SIN-CHN
SCIENTIFIC PRESS PTE.LTD.

Molecular & Cellular Biomechanics 2024, 21(3), 599.
https://doi.org/10.62617/mcb599

Article

Feature selection for intrusion detection based on an improved rime optimization algorithm

Qingyuan Peng, Xiaofeng Wang*, Ao Tang

School of Computer Science & Engineering, North Minzu University, Yinchuan 750021, China
*** Corresponding author:** Xiaofeng Wang, xfwang@nmu.edu.cn

CITATION

Peng Q, Wang X, Tang A. Feature selection for intrusion detection based on an improved rime optimization algorithm. Molecular & Cellular Biomechanics. 2024; 21(3): 599. https://doi.org/10.62617/mcb599

ARTICLE INFO

Received: 23 October 2024
Accepted: 1 November 2024
Available online: 22 November 2024

Abstract: With the rapid development of information technology, cybersecurity issues have become increasingly prominent, posing serious threats to national security, economic growth, and personal privacy. Intrusion detection systems have been widely applied to ensure network security and prevent malicious cyber-attacks. In intrusion detection, redundant and irrelevant features not only slow down the classification process but also hinder classifiers from making accurate decisions, resulting in decreased system performance. Addressing the problem of low accuracy in intrusion detection systems due to high-dimensional datasets, we propose a network intrusion detection method based on an enhanced Rime Optimization Algorithm for feature selection. Firstly, building upon the traditional Rime Optimization Algorithm, we introduce Cauchy mutation and differential mutation operations to improve both global and local search capabilities. Cauchy mutation introduces a heavy-tailed distribution to increase the probability of escaping local optima, while differential mutation, through the differential operator, further enhances solution diversity and algorithm convergence speed. Combining the two mutation operations, the optimization algorithm achieves a good balance between global search and local search, effectively avoids premature convergence and falling into local optimum, and effectively improves the feature selection results. Secondly, the improved Rime optimization algorithm (IRIME) was applied to the feature selection process of intrusion detection system, and it was combined with the decision tree classifier to construct a wrapper feature selection algorithm, which could directly optimize the classification task and avoid the mismatch between feature selection and classifier. The optimized algorithm can quickly select the most representative feature subset from the high-dimensional feature space, significantly reducing the computational cost. At the same time, the selected feature subset can more accurately reflect the inherent law of the data set, thereby improving the prediction accuracy of the classifier. Finally, NSL-KDD and UNSW-NB15 datasets were used for performance evaluation. Experimental results show that compared with several feature selection algorithms, the proposed method achieves the best binary classification performance after feature selection. Specifically, it is superior to other algorithms in terms of precision, accuracy, F1 score and recall of all evaluation metrics.

Keywords: rime optimization algorithm; differential mutation; Cauchy mutation; feature selection; intrusion detection

1. Introduction

Cybersecurity issues are becoming increasingly prominent with the rapid development and widespread application of Internet technology. Network intrusion behaviors exhibit a trend towards diversity and complexity, rendering traditional security measures such as firewalls and intrusion prevention systems ineffective against new network attacks [1]. Intrusion detection technology, a proactive defense mechanism in cybersecurity, has garnered widespread attention and research in recent

years. IDS (Intrusion Detection Systems) use specific devices or application software to predict malicious traffic behaviors within networks. The primary function of IDS is to analyze network traffic or system logs deeply, identify potential abnormal behaviors, and promptly issue alerts so that administrators can take corresponding defensive measures. This proactive defense approach effectively supplements the shortcomings of traditional security measures like firewalls and intrusion prevention systems in combating new network attacks, thereby crucial in ensuring network security.

Traditional intrusion detection methods mainly include signature-based detection and anomaly-based detection. Signature-based detection methods identify intrusion behaviors by matching predefined attack features or patterns. These signatures represent known attack characteristics or rules, typically created by security experts based on historical attack data. This method excels in accurately detecting known attacks with low false positive rates. However, its drawback lies in its inability to detect unknown or variant attacks not included in the predefined signature database. Anomaly-based detection methods establish a baseline of normal behavior and identify potential intrusions by detecting deviations from this baseline. While capable of detecting unknown and novel attacks, this method produces higher false positive rates because certain normal behavioral changes may be mistaken for attacks [2].

In the context of intrusion detection, high-dimensional data often contain numerous redundant or irrelevant features that may impede classification speed and diminish accuracy. Therefore, data dimensionality is a critical factor in intrusion detection. In the era of big data, reducing the dimensions of large datasets is crucial for improving the performance of classification systems [3,4]. Dimensionality reduction is a key challenge for any artificial intelligence system to overcome the so-called “curse of dimensionality”. One approach to achieving dimensionality reduction is feature selection (FS), which involves eliminating unnecessary features such as irrelevant and redundant ones [5]. Thus, a fundamental preprocessing step in any data mining system is the feature selection process, aiming to obtain a minimal feature subset that maximizes classification performance. FS has been applied in various classification systems including medical diagnostics [6], malware detection [7], EEG signal denoising [8], biometrics [9], software defects [10], sports [11], and IDS [12].

Given that IDS (Intrusion Detection Systems) handle vast amounts of data, which may include false positives and unnecessary or repetitive features, a reliable feature selection method is crucial for boosting accuracy and speeding up training and testing processes. By selecting optimal features, IDS systems can lower processing expenses, cut down on storage needs, and enhance comprehension of test data. As emphasized by Mohammadi et al. [2], relevant features contain vital information that significantly aids in the classification task.

Feature selection methods primarily fall into three categories: Filter Method, Wrapper Method, and Embedded Method. Each method has its unique characteristics and applications. The Filter Method selects features before model training, independent of any machine learning algorithm, by evaluating the importance of each feature using statistical and scoring criteria. This method is fast and simple, suitable for high-dimensional datasets. Common filter methods include variance thresholding, correlation coefficient, chi-square test, information gain, and mutual information. The Wrapper Method evaluates the effectiveness of feature sets using specific machine

learning algorithms, typically through iterative searching to find the optimal feature subset. Although computationally costly, this method often provides better performance. Common wrapper methods include recursive feature elimination, forward selection, backward elimination, and exhaustive search. The Embedded Method performs feature selection during the model training process, synchronously with feature selection. Embedded methods typically utilize the internal mechanisms of models to evaluate feature importance. Common embedded methods include Lasso regression, decision trees, random forests, gradient boosting trees, etc.

Metaheuristic algorithms are predominantly employed in feature selection for intrusion detection systems, owing to their adaptive searching strategies and ability to explore the entire solution space [13]. Swarm Intelligence, drawing inspiration from insect and bee collective behaviors, represents a method of artificial intelligence tailored for tackling intricate problems. Examples commonly found in literature for delivering satisfactory solutions in feature selection include Particle Swarm Optimization (PSO) [14], Grey Wolf Optimization (GWO) [15], Harris Hawks Optimization (HHO) [16], Multi-Verse Optimization (MVO) [17], and Dual Throat Optimization (DTO) [18] algorithms. The Rime Optimization Algorithm (RIME) [19] used in this paper is a nature-inspired intelligent optimization algorithm that simulates the process of frost growth. It possesses excellent global search capabilities and adaptability, is simple to implement, and has robust parallel computing capabilities, thus offering significant advantages in solving complex optimization problems. However, all metaheuristic optimization algorithms must strike a balance between exploration and exploitation stages to avoid falling into local optima or failing to converge. The stochastic nature inherent in metaheuristic algorithmic solutions also poses challenges. To address this issue, this paper incorporates differential mutation and Cauchy mutation into the Rime Optimization Algorithm to avoid local optima.

