feature.

Although NSL-KDD is a benchmark dataset in the field of network intrusion detection, some of the attack types are outdated due to the rapid development of network technology. Therefore, it hardly reflects the current real-network environment. CICIDS2017 is a novel network intrusion detection dataset released by the Canadian Institute for

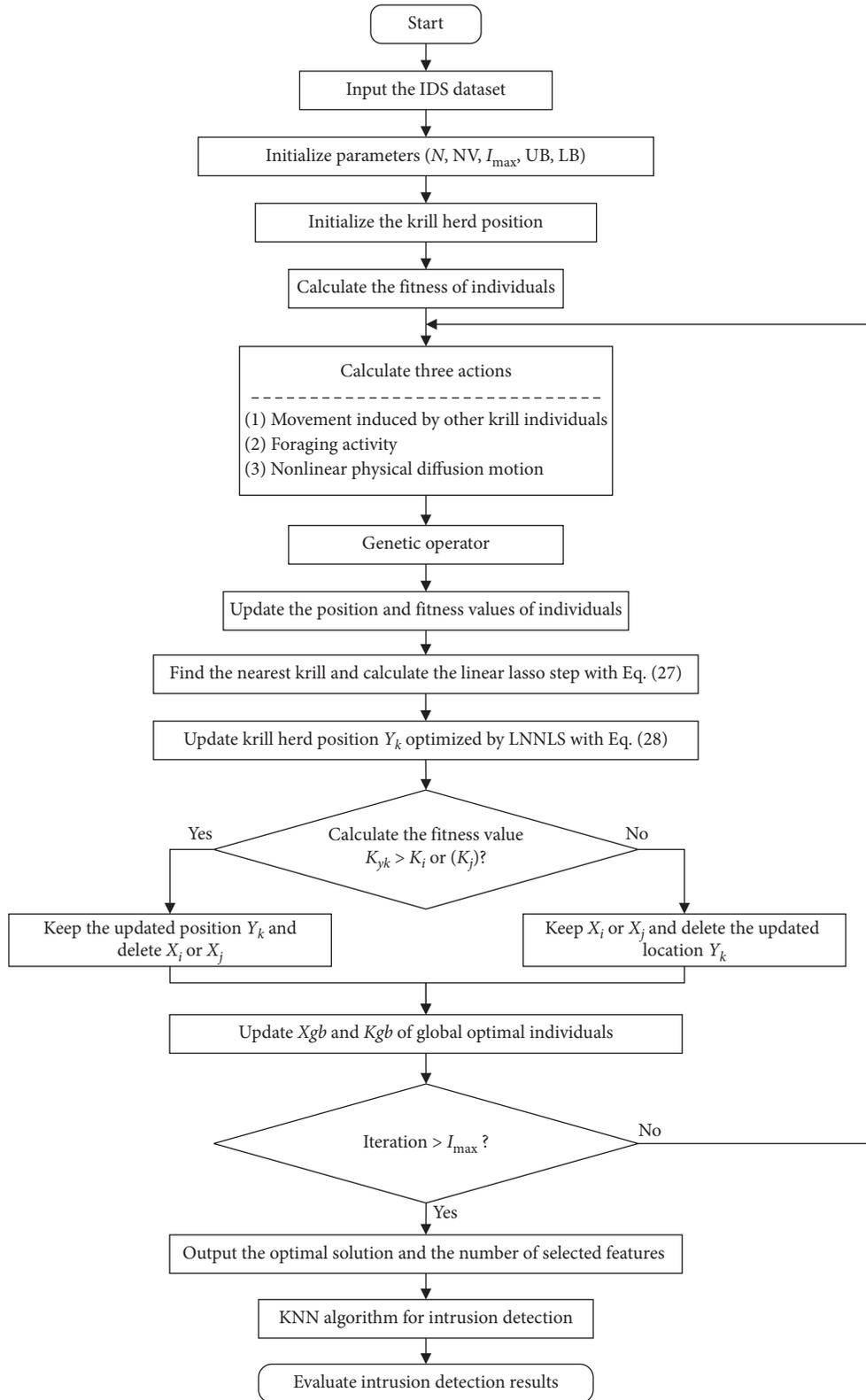


FIGURE 8: The process of LNNLS-KH algorithm for IDS feature selection.

Cybersecurity (CIC) in 2017. The dataset collected traffic data for five days, with only normal traffic on Monday and attacks occurring in the morning and afternoon from Tuesday to Friday. It includes “FTP patator,” “SSH patator,”

“DoS GoldenEye,” “DoS Slowhttptest,” “DoS Slowloris,” “Heartbleed,” “Web Attack Brute Force,” “Web Attack Sql Injection,” “Web Attack XSS,” “Infiltration Attack,” “Bot,” “DDoS,” and “PortScan,” which are common types of attacks

Input: Training set
Output: Global best solution, the number of selected features, and feature selection time
(1) Begin:
(2) Initialize algorithm parameters: $N^{\max}, V_f, D^{\max}, NV, I_{\max}, UB, LB$
(3) Initialize the krill herd position
(4) Evaluate the fitness of krill individuals and find the individuals with the best and worst fitness values
(5) for $I = 1$ to I_{\max} do
(6) for each krill individual $i (i = 1, 2, \dots, m)$ do
(7) Calculate the three components of motion:
(8) (1) The motion induced by other krill individuals
(9) (2) The foraging activity
(10) (3) The nonlinear optimized physical diffusion
(11) Implement crossover operator
(12) Update krill herd position and fitness values
(13) Calculate the linear nearest neighbor lasso step and new position using equations (24) and (25), and update new fitness values.
(14) if $Ky_k > K_i$ or (K_j)
(16) Leave K_i or (K_j) and delete Ky_k
(17) else
(18) Leave Ky_k and delete K_i or (K_j)
(19) end if
(19) end for
(20) Update X_{gb} and K_{gb} of the globally optimal individuals
(21) end for
(22) Output the global best solution, the number of selected features and feature selection time
(23) End

ALGORITHM 1: The LNNLS-KH algorithm.

TABLE 5: The distribution of sample categories.

Attack types	Attack names
DoS	Neptune, back, land, pod, smurf, teardrop, mailbomb, Apache2, processtable, udpstorm, worm
Probe	Ipsweep, nmap, portsweep, Satan, mscan, saint
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, httptunnel
U2R	buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, xterm

TABLE 6: The distribution of sample categories.

