Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062189

ARTICLE

Metaheuristic-Driven Abnormal Traffic Detection Model for SDN Based on Improved Tyrannosaurus Optimization Algorithm

Hui Xu, Jiahui Chen* and Zhonghao Hu

School of Computer Science, Hubei University of Technology, Wuhan, 430068, China

*Corresponding Author: Jiahui Chen. Email: 102301180@hbut.edu.cn

Received: 12 December 2024; Accepted: 19 February 2025; Published: 19 May 2025

ABSTRACT: Nowadays, abnormal traffic detection for Software-Defined Networking (SDN) faces the challenges of large data volume and high dimensionality. Since traditional machine learning-based detection methods have the problem of data redundancy, the Metaheuristic Algorithm (MA) is introduced to select features before machine learning to reduce the dimensionality of data. Since a Tyrannosaurus Optimization Algorithm (TROA) has the advantages of few parameters, simple implementation, and fast convergence, and it shows better results in feature selection, TROA can be applied to abnormal traffic detection for SDN. However, TROA suffers from insufficient global search capability, is easily trapped in local optimums, and has poor search accuracy. Then, this paper tries to improve TROA, namely the Improved Tyrannosaurus Optimization Algorithm (ITROA). It proposes a metaheuristic-driven abnormal traffic detection model for SDN based on ITROA. Finally, the validity of the ITROA is verified by the benchmark function and the UCI dataset, and the feature selection optimization operation is performed on the InSDN dataset by ITROA and other MAs to obtain the optimized feature subset for SDN abnormal traffic detection. The experiment shows that the performance of the proposed ITROA outperforms compared MAs in terms of the metaheuristic-driven model for SDN, achieving an accuracy of 99.37% on binary classification and 96.73% on multiclassification.

KEYWORDS: Software-defined networking; abnormal traffic detection; feature selection; metaheuristic algorithm; tyrannosaurus optimization algorithm

1 Introduction

Traditional networks are being pushed to their limitations by social media, mobile devices, and cloud computing [1]. The traditional network architecture model has many problems with protocols and closed models. Software-Defined Networking (SDN) [2] is an emerging networking paradigm that gives hope to change the limitations of current network infrastructures [3]. It was first described as a method for designing, constructing, and maintaining networks that divide the control and forwarding planes. It allows the network control to be directly programmable and the underlying infrastructure to be abstracted for network services and applications [4]. However, its open interfaces and programmability can also be used maliciously, allowing attackers to use these interfaces to codify network configurations. Techniques for abnormal traffic detection can pay close attention to network traffic patterns to identify and block such threats. SDN abnormal traffic detection becomes crucial as a result.

The categories of the two main SDN abnormal traffic detection methods are rule-based and Machine learning-based. In this rule-based method, abnormal traffic detection is performed by utilizing pre-defined rules. Common rules include bandwidth limitations, latency, packet loss rate, and other traffic-related

parameters. However, this method may not be successful in identifying certain new types of network attacks or abnormal conditions when faced with them. As a branch of artificial intelligence, machine learning has become a helpful ally [5]. It is a promising instrument for abnormal traffic detection because of its lightweight design and capacity to process massive amounts of data [6–8]. A machine learning-based method builds models for abnormal detection by training lots of traffic data, which learns the normal traffic patterns to create a rubric in the training phase and determines the abnormal discovery in new traffic based on the learned rubric in the detection phase.

Abnormal traffic detection for SDN based on machine learning has received some attention. Niyaz et al. [9] propose a deep learning-based multi-vector DDoS detection system in an SDN environment by using deep learning to reduce the features derived from network traffic and improve attack detection accuracy. Tang et al. [10] develop a deep neural network model for an intrusion detection system and trained the model using the NSL-KDD dataset for stream-based anomaly detection. The experiments have confirmed the great potential of deep learning methods for SDN abnormal detection. Wang et al. [11] propose a deep learning hybrid model-based abnormal traffic detection system to achieve fine detection of abnormal traffic from the surface and improve accuracy, recall, and false alarm rate compared to traditional SDN abnormal traffic detection methods. Arevalo-Herrera et al. [12] propose a machine-learning algorithm for classification attacks using the CES CIC IDS2018 dataset. Their analysis evaluates the performance of traditional machine-learning techniques, including decision trees, random forests, and a neural network architecture.

In recent years, there have been notable developments in Machine learning-based aberrant traffic identification for SDN. By extracting the network traffic features and using machine learning to classify and detect the traffic, researchers can quickly identify abnormal traffic detection. However, Machine learning-based SDN abnormal traffic detection may encounter performance bottlenecks when dealing with large-scale data and require additional designs to optimize the data processing flow. As a key step in data processing, feature selection aims to reduce the number of features to deal with the problem of data redundancy. The Metaheuristic algorithm (MA), such as the Whale Optimization Algorithm (WOA) and the Particle Swarm Optimization (PSO), can efficiently filter out the most important features by searching the solution space heuristically, thus improving the performance and accuracy of the detection.

MA is widely used in the feature selection phase for the SDN abnormal traffic detection model to raise the prediction accuracy. Stein et al. [13] utilize a genetic algorithm to select a subset of input features for the decision tree classifier, using the KDDCUP 99 dataset to train and test the classifier. The generated decision tree performs better than the decision tree constructed using all available functions, increasing the detection rate and reducing the false alarm rate for network intrusion detection. Zainal et al. [14] use a wrapper method that integrates a rough set and PSO to be a 2-tier structure of the feature selection process, and the experimental findings demonstrate that the feature subset suggested by the technique is robust and provides a better representation of the data. Mojtahedi et al. [15] utilize feature selection based on a blend of WOA and genetic algorithms to create a network intrusion detection system, which makes it great to extract features related to class labels. Lin et al. [16] take advantage of support vector machines and simulated annealing, in which support vector machines and simulated annealing can find the best selection of features to improve the accuracy of the abnormal traffic detection amount. Additionally, our prior work [17–21] has introduced relevant applications of MA for feature selection. We apply MAs skillfully to the feature selection phase of network traffic detection, aiming to substantially improve the efficiency and accuracy of detection by intelligently filtering and extracting the most representative subset of features. Based on the in-depth analysis of network traffic, this paper further focuses the research on abnormal traffic detection in SDN environments, aiming to provide more targeted and accurate solutions.

As found above, metaheuristic techniques have proved to be excellent methodologies [22]. MA offers solutions that are close to optimal without ensuring a global optimum [23]. Consequently, the quest for more effective optimization solutions has spurred the continual design of new MA [24]. A new MA named Tyrannosaurus (T-Rex) Optimization Algorithm (TROA) [25] was proposed by Sahu et al. in 2023, which simulates the T-Rex's hunting habits. The core of the TROA is to simulate the hunting process of the T-Rex to update the position of the population, which has significant advantages such as simple design, fast searching speed, and fast convergence. Sahu et al. demonstrate the significant performance advantages TROA exhibits over various other MAs. Through experiments and comparative analyses, they find that TROA converges faster in solving optimization problems. This advantage makes TROA stand out among similar MAs and become the tool of choice in our research. The TROA uses random numbers in the hunting and chasing phases, increasing the search's randomness. It could also result in the algorithm reaching a local optimum solution too soon within the search. To solve this problem, the TROA can be improved and optimized accordingly. Because of these problems, this paper uses four improvement strategies to improve the TROA, and the Improved Tyrannosaurus Optimization Algorithm (ITROA) is proposed. Then, a metaheuristic-driven abnormal traffic detection model for SDN based on ITROA is proposed by applying ITROA to the feature selection. The next sections of this paper are organized as follows:

- Four improvement strategies are introduced to address the limitations of TROA applied to SDN abnormal traffic detection, and ITROA is proposed. It effectively solves problems such as TROA's tendency to fall into local optimization.
- A metaheuristic-driven abnormal traffic detection model for SDN based on ITROA is proposed. In order to improve detection efficiency and address the issue of data redundancy in the abnormal traffic detection model for SDN, the model employs ITROA for feature selection.