This paper presents a wrapper-based approach to address feature selection (FS) challenges in Intrusion Detection Systems (IDS). It employs an enhanced version of the Rime Optimization Algorithm to identify the optimal set of features and employs the Decision Tree (DT) algorithm, a popular machine learning tool for classification, to evaluate the effectiveness of the selected feature set. The organization of this paper is as follows: Section 2 discusses recent research on feature selection methods that utilize intelligent optimization algorithms. Section 3 details the proposed methodology. Section 4 analyzes experimental results to validate the effectiveness of the proposed method. Finally, Section 5 summarizes the main findings and suggests directions for future research.

2. Literature review

Feature Selection is a vital data mining technique [20], and in recent times, there has been a growing focus on FS methods aimed at identifying malicious network attacks [21]. This is due to the detrimental impacts of network attacks on diverse systems, which can affect businesses, organizations, and even society at large [22]. Among the various technologies and algorithms developed, the incorporation of swarm intelligence and machine learning (classification) algorithms has significantly improved the detection precision of these methods.

Xue et al. [23] introduced a strategy to tackle large and intricate datasets characterized by numerous noisy features and multiple local optima within the feature space. Their approach is rooted in the Multi-Classifier Adaptive Parameter Particle Swarm Optimization (SPS-PSO) algorithm. Within SPS-PSO, a specific solution format and five methods for generating individual candidate solutions (CSGS) are employed. This algorithm can modify the CSGS and their parameters when dealing with large-scale feature selection challenges.

Chen et al. [24] proposed an innovative Particle Swarm Optimization (PSO) feature selection (FS) approach that iteratively enhances the quality of the population in each cycle. Guided by the relevance of current population information, an updated strategy is directed to generate superior solutions. Additionally, an agent-based solution selection strategy is employed to choose solutions with good convergence and diversity to form a new population. The results suggest that, in the majority of instances, this method is capable of choosing smaller feature sets while maintaining high classification accuracy.

Bonab et al. [25] developed an IDS system that selects optimal features using a combination of the Fruit Fly Algorithm (FFA) and Ant Lion Optimization (ALO). Evaluations on datasets like KDDCUP99, NSL-KDD, and UNSW-NB15 showed impressive results in terms of accuracy and sensitivity.

Zhou et al. [26] introduced a novel FS method, the Bat Algorithm (CFS-BA), based on ensemble learning and selecting relevant features. The CFS-BA algorithm determines the optimal feature subset by leveraging correlations between features. An ensemble classification system is built using classifiers like C4.5, RandomForest, and ForestPA.The results demonstrate accuracy levels of 99.8% for the NSL-KDD dataset, 99.52% for the AWID (AI-based Wi-Fi intrusion dataset), and 99.52% for the CIC-IDS2017 dataset.

Fatani et al. [27] created a feature extraction technique utilizing Convolutional Neural Networks (CNN). They utilized the Aquila Optimization Algorithm (AQU) to select features across four well-known public datasets: CIC2017, NSL-KDD, BoT-IoT, and KDD99. Comprehensive comparisons with various other optimization approaches, using diverse evaluation metrics, revealed that their method excelled in both multi-classification and binary classification contexts.

Mojtahedi [28] and colleagues introduced a method for selecting features by integrating the Whale Optimization Algorithm (WOA) with the Genetic Algorithm (GA), tailored for sample-centric classification approaches in network intrusion detection systems. Standard datasets such as KDDCUP1999 were used, and experimental results demonstrated higher accuracy than previous methods. The Whale Optimization Algorithm and Genetic Algorithm are capable of efficiently identifying features associated with class labels, whereas the KNN method is utilized to detect anomalous behavior nodes within wireless network intrusion detection datasets.

Nazir and Khan [29] introduced an FS method that combines Tabu Search with Random Forest (TS-RF). When tested on the UNSW-NB15 dataset, their method demonstrated enhanced classification accuracy, fewer selected features, and lower false positive rates.

Maazalahi [30] and his team introduced a two-stage hybrid approach that combines machine learning techniques with metaheuristic algorithms. In the initial

stage, data preprocessing was conducted using population-based metaheuristic algorithms, namely Atomic Search Optimization (ASO) and Equilibrium Optimization (EO), for selecting features aimed at achieving global optimization. The subsequent stage focused on detecting attacks by employing K-means clustering along with the Firefly Algorithm (FA). The performance of the proposed method was evaluated on the NSL-KDD, UNSW-NB15, and KDD CUP99 datasets, demonstrating superior accuracy and efficiency compared to alternative methods.

In summary, it is worth noting in the literature that there is no uniform feature or feature subset in the literature, and most papers only use performance metrics to define the fitness function, ignoring the number of features. Second, another challenge of using optimization problems for feature selection is the time complexity. Among other swarm intelligence algorithms, the Rime optimization algorithm is considered to be the best globally convergent algorithm. RIME algorithm is an innovative metaheuristic optimization technique, which ingeniously simulates the growth process of frost and ice in nature, especially the different formation mechanism of soft frost and hard frost, and constructs its unique search strategy based on it. In this process, the soft frost search strategy is designed to simulate the slow and uniform diffusion of frost and ice in the low temperature environment, which enables the algorithm to explore the solution space extensively in the initial stage to ensure that no potential high-quality solution regions are missed, while the hard frost piercing mechanism simulates the rapid and deep solidization of frost and ice under specific conditions. It enables the algorithm to accurately focus on those outstanding solution regions after exploring a certain depth, and perform deep mining and optimization. Furthermore, RIME innovates the traditional metaheuristic selection mechanism by introducing a positive greedy selection mechanism. The core of this mechanism is that it can dynamically evaluate the quality of the current solution and the possible improvement space in the future during the running of the algorithm, so as to avoid the premature convergence of the algorithm to the local optimal solution, the so-called “local optimum trap”. This mechanism not only enhances the global search ability of the algorithm, but also ensures that the algorithm can maintain a high degree of stability and robustness in the face of complex and multimodal optimization problems. Compared with classical algorithms such as Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Harris Hawk Optimization (HHO) and Moth Flame Optimization (MFO), RIME algorithm shows significant advantages. Its fast convergence speed enables it to find high-quality solutions in a shorter time. At the same time, RIME achieves a more skillful balance between exploration and utilization of resources, which means that it cannot only fully explore the solution space to discover new possibilities, but also effectively use the existing information to refine and optimize the solution. Therefore, when facing complex and high-dimensional optimization problems, RIME algorithm not only shows stronger optimization ability, but also shows higher stability and reliability. These characteristics make RIME algorithm have broad application prospects and potential value in many fields such as feature selection, parameter optimization, function optimization and so on.

Therefore, this paper introduces a feature selection approach tailored for intrusion detection systems, leveraging real-time optimization algorithms. The goal of this method is to enhance the precision and efficiency of intrusion detection by refining

the selection of feature subsets. Subsequently, a detailed description of the proposed method will be provided, encompassing data set collection and preprocessing, as well as the implementation of the RIME optimization algorithm and other crucial steps.