Data category	KDDTrain + samples	KDDTest + samples	Total number of samples
Normal	65120	11536	76656
DoS	36944	6251	43195
Probe	10786	2421	13207
R2L	995	2653	3648
U2R	52	67	119
All	113897	22928	136825

in modern networks. The distribution of attack time and types of CICIDS2017 dataset is shown in Table 8. We use the MachineLearningCVE file in the CICIDS2017 dataset as the dataset, which contains 78 features and an attack type label. The number and name of the feature are shown in Table 9. Compared with the NSL-KDD dataset, the attack types in the CICIDS2017 dataset are more in line with the situation of modern networks.

4.2. *Experimental Results and Discussion of NSL-KDD Dataset.* The experiment is conducted in MATLAB R2016a on Windows 64 bit operating system, with the processor of Intel (R) core (TM) i7-4790. Since the training of the algorithm requires normal and abnormal samples, we mix normal samples and different types of attack samples to construct train sets and test sets of four different attack types. In order to reduce the time of searching the optimal feature

TABLE 7: The features of NSL-KDD dataset.

Classification of features	Number	Serial number and name of features
The basic characteristics of TCP connections	9	(1) duration, (2) protocol_type, (3) service, (4) flag, (5) src_bytes, (6) dst_bytes, (7) land, (8) wrong_fragment, (9) urgent
The content characteristics of a TCP connection	13	(10) hot, (11) num_failed_logins, (12) logged_in, (13) num_compromised, (14) root_shell, (15) num_root, (16) su_attempted, (17) num_file_creations, (18) num_shells, (19) num_access_files, (20) num_outbound_cmds, (21) is_host_login, (22) is_guest_login
Time-based statistical characteristics of network traffic	9	(23) count, (24) srv_count, (25) serror_rate, (26) srv_serror_rate, (27) error_rate, (28) srv_error_rate, (29) same_srv_rate, (30) diff_srv_rate, (31) srv_diff_host_rate
Host-based network traffic statistics	10	(32) dst_host_count, (33) dst_host_srv_count, (34) dst_host_same_srv_rate, (35) dst_host_diff_srv_rate, (36) dst_host_same_src_port_rate, (37) dst_host_srv_diff_host_rate, (38) dst_host_serror_rate, (39) dst_host_srv_serror_rate, (40) dst_host_rerror_rate, (41) dst_host_srv_rerror_rate

TABLE 8: Attack time and attack types of the CICIDS2017 dataset.

Time	Type	Label	Amount	Total
Monday	Normal	BENIGN	529918	529918
Tuesday	Brute force	BENIGN	432074	445909
		FTP patator	7938	
		SSH patator	5897	
Wednesday	DoS	BENIGN	440031	692703
		DoS GoldenEye	10293	
		DoS slowhttptest	5499	
		Dos slowloris	5796	
		Heart bleed	11	
Thursday morning	Web attack	BENIGN	168186	170366
		Web attack brute force	1507	
		Web attack sql injection	21	
Thursday afternoon	Infiltration	Web attack XSS	652	288602
		BENIGN	288566	
		Infiltrationdnt	36	
Friday morning	Botnet	BENIGN	189067	191033
		Bot	1966	
Friday afternoon (1)	DDoS	BENIGN	97718	225745
		DDoS	128027	
Friday afternoon (2)	PortScan	BENIGN	127537	286467
		PortScan	158930	

subset, we randomly select 50% of Probe attack samples, 10% of DoS attack samples, 100% of U2R attack samples, and 100% of R2L attack samples in the KDDTraint + dataset as the training dataset, 100% of Probe dataset, 50% of DoS dataset, 100% of U2R dataset, and 20% of R2L dataset in the KDDTest + dataset as test dataset.

For the LNNLS-KH algorithm, the maximum number of iterations I_{\max} and quantity of krill individuals N are set to be 100 and 30, respectively. In [41], the foraging speed of krill individuals V_f is set to be 0.02, the maximum random diffusion rate D_{\max} is set to be 0.05, and the maximum induction speed N_{\max} is set to be 0.01. In [47], the threshold θ is set to be 0.7. As the LNNLS-KH algorithm is preferentially designed to ensure high accuracy and posteriorly reduce the number of features, the weight factor α in fitness function is set to be 0.02:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}, \quad (27)$$

$$\text{DR} = \frac{\text{TR}}{\text{TP} + \text{FN}}. \quad (28)$$

We adopt the iterative curve of global optimal fitness value, feature selection time, test set detection time, data dimension after feature selection, classification accuracy, detection rate (DR), and false positive rate (FPR) as evaluation measures of feature selection for IDS. The accuracy represents the ratio of the correctly classified samples to the total number of samples, which is defined as equation (19). FPR is also known as false alarm rate (FAR), which represents the ratio of samples that are incorrectly detected as intrusions to all normal samples, as shown in

