symmetry

MDPI

Article

A Feature-Selection Method Based on Graph Symmetry Structure in Complex Networks

Wangchuanzi Deng [1], Minggong Wu [1], Xiangxi Wen [1,*], Yuming Heng [2] and Liang You [3]

[1] Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an 710051, China; dwcz_2023@163.com (W.D.); wuminggong@sohu.com (M.W.)
[2] Unit of 95129, Kaifeng 475000, China; 13419886627@163.com
[3] Equipment Management and Unmanned Aerial Vehicle Engineering College, Air Force Engineering University, Xi'an 710051, China; you_liang2022@163.com
* Correspondence: wxxajy@163.com

Abstract: This study aims to address the issue of redundancy and interference in data-collection systems by proposing a novel feature-selection method based on maximum information coefficient (MIC) and graph symmetry structure in complex-network theory. The method involves establishing a weighted feature network, identifying key features using dominance set and node strength, and employing the binary particle-swarm algorithm and LS-SVM algorithm for solving and validation. The model is implemented on the UNSW-NB15 and UCI datasets, demonstrating noteworthy results. In comparison to the prediction methods within the datasets, the model's running speed is significantly reduced, decreasing from 29.8 s to 6.3 s. Furthermore, when benchmarked against state-of-the-art feature-selection algorithms, the model achieves an impressive average accuracy of 90.3%, with an average time consumption of 6.3 s. These outcomes highlight the model's superiority in terms of both efficiency and accuracy.

Keywords: feature selection; MIC; graph symmetry structure; dominating sets; UNSW-NB15

Citation: Deng, W.; Wu, M.; Wen, X.; Heng, Y.; You, L. A Feature-Selection Method Based on Graph Symmetry Structure in Complex Networks. *Symmetry* **2024**, *16*, 549. https://doi.org/10.3390/sym16050549

Academic Editor: Theodore E. Simos

Received: 21 March 2024
Revised: 24 April 2024
Accepted: 30 April 2024
Published: 2 May 2024

1. Introduction

1.1. Literature Review

Feature selection (FS) is a crucial issue in the fields of machine learning (ML) and data mining (DM) [1–3]. Selecting the most relevant and informative features from a dataset can not only enhance the predictive performance and generalization ability of the model but also reduce the complexity and computational cost. It has been widely applied in various domains, such as text classification, image recognition, and bioinformatics.

Common feature-selection methods can be categorized into Filter [4–6], Wrapper [7–9], and Embedded methods [10–12]. These methods preserve key features by selecting and retaining them from the original dataset [13]. Filter methods are simple in design and fast in computation. HAHS [14] employs significance tests on sample mean differences for feature selection, but its screening capability is limited, leading to the presence of noise and redundant features in the selected subset that are unable to guarantee the optimality of the feature subset. Wrapper methods combine classification techniques with search algorithms, continuously evaluating the classification performance of selected features to identify the best ones. GUYON [15] uses a recursive feature-elimination algorithm to find the optimal features. However, due to the inclusion of the classification process during iteration, the computational burden is often significant. Embedded methods incorporate penalties for features into the loss function, enabling the model to automatically select the most relevant features during training, complete automatic screening, and the adjustment of features. Common algorithms include Lasso regression [16], Ridge regression [17], and Elastic Net regression [18].

Apart from methods that eliminate redundant features, analyzing the interrelationships among features is also a crucial means of feature selection. Pearson [19] utilizes the Pearson correlation coefficient to measure linear relationships, which is fast and easy to compute. However, if the relationship is nonlinear, even if two variables have a one-to-one correspondence, the Pearson correlation may approach zero. The use of the Spearman correlation coefficient to eliminate redundant features and retain key ones can result in excessive deletion of information, leading to the loss of critical information. Additionally, in practical operations, these algorithms tend to have large computational burdens and longer runtimes. Reshef [20] proposed a new metric based on information theory, utilizing mutual information and the maximal information coefficient to reflect nonlinear relationships between features. However, its results on different datasets are not comparable, and it has limitations in handling continuous variables.