3 Methodology

3.1. Dataset collection

3.1.1. NSL-KDD

The KDD Cup 99 dataset originated from the 1999 International Knowledge Discovery and Data Mining Tools Competition (KDD Cup 1999) and is rooted in the DARPA 1998 intrusion detection evaluation initiative. Due to the excessive size and redundancy of the original dataset, researchers proposed an improved version, the NSL-KDD dataset. The NSL-KDD dataset removes duplicate records from the original KDD dataset, ensuring that classifiers do not achieve unfairly high accuracy due to repeated data. The NSL-KDD dataset is of moderate size, allowing researchers to conduct experiments without requiring substantial computational resources. The NSL-KDD dataset consists of 148,517 data records, with 125,973 records in the training set and 22,544 in the testing set. Each record contains 41 attributes, and the dataset labels indicate whether the network traffic record is normal or a type of attack. The number of normal and attack data is shown in **Table 1**.

Table 1. Data distribution of the NSL-KDD dataset.

Type	Normal	DoS	Probe	U2R	R2L
Training set	67,343	45,927	11,656	52	995
Test set	9711	7458	2421	200	2754

3.1.2. UNSW-NB15

The UNSW-NB15 dataset was developed in 2015 by the University of New South Wales (UNSW) and the Australian Defence Force Academy (ADFA) specifically for network intrusion detection. This dataset was generated in a simulated network environment that included realistic modern user activities and various malicious attack behaviors. Tools used included IXIA Perfect Storm, which can generate network traffic. The UNSW-NB15 dataset covers various modern network attack types and includes rich feature information that can be used for the development and evaluation of intrusion detection systems. The UNSW-NB15 dataset comprises 49 features per record and encompasses nine types of attacks along with normal traffic, as illustrated in **Table 2**.

Table 2. Data distribution of the UNSW-NB15 dataset.

Type	Normal	Fuzzers	Analysis	Backdoors	DoS	Exploits	Generic	Reconnaissance	Shellcode	Worms
Training Set	56,000	18,184	2000	1746	12,264	33,393	40,000	10,491	1133	130
Test Set	37,000	6062	677	583	4089	11,132	18,871	3496	378	44

3.2. Dataset preprocessing

Before proceeding with the training and testing process, several preprocessing

steps need to be applied to the dataset. First, class and data transmission are required to convert symbolic values into numerical ones. In the CLASS column, two types are designated: 0 for normal network traffic and 1 for malicious attack traffic. Data normalization, another crucial preprocessing step, adjusts or scales the data values of each feature to a certain range, thereby preventing any bias towards features with larger values within the dataset. Data normalization is performed using Equation (1) to normalize the datasets used to the range [0,1] [31].

$$X_{normalized} = \frac{X - X_{min}}{Xmin_{max}} \tag{1}$$

3.3. Rime optimization algorithm

In 2023, Su et al. [19] were inspired by the frost growth mechanism in nature and proposed the Rime Optimization Algorithm, drawing inspiration from the natural frost growth mechanism. The algorithm uses the randomness of soft frost and the regularity of hard frost for searching, enhancing the algorithm's performance through soft frost search strategies and hard frost puncture mechanisms. An active greedy selection mechanism is also applied to population updates to improve the quality of the global solution.

3.3.1. Soft rime search strategy

In a breeze-like setting, soft frost formation exhibits significant randomness, with frost particles freely covering a large portion of the substrate's surface but growing slowly in a uniform direction. Drawing inspiration from this phenomenon, this study introduces a soft frost search strategy that capitalizes on the randomness and extensive coverage of frost particles. This allows the algorithm to efficiently explore the entire search space during initial iterations, avoiding local minima traps. The condensation of frost particles into soft frost agents is simulated, with the process of a single particle's condensation illustrated in **Figure 1a**. The positions of these frost particles are determined using Equation (2).

$$R_{i,j}^{new} = R_{best,j} + r_1 \cdot cos\,\theta \cdot \beta\big(h \cdot \big(Ub_{ij} - Lb_{ij}\big) + Lb_{ij}\big), r_2 < E \tag{2}$$

where $R_{i,j}^{new}$ is the updated position of the particle, *i* and *j* represent the ith frost agent's *j*-th particle. $R_{best,j}$ is the *j*-th particle of the best frost agent in the frost population *R*. The parameter r_1 is a random number in the range (−1,1), which, together $cos\,\theta$, controls the particle's movement direction and changes with the number of iterations, as shown in Equation (3).

$$\theta = \pi \cdot \frac{t}{10 \cdot T} \tag{3}$$

where *t* is the current number of iterations and *T* is the maximum number of iterations of the algorithm.

β is an environmental factor that varies with the number of iterations to simulate the influence of external conditions, ensuring algorithm convergence, as shown in Equation (4).

$$\beta = 1 - \left[\frac{w \cdot t}{T}\right]/w \tag{4}$$

h represents adhesion and is a random number within the range (0,1); $Ub_{i,j}$ and $Lb_{i,j}$ denotes the upper and lower limits of the escape space, respectively; r_2 is a random number which E with co-controls whether the particle position is updated or not.

E is the attachment coefficient, the expression shown in Equation (5), which affects the cohesion probability of an individual and increases with the number of iterations.

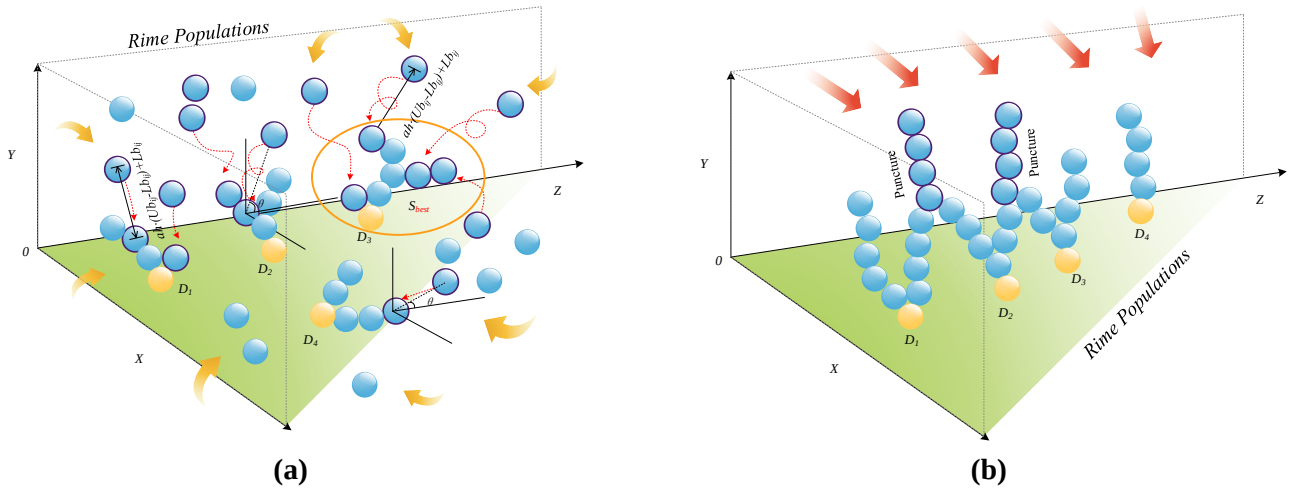
$$E = \sqrt{(t/T)} \tag{5}$$

3.3.2. Hard rime puncture mechanism

In high wind conditions, hard frost growth tends to be straightforward and predictable, whereas soft frost growth remains more unpredictable. Hard frost agents accumulate like snowballs in a uniform direction but are susceptible to penetration. Drawing from this penetration phenomenon, this study introduces a hard frost penetration mechanism designed for agent-to-agent updates within the algorithm. This mechanism facilitates particle exchange, enhancing the algorithm's convergence speed and capacity to avoid local minima. The penetration phenomenon is illustrated in **Figure 1b**, and the equation governing particle substitution is provided in Equation (6).

$$R_{i,j}^{new} = R_{best,j}, r_3 < F^{normr}(S_i) \tag{6}$$

where $F^{normr}(S_i)$ [?] denotes the normalized value of the current agent fitness value, which indicates the probability that the ith rime agent is selected r_3, it is a random number in the interval (−1, 1).