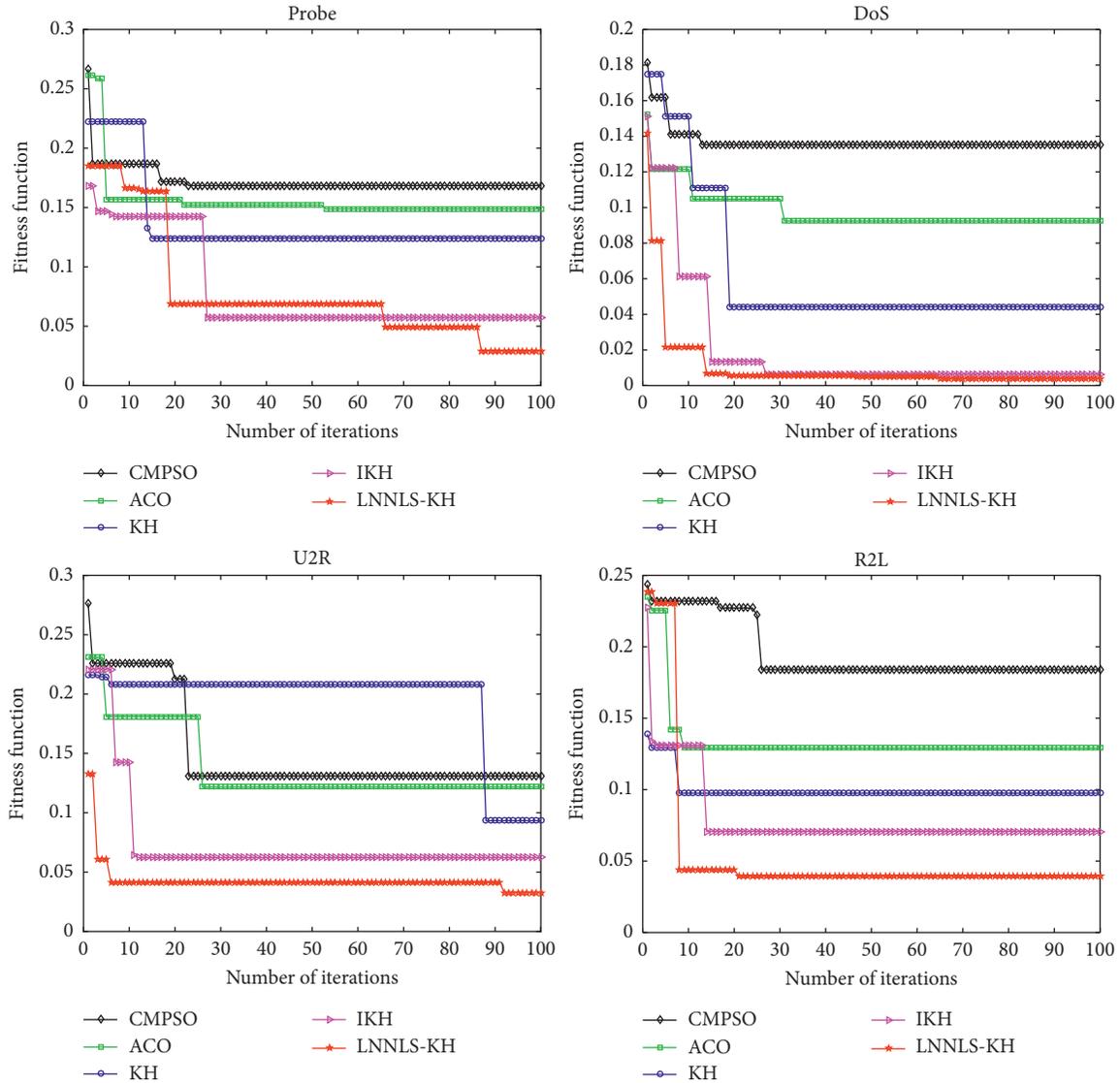


FIGURE 9: Convergence curve of fitness functions for the four attack datasets.

equation (27). DR, also known as recall or sensitivity, represents the probability of being correctly detected in all abnormalities, as shown in equation (28). The crossover-mutation PSO (CMPSO) algorithm [47], ACO algorithm [48], KH algorithm [41], and IKH algorithm [9] are set to be comparative experiments. The experimental results of Probe, DoS, R2L, and U2R dataset are shown as follows.

For reflecting the performance of the LNNLS-KH algorithm intuitively, the convergence curves of fitness function for Probe, DoS, U2R, and R2L datasets are shown in Figure 9. The results show that LNNLS-KH algorithm achieves a good fitness function value when the number of iterations reaches about 20, which demonstrates the strong exploitation ability and good convergence performance of the LNNLS-KH algorithm. As the number of iterations increases, other algorithms show varying degrees of convergence stagnation, while LNNLS-KH algorithm constantly jumps out of local optimum and finds the global optimal solution with better fitness. The fitness function values after

100 iterations achieve 0.0328, 0.0393, 0.0292, and 0.0036, respectively, for the four attack datasets, showing excellent exploration ability. Therefore, compared with the CMPSO, ACO, KH, and IKH algorithms, the LNNLS-KH algorithm exhibits faster convergence speed and stronger abilities of exploitation and exploration.

The results of different feature selection algorithms are shown in Table 10. The bold number in front of the brackets indicates the quantity of features after feature selection, and the specific feature numbers are listed in the brackets. The comparison of feature selection dimensions is shown in Figure 10, and different colours are used to distinguish the five algorithms. Obviously, the proposed LNNLS-KH algorithm marked in red is in the innermost circle of Figure 10 for Probe, DoS, U2R, and R2L datasets. It indicates that compared with the other four feature selection algorithms, LNNLS-KH algorithm retains the least features while ensuring accuracy. According to Figure 10, LNNLS-KH algorithm selects the average 7 main features of the NSL-KDD dataset, accounting

TABLE 9: The number and name of the features in the CICIDS2017 dataset.

Feature number	Feature name	Feature number	Feature name	Feature number	Feature name
1	Destination port	27	Bwd IAT mean	53	Average packet size
2	Flow duration	28	Bwd IAT std	54	Avg fwd segment size
3	Total fwd packets	29	Bwd IAT max	55	Avg bwd segment size
4	Total backward packets	30	Bwd IAT min	56	Fwd header length
5	Total length of fwd packets	31	Fwd PSH flags	57	Fwd avg bytes/bulk
6	Total length of bwd packets	32	Bwd PSH flags	58	Fwd avg packets/bulk
7	Fwd packet length max	33	Fwd URG flags	59	Fwd avg bulk rate
8	Fwd packet length min	34	Bwd URG flags	60	Bwd avg bytes/bulk
9	Fwd packet length mean	35	Fwd header length	61	Bwd avg packets/bulk
10	Fwd packet length std	36	Bwd header length	62	Bwd avg bulk rate
11	Bwd packet length max	37	Fwd Packets/s	63	Subflow fwd packets
12	Bwd packet length min	38	Bwd Packets/s	64	Subflow fwd bytes
13	Bwd packet length mean	39	Min packet length	65	Subflow bwd packets
14	Bwd packet length std	40	Max packet length	66	Subflow bwd bytes
15	Flow bytes/s	41	Packet length mean	67	Init_Win_bytes_forward
16	Flow packets/s	42	Packet length std	68	Init_Win_bytes_backward
17	Flow IAT mean	43	Packet length variance	69	act_data_pkt_fwd
18	Flow IAT std	44	FIN flag count	70	min_seg_size_forward
19	Flow IAT max	45	SYN flag count	71	Active mean
20	Flow IAT min	46	RST flag count	72	Active std
21	Fwd IAT total	47	PSH flag count	73	Active max
22	Fwd IAT mean	48	ACK flag count	74	Active min
23	Fwd IAT std	49	URG flag count	75	Idle mean
24	Fwd IAT max	50	CWE flag count	76	Idle std
25	Fwd IAT min	51	ECE flag count	77	Idle max
26	Bwd IAT total	52	Down/up ratio	78	Idle min

for 17.07% of the total number of features. Compared with CMPSO, ACO, KH, and IKH algorithms, the proposed LNNLS-KH algorithm reduces the features of 44%, 42.86%, 34.88%, and 24.32%, respectively, in the dataset of four attack types. Meanwhile, the total number of features in the four types of attack datasets is reduced by 37.43%.