The subsequent sections of this paper are structured as follows. Section 2 describes the basic TROA algorithm. Section 3 proposes a metaheuristic-driven abnormal traffic detection model for SDN based on ITROA. Section 4 provides the experimental design and analysis. Section 5 summarizes this paper and outlines the limitations of this research and future research directions.

2 Tyrannosaurus Optimization Algorithm

TROA is a new bionic optimization algorithm that mimics T-Rex's hunting behavior. It has advantages, such as fast search speed and few parameters. TROA simulates the hunting behavior of a T-Rex through three main phases: position initialization, hunt and chase, selection phase, and, ultimately, the best option. This section focuses on the second and third phases, as shown in Algorithm 1.

Algorithm 1: Pseudocode of TROA

Start TROA.

Input problem information.
Initialization of the population size (N), the iterations (T), and the dimension (D).
Calculate the fitness value and record the optimal position.
For t = 1:T
  For i = 1:N
    Phase 1: *Huntingandchasing.*
    If *rand* < *Er*
      update the T-Rex's position by Eq. (2).
    else

(Continued)

Algorithm 1 (continued)

```
10.          Randomly update the T-Rex's position.
11.        end
12.        Calculate the new fitness values.
13.        Phase 2: Selection
14.        If f(X) < f(X_new)
15.          update the prey's position, target, and fitness values.
16.        else
17.          Target equals 0.
18.        end
19.        End
20.      Save the best solution attained.
21.      End
22.      Output the best solution attained.
END TROA.
```

2.1 Hunting and Chasing

T-Rex hunting behavior resembles top predators, such as tigers and wolves. Once the nearest prey is spotted, T-Rex attempts to capture it. However, prey sometimes succeed in defending themselves or take advantage of the opportunity to escape. Therefore, T-Rex selectively attacked its prey at random during feeding.

$$X_{new} = \begin{cases} x_{new}, & if\ rand\ \ () < Er \\ Random, & else \end{cases} \tag{1}$$

where Er is the estimate of reaching the dispersed prey, $Er = randn * (1 - (t/max\_Iter))$. The prey starts to flee as T-Rex begins hunting, and T-Rex updates its location to hunt the prey, as indicated by Eq. (1).

$$X_{new} = x + rand^* sr^* \left(tpos^* tr - target^* pr\right) \tag{2}$$

In the given equation, sr represents the hunting success rate, and its value domain is [0.1, 1]. When the success rate is 0, the prey escapes, and the hunting action fails. At this time, the prey's location information needs to be updated. Moreover, $tpos$ is the location of the T-Rex. The running speed of the T-Rex is denoted by tr, between [0, 1]; $target$ is the minimum distance of the prey to the T-Rex position. As for the prey, its running speed is denoted by pr, which has a value range between [0, 1].

2.2 Selection

The selection process is based on the location of the prey, which includes the current location of the prey as well as its previous location. This process is done by evaluating the fitness function. If the T-Rex hunts successfully, the target's location information and fitness function value are updated. If the T-Rex does not hunt, the prey's location information is zeroed out, which occurs when the prey escapes or takes self-protective measures to avoid predation.

$$X_i^{k+1} = \begin{cases} update\ the\ target\ position, & if\ f(X) < f(X_{new}) \\ target\ is\ zero, & else \end{cases} \tag{3}$$

where $f(X)$ is the fitness function of the initial random prey position and $f(X_{new})$ is the fitness function of the updated prey position.

Because of the influence of the random numbers and various parameters in Eq. (1), the TROA updates the position of an individual to a certain degree of randomness. Due to this update mechanism, TROA may prematurely converge to a locally optimal solution. In addition, the selection process of TROA mainly relies on conditional judgment, which may also cause TROA to stop prematurely during the search process and thus miss the global optimal solution. The architecture of TROA's update mechanism and selection process is primarily responsible for its structural flaws, which make it simple to drift toward the local optimum. In our subsequent studies, we explored various improvement strategies to overcome this shortcoming.

3 Proposed Abnormal Traffic Detection Model for SDN

There are two sections in this section. Firstly, the ITROA is improved by TROA. Secondly, the metaheuristic-driven abnormal traffic detection model for SDN based on ITROA is proposed.

3.1 Improved Tyrannosaurus Optimization Algorithm

3.1.1 Circle Mapping Strategy

The circle mapping [26] strategy effectively avoids falling into local optimal solutions during the search process by introducing the randomness and traversal of chaotic sequences. It also enhances the TROA's global search capability. Because of the distribution of individuals in it, the initial population obtained is not uniformly distributed. Thus, there can be no fundamental limitation on the algorithmic convergence. The circle mapping strategy is used to initialize the population.

Circle mapping is a chaotic mapping that can be used to generate chaotic numbers between 0 and 1. It has the following formula:

$$x_{k+1} = mod\left(x_k + e - \frac{f}{2\pi}\sin\left(2\pi x_k\right), 1\right) \tag{4}$$

where x_k represents the kth chaotic number. $e = 0.5, f = 0.2$. mod denotes the remainder operation. Chaotic numbers between 0 and 1 are generated by circle mapping and used as random numbers for the population's initialization. It can enhance the diversity and exploration of the population.

Assuming an individual of $N = 1000$. The distribution of individuals is created by circle mapping and random sequences in the interval $[0, 1]$. As shown in Fig. 1, the circle mapping strategy has a more uniform population distribution.

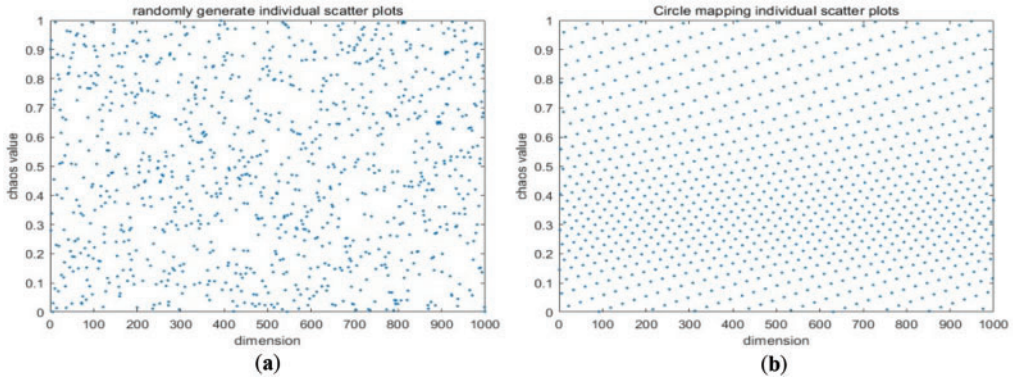

Figure 1: Binary classification results. (a) Random; (b) Circle mapping strategy

3.1.2 Elite Reverse Learning Strategy

The elite reverse learning strategy [27,28] increases the likelihood of finding a better solution, thus overcoming the limitation of local optimality. The search using the inverse solution of the current optimal solution allows for rapid exploration of unexplored regions of the solution space. This strategy is based on the advantage that elite individuals contain more valid information than ordinary individuals and use the elite individuals in the current population to generate a reverse population, which aims to increase the diversity of the population.

Definition of elite inverse solution: assume that the current general individual in the population corresponds to an elite individual at its extremum $X^e_{i,j} = X^e_{i,1}, X^e_{i,2,} \ldots, X^e_{i,d}$, its inverse solution $OP^e_{i,j} = OP^e_{i,1}, OP^e_{i,2,} \ldots, OP^e_{i,d}$. It can be defined as:

$$\alpha_j = max\left(X_{i,j}\right) \tag{5}$$

$$\beta_j = min\left(X_{i,j}\right) \tag{6}$$

$$OP^e_{i,j} = rand\left(\alpha_j + \beta_j\right) - X^e_{i,j} \tag{7}$$

where *rand* is a random number in the range [0, 1], $X^e_{i,j}\ \varepsilon\ [\alpha_j, \beta_j]$ and β_j are dynamic boundaries. The drawback of fixed boundaries, which makes search preservation challenging, is addressed by dynamic boundaries. Elite reverse learning strategy can be localized in a narrow search space, facilitating algorithmic convergence.