(a) **(b)**

Figure 1. Two stages of the rime optimization algorithm **(a)** soft frost search strategy; **(b)** hard frost puncture mechanism.

3.3.3. Proactive greedy selection mechanism

In metaheuristic optimization algorithms, the active greedy selection mechanism

is employed for updating the population. The core concept involves comparing the fitness value of an agent after an update with its fitness value before the update. If the updated fitness value surpasses the pre-update value, the updated agent takes the place of the pre-update agent. Subsequently, the fitness value of the current agent is compared to the optimal fitness value; if the current agent's fitness value is superior, it replaces the optimal agent, and the optimal fitness value is accordingly updated. This mechanism is intended to bolster the algorithm's ability to explore and exploit the global solution space while preserving its inherent strengths. Specifically, it focuses not only on the optimal result after a single update but also on potentially retaining some non-optimal individuals that help further discover potential good solutions, thus achieving better global optimization results.

3.4. Improved rime optimization algorithm

The key to obtaining high-quality global optimal solutions with the Rime Optimization Algorithm is whether the algorithm can escape local optima. To address the Rime Optimization Algorithm's tendency to get stuck in local optima, a mutation strategy is employed to enhance population diversity, boost the algorithm's ability to search globally, and broaden the exploration scope.

The Cauchy distribution function features a narrow peak at the center and extends widely on both sides, making it suitable for data perturbation due to its heavy-tail characteristics. Therefore, this paper integrates the Cauchy operator to perturb the position of the optimal individual in the Rime Optimization Algorithm, fully utilizing the effects of mutation at both ends of the Cauchy distribution function to optimize the algorithm's optimal individual, allowing the algorithm to achieve better global optima. This perturbation strategy not only maintains population diversity but also enhances the algorithm's ability to escape local optima in global search. After obtaining the current optimal solution, the mutation operation is performed on the current global optimal solution using the update Equation (7) shown below.

$$R_{best,j}^{new} = R_{best,j} + R_{best,j} \cdot f(\alpha) \tag{7}$$

where $f(\alpha)$ is calculated using the Cauchy distribution, as shown in Equation (8):

$$f(\alpha) = \frac{1}{\pi} \cdot \frac{1}{(1+\alpha^2)}, \alpha \in (0,1) \tag{8}$$

To further enhance the algorithm's exploration capability and avoid early convergence, this paper integrates the differential evolution algorithm into the algorithm. After applying the active greedy selection mechanism, the population undergoes mutation, crossover, and selection operations to increase diversity. This strategy not only helps the algorithm escape local optima but also explores a broader solution space during a global search.

The way new individuals are generated by performing the mutation operation is shown in Equation (9):

$$S_{i,j} = R_{r1,j} + MR \cdot \left(R_{r2,j} - R_{r3,j}\right) \tag{9}$$

where subscripts r_1, r_2, r_3 are randomly selected distinct integers from i; r_1, r_2, $r_3 \in$

[0,N−1] and $r_1 \neq r_2 \neq r_3$; *MR* represents the mutation probability.

The way new individuals are generated by performing the crossover operation is shown in Equation (10):

$$R_{i,j}^{new} = \begin{cases} S_{i,j}, R \leq CR \\ R_{i,j}, otherwise \end{cases} \tag{10}$$

where $S_{i,j}$ represents the value of the jth dimension of the new individual generated by the crossover mutation operation of the ith individual; $R_{i,j}$ represents the value of the jth dimension of the ith individual; *CR* is the crossover mutation probability; *R* is a random number between [0,1].

The individual is saved according to the greedy strategy, as shown in Equation (11):

$$R_{i,j} = \begin{cases} R_{i,j}^{new}, fit\left(R_{i,j}^{new}\right) < fit\left(R_{i,j}\right) \\ R_{i,j}, otherwise \end{cases} \tag{11}$$

where $fit(\cdot)$ is the fitness value calculation function.

The process of intrusion detection feature selection by the improved optimization algorithm is as follows: First, the initial population is generated by the frost ice optimization algorithm, and each individual represents a feature selection scheme. According to the characteristics and requirements of the intrusion detection system, a fitness function is set to evaluate the pros and disadvantages of each feature selection scheme. In the mutation step, a new individual is created by adding the difference in vectors between any two individuals in the population to a third individual. In the crossover step, the parent individual and the experimental individual are crossed with a certain crossover probability to generate a new offspring individual. In the selection step, the fitness function is used to evaluate the advantages and disadvantages of the new individual and the parent individual, and the better individual is selected into the next generation population. Cauchy distribution is introduced into the mutation step of differential evolution operation to enhance the diversity and robustness of search. Specifically, a Cauchy distribution based random number can be introduced into the mutation operation to adjust the mutation step size and direction, which can make the algorithm easier to jump out of the local optimal solution in the search process and improve the global search ability. According to the results of the differential evolution strategy and the evaluation of the fitness function, the parameters and search strategy of the Frost ice optimization algorithm were constantly adjusted and optimized. The memory ability and dynamic tracking characteristics of the Frost ice optimization algorithm were used to adjust the search direction according to the current search situation to improve the search efficiency, and the differential evolution operation and the adjustment steps of the frost ice optimization algorithm were repeated. Until the preset number of iterations is reached or the fitness function value meets certain stopping conditions, the optimal feature selection scheme is output for the practical application of intrusion detection system. The pseudocode of the algorithm is shown in **Algorithm 1**.

Algorithm 1 Pseudo-code of IRIME

1: **Input:** maximum number of iterations T, population size num, solution length dim
2: **Output:** Global Solution R_{best}
3: Initialize the rime population R
4: Get the current optimal agent and optimal fitness
5: ***While*** $t <= T$
6: Perform Cauchy mutation on the optimal agent
7: Coefficient of adherence $E = \sqrt{(t / T)}$
8: ***If*** $r_2 < E$
9: Update the position of the rime agent by the soft-rime search strategy
10: ***End If***
11: ***If*** $r_3 < Normalize\ fitness\ of\ S_i$
12: Cross-updating between agents by the hard-rime puncture mechanism
13: ***End If***
14: ***If*** $F\left(R_i^{new}\right) < F\left(R_i\right)$
15: Select the optimal solution and replace the suboptimal solution using the positive greedy selection mechanism
16: ***End If***
17: Perform differential mutation on the entire population and perform a greedy strategy to select the new optimal agent
18: $t = t + 1$
19: ***End While***

3.5. The proposed feature selection approach

The fitness or cost function is used to assess the quality of a solution. Feature selection methods aim to maximize classification accuracy while reducing the number of selected features and error rates. Thus, the fitness function evaluates the selected feature subset based on accuracy and the number of features. Equation (12) represents the fitness function used in this algorithm's optimization process.

$$fitness = weight_1 \cdot (1 - ACC) + weight_2 \cdot \frac{SF}{TF} \tag{12}$$

where fitness represents the fitness function value; *ACC* represents the accuracy; *SF* represents the number of selected features, and *TF* represents the total number of features; The sum $weight_i$ must equal 1, and the values of the weights are set to $weight_1$ = 0.99 and $weight_1$ = 0.01.

