Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.087054

ARTICLE

An Improved Dream Optimization Algorithm-Driven Feature Selection Model for IoT Traffic Anomaly Detection

Hui Xu, Shuang Qu* and Pan Hu

School of Computer Science and Artificial Intelligence, Hubei University of Technology, Wuhan, China

*Corresponding Author: Shuang Qu. Email: 102411193@hbut.edu.cn

Received: 09 June 2026; Accepted: 28 July 2026; Published: 15 September 2026

ABSTRACT: With the rapid growth in the number of end devices in the Internet of Things (IoT), network traffic has become increasingly complex and redundant, while multiple attack types often coexist, posing major challenges to traffic anomaly detection. Traditional machine learning-based methods for IoT traffic anomaly detection often suffer from severe feature redundancy, high computational complexity, and low detection efficiency, making it difficult to simultaneously achieve high detection accuracy and computational efficiency. To address this issue, metaheuristic algorithms are often introduced in the feature selection stage to reduce feature redundancy and improve detection efficiency. However, the original Dream Optimization Algorithm (DOA) still has limitations when handling complex optimization problems such as feature selection for IoT traffic anomaly detection, including reduced population diversity in later iterations, insufficient global exploration capability, and a tendency to fall into local optima, which in turn affect the stability of feature selection and detection performance. Therefore, this paper proposes an Improved Dream Optimization Algorithm (IDOA) based on the original DOA, and further constructs an IDOA-based IoT traffic anomaly detection model for feature selection and classification tasks. Experimental results on benchmark functions and multiple publicly available datasets demonstrate the effectiveness and competitiveness of the proposed method. On four general-purpose classification benchmarks from the University of California, Irvine (UCI) Machine Learning Repository, IDOA demonstrates competitive feature-selection and classification performance. On the NF-ToN-IoT-v2 dataset, the binary classification accuracy reaches 98.01% with only 7.4 selected features on average, and the multiclass classification accuracy reaches 96.382% with 9.20 selected features on average. The experimental results indicate that the proposed method reduces feature redundancy and improves prediction efficiency in the evaluated offline setting while maintaining competitive detection accuracy.

KEYWORDS: Internet of Things security; traffic anomaly detection; metaheuristic optimization; dream optimization algorithm; feature selection

1 Introduction

With the rapid development of Internet of Things (IoT) technology, massive heterogeneous devices have been integrated into modern networks, supporting applications such as smart cities, industrial IoT, intelligent transportation, and smart healthcare. However, large-scale interconnection also expands the attack surface. Device heterogeneity, limited resources, diverse protocols, and weak built-in security mechanisms make IoT systems vulnerable to Distributed Denial-of-Service (DDoS) attacks, botnet control, and data theft [1–3]. Recent honeypot observations further show that IoT botnets continue to exploit brute-force attacks and protocol-specific vulnerabilities in cloud-connected environments [4]. Since such attacks are commonly reflected in anomalous traffic patterns, efficient traffic anomaly detection is critical for IoT security.

Traditional anomaly detection methods mainly rely on rules or statistical feature matching, making them highly dependent on prior knowledge and difficult to adapt to evolving attacks [5]. Machine learning and deep learning methods have therefore become widely used because of their strong representation capabilities [6–8]. Recent studies on Software-Defined Networking (SDN) and IoT security have also demonstrated the potential of deep learning and graph-based models for identifying abnormal traffic patterns [9–11].

Recent cybersecurity event-detection systems increasingly focus on complex environments such as SDN-enabled networks and IoT infrastructures. Deep learning-based DDoS detection and SDN intrusion detection models have been developed to capture abnormal traffic behaviors [7,8]. Hybrid deep learning models combine different neural architectures to improve traffic representation [9,10], while graph neural networks model structural relationships in IoT traffic [11]. However, these methods are generally classifier- or representation-oriented and often rely on relatively complete feature sets. Consequently, feature redundancy, high-dimensional inputs, and unnecessary computational cost remain important issues, making feature selection an essential step in efficient IoT anomaly detection.

Learning-based methods also face challenges caused by redundant, irrelevant, and noisy traffic features. NetFlow records and packet-level statistics often contain dozens or hundreds of variables [12]. Using all features may increase computational cost, aggravate overfitting, and weaken generalization. Therefore, feature selection is important for reducing model complexity and improving detection efficiency [13,14]. Hybrid feature selection methods have also shown effectiveness in removing irrelevant and redundant features [15].

Recent work on dynamic edge networks has also emphasized lightweight decision-making for maintaining timely and usable services under changing request and resource conditions [16]. Such studies improve operational efficiency at the service-management level, whereas the present work addresses efficiency at the data-processing level by removing redundant traffic features before anomaly classification. This complementary perspective supports efficient operation in resource-constrained IoT environments.

Wrapper-based feature selection using metaheuristic algorithms has attracted considerable attention because of its global search capability [17]. Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), and Genetic Algorithm (GA) have been widely applied to feature selection and optimization [18–21]. In traffic anomaly detection, combining feature selection with intelligent classifiers can balance detection performance and computational cost [22]. Improved butterfly optimization, jump spider optimization, and dung beetle optimization have also been applied to intrusion-detection feature selection [23–25]. In addition, the Honey Badger Algorithm has been used to optimize a hybrid autoencoder and gated recurrent unit model for IoT cyber threat detection [26].

Despite these advances, metaheuristic algorithms may still suffer from exploration–exploitation imbalance, premature convergence, and population diversity loss in complex feature spaces [27,28]. Lang and Gao proposed the Dream Optimization Algorithm (DOA), inspired by human dreaming [29]. DOA is selected as the baseline optimizer because its memory, forgetting-and-replenishment, and dream-sharing mechanisms provide a suitable framework for targeted improvement. However, its deterministic memory update and limited perturbation strategy may restrict population diversity, global exploration, and convergence accuracy in traffic feature selection.

To address these limitations, this paper proposes an Improved Dream Optimization Algorithm (IDOA). By incorporating a Lévy-flight-based adaptive perturbation mechanism, an elite learning strategy, and an improved memory strategy, IDOA enhances the exploration capability and convergence stability of DOA.

An IDOA-based IoT traffic anomaly detection model is further constructed to integrate feature selection with classification. The main contributions are as follows:

(1) An improved Dream Optimization Algorithm is proposed to reduce redundant traffic features in IoT anomaly detection.
(2) An IDOA-based anomaly detection model is constructed to integrate feature selection and classification.
(3) Experiments on benchmark functions and public datasets demonstrate the convergence, stability, and detection performance of the proposed method.

The remainder of this paper is organized as follows. Section 2 introduces DOA. Section 3 presents the improvement strategies. Section 4 describes the IDOA-based IoT traffic anomaly detection model. Section 5 reports the experiments and results. Section 6 concludes the paper and discusses future work.

2 Dream Optimization Algorithm

The Dream Optimization Algorithm (DOA) is a novel metaheuristic optimization algorithm inspired by human dreaming, in which mechanisms such as memory retention, partial forgetting, and logical self-organization are mathematically modeled to achieve effective global exploration and local exploitation.

2.1 Initialization Phase

DOA first generates an initial population randomly within the search space, which is formulated as follows:

$$X_i = X_l + \text{rand} \times (X_u - X_l), \quad i = 1, 2, \ldots, N \tag{1}$$

where N is the population size, X_i denotes the ith individual, X_l and X_u are the lower and upper bounds of the search space, respectively, and rand is a Dim-dimensional vector uniformly distributed in $[0, 1]$.

Accordingly, the initialized population can be expressed as

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,Dim} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,Dim} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,Dim} \end{bmatrix} \tag{2}$$

where $x_{i,j}$ denotes the position of the ith individual in the jth dimension.

2.2 Exploration and Exploitation Phases

The optimization process of DOA consists of an exploration phase (0 to T_d) and an exploitation phase (T_d to $T_{\max}$). Both phases share the same framework but use different parameter settings.

(1) Memory strategy

In the exploration phase, the memory strategy is based on the group-best individual:

$$X_i^{t+1} = X_{best_q}^t \tag{3}$$

where $X_{best_q}^t$ denotes the best individual in the qth group at iteration t.