3.1.3 PSO Convergence Strategy

The PSO method updates a particle's position using its best position and the best position of the entire population. Therefore, this paper combines the idea of the PSO algorithm to introduce the optimal position information experienced by individual T-Rex into the position update formula. The new positional update formula is:

$$x_{(t+1)} = c1^* \text{rand}^* X_i(t) + c2^* \text{rand}^* \left(X_{ibest} - X_i(t)\right) \tag{8}$$

In the formula, $c1$ is the social learning factor, and $c2$ is the cognitive learning factor. Which represents the effect of the optimum experienced by an individual and the population optimum on the algorithm's search capability; rand is a random number in the range [0, 1]. X_{ibest} indicates where the T-Rex itself experienced the optimum.

3.1.4 Lévy Flight Strategy

The Lévy flight [29] has the property of having a long step size in conjunction with a short step size. It allows the algorithm to perform large step size jumps globally to explore new regions and fine search in small areas to optimize the quality of the solution. This property helps to jump out of the local optimum. This paper uses the Lévy flight strategy to optimize the TROA's optimization process and increase the optimization accuracy. Global detection uses a Lévy flight strategy, which distributes individuals widely in the search space to improve global optimization.

The Lévy flight strategy is a stochastic search method that obeys the Lévy distribution. It is a type of walk in which short-range searches are interspersed with occasional longer-range walks, thus contributing to the good global search capability of Lévy flights.

The position update formula for Lévy's flight is:

$$x_i(t) = x_i(t) + L \oplus Levy(\lambda) \tag{9}$$

where $x_i(t)$ denotes the ith solution in generation t; $\oplus$ denotes point-to-point multiplication; L denotes control step weights. Due to the Lévy distribution's complexity, it is often modeled using the Mantegna algorithm. The formula for calculating the step length is as follows:

$$S = \frac{\mu}{|\nu|^{\frac{1}{\gamma}}} \tag{10}$$

where μ and ν follow a normal distribution. Definitions are as follows:

$$\mu \sim N\left(0, \sigma_\mu^2\right)$$
$$\nu \sim N\left(0, \sigma_\nu^2\right)$$
$$\sigma_\mu = \left\{ \frac{\Gamma(1+\gamma)\, sin\left(\frac{\pi\gamma}{2}\right)}{\gamma * \Gamma\left[(\gamma+1)/2\right] * 2^{\frac{(\gamma-1)}{2}}} \right\}^{\frac{1}{\gamma}}$$
$$\sigma_\nu = 1$$

where γ is usually taken as 1.5.

3.1.5 ITROA Flowchart

Fig. 2 is a flowchart of the ITROA, and the pseudocode is shown in Algorithm 2.

In this paper, as shown in Fig. 2, the steps of the ITROA are shown below.

Step 1: Initialize parameters, including the population size (N), the iterations (T), and the dimension (D).

Step 2: Initialize positions by Eq. (4). Calculate the fitness value to find the nearest location of the prey. Record its location and make it a target for the T-Rex.

Step 3: Update the populations' position by Eqs. (5)–(7).

Step 4: Initiate a T-Rex hunting process to simulate the probability problem of successful hunting by Eq. (1). If the hunt is successful, the T-Rex position is updated by Eq. (8). If the hunt fails, the location of the T-Rex is randomly updated. Then, calculate new fitness values.

Step 5: Compare the random fitness value $f(X)$ with the updated fitness value $f(X_{new})$ by Eq. (3). If $f(X) < f(X_{new})$, update the prey's position, target and fitness values. Then, the position is updated by Eqs. (9) and (10). Else, the target is equal to 0.

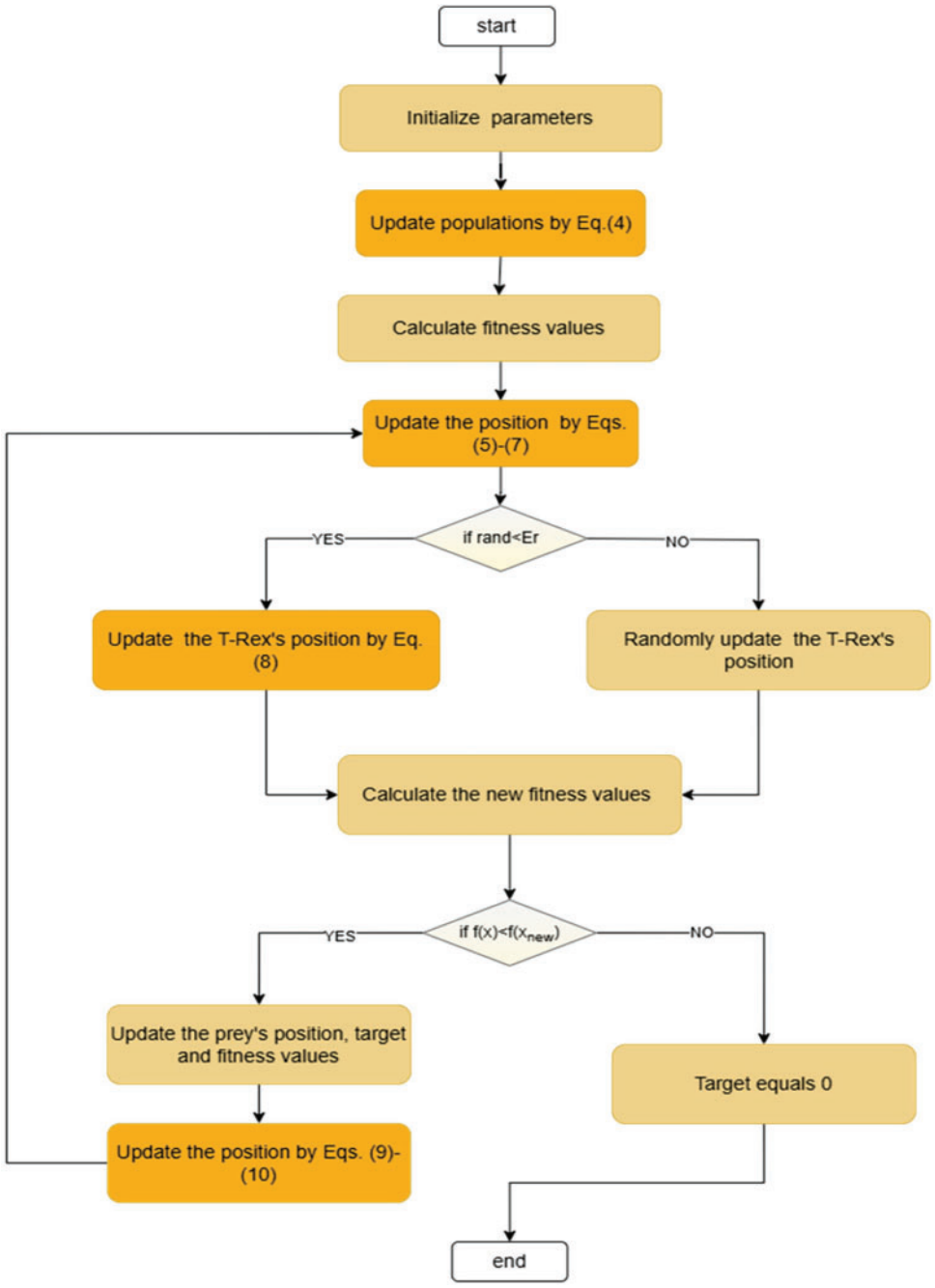

Figure 2: ITROA flowchart

Algorithm 2: Pseudocode of ITROA

Start ITROA.

1. Input problem information.
2. Initialization of the population size (N), the iterations (T), and the dimension (D).
3. Update populations by Eq. (4).