The initial RIME algorithm was designed for ongoing optimization tasks. However, feature selection works within a binary context, requiring specific operators to modify the RIME algorithm for binary optimization. The proposed RIME feature selection algorithm defines a solution as a fixed-length vector (matching the number of features), with initial values at each position randomly assigned between 0 and 1. To determine the optimal set of features, an S-shaped function is commonly utilized to convert the obtained optimal solution into binary values of 0 or 1. Specifically, a value of 0 means the corresponding feature is not chosen, while a value of 1 means it is selected. The S-shaped function is detailed in Equation (13).

$$X_{i,j}^t = \begin{cases} 0, S\left(X_{i,j}^t\right) < R \\ 1, otherwise \end{cases} \tag{13}$$

where R is a random number between [0, 1]; The definition $S(\cdot)$ is shown in Equation

(14):

$$S(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

After the binarization operation, the fitness function proposed above is used to evaluate the current solution (i.e., the feature subset). In feature selection problems, the fitness function is usually defined based on the performance of the selected features on tasks such as classification, regression, or clustering. RIME algorithm performs global search in the search space by simulating the growth mechanism of rime. In each iteration, the algorithm will perform crystal growth, selection, crossover and mutation operations on the individuals in the population to generate new individuals. These operations aim to retain excellent feature combinations while exploring new feature combination space. The algorithm ends and outputs the optimal solution.

In the RIME feature selection algorithm, the importance of different features can be analyzed in the following way:

A. Final selected features:

After the algorithm, the final selected feature subset is usually the most important set of features. These features have a high contribution in tasks such as classification, regression or clustering.

B. Feature frequency:

During the running of the algorithm, it is possible to record how often each feature is selected. Features with higher frequencies are generally more important because they are preserved across different solutions.

C. Feature contribution:

Model-based feature importance evaluation methods can be used to quantify the contribution of each feature. These methods provide more specific measures of feature importance that help to understand the impact of each feature on model performance.

D. Key feature identification:

By analyzing the final selected feature subset and feature contribution, the features that play a key role in improving the detection accuracy can be identified. These features usually have high contribution and selection frequency, and show good performance across different datasets and tasks.

Figure 2 shows the feature selection mechanism based on the improved binary RIME.

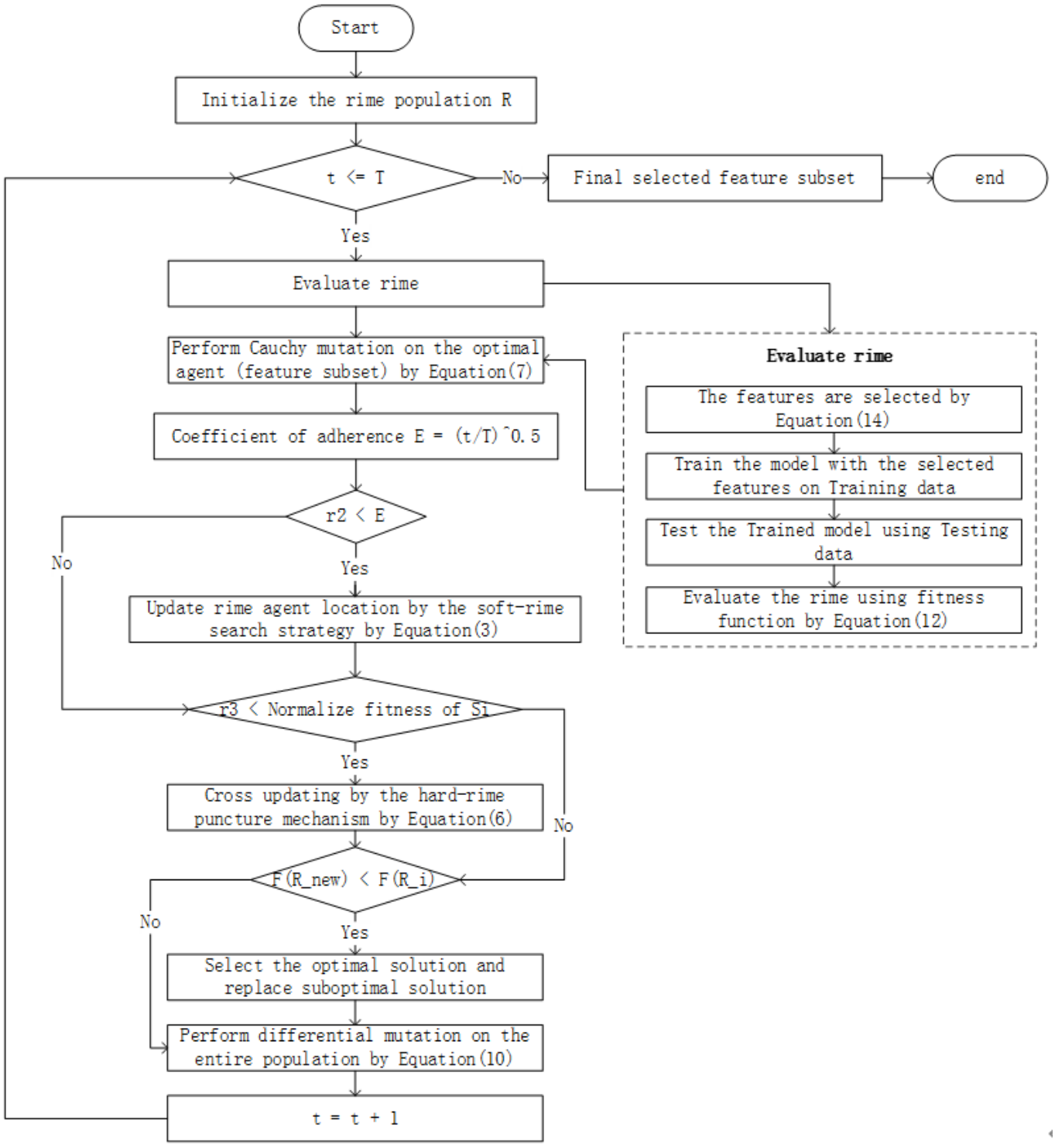


Figure 2. Feature selection mechanism based on improved binary RIME.

4. Experimental results

4.1. Experimental setup

The experiments took place on a Windows 11 laptop equipped with a 2.60GHz Intel Core i5 processor and 16 GB of RAM, using Python as the programming tool. In this paper, a Decision Tree (DT) model was utilized to train and evaluate the feature subset suggested by the proposed feature selection approach. Additionally, the

proposed feature selection algorithms were compared against several others derived from recent related research.