In summary, there are still some limitations to current feature-selection methods. (1) The curse of dimensionality. As the number of features increases, the difficulty of feature selection also rises. The computational complexity of searching for the optimal feature subset in high-dimensional spaces increases exponentially, and more data are required to support model training and generalization. (2) Ignoring feature interactions. Some features may seem unimportant when considered individually but may provide crucial information when combined with other features [21,22]. Common feature-selection methods require pre-determined feature combinations, potentially ignoring complex interactive relationships among features. (3) Limited methods for handling continuous variables and nonlinear features.

1.2. Related Works

Based on the aforementioned issues, this paper presents a graph symmetric structure complex-network-based feature-selection (CNBFS) approach with three key advantages. (1) Uncover non-linear relationships. Extract the inherent connections among features from a global viewpoint to more effectively capture the intricate structure between features. (2) Adapt to high-dimensional sparse data. Adapting to high-dimensional sparse data improves its robustness, facilitating better coping with high-level and sparse data, identifying potentially significant features and feature combinations, and being able to handle large-scale feature space. (3) Strong interpretability. To begin, the MIC is utilized to determine the connected edges and weights of the symmetric complex network. Next, determine the optimal feature set by combining the dominating set and discrete binary particle-swarm optimization (BPSO) algorithm. Finally, confirm the validity of the experiment by evaluating the UNSW-NB15 dataset and least squares support vector machine (LS-SVM) classification algorithm. The experimental results demonstrate that the method adeptly and efficiently identifies the most representative features, thus significantly improving classification accuracy, reducing redundant features, and enhancing classification performance and generalization capabilities.

2. Construction of Feature Network

As complex relationships exist between features in a dataset, it is possible to construct a feature network by analyzing their correlations. The feature network is a symmetric complex network consisting of features (nodes) and correlations between features (edges), denoted as $G = (V, E, W)$. The set $V = \{v_1, v_2, \ldots, v_n\}$ represents the nodes in the network, which correspond to the various features in the system. The set $E = \{e_1, e_2, \ldots, e_n\}$ represents the edges, which reflect the interrelationships between features, and the set $W = \{w_1, w_2, \ldots, w_n\}$ represents the edge weights, which indicate the degree of association between features in the feature network. When constructing a feature network, it is necessary to identify the nodes and the connected edges. Since the nodes of a complex network are the features of that dataset, the following subsections will focus on explaining how to determine the connected edges.

2.1. Determination of the Maximum Information Coefficient

The key to building a feature network model is determining the connected edges. The MIC not only measures the linear and nonlinear relationships between the features but also reflects the non-functional dependencies between the features. In this paper, we determine the connected edges of the feature network based on the MIC values between the features.

The MIC is obtained by mutual information and mesh-partitioning methods. The specific calculation process of the MIC is as follows.

Given a finite set of ordered pairs $D\{(x_i, y_i), i = 1, 2 \ldots n\}$, the scatterplot containing x_i and y_i in D is gridded with $x \times y$, and the mutual information in each grid is calculated. For two random variables $X = \{x_i, i = 1, 2 \ldots n\}$ and $Y = \{y_i, i = 1, 2 \ldots n\}$, n is the number of samples. Equation (1) is used to calculate the mutual information.

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{1}$$

where p(X, Y) is the joint probability densities of X and Y, and p(X) and p(Y) are the edge probability densities of X and Y, respectively. max(I(X, Y)) denotes the maximum value of I(X, Y) under different information, which is selected as the mutual information value for dividing the $x \times y$ grid. Then, mic(I(X, Y)) represents the result obtained after its normalization.

Equation (2) is used to calculate the MIC.

$$mic(X : Y) = \max_{|X||Y| < B} \frac{\max(I(X, Y))}{\log_2(\min(|X|, |Y|))} \tag{2}$$

The matrix B in Equation (4) is the upper limit of the grid division $x \times y$, which is correlated with the number of samples. The best outcome is demonstrated empirically for $n^{0.6}$, and the same approach is used in the remaining portion of this study. Additionally, $mic(I(X : Y)) \in [0, 1]$: when mic = 0, it indicates that two features are probabilistically independent; when mic = 1, it implies that two features are fully correlated and interchangeable. As the maximum MIC rises, any two traits become more relevant to one another. Buda Andrzej's analysis of the use of the correlation coefficient in his work showed that the correlation coefficient is at [0, 0.2], and the correlation between the two features is extremely weak or does not exist [23]. Therefore, in this paper, the MIC is used to evaluate the feature-to-feature correlation when it is used to determine whether two nodes i and j have a connected edge. When $mic(i, j) < 0.2$, there is no edge between the two feature nodes, and the correlation is seen as being small. There is a connected edge, and the two nodes are regarded as correlated when $mic(i, j) \geq 0.2$. The edge weight between any two nodes is set to $mic(i, j)$.