To further evaluate the performance of the feature selection algorithms, we show the feature selection time and detection time of five different algorithms in Table 11. Feature selection time represents the time of filtering out redundant features. The detection time represents the time from inputting the most representative feature subsets into KNN classifier to the end of detection. It can be seen from Table 11 that the feature selection time of standard KH algorithm is shorter than that of CMPSO algorithm and ACO algorithm, which indicates that KH algorithm achieves faster speed and better performance. In addition, compared with standard KH algorithm, the feature selection time of LNNLS-KH algorithm is longer, which is mainly due to the nonlinear optimization of physical diffusion motion and the optimization of linear neighbor lasso step after the krill herd position is updated. Although part of the feature selection time is increased, the convergence speed and global search ability are greatly improved. At the same time, LNNLS-KH algorithm removes redundant features, which considerably increases the detection speed. In comparison to other four feature selection algorithms, the detection time of LNNLS-KH algorithm is reduced by 16.83%, 16.91%, 8.94%, and 6.96% on average in test dataset samples of Probe, DoS, R2L, and U2R.

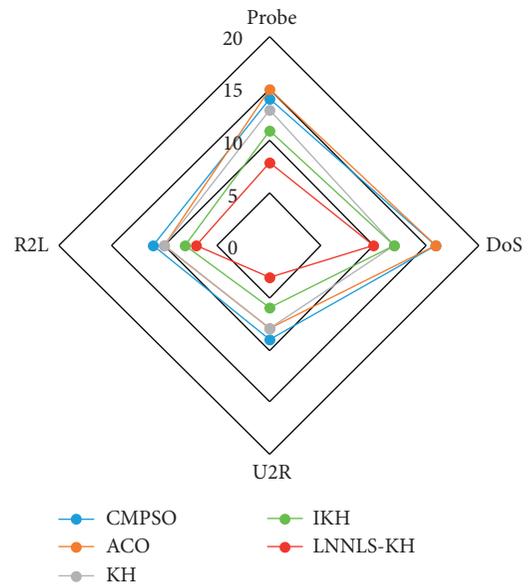


FIGURE 10: Comparison of feature selection dimensions produced by different algorithms.

The selection results of CMPSO, ACO, KH, IKH, and LNNLS-KH algorithms are used as feature subsets, and the test dataset is detected using KNN classifier. The classification accuracy of different algorithms is shown in Table 12. Comparing the accuracy of results, it is found that LNNLS-KH feature selection algorithm achieves a classification accuracy of above 90% for Probe, DoS, U2R, and R2L test

TABLE 10: The feature selection results of different feature selection algorithms (NSL-KDD dataset).

Data categories	CMPSO	ACO	KH	IKH	LNNLS-KH
Probe	14 (2, 3, 4, 7, 8, 10, 11, 17, 19, 20, 21, 27, 30, 33)	15 (1, 3, 4, 6, 15, 16, 17, 19, 21, 23, 29, 35, 39, 40, 41)	13 (3, 4, 5, 7, 8, 13, 14, 18, 19, 21, 26, 28, 40)	11 (2, 3, 5, 8, 10, 17, 18, 29, 34, 35, 41)	8 (3, 4, 8, 11, 15, 29, 34, 40)
DoS	16 (3, 4, 5, 6, 8, 13, 14, 17, 18, 22, 23, 26, 30, 32, 35, 41)	16 (3, 4, 7, 12, 14, 19, 20, 25, 27, 28, 30, 33, 34, 37, 40, 41)	12 (2, 3, 4, 5, 8, 9, 12, 15, 19, 24, 26, 30)	12 (2, 3, 4, 6, 12, 18, 20, 22, 27, 28, 30, 31)	10 (3, 4, 6, 15, 17, 19, 20, 21, 30, 37)
U2R	9 (3, 4, 5, 9, 12, 19, 32, 33, 41)	8 (3, 4, 6, 8, 20, 24, 33, 36)	8 (3, 4, 10, 12, 19, 23, 31, 32)	6 (3, 10, 11, 21, 36, 39)	3 (3, 33, 36)
R2L	11 (2, 3, 4, 8, 21, 22, 25, 27, 37, 40, 41)	10 (3, 4, 7, 12, 17, 21, 29, 37, 38, 40)	10 (2, 3, 4, 6, 13, 18, 19, 22, 32, 41)	8 (3, 4, 5, 8, 11, 14, 21, 31)	7 (2, 3, 4, 10, 15, 21, 36)

TABLE 11: Feature selection time and detection time of different feature selection algorithms (NSL-KDD dataset).

Data categories	Time of feature selection (second)					Time of detection (second)				
	CMPSO	ACO	KH	IKH	LNNLS-KH	CMPSO	ACO	KH	IKH	LNNLS-KH
Probe	5231.78	4998.14	4745.33	5348.87	5490.48	37.13	38.23	35.30	34.05	31.06
DoS	7892.35	7630.86	7168.52	8038.16	8296.92	118.69	118.15	106.66	105.14	98.44
U2R	154.87	147.29	144.18	157.79	172.24	0.087	0.086	0.086	0.086	0.078
R2L	2556.75	2369.08	2240.92	2669.51	2727.70	9.55	9.13	9.07	8.62	8.03

TABLE 12: The classification accuracy of different feature selection algorithms (NSL-KDD dataset).

Data categories	CMPSO (%)	ACO (%)	KH (%)	IKH (%)	LNNLS-KH (%)
Probe	80.46	86.56	92.42	93.74	98.24
DoS	81.74	83.36	86.03	88.74	97.01
U2R	82.74	84.57	85.59	91.89	95.67
R2L	78.70	81.62	88.78	90.49	93.56

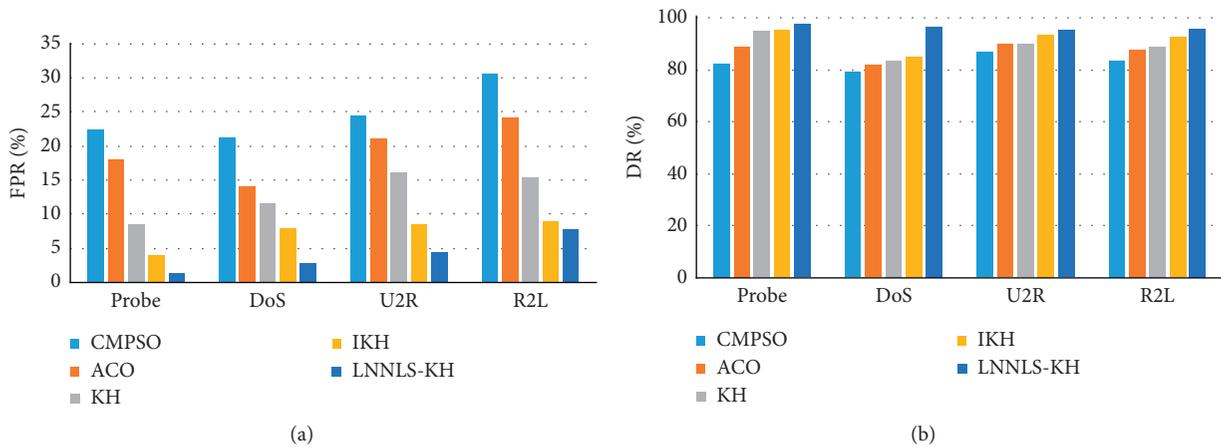


FIGURE 11: Comparison of classification FPR and DR of different feature selection algorithms: (a) FPR of different feature selection algorithms; (b) DR of different feature selection algorithms.

dataset samples. Furthermore, LNNLS-KH algorithm improves the average classification accuracy of Probe, DoS, U2R, and R2L test dataset samples by 9.95%, 12.04%, 9.47%, and 8.66%.