In the exploitation phase, the memory strategy is based on the global-best individual:

$$X_i^{t+1} = X_{best}^t \tag{4}$$

where X_{best}^t denotes the position of the best individual in the entire population at iteration t.

(2) Forgetting-and-replenishment strategy

Both phases use the following update rule:

$$x_{i,j}^{t+1} = x_{\text{best},j}^t + \left(x_{l,j} + \text{rand}(x_{u,j} - x_{l,j})\right) \frac{1}{2}\left(\cos\left(\pi \frac{t+\Delta}{T_{\max}}\right) + 1\right) \tag{5}$$

where $\Delta = T_{\max} - T_d$ and $x_{\text{best},j}^t = x_{\text{best}_q,j}^t$ during exploration, whereas $\Delta = 0$ and $x_{\text{best},j}^t$ denotes the corresponding dimension of the global-best solution during exploitation.

The cosine term adaptively balances global exploration and local exploitation. In feature selection, a "forgotten dimension" refers to a decision variable selected for replacement, and the replenished value may change whether the corresponding feature is retained after binarization.

(3) Dream-sharing strategy

Both phases employ the same sharing mechanism:

$$x_{i,j}^{t+1} = \begin{cases} x_{m,j}^{t+1}, & m \le i \\ x_{m,j}^t, & i < m \le N \end{cases} \qquad j \in \{K_1, K_2, \ldots, K_k\} \tag{6}$$

where m denotes the index of a randomly selected individual, and k is set to k_q in the exploration phase and k_r in the exploitation phase. In feature selection, dream sharing means copying selected decision-variable values from another candidate solution, thereby exchanging feature-subset information within the population rather than sharing IoT traffic data.

2.3 Parameter Settings

The key parameter settings of DOA are given as follows:

- Maximum number of iterations in the exploration phase:

$$T_d = \frac{9}{10} \times T_{\max} \tag{7}$$

- Number of forgotten dimensions for each group in the exploration phase:

$$k_q = \text{randi}\left(\frac{Dim}{8 \times q}, \max\left\{2, \frac{Dim}{3 \times q}\right\}\right) \tag{8}$$

  where $\text{randi}(a, b)$ denotes an integer randomly selected from the interval $[a, b]$, and $q = 1, 2, 3, 4, 5$ represents the group index.
- Number of forgotten dimensions in the exploitation phase:

$$k_r = \text{randi}\left(2, \max\left\{2, \frac{Dim}{3}\right\}\right) \tag{9}$$

- Strategy selection probability: $u = 0.9$. When the random number satisfies rand $< u$, the forgetting-and-replenishment strategy is executed; otherwise, the dream-sharing strategy is performed.

2.4 Boundary Handling

DOA employs two boundary-handling strategies.

- Handling for low-dimensional problems ($Dim \leq 15$):

$$x_{i,j}^{t+1} = x_{l,j} + \text{rand} \times (x_{u,j} - x_{l,j}) \tag{10}$$

- Handling for high-dimensional problems ($Dim > 15$):

$$x_{i,j}^{t+1} = \begin{cases} x_{m,j}^{t+1}, & m < i \\ x_{m,j}^{t}, & i < m \leq N \end{cases} \tag{11}$$

These strategies ensure that all variables remain within the search space.

2.5 Algorithm Complexity Analysis

The computational cost of DOA mainly arises from population initialization, position updating, and fitness evaluation. Population initialization requires $\mathcal{O}(N \times D)$, while position updating over all iterations requires $\mathcal{O}(T_{\max} \times N \times D)$. Fitness evaluation requires $\mathcal{O}(T_{\max} \times N \times E)$, where N is the population size, D is the search-space dimension, and E is the cost of one fitness evaluation. Therefore, the dominant time complexity of DOA is

$$\text{TC}_{\text{DOA}} = \mathcal{O}(T_{\max} \times N \times (D + E)). \tag{12}$$

3 Improvement Strategies

DOA provides a flexible framework for balancing exploration and exploitation through its memory, forgetting-and-replenishment, and dream-sharing mechanisms. However, its deterministic memory reset, cosine-based perturbation, and limited use of elite information may reduce population diversity and convergence performance in complex feature-selection tasks. Therefore, IDOA introduces adaptive Lévy-flight perturbation, elite learning, and probabilistic memory updating to improve global exploration and convergence stability.

3.1 Lévy-Flight-Based Adaptive Perturbation Strategy

3.1.1 Lévy Flight Strategy

Lévy flight is a heavy-tailed random walk with occasional long-range jumps, which helps enhance global exploration and avoid local optima. In IDOA, it replaces the random perturbation mechanism in both the exploration and exploitation phases of DOA. A smaller β is used in the exploration phase to promote long-range jumps, whereas a larger β is used in the exploitation phase to support fine-grained local search.

The step length of Lévy flight is generated by the Mantegna algorithm:

$$\text{step} = \frac{u \cdot \sigma}{|v|^{1/\beta_{\text{levy}}}} \tag{13}$$

where σ is calculated as

$$\sigma = \left[\frac{\Gamma(1 + \beta_{\text{levy}}) \sin(\pi \beta_{\text{levy}}/2)}{\Gamma((1 + \beta_{\text{levy}})/2) \beta_{\text{levy}} \cdot 2^{(\beta_{\text{levy}}-1)/2}} \right]^{1/\beta_{\text{levy}}} \tag{14}$$

3.1.2 Adaptive Mechanism

The adaptive step-size scaling factor is defined as

$$\text{current\_levy\_scale} = \text{levy\_scale\_factor\_initial} - (\text{levy\_scale\_factor\_initial} - \text{levy\_scale\_factor\_final}) \times \frac{i_{\text{iter}}}{T} \tag{15}$$

and the adaptive Lévy distribution parameter is given by

$$\beta_{\text{levy}} = \beta_{\text{levy\_initial}} + (\beta_{\text{levy\_final}} - \beta_{\text{levy\_initial}}) \times \frac{i_{\text{iter}}}{T} \tag{16}$$

These adaptive parameters are updated according to the iteration progress and do not require additional problem-specific tuning.

3.2 Elite Learning Strategy

The elite learning strategy guides individuals toward the current best solution to improve convergence, while a random perturbation term is introduced to maintain population diversity. At iteration t, the elite individual and the current individual are denoted as

$$X_{\text{best}}^{t} = \left(x_{\text{best},1}^{t}, x_{\text{best},2}^{t}, \ldots, x_{\text{best},d}^{t}\right). \tag{17}$$

$$X_{i}^{t} = \left(x_{i,1}^{t}, x_{i,2}^{t}, \ldots, x_{i,d}^{t}\right). \tag{18}$$

The elite learning update is formulated as

$$X_{i}^{t+1} = X_{i}^{t} + \alpha \cdot \left(X_{\text{best}}^{t} - X_{i}^{t}\right) + \beta \cdot \text{rand}, \tag{19}$$

where α controls the movement toward the elite solution, β is the perturbation factor, and rand denotes a random number in $[0, 1]$.

3.3 Improved Memory Strategy

In the original DOA, individuals are deterministically reset to the group-best solution in the exploration phase or the global-best solution in the exploitation phase before perturbation, which may cause premature aggregation. IDOA introduces a probabilistic memory mechanism, using a high reset probability in the early stage to accelerate convergence and a lower probability later to preserve population diversity.

The reset probability is linearly decreased as

$$p_{\text{memory}}^{t} = p_{\text{memory}}^{\text{initial}} - (p_{\text{memory}}^{\text{initial}} - p_{\text{memory}}^{\text{final}}) \times \frac{t}{T} \tag{20}$$

During exploration, an individual is reset to the group-best solution $\mathbf{s}_g$; during exploitation, it is reset to the global-best solution $\mathbf{s}^*$. If memory reset is not triggered, the current position is retained for subsequent perturbation.

3.4 Algorithm Pseudocode

Algorithm 1 summarizes the main procedure of IDOA, including adaptive parameter updating, improved memory updating, Lévy-flight perturbation, elite learning, boundary handling, and best-solution updating.