(Continued)

Algorithm 2 (continued)

Calculate the fitness value and record the optimal position.
Update populations' position by Eqs. (5)–(7).
For t = 1:T
  For i = 1:N
    Phase 1: *Hunting and chasing.*
    If *rand* < *Er*
      update the T-Rex's position by Eq. (8).
    else
      Randomly update the T-Rex's position.
    end
    Calculate the new fitness values.
    Phase 2: *Selection*
    If $f(X) < f(X_{new})$
      up date the prey's position, target, and fitness values.
      Update the position by Eqs. (9), (10).
    else
      Target equals 0.
    end
  End
Save the best solution attained.
End
Output the best solution attained.
END ITROA.

3.2 Metaheuristic-Driven Abnormal Traffic Detection Model

As for SDN abnormal traffic detection, Fig. 3 proposes a metaheuristic-driven abnormal traffic detection model for SDN based on ITROA. Fig. 4 shows the feature selection flowchart of ITROA. As shown in Fig. 3, the metaheuristic-driven abnormal traffic detection model for SDN based on ITROA can be divided into four phases: dataset acquisition phase, data processing phase, feature selection based on ITROA phase, and results evaluation phase.

(1) Data acquisition phase

Use tools to acquire an abnormal traffic detection dataset for SDN to obtain data for later analysis. This paper uses an abnormal traffic dataset in an SDN environment to simulate realistic network data and validate abnormal traffic detection for SDN.

(2) Data processing phase

Typically, the collected data contains issues like incompleteness, inconsistency, and redundancy. Therefore, data processing before the experiment is crucial. The InSDN dataset should first be cleaned, which includes removing duplicate data and addressing missing or inaccurate data. Then, to prevent the occurrence of format inconsistency problems, the InSDN dataset needs to be processed uniformly, and it is converted into numerical classes. Finally, to address the significant variations in the InSDN dataset's dimensions. A normalization function is used to process the data and map all data to [0, 1].

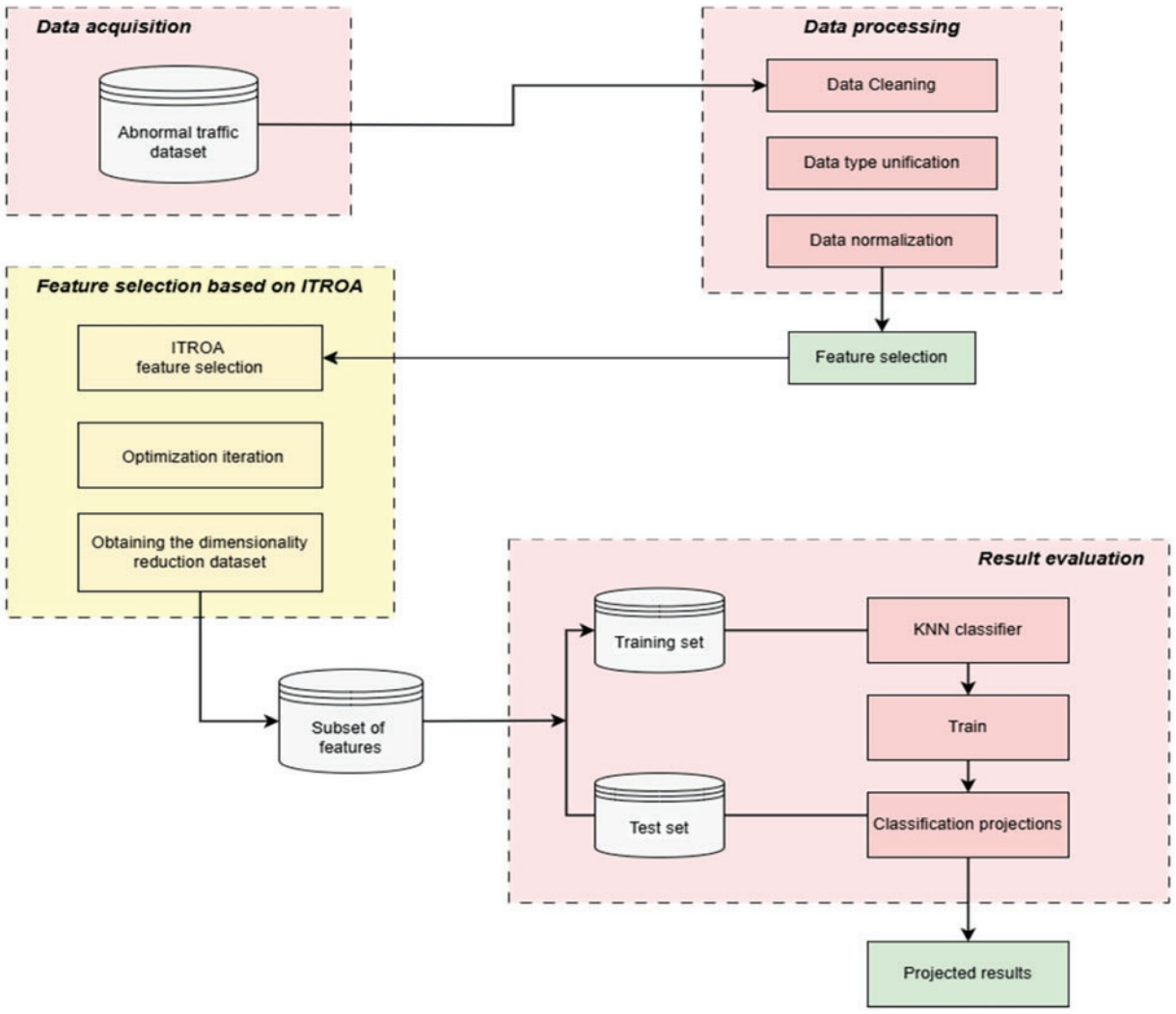

Figure 3: The abnormal traffic detection model for SDN based on ITROA

(3) Feature selection based on ITROA phase

Many invisible redundant features remain because of the vast amount of acquired data processed. Therefore, feature selection of the InSDN dataset effectively reduces redundant features and can improve the accuracy of abnormal traffic detection. This phase uses the ITROA for feature selection, as shown in Fig. 4. Iterative optimization search by the ITROA. The optimal subset's index is obtained at the end of the iteration. The phase now acquires the optimal subset of features. Thus, the purpose of removing redundant features is achieved, and the efficiency of abnormal traffic detection is improved.

(4) Result evaluation phase

The classifier is an important tool for evaluating the performance. In this paper, the KNN classifier is used for evaluation. As described in Fig. 4, firstly, the feature-selected InSDN dataset is divided into training and test sets. Then, the KNN classifier is used to classify and obtain the assessment indicators, including accuracy, recall, precision, and F1-score. Finally, projected results are reflected through the assessment indicators.