4.2. Evaluation results

4.2.1. Performance test of IRIME

To experimentally verify the effectiveness of the proposed IRIME algorithm, six internationally commonly used benchmark test functions were selected for simulation experiments, as shown in **Table 3**:

Table 3. Test function parameter design table.

Function number	function expression	dimensionality	Variable Range Values	optimum value
F1	$f_1(x)=\sum_{i=1}^{n}x_i^2$	30	[-100,100]	0
F2	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10,10]	0
F3	$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i}x_j\right)^2$	30	[-100,100]	0
F4	$f_4(x)=\max_i\left\{\lvert x_i\rvert,1\le i\le n\right\}$	30	[-10,10]	0
F5	$f_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-30,30]	0
F6	$f_6(x)=\sum_{i=1}^{n}\left[x_i+0.5\right]^2$	30	[-100,100]	0

Figure 3 shows the comparison of convergence curves between RIME and IRIME on six test functions. It can be seen from the figure that the IRIME algorithm quickly finds the global optimal solution, converging faster and with higher search accuracy than the RIME algorithm, demonstrating superior performance. The IRIME algorithm has significant advantages, accelerating global convergence to some extent, quickly focusing on the optimal region for exploration, and easily bypassing the stability of the algorithm in local optima, ensuring fast convergence speed and high precision.

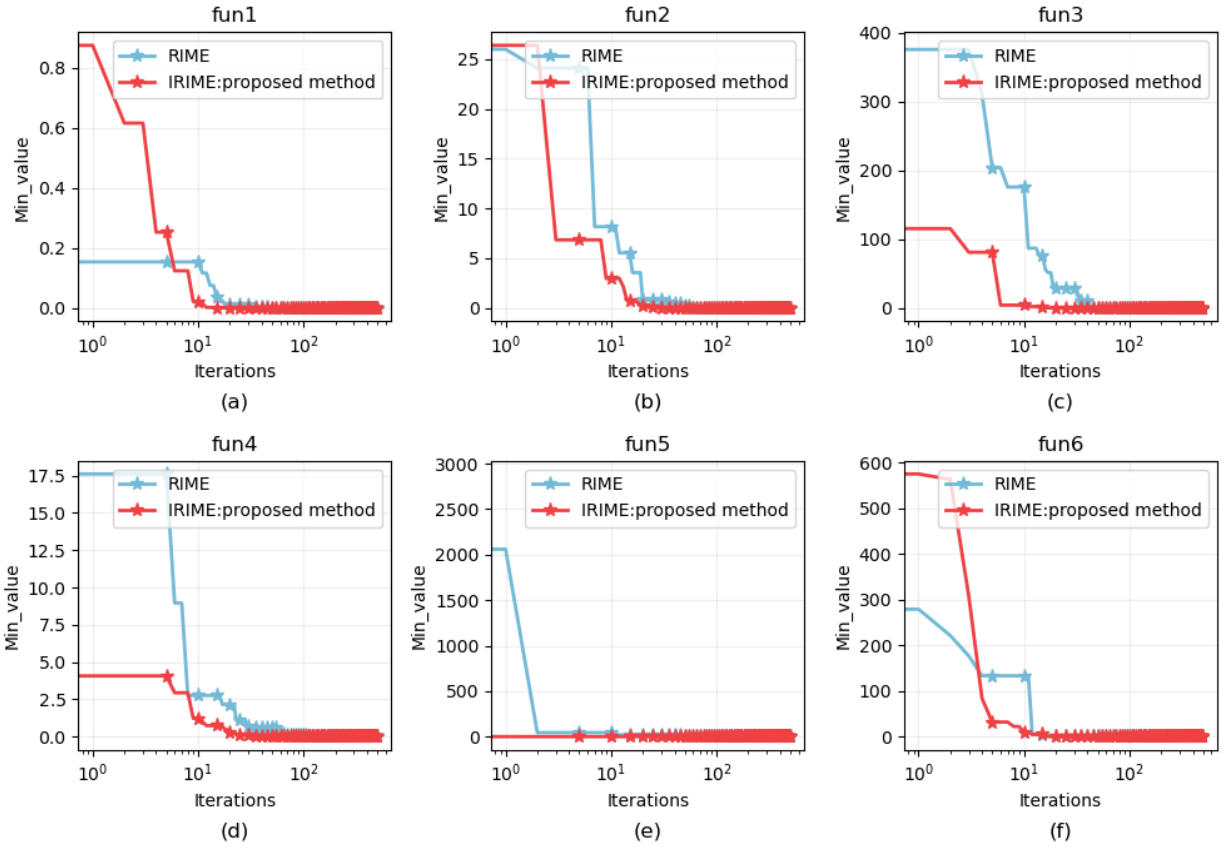


Figure 3. Comparison of algorithm convergence curves.

4.2.2. Performance test of IRIME feature selection method

In this section, we evaluate the IRIME feature selection algorithm using two datasets: NSL-KDD and UNSW-NB15. Our initial tests on these datasets involved applying a Decision Tree (DT) classifier without feature selection (FS). **Table 4** shows the performance of the DT classifier without FS on both datasets. The NSL-KDD dataset, used first to assess the proposed FS algorithms, is an improved version of the KDDCUP 99 dataset and retains its feature set. **Table 4** outlines the feature sets selected from the NSL-KDD dataset by various FS algorithms, including our proposed method. As noted in **Table 4**, each FS algorithm picks a varying number of features.

Table 4. Performance of DT without feature selection on two datasets.

Dataset	accuracy	precision	recall	*f*-score
NSL-KDD	0.797	0.812	0.813	0.797
UNSW-NB15	0.895	0.874	0.913	0.886

The algorithm was benchmarked against the Particle Swarm Optimization and Harris Hawk Optimization algorithms, which are frequently employed in network intrusion detection systems. The assessment of all feature selection algorithms was conducted using the Decision Tree (DT) classifier from Python's sci-kit-learn library, given DT's superior handling of feature interactions compared to other basic classifiers. To maintain consistency, identical data preprocessing steps were applied

across all algorithms under investigation. It's important to note that the experimental outcomes may vary from those reported in prior research.

The NSL-KDD dataset, used as the primary dataset for assessing the proposed feature selection algorithm, is an improved version of the KDDCUP 99 dataset that retains its original features. **Table 5** lists the feature sets selected by various feature selection algorithms, including the proposed method, from the NSL-KDD dataset. As shown in **Table 5**, each algorithm picks a different number of features.

Table 5. Selected feature subsets of different algorithms on the NSL-KDD Dataset.

Reference	Technique	Number	Selected set of features
[32]	Hybrid association rules	11	[2, 5, 6, 7,12, 16, 23, 28, 31, 36, 37]
[33]	IGFS	8	[5, 3, 6, 4, 30, 29, 33, 34]
[34]	PSOFS	37	[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17,18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31,32, 33, 34, 35, 36, 37, 38, 39, 40, 41]
[35]	CosinePIOFS	5	[2, 6, 10, 22, 27]
[36]	CossimMFOFS	4	[2, 6, 10, 22]
Proposed approach	RIMEFS	18	[0, 2, 5, 8, 10, 12, 14, 17, 19, 20, 21, 22, 24, 25, 28, 31, 37, 39]
Proposed approach	IRIMEFS	17	[0, 2, 4, 5, 7, 8, 12, 13, 18, 22, 23, 26, 28, 29, 30, 31, 38]

Table 6 displays the mean outcomes of training and testing the DT classifier 30 times, utilizing the features picked by each algorithm as shown in **Table 5** on the NSL-KDD dataset. It compares the accuracy, precision, recall, and *F*-Score metrics. As shown in **Table 6**, the proposed IRIME algorithm achieves the highest recall, while the Hybrid Association algorithm achieves the lowest. Accuracy and *F*-Score are good indicators for comparing the algorithms being tested, with IRIME achieving the best results with accuracy and *F*-Score values of 0.901 and 0.900, respectively. RIME achieves the second-best results with values of 0.895 and 0.895, respectively, while CossimMFO ranks third in accuracy and *F*-Score compared to other algorithms listed in **Table 6**.