Algorithm 1: Pseudo-code of IDOA

Require: Population size N, maximum iterations T, bounds **lb**, **ub**, dimension D, objective function f
Ensure: Best solution $\mathbf{s}^*$ and best fitness f^*
Initialize population $\mathbf{X}_i$
Evaluate all individuals and initialize global best $\mathbf{s}^*$, f^*
Divide the population into five groups
for $t = 1 : T$ **do**
  Update adaptive parameters
  if $t \leq 0.9T$ **then**
    Set exploration phase
    for each group g **do**
      for each individual $\mathbf{X}_i$ in group g **do**
        Select memory base from $\mathbf{s}_g$
        Apply Lévy-flight perturbation
        Apply elite learning using Eq. (19)
        Apply boundary handling
      end for
    end for
  else
    Set exploitation phase
    for each individual $\mathbf{X}_i$ **do**
      Select memory base from $\mathbf{s}^*$
      Apply Lévy-flight perturbation
      Apply elite learning using Eq. (19)
      Apply boundary handling
    end for
  end if
  Evaluate updated individuals and update group bests and global best
end for
return $\mathbf{s}^*$, f^*

3.5 Time Complexity Analysis of the Algorithm

Population initialization requires $\mathcal{O}(N \times D)$, while fitness evaluation requires $\mathcal{O}(T_{\max} \times N \times E)$. Both the exploration and exploitation phases require at most $\mathcal{O}(N \times D)$ position-update operations per iteration. Therefore, the time complexity of IDOA is

$$\mathrm{TC}_{\mathrm{IDOA}} = \mathcal{O}(N \times D) + \mathcal{O}(0.9T_{\max} \times N \times D) + \mathcal{O}(0.1T_{\max} \times N \times D) + \mathcal{O}(T_{\max} \times N \times E) \tag{21}$$

$$= \mathcal{O}(T_{\max} \times N \times (D + E)). \tag{22}$$

The Mantegna-based Lévy-flight calculation and adaptive-parameter updating introduce only scalar operations, while Lévy perturbation, elite learning, and probabilistic memory updating require at most linear

position updates over the population and feature dimensions. These strategies therefore do not change the dominant asymptotic order. Under the same wrapper-based evaluation setting, IDOA has the same dominant complexity order as DOA, PSO, GWO, and GA.

4 IDOA-Based Traffic Anomaly Detection Model

To improve the effectiveness and efficiency of IoT traffic anomaly detection, an IDOA-based traffic anomaly detection model is proposed. The proposed model employs IDOA for traffic feature selection to obtain a compact and discriminative feature subset, followed by KNN-based traffic anomaly detection. The model consists of four stages: data acquisition, data preprocessing, feature selection, and traffic classification. The overall architecture is shown in Fig. 1.

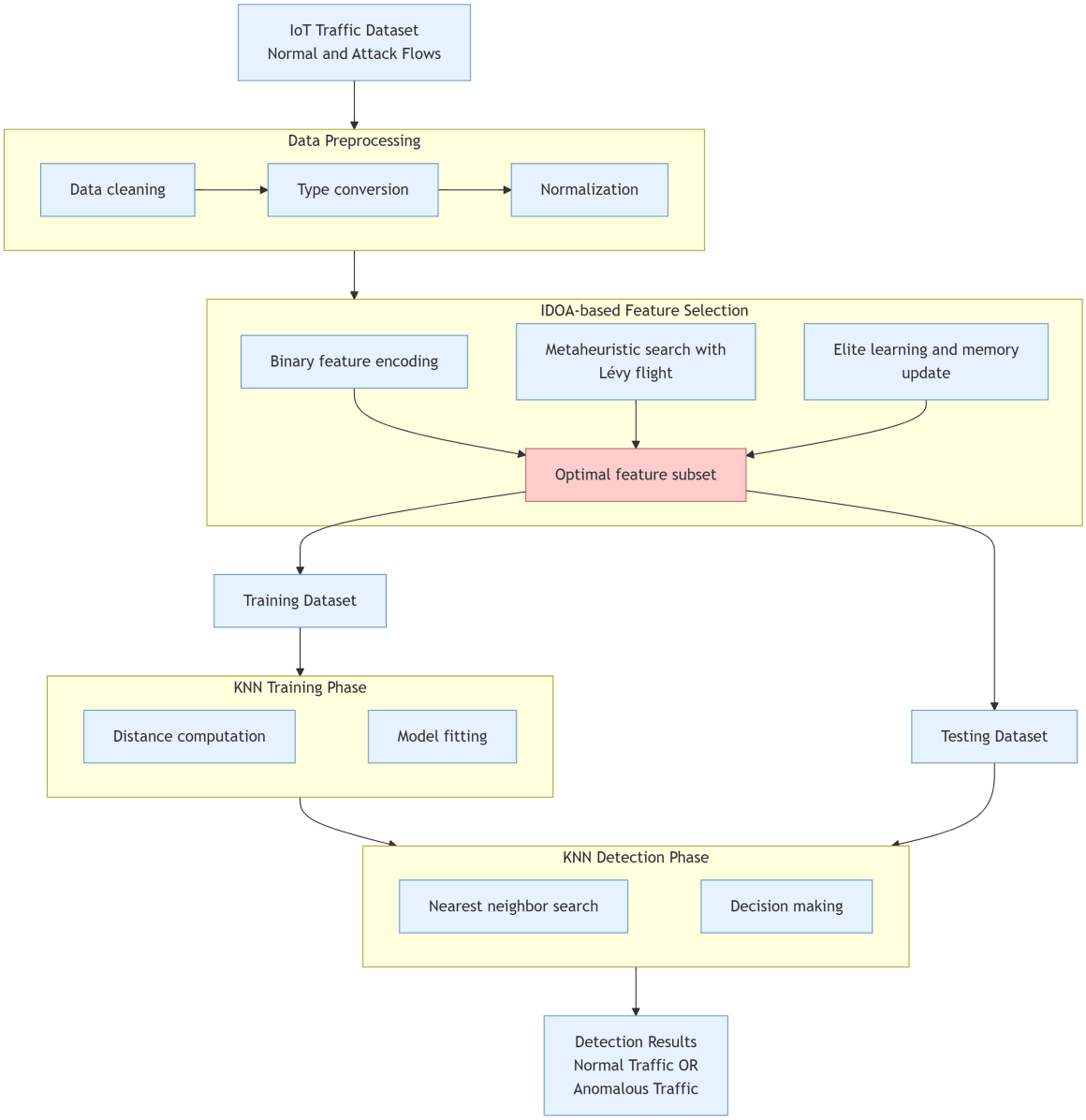


Figure 1: Overall workflow of the proposed IDOA-based IoT traffic anomaly detection model. IDOA and KNN denote the improved dream optimization algorithm and k-nearest neighbor classifier, respectively.

(1) Data Acquisition Module

The data acquisition module is responsible for collecting raw IoT network traffic data. In this study, the NF-ToN-IoT-v2 dataset is used as the input, containing both normal and malicious traffic samples, thereby providing the data source for subsequent traffic anomaly detection.

(2) Data Preprocessing Module

The raw traffic data are first preprocessed to improve data quality and remove irrelevant information. The preprocessing procedure mainly includes the following steps:

- Data Cleaning: removing missing values, outliers, and redundant records;
- Type Conversion: converting categorical attributes into numerical representations;
- Normalization: scaling all features to a unified range before model training.

After preprocessing, the traffic data are transformed into normalized feature vectors for subsequent feature selection and classification.

(3) Feature Selection Module

After preprocessing, the improved Dream Optimization Algorithm (IDOA) is employed for IoT traffic feature selection, as shown in Fig. 2. Each individual is represented as a continuous position vector. Since feature selection is a binary optimization problem, each continuous position vector is converted into a binary feature mask before fitness evaluation according to the following thresholding rule:

$$b_{i,j} = \begin{cases} 1, & x_{i,j} > 0.5, \\ 0, & x_{i,j} \le 0.5, \end{cases} \tag{23}$$

where $b_{i,j} = 1$ indicates that the j-th feature is selected, while $b_{i,j} = 0$ indicates that the feature is discarded. Unlike S-shaped or V-shaped transfer functions commonly used in binary metaheuristic algorithms, this study adopts a deterministic thresholding rule with a threshold of 0.5. In this way, long-range jumps generated by Lévy flight in the continuous space can be mapped to different binary feature subsets after thresholding, thereby enhancing the exploration capability of IDOA.