Table 13 shows the false positive rate and detection rate of feature subset produced by different feature selection algorithms. To visualize the difference, we show the

comparison in Figure 11. For Probe, DoS, U2R, and R2L datasets, the average false positive rate of LNNLS-KH feature selection algorithm is 4.00%. It reduces by 20.70%, 15.30%, 8.88%, and 3.34%, respectively, compared with CMPSO, ACO, and IKH algorithms. Similarly, for the detection rate, the proposed LNNLS-KH feature selection algorithm exhibits excellent performance. The average detection rate of

TABLE 13: The classification FPR and DR of different feature selection algorithms (NSL-KDD dataset).

Data categories	False positive rate (FPR) (%)					Detection rate (DR) (%)				
	CMPSO	ACO	KH	IKH	LNNLS-KH	CMPSO	ACO	KH	IKH	LNNLS-KH
Probe	22.37	18.04	8.50	4.05	1.18	82.32	89.18	95.01	95.22	97.73
DoS	21.27	14.08	11.45	7.88	2.85	79.12	82.08	83.77	85.23	96.80
U2R	24.51	21.04	16.13	8.45	4.30	87.02	89.79	90.14	93.67	95.52
R2L	30.66	24.05	15.42	8.99	7.67	83.56	87.56	88.91	92.89	95.85

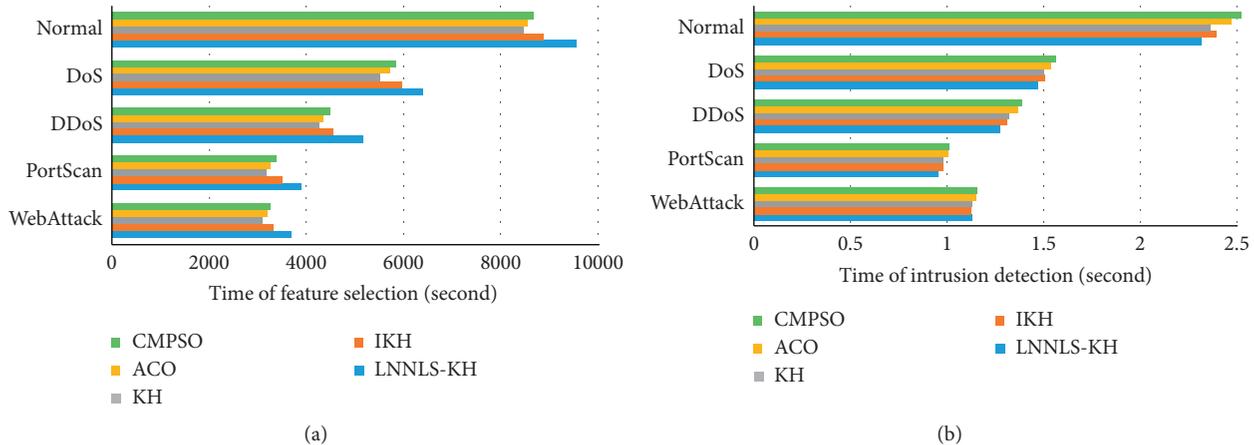


FIGURE 12: Comparison of feature selection time and intrusion detection time for different feature selection algorithms. (a) Feature selection time for different feature selection algorithms. (b) Intrusion detection time of different feature selection algorithms.

the LNNLS-KH algorithm is 96.48%, which is 13.47%, 9.32%, 7.02%, and 4.72% higher than the CMPSO, ACO, KH, and IKH feature selection algorithms, respectively.

In conclusion, LNNLS-KH feature selection algorithm performs excellent in the global optimal fitness iteration curve, test set detection time, number of dimensions of feature subset, classification accuracy, false positive rate, and detection rate. Although the offline training time of the LNNLS-KH algorithm is longer than the CMPSO, ACO, KH, and IKH algorithms, its lower feature dimension reduces the detection time. Moreover, the algorithm has faster convergence speed, higher detection accuracy, and lower classification false positive rate and detection rate.

4.3. Experimental Results and Discussion of CICIDS2017 Dataset. The experiment is conducted in MATLAB R2016a on Windows 64 bit operating system, with the processor of Intel (R) core (TM) i7-4790. The MachineLearningCVE file in the CICIDS2017 dataset includes 8 csv files of all traffic data, which contain 78 features plus an attack type tag by removing some duplicate features. We annotate traffic records according to different attack periods and types and standardize and normalize the dataset. Due to the excessive amount of data contained in the analyzed CSV file, problems such as excessively long time consuming and slow convergence rate of the model will occur when the host is used for model training. Therefore, we simplified and reintegrated these CSV data files while preserving the original attack

timing features. We selected a total of 12090 records and 5 types of traffic, including 1 type of normal traffic and 4 types of attack traffic, respectively: “DoS,” “DDoS,” “PortScan,” and “WebAttack.” The data are randomly divided into training sets and test sets in a 2 : 1 ratio with independent and repeated experiments.