Table 6. Performance comparison of various feature selection algorithms based on decision trees on the NSL-KDD dataset.

Approach	Accuracy	Precision	Recall	*f*-score
Hybrid association rules	0.796	0.758	0.665	0.795
IGFS	0.808	0.706	0.707	0.808
PSOFS	0.782	0.774	0.637	0.781
CosinePIOFS	0.883	0.873	0.866	0.882
CossimMFOFS	0.897	0.892	0.891	0.892
RIMEFS	0.898	0.894	0.902	0.895
IRIMEFS	0.901	0.899	0.907	0.900

UNSW-NB15 serves as the second dataset in this paper for evaluating the proposed IRIME feature selection algorithm. **Table 7** presents the feature subsets selected from the UNSW-NB15 dataset by five different feature selection algorithms, including the proposed IRIME algorithm. Each row details the number of chosen

features, their respective indices, and the selection method used. The indices can be matched to the corresponding feature names listed in **Table 7**.

Table 7. Selected feature subsets of different algorithms on the UNSW-NB15 dataset.

Reference	Technique	Number	Selected set of features
[32]	Hybrid association rules	11	[6, 10, 11, 19, 20, 27, 34, 37, 42, 44, 46]
[34]	PSOFS	19	[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17,18, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31,32, 33, 34, 35, 36, 37, 38, 39, 40, 41]
[37]	Rule-based	13	[5, 8, 9, 10, 13, 14, 32, 41, 42, 43, 45, 46, 47]
[35]	CosinePIOFS	5	[2, 6, 10, 22, 27]
[36]	CossimMFOFS	4	[3, 4, 8, 12]
Proposed approach	RIMEFS	10	[1, 2, 3, 6, 10, 14, 26, 29, 31, 34]
Proposed approach	IRIMEFS	8	[1, 2, 4, 6, 14, 29, 31, 34]

Table 8 presents the average training and testing results of the DT classifier, which was run 30 times using the features selected by each algorithm from the UNSW-NB15 dataset as shown in **Table 7**. The results indicate that the proposed IRIME algorithm outperforms other feature selection methods, achieving the highest precision (0.912) and recall (0.943). Additionally, IRIME yields the best accuracy (0.942) and *F*-Score (0.923) among all the algorithms.

Table 8. Performance comparison of various feature selection algorithms based on decision trees on the UNSW-NB15 dataset.

Approach	Accuracy	Precision	Recall	*f*-score
Hybrid association rules	0.792	0.758	0.721	0.784
PSOFS	0.895	0.852	0.863	0.886
Rule-Based	0.884	0.831	0.889	0.870
CosinePIOFS	0.924	0.876	0.921	0.942
CossimMFOFS	0.917	0.865	0.894	0.909
RIMEFS	0.923	0.903	0.935	0.915
IRIMEFS	0.942	0.912	0.943	0.923

When dealing with NSL-KDD and UNSW-NB15 datasets, IRIME algorithm significantly enhances its global search ability by incorporating Cauchy mutation technology, while maintaining efficient search performance. As a heuristic search strategy, Cauchy mutation uses Cauchy distribution to generate larger mutation step size, which effectively broaden the search range of the algorithm. In the feature selection task, this property enables the IRIME algorithm to easily jump out of local optimal solutions and explore more potential feature combinations. At the same time, IRIME algorithm also uses differential evolution strategy to generate and screen out better solutions by simulating the mutation, crossover and selection mechanism in the process of biological evolution. This strategy not only maintains the diversity of the population, but also promotes the algorithm to approach the global optimal solution gradually. Differential evolution, as a population-based global optimization algorithm, provides powerful optimization capabilities for IRIME algorithm.

Combining Cauchy mutation and differential evolution strategy, IRIME algorithm can quickly locate the optimal feature combination in the broad search space. This advantage makes the IRIME algorithm perform well when dealing with large-scale data sets, and can obtain better results in a limited time. At the same time, the combination of these two strategies also gives the IRIME algorithm a stronger global search ability to avoid falling into the dilemma of local optimal solutions, so as to screen out a better feature subset.

By selecting these better feature subsets, IRIME algorithm constructs intrusion detection models with better performance, which show higher accuracy, accuracy and F1 score and other evaluation indicators on classification, regression or clustering tasks. The experimental results show that the IRIME algorithm is superior to other algorithms in all evaluation indicators on NSL-KDD and UNSW-NB15 datasets, which fully verifies the effectiveness of the improvement of the algorithm. Specifically, the IRIME algorithm significantly improves the prediction precision and accuracy of the model by optimizing the feature selection process, making the model perform well in identifying intrusion data. At the same time, the IRIME algorithm further reduces the false positive and false negative by improving the F1 score and recall rate, thereby improving the reliability and practicability of the intrusion detection model. In summary, IRIME algorithm successfully improves the global search ability and search efficiency of the algorithm by combining Cauchy mutation and differential evolution strategy, which provides strong support for processing large-scale data sets and building high-performance intrusion detection models.

5. Conclusion

This paper introduces a IRIME-based feature selection algorithm designed for Intrusion Detection Systems (IDS). The objective of the IRIME algorithm is to reduce the number of features required for building a reliable IDS, while maintaining high detection rates, accuracy, and minimizing false positives. By applying the IRIME algorithm, the number of features is reduced from 41 to 17 for the NSL-KDD dataset and from 49 to 8 for the UNSW-NB15 dataset. This reduction maintains high precision and accuracy while significantly shortening the model construction time.

Future research can explore further improvements to the Rime Optimization Algorithm to adapt to more complex datasets and application scenarios. Additionally, combining other optimization algorithms or considering multi-objective optimization problems is an interesting direction.

Author contributions: Conceptualization, QP, XW and AT; methodology, QP; software, QP; validation, QP, XW and AT; data curation, QP; writing—original draft preparation, QP; writing—review and editing, QP, XW and AT; supervision, XW; project administration, QP; funding acquisition, QP and XW. All authors have read and agreed to the published version of the manuscript.

Funding: The Project of Ningxia Natural Science Foundation (2024AAC03165, 2024AAC03169); The National Natural Science Foundation of China under Grant (62062001); Ningxia Youth Top Talent Project (2021).

Ethical approval: Not applicable.

Informed consent: Not applicable.

Conflict of interest: The authors declare no conflict of interest.