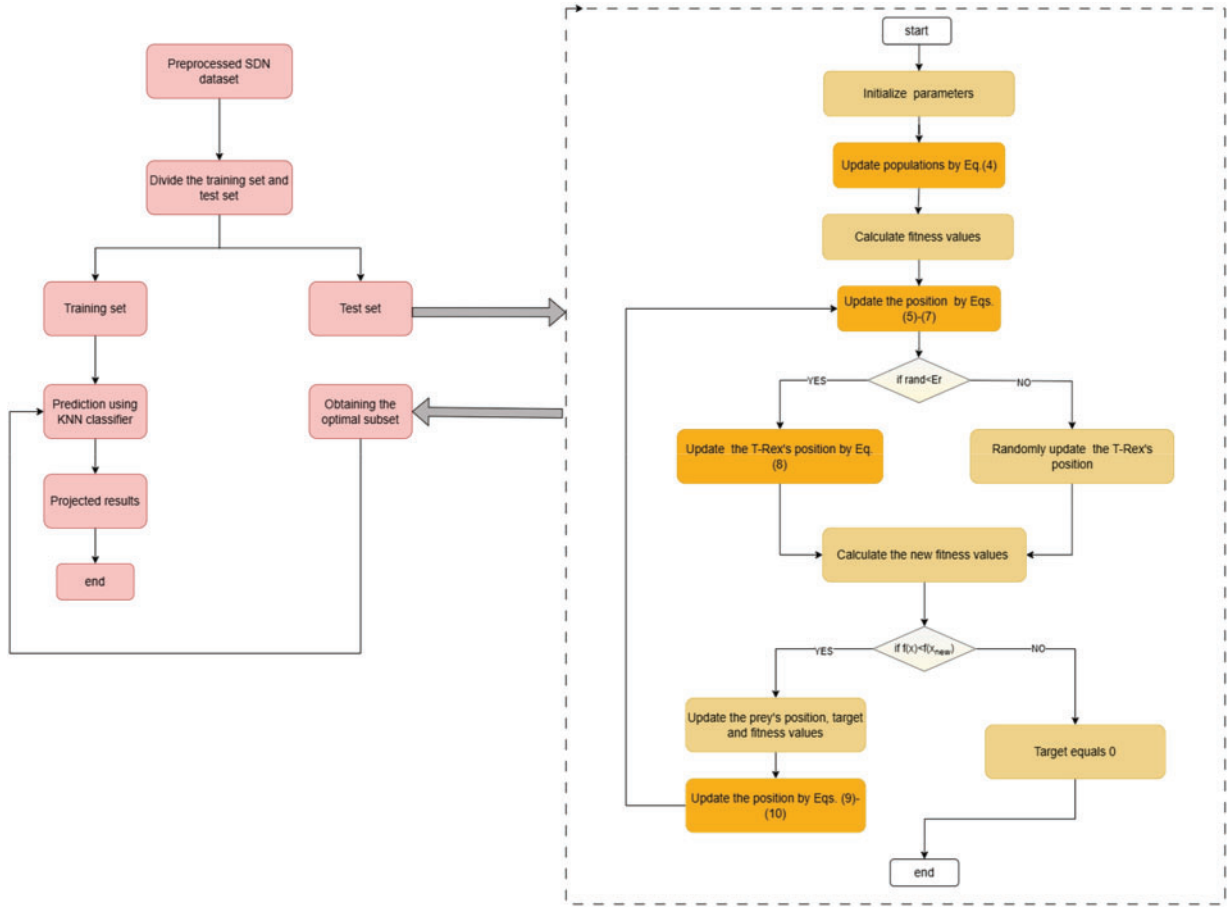

Figure 4: Feature selection flowchart of ITROA

4 Experimental Results and Analysis

There are three sections in this section. The benchmark functions are used in the first part to analyze the convergence of ITROA. In the second part, the efficiency of the ITROA is verified on the UCI dataset. In the third part, the performance of ITROA on abnormal traffic detection is validated on the InSDN dataset. The experiment is conducted in MATLABR2022b, using Intel(R) Core(TM) i5-8250U CPU@1.60 GHz 1.80 GHz, with a processor with Windows 11 as the software environment.

4.1 Benchmark Function Experiment

The algorithm is analyzed using simulation experiments to evaluate the ITROA's performance for optimization search. Table 1 selects six distinct kinds of benchmark functions. Table 2 shows the test function experiment results. Table 3 verifies that ITROA significantly outperforms the original TROA regarding optimization performance using the Wilcoxon sign rank test analysis. Fig. 5 shows the fitness convergence curves of six benchmark functions.

As presented in Table 1, three single-peak functions (F1–F3) and three multi-peak functions (F4–F6) are included. Set the parameters of the public part of the algorithm as follows: number of iterations T = 500. Considering the random nature of the algorithms, all the above algorithms are run 20 times repeatedly. Table 2 shows the calculated mean and standard deviation of the 20 results. The test results are shown in Table 3. The Wilcoxon sign rank test $p < 0.05$, h = 1 on the benchmark function. From

the results of the Wilcoxon sign rank test, ITROA and TROA perform significantly differently. As shown in Fig. 5, comparative experiments are conducted using PSO, Whale Optimization Algorithm (WOA), TROA, and ITROA.

Table 1: Benchmark function expressions

Type	Function	Dim	Range	Min
Unimodal	$F1(z) = \sum_{i=1}^{n} z_i^2$	30	$[-100, 100]$	0
Unimodal	$F2(z) = \sum_{i=0}^{n} \lvert Z_i \rvert + \prod_{i=0}^{n} \lvert Z_i \rvert$	30	$[-10, 10]$	0
Unimodal	$F3(z) = \max \lvert z_i \rvert, 1 \le i \le n$	30	$[-100, 100]$	0
Multimodal	$F4(z) = 20 \exp\left(0.2\sqrt{\frac{1}{n}\Sigma_{I=1}^{n} Z_i^2}\right) - \exp\left(\frac{1}{n}\Sigma_{I=1}^{n} \cos(2\pi z_i)\right) + 20 + e$	30	$[-3.2, 3.2]$	0
Multimodal	$F5(z) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \Sigma_{i-1}^{2}} \left(z_i - a_{ij}\right)\right)$	30	$[-65, 65]$	1
Multimodal	$F6(z) = 4z_1^2 - 2.1z_1^4 + \frac{1}{3}z_1^6 + z_1 z_2 - 4z_z^2 + 4z_2^4$	30	$[-5, 5]$	−1.0316

Table 2: Test function experiment results

Function	Type	PSO	WOA	TROA	ITROA
F1	Mean	2.41E−01	5.99E−73	1.85E−199	0.00E+00
	Std	6.58E−02	1.68E−72	0.00E+00	0.00E+00
F2	Mean	2.54E+00	4.78E−48	4.27E−105	4.31E−174
	Std	4.60E−01	1.24E−47	1.91E−104	0.00E+00
F3	Mean	5.48E−01	2.41E−01	6.34E−108	2.41E−177
	Std	4.01E−02	1.43E−01	2.82E−107	0.00E+00
F4	Mean	1.98E+00	4.17E−15	4.44E−16	4.44E−16
	Std	3.59E−01	2.44E−15	0.00E+00	0.00E+00
F5	Mean	1.26E+01	4.63E+00	1.26E+01	1.01E+01
	Std	3.23E−12	3.80E+00	1.87E−10	1.32E−02
F6	Mean	−1.03E+00	−1.03E+00	−9.99−01	−1.03E+00
	Std	9.92E−07	4.90E−09	1.45E−04	6.15E−05

Table 3: The Wilcoxon sign rank test results

Value	F1	F2	F3	F4	F5	F6
p	6.56E−24	2.13E−16	5.51E−22	2.68E−2	1.81E−14	3.93E−167
h	1	1	1	1	1	1