The feature selection problem is formulated as a minimization problem by jointly considering the classification error and the number of selected features. The fitness function is defined as

$$Fitness = \alpha \cdot Err + \beta \cdot \frac{|S|}{D} \tag{24}$$

where Err denotes the average classification error obtained by a 5-fold cross-validation using a KNN classifier with $k = 5$, Euclidean distance, and feature standardization enabled, $|S|$ is the number of selected features, and D is the total number of available features. In this study, $\alpha = 0.99$ and $\beta = 0.01$, which give priority to classification performance while introducing a mild penalty on the number of selected features.

Based on this fitness function, IDOA searches the feature space to obtain a compact and discriminative feature subset, thereby reducing feature redundancy and computational complexity while improving subsequent classification performance.

(4) Result Evaluation Module

The dataset is first partitioned using a stratified 5-fold cross-validation protocol. Within each fold, preprocessing steps requiring data-dependent parameter estimation, including normalization, are fitted exclusively on the training subset and then applied to the corresponding test subset. For each fold, feature selection is performed exclusively on the training subset using IDOA, while the corresponding test subset remains unseen during the feature selection process. The normalization parameters are estimated only

from the training subset and then applied to the corresponding test subset. This protocol ensures that neither feature selection nor normalization uses any information from the test data, thereby preventing information leakage.

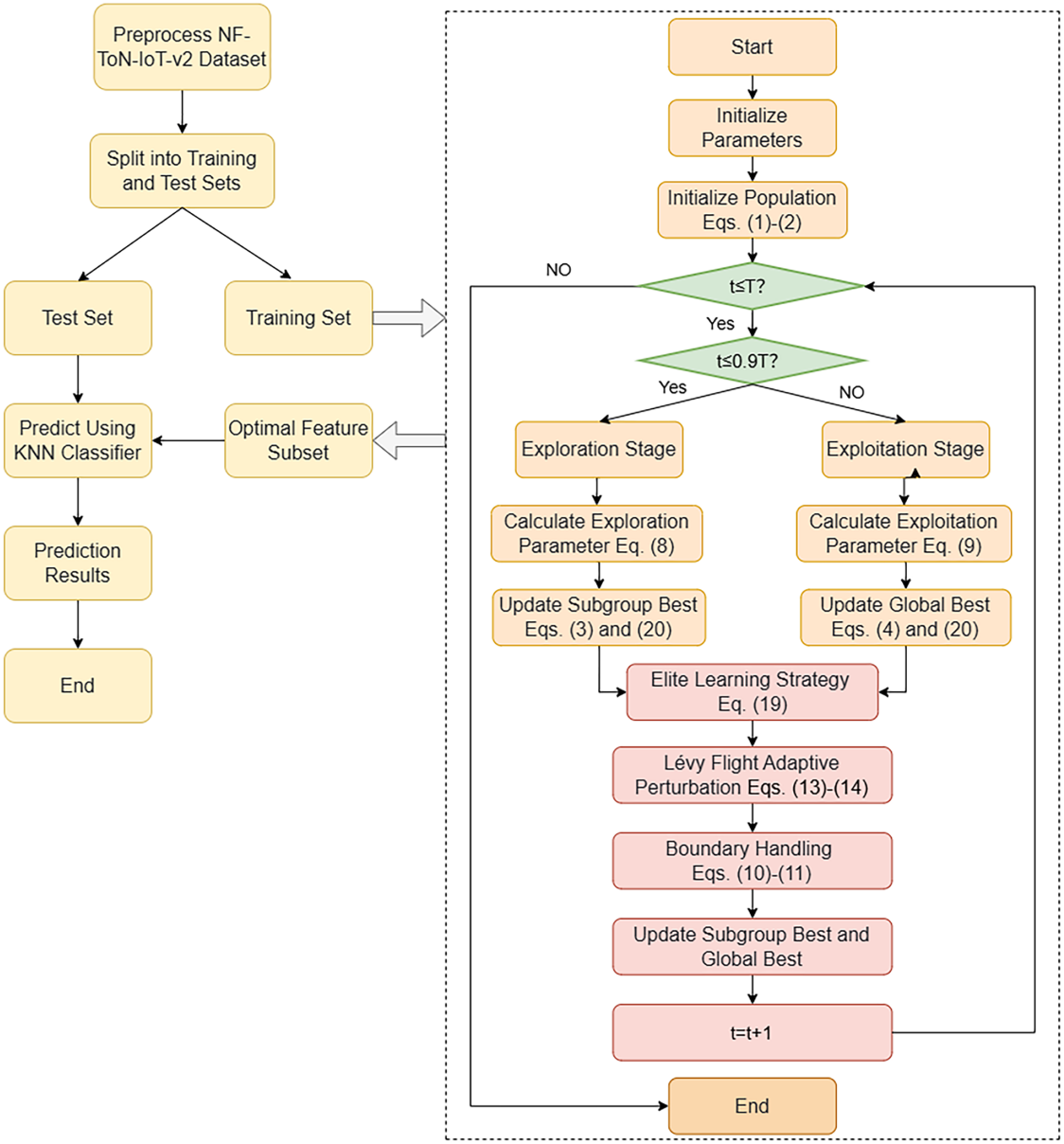


Figure 2: Feature selection flowchart of IDOA. IDOA and KNN denote the improved dream optimization algorithm and k-nearest neighbor classifier, respectively. The workflow includes data splitting, IDOA-based feature selection, binary feature mask generation, and KNN-based prediction.

After obtaining the optimal feature subset, the KNN classifier is trained using the selected features from the training subset and evaluated on the corresponding test subset. This module consists of two stages:

- KNN Training Stage: model training based on the selected feature subset;
- KNN Detection Stage: classification of the test samples using the trained KNN classifier.

Finally, the network traffic is classified as normal or anomalous. The detection performance is evaluated using Accuracy, Recall, Precision, and F1-score.

5 Experimental Results and Analysis

This chapter evaluates the proposed IDOA algorithm in three aspects. First, benchmark functions are used to assess its convergence performance. Second, UCI datasets are adopted to verify its classification performance and generalization ability. Third, the NF-ToN-IoT-v2 dataset is used to validate its effectiveness in IoT traffic anomaly detection. KNN is adopted as the classifier in the classification experiments because of its simplicity and sensitivity to feature quality, making it suitable for evaluating the effectiveness of the proposed feature selection algorithm. All experiments were conducted on MATLAB R2024b using an Intel(R) Core(TM) i7-14650HX processor with a main frequency of 2.20 GHz.

5.1 Parameter Settings

The parameter settings of all optimization algorithms are summarized in Table 1. For a fair comparison, all algorithms used the same population size, iteration budget, and stopping criterion.

Table 1: Main parameter settings of the optimization algorithms.

Algorithm	Population Size	FS Iterations	BF Iterations	Stopping Criterion	Algorithm-Specific Parameters
IDOA	30	50	500	Maximum iterations; no early stopping	Lévy scale factor: 0.2 → 0.005; elite learning factor: 0.6 → 0.1; memory probability: 0.9 → 0.2.
DOA	30	50	500	Maximum iterations; no early stopping	$T_d = 0.9T_{\max}$; strategy probability $u = 0.9$; number of groups = 5; k_q and k_r are determined according to Eqs. (8) and (9), respectively.
PSO	30	50	500	Maximum iterations; no early stopping	Inertia weight $w = 0.7298$; $c_1 = c_2 = 1.49618$; $V_{\max} = 0.2(ub - lb)$.
GWO	30	50	500	Maximum iterations; no early stopping	Control parameter a decreases linearly from 2 to 0.
GA	30	50	500	Maximum iterations; no early stopping	Crossover probability $p_c = 0.80$; mutation probability $p_m = 1/D$; $\eta_c = \eta_m = 20$; tournament size = 3; elite ratio = 5%.