CMPSO, ACO, KH, and IKH algorithms are used as the comparison of LNNLS-KH algorithm. The preprocessed Normal, DoS, DDoS, PortScan, and WebAttack subsets are input into the algorithm model successively, and the dimension and feature subsets of feature selection are obtained. We adopt the KNN classification model as the classifier and get the accuracy of intrusion detection through test set data. The results of feature selection dimension for the CICIDS2017 dataset are shown in Table 14. According to different attack types, LNNLS-KH algorithm selects different features. For example, the selected features of DOS subset are “Total Length of Bwd Packets,” “Fwd Packet Length Min,” “Flow IAT Min,” “FIN Flag Count,” “RST Flag Count,” “URG Packets/Bulk,” “Bwd Avg Packets/Bulk,” “Idle Mean” and “Idle Std.” For WebAttack subset, “Total Fwd Packets,” “Bwd IAT Max,” “Bwd PSH Flags,” “Fwd Packets/s,” “Bwd Avg Packets/Bulk,” “Subflow Fwd Bytes,” “Active Max,” and “Idle Max” are selected as attack features by LNNLS-KH algorithm. It reduces the feature dimension of IDS dataset while ensuring high accuracy. The average feature dimension selected by LNNLS-KH algorithm is 10.2, accounting for 13.08% of the total number of features in CICIDS2017 dataset. It decreases the

TABLE 14: The number of feature selection for different algorithms (CICIDS2017 dataset).

Data categories	CMPSO	ACO	KH	IKH	LNNLS-KH
Normal	28 (3, 7, 13, 15, 16, 17, 20, 22, 24, 26, 30, 35, 37, 38, 42, 43, 44, 45, 46, 49, 50, 56, 59, 62, 63, 64, 65, 76)	25 (1, 3, 4, 7, 10, 11, 12, 13, 15, 19, 29, 32, 34, 35, 37, 43, 46, 47, 51, 55, 56, 58, 73, 76, 78)	14 (11, 19, 33, 39, 43, 49, 55, 56, 58, 65, 66, 68, 71, 73)	14 (5, 10, 19, 20, 21, 23, 27, 33, 43, 56, 69, 70, 73, 78)	8 (6, 12, 16, 32, 38, 50, 54, 73)
DoS	24 (1, 3, 4, 13, 16, 17, 24, 26, 30, 33, 35, 39, 40, 44, 48, 51, 53, 57, 58, 59, 60, 62, 67, 70)	19 (3, 6, 12, 13, 15, 26, 35, 39, 51, 55, 60, 61, 66, 69, 71, 73, 75, 77, 78)	13 (8, 16, 21, 30, 45, 50, 52, 57, 59, 63, 66, 67)	14 (2, 12, 15, 16, 19, 21, 32, 34, 44, 46, 65, 68, 76, 77)	9 (6, 8, 20, 44, 46, 49, 61, 75, 76)
DDoS	29 (15, 18, 19, 20, 23, 25, 26, 33, 34, 35, 38, 39, 42, 43, 46, 47, 49, 51, 55, 56, 57, 59, 60, 61, 62, 63, 71, 72, 78)	27 (6, 9, 10, 13, 16, 19, 24, 28, 31, 41, 42, 45, 47, 48, 50, 51, 52, 53, 54, 56, 59, 60, 61, 62, 65, 68, 72)	21 (10, 12, 13, 15, 18, 23, 27, 30, 34, 35, 41, 42, 45, 55, 61, 63, 65, 66, 68, 70, 76)	18 (1, 11, 13, 14, 19, 24, 32, 35, 36, 40, 42, 47, 51, 57, 60, 69, 70, 75)	14 (2, 5, 8, 9, 11, 22, 26, 33, 41, 43, 47, 51, 74, 77)
PortScan	24 (1, 3, 6, 15, 16, 28, 30, 33, 35, 37, 44, 45, 52, 56, 59, 60, 61, 63, 65, 68, 70, 75, 77, 78)	21 (1, 2, 6, 10, 15, 17, 26, 27, 29, 39, 42, 43, 46, 49, 58, 61, 66, 69, 70, 71, 76)	14 (15, 20, 22, 27, 37, 44, 49, 50, 53, 59, 62, 65, 67, 78)	15 (1, 24, 30, 32, 33, 43, 49, 53, 54, 58, 60, 61, 63, 64, 69)	12 (2, 6, 15, 24, 25, 28, 32, 57, 59, 63, 66, 76)
WebAttack	16 (2, 7, 26, 29, 45, 47, 50, 52, 53, 54, 63, 66, 68, 69, 72, 78)	15 (3, 9, 10, 12, 19, 26, 40, 46, 50, 54, 64, 65, 68, 69, 73)	8 (1, 17, 19, 36, 48, 49, 53, 60)	7 (14, 17, 35, 39, 44, 48, 54)	8 (3, 29, 32, 37, 61, 64, 73, 77)

TABLE 15: The classification accuracy of different feature selection algorithms (CICIDS2017 dataset).

Data categories	CMPSO (%)	ACO (%)	KH (%)	IKH (%)	LNNLS-KH (%)
Normal	89.78	89.06	92.70	94.58	94.64
DoS	77.03	82.69	90.90	93.34	94.51
DDoS	81.73	86.94	91.85	88.19	95.76
PortScan	92.38	95.64	95.05	97.35	97.55
WebAttack	89.12	93.08	93.77	94.26	96.85

TABLE 16: The classification FPR and DR of different feature selection algorithms (CICIDS2017 dataset).

Data categories	False positive rate (FPR) (%)					Detection rate (DR) (%)				
	CMPSO	ACO	KH	IKH	LNNLS-KH	CMPSO	ACO	KH	IKH	LNNLS-KH
Normal	9.25	8.72	6.41	4.93	3.67	88.05	88.51	89.25	92.46	93.89
DoS	5.41	4.48	4.06	2.83	1.94	72.57	82.89	87.86	92.56	92.64
DDoS	6.85	4.92	4.54	6.33	3.18	79.03	83.47	90.22	87.52	92.98
PortScan	4.65	3.02	2.84	1.86	1.16	88.25	93.80	94.33	95.14	95.42
WebAttack	5.33	3.16	2.52	2.11	1.60	87.40	91.35	92.19	92.94	94.77

number of features by 57.85%, 52.34%, 27.14%, and 25%, respectively, compared with the CMPSO, ACO, KH, and IKH algorithms.

Figure 12 shows the feature selection time and intrusion detection time of 5 different feature selection algorithms to further evaluate the performance of the feature selection algorithm. It can be seen from Figure 12(a) that, in the feature selection stage, the LNNLS-KH algorithm consumes a long time in finding the optimal feature subset due to the linear nearest neighbor lasso step optimization after the position update of the krill herd. Compared with the KH and IKH algorithms, it increases the time by an average of 14.38% and 9.32%. Although the LNNLS-KH algorithm occupies more calculation time, the convergence speed and global search ability have been improved. Figure 12(b) shows the intrusion detection time of 5 different feature selection algorithms. It is the detection time of the sample dataset by the KNN classifier after the feature subset is searched,

excluding the time of searching for the optimal feature subset. The feature dimension of LNNLS-KH algorithm is low and the amount of data processed in the classification of detection sample dataset is small, which result *s* in the reduction of classification detection time. Compared with the CMPSO, ACO, KH, and IKH algorithms, the intrusion detection time of the LNNLS-KH algorithm is reduced by 6.52%, 5.17%, 2.14%, and 2.28% on average.