2.2. Analysis of Network Topology

The collected features are constructed as a feature network through the feature nodes and correlations between them. This network is an undirected weighted network because the MIC between any two features are equivalent. The adjacency matrix of the feature network is denoted as A and the weights are stored in matrix B. Equations (3) and (4) are used to derive matrices A and B.

$$A = \begin{vmatrix} 0 & a_{12} & \cdots & a_{1N} \\ a_{21} & 0 & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & 0 \end{vmatrix}, \begin{cases} a_{ij} = 0 \text{ if } mic_{ij} < 0.2 \\ a_{ij} = 1 \text{ if } mic_{ij} \geq 0.2 \end{cases} \tag{3}$$

$$B = \begin{vmatrix} 0 & b_{12} & \cdots & b_{1N} \\ b_{21} & 0 & \cdots & b_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & 0 \end{vmatrix}, \begin{cases} b_{ij} = 0 \text{ if } mic_{ij} < 0.2 \\ b_{ij} = 1 \text{ if } mic_{ij} \geq 0.2 \end{cases} \quad (4)$$

Figure 1 shows a typical topology diagram of a feature network. The following feature-node analysis is performed with the help of this diagram.

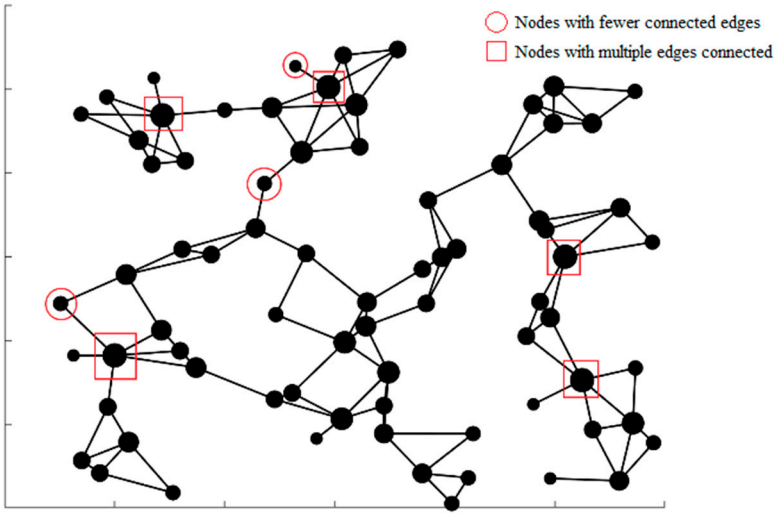


Figure 1. A Feature-network topology.

Observing Figure 1, it is noteworthy that the nodes' size and the number of connected edges denote the correlation of features between different nodes. The larger nodes and the more connected edges indicate a higher number of features between the nodes, as shown by the highlighted nodes in boxes and circles. The nodes identified by the boxes exhibit greater connectivity to other nodes, suggesting that they hold more informative value and potentially serve as the optimal feature selection. Furthermore, these nodes serve as the genesis and foundation of the concepts presented in this paper.

3. The Proposed Algorithm

In the feature-selection problem, on the one hand, it is necessary to identify features that contain as much information as possible about all features in order to achieve better accuracy in later classification. On the other hand, it is important to reduce the redundant features as much as possible. When evaluating redundancy, if there is a high correlation between two feature points, then there is redundant information. This approach can effectively reduce the cost and computational complexity of subsequent algorithms and improve efficiency.