Note: FS denotes feature-selection experiments, and BF denotes benchmark-function experiments.

These parameters were kept fixed across all datasets and were not tuned separately for individual datasets. For reproducibility, the UCI experiments were repeated using random seeds 101–130, and the NF-ToN-IoT-v2 experiments were repeated using random seeds 101–106.

5.2 Benchmark Function Experiments

To evaluate the optimization performance and exploration capability of IDOA, six classical benchmark functions, including two unimodal (F1–F2) and four multimodal (F3–F6) functions, were employed. Table 2

summarizes their mathematical expressions, search ranges, and theoretical optimal values. The experimental results of IDOA and the comparison algorithms (DOA, PSO, GWO, and GA) are reported in Table 3, and the corresponding convergence curves are shown in Fig. 3.

Table 2: Benchmark function expressions.

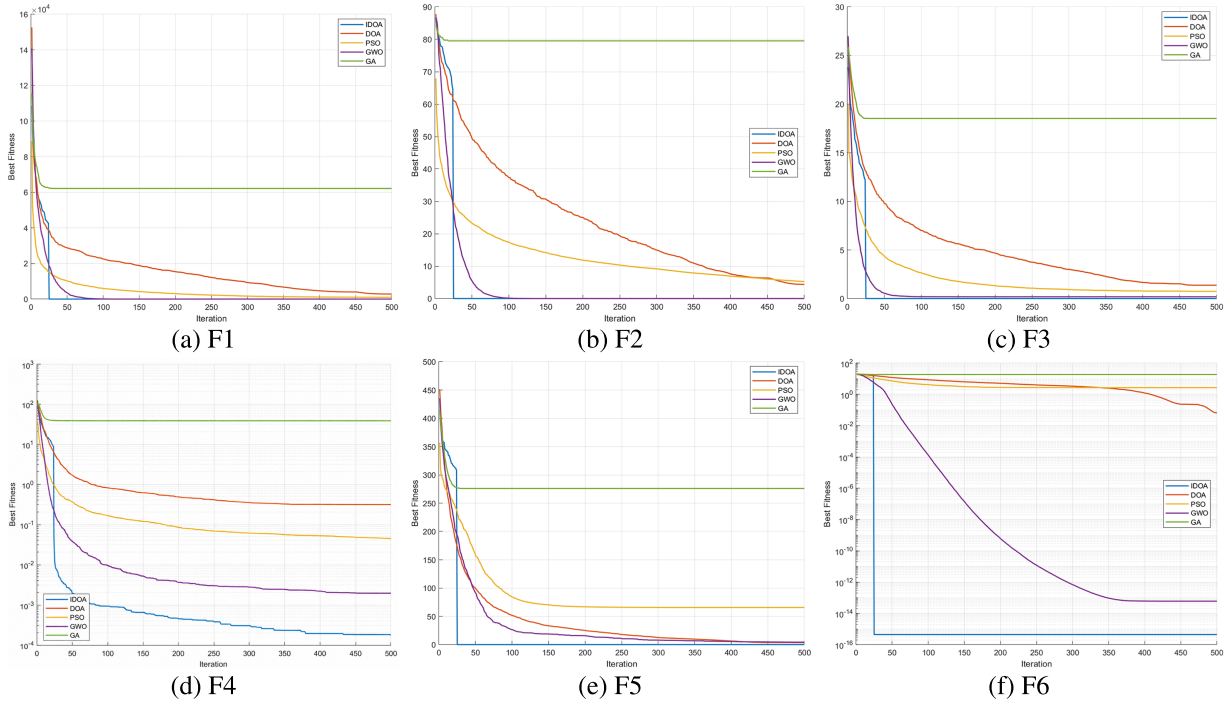
Function Expressions	Dimension	Range	Min
$F_1(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]$	0
$F_2(\mathbf{x}) = \max_{1 \le i \le D} \lvert x_i \rvert$	30	$[-100, 100]$	0
$F_3(\mathbf{x}) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$	30	$[-100, 100]$	0
$F_4(\mathbf{x}) = \sum_{i=1}^{D} i x_i^4 + \varepsilon,\ \varepsilon \sim U(0,1)$	30	$[-1.28, 1.28]$	0
$F_5(\mathbf{x}) = 10D + \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i)\right)$	30	$[-5.12, 5.12]$	0
$F_6(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0

Table 3: Test function experiment results.

Function	Type	PSO	DOA	GWO	IDOA	GA
F1	Mean	2.41E+02	3.05E+03	1.00E−05	0.00E+00	3.84E+04
	Std	1.75E+02	1.04E+03	2.77E−05	0.00E+00	1.50E+04
F2	Mean	6.11E+00	4.58E+00	1.12E−06	0.00E+00	7.01E+01
	Std	2.31E+00	9.10E−01	1.63E−06	0.00E+00	4.62E+00
F3	Mean	1.00E+00	1.40E+00	1.88E−01	0.00E+00	1.45E+01
	Std	4.59E−01	2.54E−01	4.53E−02	0.00E+00	2.29E+00
F4	Mean	4.03E−02	3.46E−01	2.07E−03	1.73E−04	1.41E+01
	Std	2.07E−02	1.77E−01	1.15E−03	1.45E−04	1.71E+01
F5	Mean	6.92E+01	3.51E+00	2.55E+00	0.00E+00	2.07E+02
	Std	1.94E+01	1.81E+00	4.28E+00	0.00E+00	3.49E+01
F6	Mean	2.24E+00	7.27E−02	5.96E−14	4.44E−16	1.80E+01
	Std	8.58E−01	1.98E−02	1.45E−14	0.00E+00	5.45E−01

For all algorithms, the maximum number of iterations was set to $T = 500$. Each algorithm was independently executed 20 times to reduce the influence of randomness, and the mean (Mean) and standard deviation (Std) of the results are reported in Table 3.

As shown in Table 3 and Fig. 3, IDOA achieves competitive optimization performance on the selected benchmark functions. It reaches the theoretical optimum on F1, F2, F3, and F5, while obtaining the best or near-best results on F4 and F6. The convergence curves indicate that IDOA generally converges faster and reaches lower fitness values than the comparison algorithms.



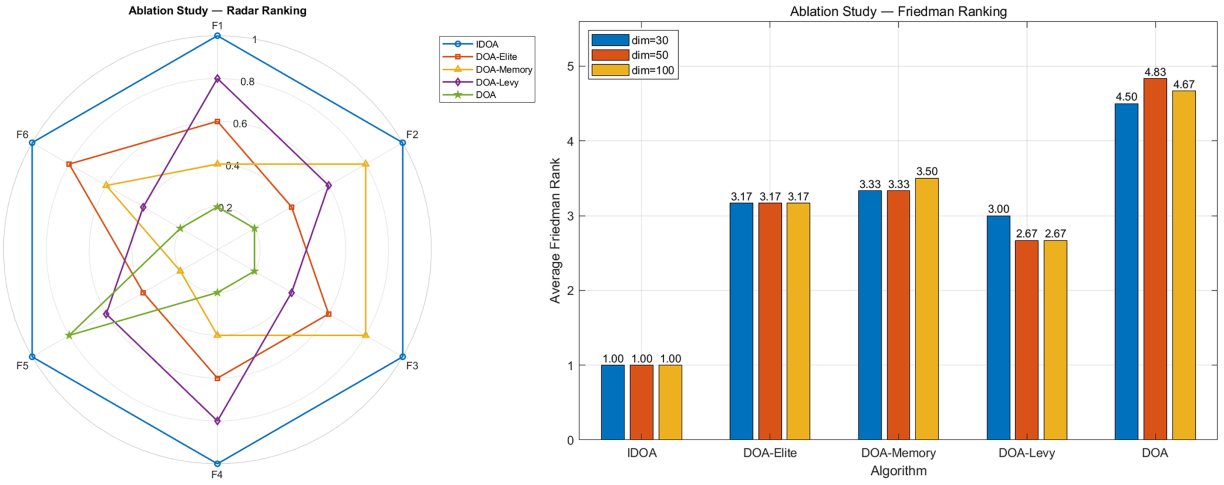
(a) F1 (b) F2 (c) F3

(d) F4 (e) F5 (f) F6

Figure 3: Convergence curves of fitness.