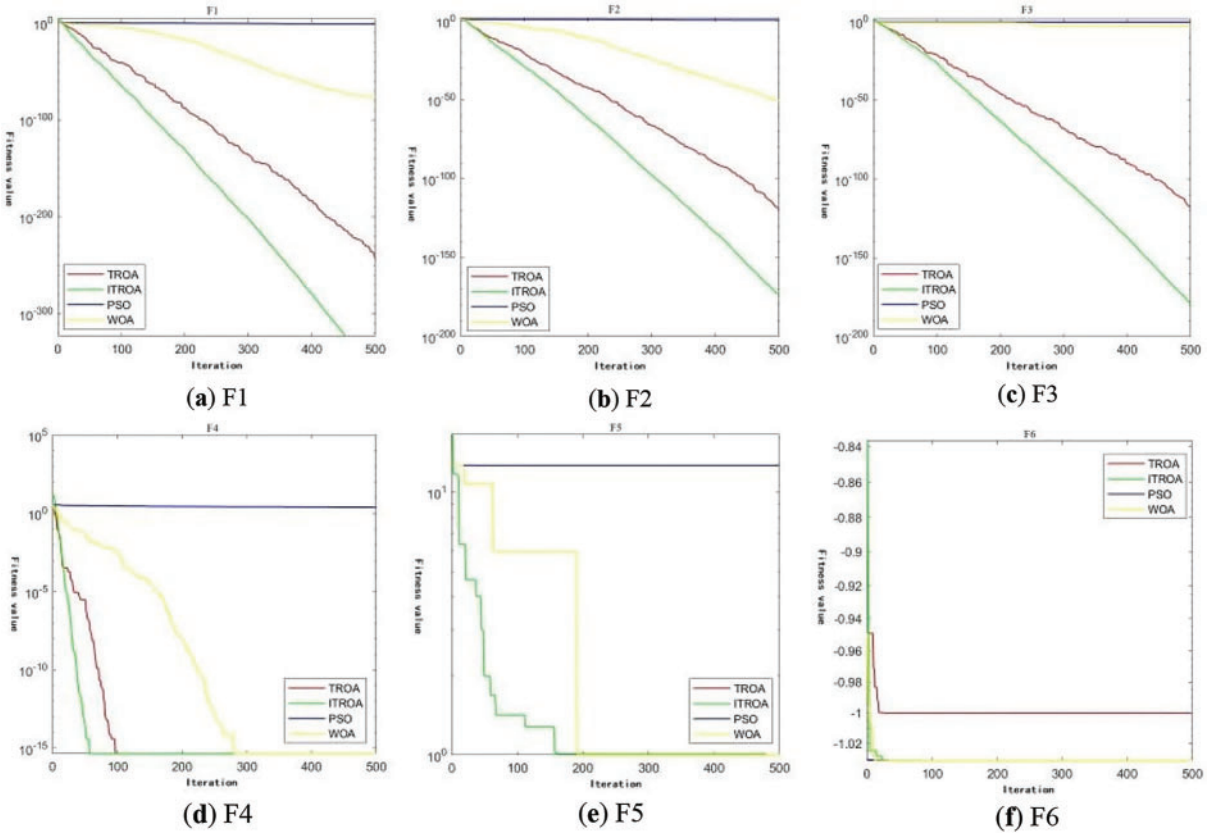

Figure 5: Convergence curves of fitness

The findings are examined by choosing one test function at random from the optimization search procedure to give more information on the ITROA's convergence rate. The results show that the ITROA exhibits better performance and stronger robustness under the same iterative optimization conditions.

4.2 Experiments Using UCI Datasets

The UCI datasets are used to validate the effectiveness of the ITROA. As described in Table 4, three datasets are selected for testing. The three datasets in Table 5 display the experimental outcomes of various algorithms.

Table 4: UCI datasets

Number	Dataset name	Sample size	Number of features
1	Ionosphere	351	34
2	Heatstatlog	270	13
3	Sonar	208	61

Table 5: Experimental results for the UCI datasets

Algorithms	Metrics	Ionosphere	Heatstatlog	Sonar
WOA	Accuracy	92.4	87.6	93.3
	Recall	98.5	86.1	94.1
	F1-score	94.4	86.1	94.8
	Precision	90.5	86.1	95.5
PSO	Accuracy	93.3	91.4	93.3
	Recall	98.5	86.1	97.1
	F1-score	95.0	89.9	95.0
	Precision	91.8	94.0	93.0
TROA	Accuracy	94.3	88.9	88.6
	Recall	97.1	**91.7**	89.7
	F1-score	95.7	88.0	91.0
	Precision	94.3	84.6	92.4
ITROA	Accuracy	**96.1**	**93.8**	**97.1**
	Recall	**1.0**	88.9	**98.5**
	F1-score	**97.1**	**92.8**	**97.8**
	Precision	**94.4**	**97.0**	**97.1**

As presented in Table 5, on the Ionosphere, Heatstatlog, and Sonar datasets, ITROA achieves the best values for almost all assessment indicators. The data shown in bold are the excellent values for each indicator. In the Heatstatlog dataset, the recall of ITROA is slightly worse than the TROA algorithm. However, ITROA significantly outperforms other algorithms in other evaluation metrics. Taken as a whole, ITROA demonstrates effectiveness.

4.3 Experiment Using InSDN Dataset

The InSDN dataset is separated into normal and abnormal traffic categories. Numerous socket details, including source IP, destination IP, stream ID, and others, are included in the InSDN dataset. In addition to the labeled columns, remove all socket functions to avoid overfitting problems. There are 49 distinct features in the finished dataset. As these features have different range intervals, it is necessary to normalize these data and restrict the proportion of these values to between 0 and 1.

4.3.1 Binary Classification Results

In this section, the binary classification problem is considered. Classify input data accurately into two categories: normal and abnormal. Ten thousand data points are chosen at random from the dataset for testing. Binary encodings are utilized in computer systems to represent these two categories of data appropriately. Specifically, the data points of the normal class are coded as 0, and the data points of the abnormal class are coded as 1.

Table 6 shows the Wilcoxon signed rank test analysis between TROA and ITROA. Based on the final experimental results, the InSDN dataset exhibits a high level of effectiveness on each algorithm, as shown in Table 7. The values of the fitness convergence curves for the four algorithms PSO, WOA, TROA, and ITROA on the InSDN dataset are displayed in Fig. 6a. As shown in Fig. 6b, data visualization shows the various evaluation performance metrics, and Fig. 6c shows the confusion matrix for binary classification.

Table 6: The Wilcoxon sign rank test results

	1	2	3	4	5	6	7
p	3.34E−10	1.07E−04	4.72E−12	6.45E−07	3.93E−08	1.06E−04	6.59E−09
h	1	1	1	1	1	1	1

Table 7: Binary classification results

Metrics	WOA	PSO	TROA	ITROA
Accuracy	98.87	99.08	99.18	**99.37**
Recall	99.35	**99.46**	98.24	99.35
Precision	98.93	99.14	99.35	**99.67**
F1-score	99.14	99.30	99.31	**99.51**
Number of features	7	17	9	**5**
Prediction times (s)	0.0464	0.0520	0.0489	**0.204**

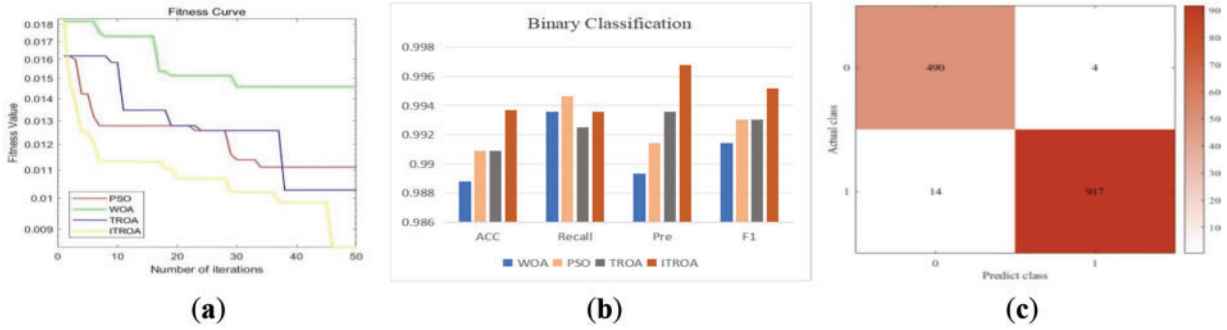

Figure 6: Binary classification results. (a) Convergence curves; (b) Histogram; (c) Confusion matrix

From Table 6, the results of 7 randomized experiments were taken, and it is known that all the results $p < 0.5$, h = 1. There is a significant gap between ITROA and TROA. As shown in Fig. 6a, ITROA has shown clear advantages. As described in Table 7, the ITROA achieves significant optimization in feature selection. The data shown in bold are the excellent values for each indicator. The quantity of features it selects is reduced to five, significantly reducing the data's redundancy. Due to feature selection, which eliminates many redundant features, ITROA has the fastest prediction time. Thus, it increases the algorithm's operational effectiveness and prediction accuracy. In terms of indicators, ITROA was the second most effective in recall. However, the ITROA shows clear advantages in terms of assessment indicators. It features an approximate 0.5% rise in accuracy, a 0.37% rise in F1-score, and a 0.53% rise in precision. The ITROA shows good classification performance with optimal results in abnormal traffic detection.