The selection results of CMPSO, ACO, KH, IKH, and LNNLS-KH algorithms are used as feature subsets, and the KNN classifier is used to detect the test dataset. The classification accuracy of different algorithms is shown in Table 15. For five types of subsets, the average classification accuracy of the proposed LNNLS-KH algorithm is 95.86%. In particular, the classification accuracy reached 97.55% for the PortScan subset. Compared with the other four feature selection methods, the LNNLS-KH algorithm has an average increase of 3.11%, 8.52%, 8.58%, 2.45%, and 4.29% on the

Normal, DoS, DDoS, PortScan, and WebAttack subsets, respectively. Table 16 shows the classification FPR and DR of different feature selection algorithms on the test sets. Based on the detection of five different test sets, the LNNLS-KH algorithm has lower FPR and higher DR than other four algorithms.

We propose the LNNLS-KH algorithm, a novel feature selection algorithm for intrusion detection. Experiments based on NSL-KDD and CICIDS2017 datasets show that the algorithm has good feature selection performance and improves the efficiency of intrusion detection.

5. Conclusions

With the rapid development of network technology, intrusion detection plays an increasingly important role in network security. However, the “dimensional disaster” was caused by massive data results in problems such as slow response and poor accuracy of the intrusion detection system. KH algorithm is a new swarm intelligence optimization method based on population, which shows good performance in high-dimensional data processing, providing a new approach for reducing the dimension of intrusion detection data and selecting useful features. In this paper, an improved KH algorithm, named LNNLS-KH, is proposed for feature selection of IDS datasets by linear nearest neighbor lasso optimization. The LNNLS-KH algorithm introduces a new fitness function which is composed of the number of feature selection dimensions and classification accuracy. Nonlinear optimization is introduced into the physical diffusion motion of krill individuals to accelerate the convergence speed of the algorithm. Moreover, the linear neighbor lasso step optimization is proposed to balance the exploration and exploitation abilities and obtain the global optimal solution of the feature subset effectively. Experiments based on NSL-KDD and CICIDS2017 datasets show that the LNNLS-KH algorithm retains 7 and 10.2 features on average, which greatly reduces the dimension of the features. In the NSL-KDD dataset, features are reduced by 44%, 42.86%, 34.88%, and 24.32% compared with CMPSO, ACO, KH, and IKH algorithms. And in the CICIDS2017 dataset, they are reduced by 57.85%, 52.34%, 27.14%, and 25%, respectively. In addition, the classification accuracy of the LNNLS-KH feature selection algorithm is increased by 10.03% and 5.39%, and the time of intrusion detection is reduced by 12.41% and 4.03% on the two datasets. Furthermore, LNNLS-KH algorithm enhances the ability of jumping out of the local optimal solution and shows good performance in the optimal fitness iteration curve, false positive rate of detection, and convergence speed, which demonstrated that the proposed LNNLS-KH algorithm is an efficient feature selection method for network intrusion detection.

In this research, we realized that the initialization of the LNNLS-KH algorithm has a certain degree of randomness. Therefore, we conducted independent and repeated experiments to solve the problem, and the results were reasonable and convincing. Although the proposed algorithm shows encouraging performance, it could be further improved.

In future work, we consider using data balancing techniques to preprocess the experimental dataset to obtain more accurate feature selection results and stronger algorithm stability. Meanwhile, we will combine the LNNLS-KH with other algorithms to improve the exploration and exploitation abilities, thereby further shortening the time of training feature subset and classification detection. On the contrary, as the LNNLS-KH algorithm is universally applicable, the LNNLS-KH algorithm can be applied to more feature selection systems and solve optimization problems in other fields.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work was sponsored by the National Key Research and Development Program of China (Grants 2018YFB0804002 and 2017YFB0803204), National Natural Science Foundation of PR China (Grant 72001191), Henan Natural Science Foundation (Grant 202300410442), and Henan Philosophy and Social Science Program (Grant 2020CZH009).

References

- [1] W. Wei and C. Guo, “A text semantic topic discovery method based on the conditional co-occurrence degree,” *Neurocomputing*, vol. 368, pp. 11–24, 2019.
- [2] C.-R. Wang, R.-F. Xu, S.-J. Lee, and C.-H. Lee, “Network intrusion detection using equality constrained-optimization-based extreme learning machines,” *Knowledge-Based Systems*, vol. 147, pp. 68–80, 2018.
- [3] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and D. Gong, “A comprehensive review of krill herd algorithm: variants, hybrids and applications,” *Artificial Intelligence Review*, vol. 51, no. 1, pp. 119–148, 2019.
- [4] J. Amudhavel, D. Sathian, R. S. Raghav et al., “A fault tolerant distributed self-organization in peer to peer (p2p) using krill herd optimization,” in *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*, pp. 1–5, Unnao, India, 2015.
- [5] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, “Hybrid clustering analysis using improved krill herd algorithm,” *Applied Intelligence*, vol. 48, no. 11, pp. 4047–4071, 2018.
- [6] P. A. Kowalski and S. Łukaszik, “Training neural networks with krill herd algorithm,” *Neural Processing Letters*, vol. 44, no. 1, pp. 5–17, 2016.
- [7] C. Stasinakis, G. Sermpinis, I. Psaradellis, and T. Verousis, “Krill-Herd Support Vector Regression and heterogeneous autoregressive leverage: evidence from forecasting and trading commodities,” *Quantitative Finance*, vol. 16, no. 12, pp. 1901–1915, 2016.