Ablation Study on Improvement Strategies

Ablation experiments on the six benchmark functions compared IDOA with four variants: DOA, DOA-Levy, DOA-Elite, and DOA-Memory. As shown in Fig. 4, IDOA achieves the best overall performance, while DOA-Levy performs best among the single-strategy variants. These results suggest that combining the three improvement strategies provides the greatest performance gain.



(a) Radar ranking chart (b) Friedman average rank bar chart

Figure 4: Ablation study results on six benchmark functions.

5.3 UCI Datasets

To evaluate the generic feature-selection capability of IDOA, four general-purpose UCI datasets, namely Ionosphere, WDBC, Heart Statlog, and Sonar, were selected as classification benchmarks. These datasets are not regarded as IoT traffic datasets; domain-specific validation is conducted separately on the NF-ToN-IoT-v2 dataset. To reduce the influence of randomness, 30 independent runs were conducted, and the average values of the evaluation metrics were reported. All four datasets were obtained from the UCI Machine Learning Repository. Table 4 summarizes their basic information and class labels, while Table 5 presents the comparative classification results.

Table 4: Selected UCI datasets.

No.	Dataset Name	Source	Sample Size	Number of Features	Class Labels
1	Ionosphere	UCI	351	34	Good/Bad
2	Heart Statlog	UCI	270	13	Absence/Presence
3	WDBC	UCI	569	30	Benign/Malignant
4	Sonar	UCI	208	60	Rock/Mine

Table 5: Experimental results for the UCI datasets.

Algorithms	Metrics	Ionosphere	WDBC	Heart Statlog	Sonar
PSO	Accuracy (%)	94.29	97.26	89.25	93.55
	Recall (%)	96.77	96.93	90.87	91.68
	F1-score (%)	94.99	94.90	89.02	92.88
	Precision (%)	93.26	92.93	87.23	94.12
GWO	Accuracy (%)	91.43	97.43	91.36	87.10
	Recall (%)	98.51	97.18	**92.33**	89.66
	F1-score (%)	**96.92**	95.85	92.75	92.37
	Precision (%)	**95.38**	94.57	93.18	95.24
GA	Accuracy (%)	93.84	97.01	88.96	92.44
	Recall (%)	95.62	96.40	90.11	90.78
	F1-score (%)	94.13	94.94	88.89	91.90
	Precision (%)	92.73	93.52	87.69	93.05
DOA	Accuracy (%)	92.38	96.33	87.65	91.94
	Recall (%)	95.42	95.41	89.56	92.13
	F1-score (%)	93.95	94.30	87.77	92.65
	Precision (%)	92.54	93.18	86.05	93.17
IDOA	Accuracy (%)	**96.19**	**98.24**	**92.42**	**95.32**
	Recall (%)	**98.72**	**97.72**	91.78	**93.10**
	F1-score (%)	96.50	**97.02**	**92.82**	**94.27**
	Precision (%)	94.37	**96.34**	**93.89**	**95.46**

Note: Bold values indicate the optimal results. Tied optimal values are not bolded.

As shown in Table 5, IDOA achieves the highest classification accuracy on all four UCI datasets, demonstrating its stable performance across datasets with different sample sizes and feature dimensions.

In addition, IDOA obtains the best Recall, Precision, or F1-score on most datasets. Although GWO achieves the highest Precision on the Ionosphere dataset and the highest Recall on the Heart Statlog dataset, IDOA maintains competitive performance across all evaluation metrics. These results suggest that the proposed feature selection strategy exhibits good generalization capability on different classification tasks.

5.4 Experiments on the NF-ToN-IoT-v2 Dataset

To further evaluate the effectiveness of the proposed method, experiments were conducted on the NF-ToN-IoT-v2 dataset, a publicly available network flow-based dataset designed for IoT traffic anomaly detection (https://espace.library.uq.edu.au/view/UQ:38a2d07). The dataset contains 43 extracted NetFlow features and multiple categories of normal and malicious traffic, and has been widely used for binary and multiclass traffic anomaly detection. After removing non-feature fields during preprocessing, 41 features were retained for subsequent experiments. For binary classification, all attack categories were merged into a single attack class, where normal and attack traffic were labeled as class 0 and class 1, respectively. For multiclass classification, the original attack labels were retained to distinguish different attack types.

Before cross-validation, 10,000 flow records were selected from the NF-ToN-IoT-v2 dataset using stratified random sampling without replacement while preserving the original class distribution. The sampled dataset was then partitioned using repeated stratified 5-fold cross-validation. Therefore, stratified sampling was completed before cross-validation. A duplicate-record check was also performed, confirming that no identical flow records appeared simultaneously in the training and test folds. For each fold, feature selection and normalization were performed exclusively on the training subset, while the corresponding test subset remained unseen until final evaluation.

5.4.1 Binary Classification Results

This subsection addresses binary classification, where normal and attack traffic are labeled as class 0 and class 1, respectively. The NF-ToN-IoT-v2 dataset contains 16,940,496 samples with 41 processed features, including 6,099,469 normal samples (36.01%) and 10,841,027 attack samples (63.99%).

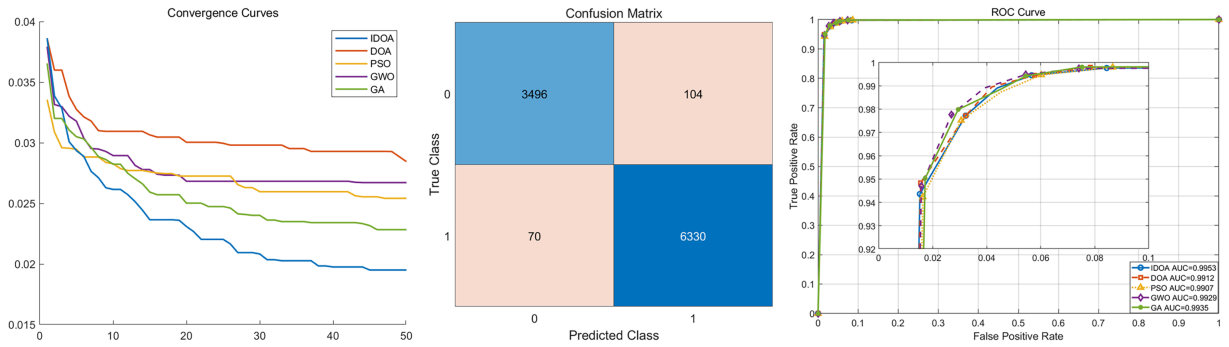
Table 6 compares all algorithms in terms of classification performance, feature size, and computational cost. Fig. 5 presents the convergence curves of all compared algorithms, the confusion matrix of IDOA, and the ROC curves of all compared algorithms.

Table 6: Binary classification results on the NF-ToN-IoT-v2 dataset.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Features	Opt Time (s)	Train Time (s)	Pred Time (s)
IDOA	**98.01**	**97.953**	**98.976**	**98.462**	**7.4**	145.09	0.054494	0.12948
DOA	96.23	95.424	98.859	97.111	23.0	216.27	0.26416	0.24220
PSO	96.41	95.781	98.734	97.235	18.8	169.48	0.16347	0.20325
GWO	96.75	96.132	98.906	97.499	16.8	209.70	0.11461	0.23994
GA	97.39	97.034	98.953	97.984	11.0	161.75	0.081597	0.15105

Note: Bold values indicate the optimal results. Tied optimal values are not bolded.

As shown in Table 6, IDOA achieves the best overall performance, obtaining the highest Accuracy, Precision, Recall, and F1-score while selecting the smallest feature subset, with 7.4 features on average. These results demonstrate a favorable balance between classification performance, feature compactness, and computational efficiency.



(a) Convergence curves (b) Confusion matrix (c) ROC curve

Figure 5: Binary classification results.

5.4.2 Multiclass Classification Results

Multiclass classification was conducted on the stratified sample of 10,000 NF-ToN-IoT-v2 flow records described in Table 7.