4.3.2 Multi-Classification Results

This section performs a multi-classification operation on the InSDN dataset by selecting 10,000 dates. The feature column labels of these selected data are encoded against their feature column labels, encoding them systematically as numerical labels from 0 to 7. It provides a solid basis for subsequent multi-classification operations.

Table 8, like Section 4.2, shows the Wilcoxon signed rank test analysis between TROA and ITROA. Table 9 shows the results of the multi-classification experiment. The data shown in bold are the excellent values for each indicator. Table 10 shows the model prediction times for the four algorithms after feature selections. In the process of selecting the data, it is found that the three numbers of types of data, Web-Attack, BOTNET, and UR2, are relatively small. Due to the insufficient sample size, these three data sets are not sufficiently referential and convincing in the analyses and comparisons. Therefore, they were ignored for the time being in the subsequent study. The convergence curves of the four algorithms' fitness values are displayed in Fig. 7a. From Fig. 7c–f, it can be seen that the ITROA shows optimal results whether dealing with normal or abnormal traffic.

Table 8: The Wilcoxon sign rank test results

	1	**2**	**3**	**4**	**5**	**6**	**7**
p	3.03E−08	1.15E−15	5.06E−15	5.24E−10	2.94E−12	1.28E−09	1.04E−04
h	1	1	1	1	1	1	1

Table 9: Multi-classification results

		Normal	**BFA**	**DDos**	**Dos**	**Probe**	**Web-Attack**	**BOTNET**	**UR2**
Precision	WOA	98.31	90.63	98.84	98.29	94.54	73.33	92.45	100.0
	PSO	98.25	91.08	99.84	98.61	93.81	71.74	90.74	66.67
	TROA	98.52	90.35	**99.76**	97.29	94.23	76.09	89.09	100.0
	ITROA	**98.46**	**92.29**	99.53	**98.64**	**95.01**	74.47	90.74	80.00
Recall	WOA	98.73	**93.80**	99.30	95.68	93.77	60.00	100.0	20.00
	PSO	98.80	93.53	99.30	94.68	**94.58**	60.00	100.0	40.00
	TROA	98.66	92.27	99.53	95.68	92.95	63.64	100.0	60.00
	ITROA	**99.30**	93.53	**99.60**	**96.34**	92.95	63.64	100.0	80.00
F1-score	WOA	98.52	91.19	99.07	96.97	94.15	66.00	96.08	33.33
	PSO	98.53	**92.90**	99.07	96.61	94.17	65.35	95.15	50.00
	TROA	98.59	91.05	**99.64**	96.48	93.58	69.31	94.23	75.00
	ITROA	**98.88**	92.29	99.34	**97.48**	**94.97**	68.63	95.15	80.00
Accuracy	WOA				96.47				
	PSO				96.50				
	TROA				96.37				
	ITROA				**96.73**				

Table 10: The prediction times

	WOA	**PSO**	**TROA**	**ITROA**
Number of features	21	18	16	6
Prediction times (s)	0.1040	0.0852	0.0839	0.0575

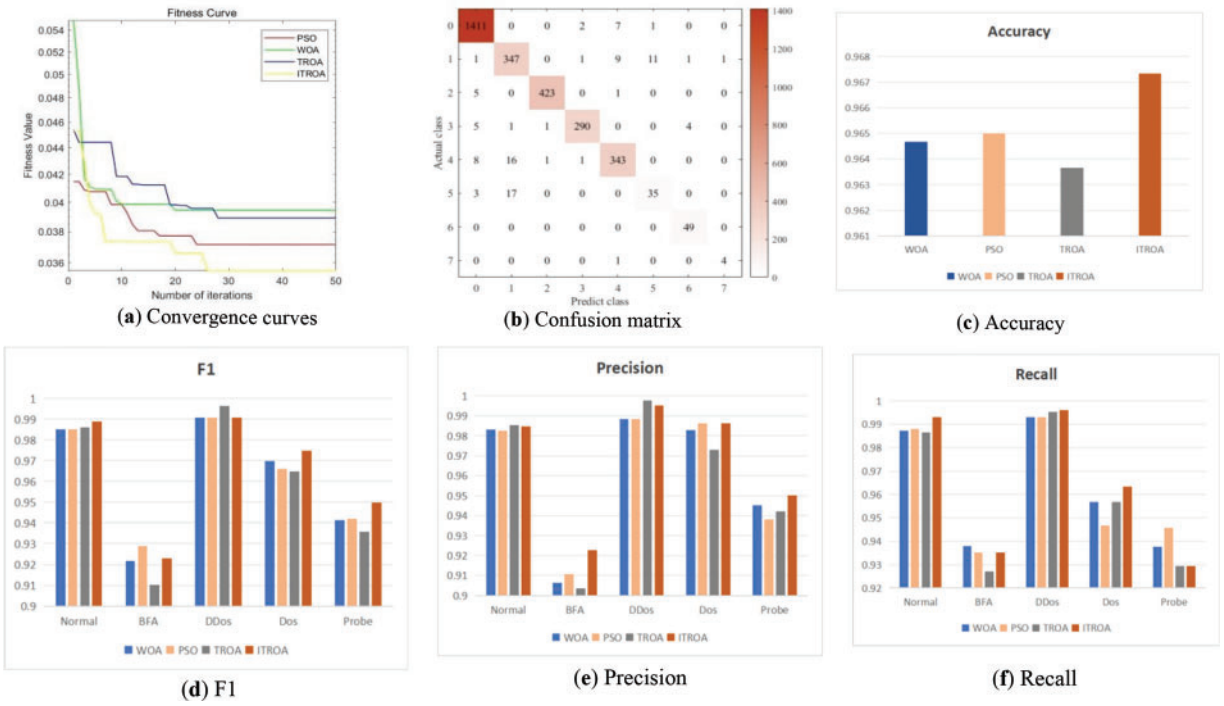

Figure 7: Multi-classification results

The core research in this section is to analyze and explore the abnormal traffic in depth. Table 9 shows a significant gap between ITROA and TROA based on the values of p and h. Table 10 shows the model prediction times for the four algorithms. The ITROA exhibits excellent time cost and dramatically improves the overall performance. The results are shown in Fig. 7a. TROA does not converge as well as PSO in the early stage, but it is comparable to PSO in the middle stage but outperforms PSO later on. However, the convergence of the ITROA algorithm shows a clear advantage. Fig. 7b shows the confusion matrix for abnormal traffic detection under ITROA optimization conditions. As shown in Fig. 7c–f, ITROA shows a clear advantage in recall, F1-score, precision, and accuracy in the Normal category. On the DoS attack category, ITROA also shows good detection. There was an improvement of about 1.3% in precision, 1.7% in recall, and 1% in F1-score. It also improves accuracy by about 0.4%. The Probe attack category, excluding recall, significantly improves on the other three evaluation metrics: accuracy, precision, and F1-score. ITROA shows good detection. There was an improvement of about 0.8% in precision and F1-score. On the BFA and DDos attack categories, although some of the indicators are not as good as hoped, ITROA performs better when the four indicators are combined. Although its effect is not significant regarding recall and F1 in the BFA attack category, it shows some advantages regarding precision and accuracy. In the precision rate, it increased by 1.5%. On the DoSS attack category, the combined effect of ITROA is better.

Experiments show that the ITROA improves the original algorithm and outperforms other algorithms. It shows an advantage in abnormal traffic detection.

5 Conclusions and Future Work

Considering the problem of the gradual complexity of abnormal traffic detection. This paper proposes a metaheuristic-driven abnormal traffic detection model for SDN based on ITROA to improve the efficiency of abnormal traffic detection. ITROA is used during the feature selection stage of the model to solve the

problem of redundant abnormal traffic data for SDN. Then, the UCI datasets and benchmark functions are used to verify the effectiveness of ITROA. Finally, the experiment on the InSDN dataset shows that compared to TROA, WOA, and PSO. In the binary classification experiments, ITROA showed excellent performance on all metrics. In the multi-classification experiments, ITROA demonstrates better results comprehensively. However, as a specific dataset for SDN, the InSDN dataset still has some limitations. Therefore, we will continue exploring new dataset construction methods for SDN abnormal traffic detection to overcome the limitations in the subsequent work.