- [8] L. Wang, P. Jia, T. Huang, S. Duan, J. Yan, and L. Wang, "A novel optimization technique to improve gas recognition by electronic noses based on the enhanced krill herd algorithm," *Sensors*, vol. 16, no. 8, p. 1275, 2016.
- [9] R. Jensi and G. W. Jiji, "An improved krill herd algorithm with global exploration capability for solving numerical function optimization problems and its application to data clustering," *Applied Soft Computing*, vol. 46, pp. 230–245, 2016.
- [10] H. Pulluri, R. Naresh, and V. Sharma, "Application of stud krill herd algorithm for solution of optimal power flow problems," *International Transactions on Electrical Energy Systems*, vol. 27, no. 6, Article ID e2316, 2017.
- [11] D. Rodrigues, L. A. M. Pereira, J. P. Papa et al., "A binary krill herd approach for feature selection," in *Proceedings of the 2014 22nd International Conference on Pattern Recognition*, pp. 1407–1412, IEEE, Stockholm, Sweden, August 2014.
- [12] A. Mukherjee and V. Mukherjee, "Chaotic krill herd algorithm for optimal reactive power dispatch considering FACTS devices," *Applied Soft Computing*, vol. 44, pp. 163–190, 2016.
- [13] S. Sun, H. Qi, F. Zhao, L. Ruan, and B. Li, "Inverse geometry design of two-dimensional complex radiative enclosures using krill herd optimization algorithm," *Applied Thermal Engineering*, vol. 98, pp. 1104–1115, 2016.
- [14] S. Sultana and P. K. Roy, "Oppositional krill herd algorithm for optimal location of capacitor with reconfiguration in radial distribution system," *International Journal of Electrical Power & Energy Systems*, vol. 74, pp. 78–90, 2016.
- [15] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Applied Sciences*, vol. 8, no. 9, 2018.
- [16] D. Smith, Q. Guan, and S. Fu, "An anomaly detection framework for autonomic management of compute cloud systems," in *Proceedings of the 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, pp. 376–381, IEEE, Seoul, South Korea, July 2010.
- [17] Y. Zhao, Y. Zhang, W. Tong et al., "An improved feature selection algorithm based on MAHALANOBIS distance for network intrusion detection," in *Proceedings of 2013 International Conference on Sensor Network Security Technology and Privacy Communication System*, pp. 69–73, IEEE, Nan-gang, China, May 2013.
- [18] P. Singh and A. Tiwari, "An efficient approach for intrusion detection in reduced features of KDD99 using ID3 and classification with KNN," in *Proceedings of the 2015 Second International Conference on Advances in Computing and Communication Engineering*, pp. 445–452, IEEE, Dehradun, India, May 2015.
- [19] M. A. Ambusaidi, 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.
- [20] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
- [21] Y. Xue, W. Jia, X. Zhao et al., "An evolutionary computation based feature selection method for intrusion detection," *Security and Communication Networks*, vol. 2018, Article ID 2492956, 10 pages, 2018.
- [22] Z. Shen, Y. Zhang, and W. Chen, "A bayesian classification intrusion detection method based on the fusion of PCA and LDA," *Security and Communication Networks*, vol. 2019, Article ID 6346708, 11 pages, 2019.
- [23] P. Sun, P. Liu, Q. Li et al., "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," *Security and Communication Networks*, vol. 2020, Article ID 8890306, 11 pages, 2020.
- [24] G. Farahani, "Feature selection based on cross-correlation for the intrusion detection system," *Security & Communication Networks*, vol. 2020, Article ID 8875404, 17 pages, 2020.
- [25] F. G. Mohammadi, M. H. Amini, and H. R. Arabnia, "Applications of nature-inspired algorithms for dimension Reduction: enabling efficient data analytics," in *Advances in Intelligent Systems and Computing, Optimization, Learning, and Control for Interdependent Complex Networks*, pp. 67–84, Springer, Cham, Switzerland, 2020.
- [26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-International Conference on Neural Networks*, no. 4, pp. 1942–1948, IEEE, Perth, WA, Australia, December 1995.
- [27] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [28] R. Rajabioun, "Cuckoo optimization algorithm," *Applied Soft Computing*, vol. 11, no. 8, pp. 5508–5518, 2011.
- [29] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 965–997, 2014.
- [30] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-tr06, Erciyes university, Engineering Faculty, Computer Engineering Department, Kayseri, Turkey, 2005.
- [31] W.-T. Pan, "A new Fruit Fly Optimization Algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.
- [32] R. Zhao and W. Tang, "Monkey algorithm for global numerical optimization," *Journal of Uncertain Systems*, vol. 2, no. 3, pp. 165–176, 2008.
- [33] X. S. Yang and X. He, "Bat algorithm: literature review and applications," *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, pp. 141–149, 2013.
- [34] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [35] K. Ahmed, A. E. Hassanien, and S. Bhattacharyya, "A novel chaotic chicken swarm optimization algorithm for feature selection," in *Proceedings of the 2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 259–264, IEEE, Kolkata, India, November 2017.
- [36] S. Tabakhi, P. Moradi, F. Akhlaghian et al., "An unsupervised feature selection algorithm based on ant colony optimization," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 112–123, 2014.
- [37] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Systems with Applications*, vol. 116, pp. 147–160, 2019.
- [38] C. Yan, J. Ma, H. Luo, and A. Patel, "Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets," *Chemo-metrics and Intelligent Laboratory Systems*, vol. 184, pp. 102–111, 2019.
- [39] G. I. Sayed, A. Tharwat, and A. E. Hassanien, "Chaotic dragonfly algorithm: an improved metaheuristic algorithm for

- feature selection,” *Applied Intelligence*, vol. 49, no. 1, pp. 188–205, 2019.
- [40] Z. Zhang, P. Wei, Y. Li et al., “Feature selection algorithm based on improved particle swarm joint taboo search,” *Journal of Communication*, vol. 39, no. 12, pp. 60–68, 2018.
- [41] A. H. Gandomi and A. H. Alavi, “Krill herd: a new bio-inspired optimization algorithm,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
- [42] Q. Tan and Z. Huang, “Krill herd with nearest neighbor lasso operator,” *Computer Engineering and Applications*, vol. 55, no. 9, pp. 124–129, 2019.
- [43] Q. Wang, C. Ding, and X. Wang, “A hybrid data clustering algorithm based on improved krill herd algorithm and KHM clustering,” *Control and Decision*, vol. 35, no. 10, pp. 2449–2458, 2018.
- [44] Q. Li and B. Liu, “Clustering using an improved krill herd algorithm,” *Algorithms*, vol. 10, no. 2, p. 56, 2017.
- [45] G.-G. Wang, A. H. Gandomi, and A. H. Alavi, “Stud krill herd algorithm,” *Neurocomputing*, vol. 128, pp. 363–370, 2014.
- [46] J. Li, Y. Tang, C. Hua, and X. Guan, “An improved krill herd algorithm: krill herd with linear decreasing step,” *Applied Mathematics and Computation*, vol. 234, pp. 356–367, 2014.
- [47] H. B. Nguyen, B. Xue, P. Andreae et al., “Particle swarm optimisation with genetic operators for feature selection,” in *Proceedings of the 17 IEEE Congress on Evolutionary Computation (CEC)*, pp. 286–293, IEEE, San Sebastian, Spain, June 2017.
- [48] M. H. Aghdam and P. Kabiri, “Feature selection for intrusion detection system using ant colony optimization,” *International Journal of Network Security*, vol. 18, no. 3, pp. 420–432, 2016.