Table 7: Class distribution of the sampled NF-ToN-IoT-v2 dataset.

	Benign	Backdoor	DDoS	DoS	Injection	MITM	Password	Ransomware	Scanning	XSS	Total
Original count	6,099,469	16,809	2,026,234	712,609	684,465	7723	1,153,323	3425	3,781,419	2,455,020	16,940,496
Sampled count	3600	10	1196	421	404	5	681	2	2232	1449	10,000

Table 8 reports the classification results of the five feature selection algorithms, with the best values highlighted in bold. The corresponding Wilcoxon signed-rank test results between IDOA and the other algorithms are presented in Table 9.

Table 8: Multiclass classification results on the NF-ToN-IoT-v2 dataset.

Metric	Algorithm	Benign	Backdoor	DDoS	DoS	Injection	MITM	Password	Ransomware	Scanning	XSS
Recall (%)	IDOA	**96.583**	100.000	**96.321**	**96.913**	**96.664**	0.000	**95.363**	20.000	97.311	**95.477**
	DOA	96.500	100.000	95.484	94.294	93.457	0.000	94.918	20.000	97.356	93.614
	PSO	96.278	100.000	96.154	94.532	93.204	0.000	94.626	20.000	97.356	95.131
	GWO	96.417	100.000	96.153	94.056	93.454	0.000	94.481	20.000	97.222	94.994
	GA	96.361	100.000	95.903	96.437	94.682	0.000	94.774	20.000	97.311	95.202
Precision (%)	IDOA	**98.643**	100.000	**97.290**	**95.772**	85.516	0.000	88.659	20.000	97.497	**98.169**
	DOA	98.503	100.000	96.452	95.139	83.041	0.000	87.354	20.000	97.577	96.286
	PSO	98.528	100.000	96.818	94.063	**85.717**	0.000	87.397	20.000	97.359	96.564
	GWO	98.561	100.000	96.396	94.657	84.388	0.000	88.051	20.000	**97.620**	96.389
	GA	98.444	100.000	96.878	95.528	83.991	0.000	**89.369**	20.000	97.011	97.912
F1-score (%)	IDOA	**97.600**	100.000	**96.786**	**96.335**	**90.752**	0.000	91.866	20.000	97.401	**96.793**
	DOA	97.488	100.000	95.955	94.681	87.931	0.000	90.938	20.000	**97.464**	94.916
	PSO	97.386	100.000	96.492	94.290	89.264	0.000	90.839	20.000	97.357	95.840
	GWO	97.473	100.000	96.266	94.301	88.626	0.000	91.133	20.000	97.418	95.677
	GA	97.389	100.000	96.375	95.975	88.906	0.000	**91.953**	20.000	97.159	96.531

(Continued)

Table 8 (continued)

Metric	Algorithm	Benign	Backdoor	DDoS	DoS	Injection	MITM	Password	Ransomware	Scanning	XSS
Accuracy (%)	IDOA	**96.382**	Selected Features		**9.20**	Train time (s)	**0.011862**	Pred time (s)	**0.032688**		
	DOA	95.769			19.60		0.022856		0.037937		
	PSO	95.961			18.20		0.020519		0.036209		
	GWO	95.940			15.80		0.013645		0.034168		
	GA	96.156			15.20		0.012229		0.034442		

Note: Bold values indicate the optimal results. Tied optimal values are not bolded.

Table 9: Wilcoxon signed-rank test results for pairwise comparisons.

	IDOA vs. DOA	IDOA vs. PSO	IDOA vs. GWO	IDOA vs. GA
p-value	1.74E−06	1.93E−06	2.60E−06	3.72E−05
h	1	1	1	1

As shown in Table 8, IDOA achieves the highest overall classification accuracy while selecting the fewest features among the compared algorithms. Although some algorithms perform best on specific categories, IDOA maintains competitive performance across most traffic categories with a much smaller feature subset. Since several minority attack classes, such as MITM and ransomware, contain very few sampled instances, their per-class Recall and F1-score values may fluctuate and should be interpreted with caution. Therefore, the overall comparison is mainly based on weighted metrics and repeated cross-validation results.

Table 9 reports the Wilcoxon signed-rank test results between IDOA and the other four feature selection algorithms. Here, $h = 1$ indicates that the null hypothesis is rejected at the 0.05 significance level, and the test was conducted using paired accuracy results from repeated cross-validation runs.

5.5 Additional Baseline Comparison

5.5.1 Binary Classification

To further evaluate the proposed method, three representative baseline methods, namely Full-KNN, MI-KNN, and RF-All, were introduced. Full-KNN uses all original features without feature selection. MI-KNN employs mutual information (MI) as a filter-based feature selection method, where the top 10 ranked features are selected for classification. RF-All uses the complete feature set with a Random Forest classifier as a non-KNN baseline.

All methods were evaluated using the same sampled dataset, preprocessing procedure, repeated stratified 5-fold cross-validation protocol, and evaluation metrics described in Section 5.4. Table 10 summarizes the classification results and accumulated running times. Since NF-ToN-IoT-v2 provides pre-extracted NetFlow features, the reported feature-preparation latency refers to the time required to construct the selected feature subset from the available feature matrix, rather than the time required to extract NetFlow attributes from raw packets. Classification latency is calculated as $T_{\text{pred}}/N_{\text{test}} \times 10^6$, where T_{pred} denotes the accumulated prediction time in seconds and $N_{\text{test}} = 10000$ denotes the number of evaluated flows. Table 11 further reports feature-subset preparation latency, classification latency, and their sum as the total online processing latency.

Table 10: Binary classification baseline comparison on the NF-ToN-IoT-v2 dataset.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Features	Opt. Time (s)	Train Time (s)	Pred. Time (s)
Full-KNN	95.760	95.880	97.578	96.718	41.0	0	0.370	0.401
MI-KNN	95.280	95.564	97.140	96.342	10.0	0.230	0.024	0.118
RF-All	**98.150**	**98.503**	98.609	**98.555**	41.0	0	8.365	0.519
IDOA-KNN	98.010	97.953	**98.976**	98.462	**7.4**	145.090	0.054	0.129

Note: Bold values indicate the optimal results. Tied optimal values are not bolded.

Table 11: Online processing efficiency comparison between full-feature and selected-feature models for binary classification.

Method	NumFeatures	Feature-Subset Preparation Latency (μs/flow)	Classification Time (s)	Classification Latency (μs/flow)	Total Online Latency (μs/flow)
Full-KNN	41.0	0.00	0.4010	40.10	40.10
IDOA-KNN	7.4	0.73	0.1290	12.90	13.63

As shown in Table 10, IDOA-KNN achieves higher overall classification performance than Full-KNN while reducing the average number of features from 41 to 7.4. Although RF-All obtains slightly higher Accuracy, Precision, and F1-score, it uses all 41 features and requires longer training and prediction times. Table 11 shows that the classification latency of IDOA-KNN is reduced from 40.10 to 12.90 μs/flow. Constructing the selected feature subset from the pre-extracted feature matrix requires an additional 0.73 μs/flow, resulting in a total online processing latency of 13.63 μs/flow. This value is approximately 66.0% lower than that of Full-KNN under the evaluated offline setting. These results suggest that IDOA-KNN provides a favorable trade-off among classification performance, feature compactness, and flow-level processing efficiency.

5.5.2 Multiclass Classification

All methods were evaluated under the same preprocessing procedure, sampling strategy, repeated stratified 5-fold cross-validation protocol, and evaluation metrics described in the previous subsection. Table 12 summarizes the multiclass classification results and accumulated running times. Classification latency is calculated using the same formula. Table 13 further reports Feature-Subset Preparation Latency, classification latency, and their sum as total online latency.

Table 12: Multiclass classification baseline comparison on the NF-ToN-IoT-v2 dataset.

Method	Accuracy (%)	Weighted Precision (%)	Weighted Recall (%)	Weighted F1-Score (%)	Features	Train Time (s)	Pred. Time (s)
Full-KNN	93.161	93.630	93.161	93.300	41.0	0.096	0.289
MI-KNN	83.793	87.717	83.793	84.697	10.0	0.116	0.060
RF-All	95.920	**95.870**	95.920	**95.840**	41.0	7.580	0.654
IDOA-KNN	**96.382**	95.384	**96.382**	95.158	**9.2**	0.058	0.1634

Note: Bold values indicate the optimal results. Tied optimal values are not bolded.