Acknowledgement: The authors thank the editor and reviewers for their valuable comments.

Funding Statement: This work has been supported by the National Natural Science Foundation of China under Grant 61602162 and the Hubei Provincial Science and Technology Plan Project under Grant 2023BCB041.

Author Contributions: Study conception and design: Hui Xu, Jiahui Chen, and Zhonghao Hu; data collection: Jiahui Chen; analysis and interpretation of results: Jiahui Chen and Hui Xu; draft manuscript preparation: Hui Xu, Jiahui Chen, and Zhonghao Hu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The dataset used in this study is openly accessible and reliable. It can be obtained from the following website: http://aseados.ucd.ie/datasets/SDN/ (accessed on 1 January 2025).

Ethics Approval: Not applicable.

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

References

1. Emad Ali T, Hussein Morad A, Abdala MA. Load balance in data center SDN networks. Int J Electr Comput Eng (IJECE). 2018;8(5):3084. doi:10.11591/ijece.v8i5.pp3084-3091.
2. Scott-Hayward S, O'Callaghan G, Sezer S. Sdn security: a survey. In: 2013 IEEE SDN for Future Networks and Services (SDN4FNS); 2013 Nov 11–13; Trento, Italy.
3. Kreutz D, Ramos FMV, Veríssimo PE, Rothenberg CE, Azodolmolky S, Uhlig S. Software-defined networking: a comprehensive survey. Proc IEEE. 2015;103(1):14–76. doi:10.1109/JPROC.2014.2371999.
4. Feamster N, Rexford J, Zegura E. The road to SDN: an intellectual history of programmable networks. ACM Sigcomm Comp Com. 2014;44(2):87–98. doi:10.1145/2602204.2602219.
5. Alabdulatif A, Thilakarathne NN, Aashiq M. Machine learning enabled novel real-time IoT targeted DoS/DDoS cyber attack detection system. Comput Mater Contin. 2024;80(3):3655–83. doi:10.32604/cmc.2024.054610.
6. Mihoub A, Ben Fredj O, Cheikhrouhou O, Derhab A, Krichen M. Denial of service attack detection and mitigation for Internet of Things using looking-back-enabled machine learning techniques. Comput Electr Eng. 2022;98:107716. doi:10.1016/j.compeleceng.2022.107716.
7. Aljuhani A. Machine learning approaches for combating distributed denial of service attacks in modern networking environments. IEEE Access. 2021;9:42236–64. doi:10.1109/ACCESS.2021.3062909.
8. Ali MH, Jaber MM, Abd SK, Rehman A, Awan MJ, Damaševičius R, et al. Threat analysis and distributed denial of service (DDoS) attack recognition in the Internet of Things (IoT). Electronics. 2022;11(3):494. doi:10.3390/electronics11030494.
9. Niyaz Q, Sun W, Javaid AY. A deep learning based DDoS detection system in software-defined networking (SDN). ICST Trans Secur Saf. 2017;4(12):153515. doi:10.4108/eai.28-12-2017.153515.
10. Tang TA, Mhamdi L, McLernon D, Ali Raza Zaidi S, Ghogho M. Deep learning approach for network intrusion detection in software defined networking. In: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM); Oct 26–29; Fez, Morocco.
11. Wang K, Fu Y, Duan X, Liu T, Xu J. Abnormal traffic detection system in SDN based on deep learning hybrid models. Comput Commun. 2024;216:183–94. doi:10.1016/j.comcom.2023.12.041.

12. Arevalo-Herrera J, Camargo Mendoza JE, Martinez Torre JI. Network anomaly detection with machine learning techniques for SDN networks. In: 2022 the 7th International Conference on Information and Education Innovations (ICIEI); 2022 Apr 14–16; Belgrade, Serbia.
13. Stein G, Chen B, Wu AS, Hua KA. Decision tree classifier for network intrusion detection with GA-based feature selection. In: Proceedings of the 43rd Annual Southeast Regional Conference—Volume 2; 2005 Mar 19–20; Kennesaw, Georgia.
14. Zainal A, Maarof MA, Shamsuddin SM. Lecture notes in computer science. Berlin/Heidelberg, Germany: Springer; 2007.
15. Mojtahedi A, Sorouri F, Souha AN, Molazadeh A, Mehr SS. Feature selection-based intrusion detection system using genetic whale optimization algorithm and sample-based classification. arXiv:2201.00584. 2022.
16. Lin SW, Ying KC, Lee CY, Lee ZJ. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Appl Soft Comput. 2012;12(10):3285–90. doi:10.1016/j.asoc.2012.05.004.
17. Xu H, Lu Y, Guo Q. Application of improved butterfly optimization algorithm combined with black widow optimization in feature selection of network intrusion detection. Electronics. 2022;11(21):3531. doi:10.3390/electronics11213531.
18. Xu H, Hu Y, Cao W, Han L. An improved jump spider optimization for network traffic identification feature selection. Comput Mater Contin. 2023;76(3):3239–55. doi:10.32604/cmc.2023.039227.
19. Xu H, Chai X, Liu H. A multi-controller placement strategy for hierarchical management of software-defined networking. Symmetry. 2023;15(8):1520. doi:10.3390/sym15081520.
20. Li F, Xu H, Qiu F. Modified artificial rabbits optimization combined with bottlenose dolphin optimizer in feature selection of network intrusion detection. Electron Res Arch. 2024;32(3):1770–800. doi:10.3934/era.2024081.
21. Wu Q, Xu H, Liu M. Applying an improved dung beetle optimizer algorithm to network traffic identification. Comput Mater Contin. 2024;78(3):4091–107. doi:10.32604/cmc.2024.048461.
22. Yusta SC. Different metaheuristic strategies to solve the feature selection problem. Pattern Recognit Lett. 2009;30(5):525–34. doi:10.1016/j.patrec.2008.11.012.
23. Iba K. Reactive power optimization by genetic algorithm. IEEE Trans Power Syst. 1994;9(2):685–92. doi:10.1109/59.317674.
24. Hamadneh T, Kaabneh K, AbuFalahah I, Bektemyssova G, Shaikemelev G, Umutkulov D, et al. Magnificent frigatebird optimization: a new bio-inspired metaheuristic approach for solving optimization problems. Comput Mater Contin. 2024;80(2):2721–41. doi:10.32604/cmc.2024.054317.
25. Sahu VSDM, Samal P, Panigrahi CK. *Tyrannosaurus* optimization algorithm: a new nature-inspired meta-heuristic algorithm for solving optimal control problems. e-Prime—Adv Electr Eng Electron Energy. 2023;5:100243. doi:10.1016/j.prime.2023.100243.
26. Yuan W, Liao H, Yuan X. Optimization of vibration and sound insulation in GPLRC honeycomb structures based on circle chaos mapping and Levy flight-enhanced YDSE with constraints. Appl Math Model. 2024;134:752–75. doi:10.1016/j.apm.2024.06.018.
27. Tumari MZM, Ahmad MA, Mohamed Z. Identification of the continuous-time Hammerstein models with sparse measurement data using improved marine predators algorithm. Arab J Sci Eng. 2024;358:4546. doi:10.1007/s13369-024-09692-1.
28. Mohd Tumari MZ, Ahmad MA, Suid MH, Ghazali MR, Tokhi MO. An improved marine predators algorithm tuned data-driven multiple-node hormone regulation neuroendocrine-PID controller for multi-input-multi-output gantry crane system. J Low Freq Noise Vib Act Contr. 2023;42(4):1666–98. doi:10.1177/14613484231183938.
29. Biswas S, Shaikh A, Ezugwu AE, Greeff J, Mirjalili S, Bera UK, et al. Enhanced prairie dog optimization with Levy flight and dynamic opposition-based learning for global optimization and engineering design problems. Neural Comput Appl. 2024;36(19):11137–70. doi:10.1007/s00521-024-09648-4.