References

1. N. Sun, M. Ding, J. Jiang, et al., "Cyber threat intelligence mining for proactive cybersecurity defense: a survey and new perspectives," IEEE Communications Surveys & Tutorials, vol. 25, no. 3, pp. 1748-1774, May. 2023.
2. S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," Journal of Information Security and Applications, vol. 44, pp. 80-88, Feb. 2019.
3. S. Solorio-Fernández, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," Artificial Intelligence Review, vol. 53, no. 2, pp. 907-948, Jan. 2020.
4. J. M. Valls, R. Aler, I. M. Galván, D. Camacho, "Supervised data transformation and dimensionality reduction with a 3-layer multi-layer perceptron for classification problems," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 4, pp. 10515–10527, Jan. 2021.
5. R. Abu Khurma, I. Aljarah, A. Sharieh, M. Abd Elaziz, R. Damaševičius, T. Krilavičius, "A review of the modification strategies of the nature inspired algorithms for feature selection problem," Mathematics, vol. 10, no. 3, pp. 464, Jan. 2022.
6. R. A. Khurma, I. Aljarah, A. Sharieh, "Rank based moth flame optimisation for feature selection in the medical application," in Proc. of the IEEE congress on evolutionary computation, Glasgow, UK, 2020, pp. 1–8.
7. A. Martín, R. Lara-Cabrera, D. Camacho, "Android malware detection through hybrid features fusion and ensemble classifiers: the AndroPyTool framework and the OmniDroid dataset," Information Fusion, vol. 52, no. 4, pp. 128–142, Dec. 2019.
8. Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, X. S. Yang, et al. "Multi-objective flower pollination algorithm: a new technique for EEG signal denoising," Neural Computing and Applications, vol. 35, pp. 7943–7962, Jan. 2022.
9. J. Fierrez, A. Morales, R. Vera-Rodriguez, D. Camacho, "Multiple classifiers in biometrics. part 1: Fundamentals and review," Information Fusion, vol. 44, pp. 57–64, Nov. 2018.
10. R. A. Khurma, H. Alsawalqah, I. Aljarah, M. A. Elaziz, R. Damaševičius, "An enhanced evolutionary software defect prediction method using island moth flame optimization," Mathematics, vol. 9, no.15, pp. 1722, Jul. 2021.
11. G. J. Lee, G. Li, D. Camacho, J. J. Jung, "Discovering synergic association by feature clustering from soccer players," in Proc. of the Int. Conf. on research in adaptive and convergent systems, New York, NY, USA, 2020, pp. 107–112.
12. R. Abu Khurma, I. Almomani, I. Aljarah, "IoT botnet detection using salp swarm and ant lion hybrid optimization model," Symmetry, vol. 13, no. 8, pp. 1377, Jul. 2021.
13. A. Al Shorman, H. Faris, I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and grey wolf optimization for IoT botnet detection," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 2809–2825, Jul. 2019.
14. J. Kennedy, R. Eberhart. "Particle swarm optimization," in Proc. of the ICNN'95- Int. Conf. on neural networks, Perth, WA, Australia, 1995, pp. 1942-1948.
15. S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey wolf optimizer," Advances in engineering software, vol. 69, pp. 46-61, Mar. 2014.
16. A. A. HEIDARI, S. MIRJALILI, H. FARIS, et al., "Harris hawks optimization: algorithm and applications," Future Generation Computer Systems, vol. 97, pp. 849-872, Aug. 2019.
17. S. Mirjalili, S. M. Mirjalili, A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," Neural Computing and Applications, vol. 27, no. 2, pp. 495-513, 2016.
18. A. E. Takieldeen, E. S. M. El-kenawy, M. Hadwan, M. Hadwan, R. M. Zaki, "Dipper throated optimization algorithm for unconstrained function and feature selection," Comput. Mater. Contin, vol. 72, pp. 1465-1481, 2022.
19. H. Su, D. Zhao, A. A. Heidari, et al., "RIME: A physics-based optimization," Neurocomputing, vol. 532, pp. 183-214, May. 2023.
20. M. Alazab, "Automated malware detection in mobile app stores based on robust feature generation," Electronics, vol. 9, no. 3, pp. 435, Mar. 2020.M.
21. M. Alazab, S. Alhyari, A. Awajan, A. B. Abdallah, "Blockchain technology in supply chain management: an empirical study of the factors affecting user adoption/acceptance," Cluster Computing, vol. 24, no. 1, pp. 83–101, Mar. 2021.

22. R. A. Khurma, I. Aljarah, A. Sharieh, M. A. Elaziz, R. Damaševičius, T. Krilavičius, "A review of the modification strategies of the nature inspired algorithms for feature selection problem," Mathematics, vol. 10, no. 3, pp. 464, 2022.
23. Y. Xue, T. Tang, W. Pang, A. X. Liu, "Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers," Applied Soft Computing, vol. 88, pp. 106031, Mar. 2020.
24. K. Chen, B. Xue, M. Zhang, F. Zhou, "Correlation-guided updating strategy for feature selection in classification with surrogate-assisted particle swarm optimization," IEEE Transactions on Evolutionary Computation, vol. 26, no. 5, pp. 1015-1029, Oct. 2022.
25. M. S. Bonab, A. Ghaffari, F. S. Gharehchopogh, P. Alemi, "A wrapper-based feature selection for improving performance of intrusion detection systems," International Journal of Communication Systems, vol. 33, no. 12, pp. e4434, Apr. 2020.
26. Y. Zhou, G. Cheng, S. Jiang, M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks, vol. 174, pp. 107247, Jun. 2020.
27. A. Fatani, A. Dahou, M. A. Al-Qaness, S. Lu, M. A. Elaziz, "Advanced feature extraction and selection approach using deep learning and aquila optimizer for IoT intrusion detection system," Sensors, vol. 22, no. 1, pp. 140, 2022.
28. A. Mojtahedi, F. Sorouri, A. N. Souha, A. Molazadeh, S. S. Mehr, "Feature selection-based intrusion detection system using genetic whale optimization algorithm and sample-based classification," arXiv preprint arXiv:2201.00584, 2022. doi: 10.48550/arXiv.2201.00584.
29. A. Nazir, R. A. Khan, "A novel combinatorial optimization based feature selection method for network intrusion detection," Computers & Security, vol. 102, pp. 102164, Mar. 2021.
30. M. Maazalahi, S. Hosseini, "K-means and meta-heuristic algorithms for intrusion detection systems," Cluster Computing, pp. 1-43, May. 2024.
31. A. R. Al Shorman, H. Faris, P. Castillo, J. Merelo Guervs, N. Al-Madi, "The influence of input data standardization methods on the prediction accuracy of genetic programming generated classifiers," in Proc. IJCCI, 2018, pp. 79-85.
32. N. Moustafa, J. Slay, "A hybrid feature selection for network intrusion detection systems: Central points," arXiv preprint arXiv:1707.05505, 2017. doi: 50/arXiv.1707.05505.
33. S. Aljawarneh, M. Aldwairi, M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," Journal of Computational Science, vol. 25, pp. 152-160, Mar. 2018.
34. A. Tama, M. Comuzzi, K.-H. Rhee, "Tse-ids: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," IEEE Access, vol. 7, pp. 94497–94507, 2019.
35. H. Alazzam, A. Sharieh, K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," Expert systems with applications, vol. 148, pp. 113249, Jun. 2020.
36. M. Alazab, R. A. Khurma, A. Awajan, D. Camacho, "A new intrusion detection system based on Moth-Flame Optimizer algorithm," Expert Systems with Applications, vol. 210, pp. 118439, Dec. 2022.
37. V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," Cluster Computing, vol. 23, no. 2, pp. 1397-1418, Jun. 2020.