Table 13: Online processing efficiency comparison between full-feature and selected-feature models for multiclass classification.

Method	NumFeatures	Feature-Subset Preparation Latency (μs/flow)	Classification Time (s)	Classification Latency (μs/flow)	Total Online Latency (μs/flow)
Full-KNN	41.0	0.00	0.2892	28.92	28.92
IDOA-KNN	9.2	0.93	0.1634	16.34	17.27

As shown in Table 12, IDOA-KNN achieves the highest Accuracy and Weighted Recall among the compared methods while using 9.2 features on average. Compared with Full-KNN, its Accuracy increases from 93.161% to 96.382%, while the classification latency decreases from 28.92 to 16.34 μs/flow. As shown in Table 13, constructing the selected feature subset from the pre-extracted feature matrix requires 0.93 μs/flow. After combining this value with the classification latency of 16.34 μs/flow, the total online processing latency of IDOA-KNN is 17.27 μs/flow, which is approximately 40.3% lower than that of Full-KNN under the evaluated experimental setting.

6 Conclusions and Future Work

To address the problems of severe redundancy in high-dimensional features, high computational complexity, and limited detection efficiency in IoT traffic anomaly detection, this paper proposed an Improved Dream Optimization Algorithm (IDOA) and constructed an IoT traffic anomaly detection model based on IDOA. By incorporating a Lévy-flight-based adaptive perturbation mechanism, an elite learning strategy, and a probabilistic memory update mechanism, the proposed algorithm enhances global exploration capability, convergence accuracy, and stability without increasing the asymptotic order of time complexity.

The benchmark-function experiments verify the superior convergence performance of IDOA on both unimodal and multimodal optimization problems. In addition, the classification results on the UCI datasets and the real-world NF-ToN-IoT-v2 dataset demonstrate that the proposed IDOA-based feature selection method can effectively reduce feature redundancy while maintaining competitive detection performance. The reduced feature subset also leads to lower prediction time in the offline evaluation, indicating its potential for lightweight IoT traffic anomaly detection.

However, wrapper-based feature selection still requires repeated fitness evaluations during the offline optimization stage. Therefore, the proposed method is more suitable for offline feature subset optimization followed by lightweight online inference. Future work will further investigate deployment-level latency on edge devices, streaming IoT traffic scenarios, and more diverse classifiers to improve the robustness and practical applicability of the proposed method.

Acknowledgement: Not applicable.

Funding Statement: This study received no specific funding.

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

Availability of Data and Materials: The UCI datasets used in this study are available from the UCI Machine Learning Repository. The NF-ToN-IoT-v2 dataset is publicly available at https://espace.library.uq.edu.au/view/UQ:38a2d07. The MATLAB implementation and experimental configurations will be made publicly available at https://github.com/magic-nine/IDOA-IoT-Anomaly-Detection upon acceptance.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ahmed M, Mahmood AN, Hu J. A survey of network anomaly detection techniques. J Netw Comput Appl. 2016;60(1):19–31. doi:10.1016/j.jnca.2015.11.016.
2. Tsogbaatar E, Bhuyan MH, Taenaka Y, Fall D, Gonchigsumlaa K, Elmroth E, et al. DeL-IoT: a deep ensemble learning approach to uncover anomalies in IoT. Internet Things. 2021;14(2):100391. doi:10.1016/j.iot.2021.100391.
3. Johnpeter T, Karuppanan S. Fuzzy-rule based optimized hybrid deep learning model for network intrusion detection in SDN enabled IoT network. Comput Secur. 2025;152(10):104372. doi:10.1016/j.cose.2025.104372.
4. Banoth R, Addula SR, Godishala AK, Sannapu R, Sajja GS, Kumar D, et al. Evolving IoT botnet threats and practical honeypot observation: a summary review and experimental study. J Cybersecur Priv. 2026;6(3):82. doi:10.3390/jcp6030082.
5. Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor. 2016;18(2):1153–76. doi:10.1109/COMST.2015.2494502.
6. Kwon D, Kim H, Kim J, Suh S, Kim I, Kim K. A survey of deep learning-based network anomaly detection. Clust Comput. 2019;22(1):949–61. doi:10.1007/s10586-017-1117-8.
7. Niyaz Q, Sun W, Javaid AY. A deep learning based DDoS detection system in software-defined networking (SDN). EAI Endorsed Trans Security Saf. 2017;4(12):153515. doi:10.4108/eai.28-12-2017.153515.
8. Tang TA, Mhamdi L, McLernon D, Zaidi SAR, Ghogho M, El Moussa F. Deep learning approach for network intrusion detection in software defined networking. In: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM); 2016 Oct 26–29; Fez, Morocco. Piscataway, NJ, USA: IEEE; 2016. p. 258–63. doi:10.1109/WINCOM.2016.7777224.
9. 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(5):183–94. doi:10.1016/j.comcom.2023.12.041.
10. Lu J, Wang J, Wei X, Wu K, Liu G. Deep anomaly detection based on variational deviation network. Future Internet. 2022;14(3):80. doi:10.3390/fi14030080.
11. Gao M, Wu L, Li Q, Chen W. Anomaly traffic detection in IoT security using graph neural networks. J Inf Secur Appl. 2023;76(5):103532. doi:10.1016/j.jisa.2023.103532.
12. Krupski J, Iwanowski M, Graniszewski W. Extraction of minimal set of traffic features using ensemble of classifiers and rank aggregation for network intrusion detection systems. Appl Sci. 2024;14(16):6995. doi:10.3390/app14166995.
13. Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;3:1157–82.
14. Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, et al. Feature selection: a data perspective. ACM Comput Surv. 2017;50(6):94. doi:10.1145/3136625.
15. Shreem SS, Abdullah S, Nazri MZA. Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm. Int J Syst Sci. 2016;47(6):1312–29. doi:10.1080/00207721.2014.924600.
16. Li X, Zhang S, Luo H, Ma X, He J. Placing timely refreshing services at the network edge. IEEE Internet Things J. 2023;10(18):16450–64. doi:10.1109/JIOT.2023.3268308.
17. Agrawal P, Abutarboush HF, Ganesh T, Mohamed AW. Metaheuristic algorithms on feature selection: a survey of one decade of research (2009–2019). IEEE Access. 2021;9:26766–91. doi:10.1109/ACCESS.2021.3056407.

18. Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of ICNN'95—International Conference on Neural Networks; 1995 Nov 27–Dec 1; Perth, Australia. Piscataway, NJ, USA: IEEE; 1995. p. 1942–8. doi:10.1109/ICNN.1995.488968.
19. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw. 2014;69:46–61. doi:10.1016/j.advengsoft.2013.12.007.
20. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw. 2016;95(12):51–67. doi:10.1016/j.advengsoft.2016.01.008.
21. 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; 2005 Mar 18–20; Kennesaw, GA, USA. New York, NY, USA: ACM; 2005. p. 136–41. doi:10.1145/1167253.1167288.
22. Nakashima M, Sim A, Kim Y, Kim J, Kim J. Automated feature selection for anomaly detection in network traffic data. ACM Trans Manag Inf Syst. 2021;12(3):18. doi:10.1145/3446636.
23. 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.
24. 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.
25. 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.
26. Addula SR, Meesala MK, Ravipati P, Sajja GS. A hybrid autoencoder and gated recurrent unit model optimized by honey badger algorithm for enhanced cyber threat detection in IoT networks. Secur Priv. 2025;8(6):e70086. doi:10.1002/spy2.70086.
27. Yang XS. Nature-inspired metaheuristic algorithms. 2nd ed. Frome, UK: Luniver Press; 2010.
28. 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.
29. Lang Y, Gao Y. Dream optimization algorithm (DOA): a novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. Comput Methods Appl Mech Eng. 2025;436(2):117718. doi:10.1016/j.cma.2024.117718.