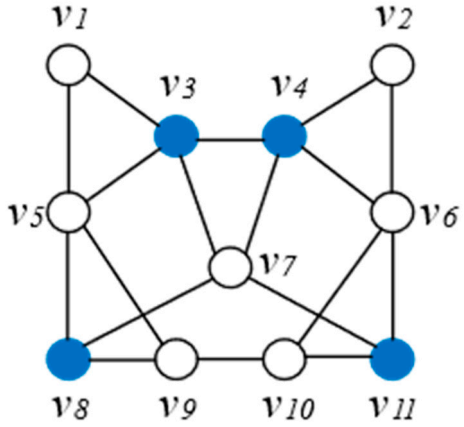
3.1. The Concept of Dominating Set

From the process of constructing the feature network in the previous section, it can be seen that there are edges between nodes in the feature network, indicating that there is a correlation between nodes and that they contain each other's information. The higher the weight of the edges, the higher the correlation between the nodes. To find the minimum set of features that contains all the information, it is required that the selected set of feature points in the feature network have edges to all the remaining points, which is consistent with the concept of a dominating set in graph theory. The concept of the dominating set is introduced below.

Definition 1. *In an undirected graph G, if $S \subseteq V$ ($S \neq \emptyset$), and for $\forall x \in V - S$, x are directly connected to at least one node in S, then S is considered to be a dominating set of G.*

According to the analysis in previous sections, the feature subset that contains all the information with the least correlation is selected to be the dominating set in the feature network, which can also be denoted as the optimal feature set.

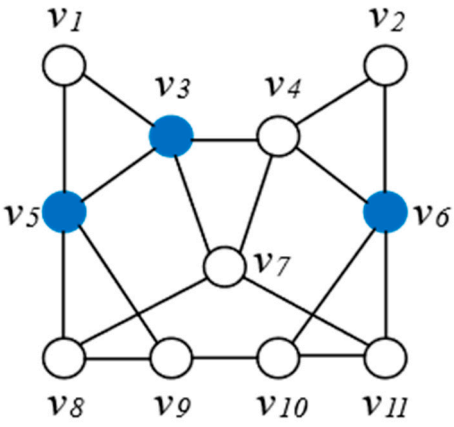
Figure 2 illustrates the dominant set with two typical dominating sets of a graph represented by the blue nodes. It can be observed that a graph may have multiple dominating sets. Both network features (a) and (b) have 11 nodes $v_1, v_2, \cdots, v_{11}$, but their dominating sets differ. $S_a = \{v_3, v_4, v_8, v_{11}\}$ has four nodes, and $S_b = \{v_3, v_5, v_6\}$ has three nodes. It is evident that, for a symmetric complex network G, there is no unique dominating set. Consequently, it remains unclear which obtained dominating set represents the optimal feature set. In the selection process of the dominating set, when the connected edges (i.e., whether there is a nonlinear relationship between features) are considered but their weights (i.e., the correlation coefficients between the features) are not, the correlations between the features cannot be reflected. Feature selection hopes that the identified nodes contain as much classification information as possible [24]. Therefore, we choose to construct the connected edges based on the correlations between the features. If the weight of a connected edge connected to a node is larger, it means that the number of nodes connected to the node is larger, and the correlation is larger. This indicates that this node contains more information. The node strength of a selected feature node is explained as follows.



Nodes: $V_1, V_2, \cdots, V_{11}$

Dominating sets: $S_a = \{V_3, V_4, V_8, V_{11}\}$

(**a**) A typical dominating set with four nodes



Nodes: $V_1, V_2, \cdots, V_{11}$

Dominating sets: $S_b = \{V_3, V_5, V_6\}$

(**b**) A typical dominating set with three nodes

Figure 2. Schematic of dominant set.

Definition 2. *The node strength of node v_i is the sum of the edge powers w_{ij} of the connected edges through the node, denoted as S(i). Equation (5) is used to calculate the node strength:*

$$S(i) = \sum_{j \in P_i} w_{ij} \tag{5}$$

where P_i is the set of nodes that form a connected edge with node v_i. The node strength reflects the total amount of influence that neighboring nodes have on it. The higher the node strength, the higher the correlation between the feature node and the features, indicating that it contains more information. Therefore, when selecting the dominant set, it is necessary to find as many nodes with high node strength as possible as the final features.

3.2. Optimal Dominating Set Based on BPSO Algorithm

The optimal dominating set is defined as the minimum number of nodes within a graph required to form a set that covers every node in the graph. This set, known as the dominating set, can be either directly or indirectly connected to other nodes in

the graph. The objective of identifying the optimal dominating set is to locate a set of nodes that dominates the entire graph while utilizing minimal resources and incurring minimal cost. The set must be composed of the least possible number of nodes to ensure comprehensive coverage.

Taking the above considerations into account, when the feature network is analyzed to find the optimal set of features, the selected set of features needs to meet the following requirements.

1. The selection of features should be minimized to improve the classification efficiency, which is beneficial for later stages;
2. The node strengths of the selected features should be as large as possible so that the selected features can contain more classification information.

Meanwhile, the selected set of feature nodes must be the dominant set of the whole network.

The essence of this problem is to find a subset of features from N feature nodes that meets the requirements. Thus, the solution space of the problem can be transformed into the selection of an N-dimensional vector. The vector x has N dimensions, representing N features. The value of the x_i dimension corresponds to whether the node v_i in the feature network is selected, with one indicating the node is selected and zero indicating the node is not selected. In this way, the problem is transformed into a 0–1 programming problem. In order to solve this problem, the binary particle-swarm optimization algorithm (BPSO algorithm) is selected here.

In this paper, we outline the steps to solve the optimal dominating set based on the BPSO algorithm; they are as follows:

Step 1: Construct the adjacency matrix.

Use the MIC to create the adjacency matrix A for the feature network. The element A_{ij} in the matrix indicates whether node i and node j have connecting edges. If a connecting edge exists between i and j, the value of A_{ij} is 1. Otherwise, the value of A_{ij} is 0;

Step 2: Generate Random Binary Strings.

Create multiple random binary strings with a size of n, where n denotes the number of nodes. These strings pertain to the possible nodes in a dominating set with a length of n. For the i-th node, the i-th bit equates to one to signify that it will serve as the dominating point and zero to signify that it will not;

Step 3: Encode the objective function.

Record the optimal set of domination points as the mechanics optimization objective;

Step 4: Update string velocity and position.

Transcoding is accomplished by using the Sigmoid function to determine the ideal location of each particle and the overall optimal position by updating the velocity and position of each binary string while maintaining the original set of dominant optimal points;

Step 5: Iterative updating.

Iterate until the maximum number of iterations is reached or the optimal solution is found. After each iteration, the current set of dominating points should be compared to the previous iteration's set, and the optimal set of dominating points should be updated if the current set is better;

Step 6: Return the optimal set of features.

The global set of optimal dominating points formed by these nodes, i.e., the dominating set, is used to cover all nodes in the graph.

The BPSO algorithm is based on the basic particle-swarm algorithm, which specifies that zero and one are only considered as the values that the particles can pick up and change in the state space and is transcoded by the Sigmoid function. Each dimension of the velocity v_{ij} represents the possibility of taking one for each bit of the position x_{ij}. Therefore, the v_{ij} update formula in the continuous particle swarm remains the same, but the individual extremum as p_{best} and the global optimal solution g_{best} consist of zero and one only. Equations (6) and (7) are used to calculate the position updates.

$$s(v_{i,j}) = 1/[1 + \exp(-v_{i,j})] \quad (6)$$

$$x_{i,j} = \begin{cases} 1, & r < s(v_{i,j}) \\ 0, & \text{others} \end{cases} \tag{7}$$

In Equation (7), r is the random number generated in the $U(0,1)$ distribution.

Equation (8) is used to calculate the velocity updates.

$$v_{i,j} = \omega \cdot v_{i,j} + c_1 \cdot \text{rand}() \cdot (p_{i,j} - x_{i,j}) + c_2 \cdot \text{rand}() \cdot (p_{g,j} - x_{g,j}) \tag{8}$$

where $p_{i,j}$, $x_{i,j}$, $p_{g,j}$, $x_{g,j}$, respectively, represent the individual optimal position, individual current position, global optimal position, and the current position of the population. The rand() is the random number that is the same as the r in Equation (7).

Based on the analysis in Section 3, the optimization objective is to find a dominating set from all the points in the network, with as few points as possible, while maximizing their total point strength. As mentioned earlier, the solution x is a binary string of length 41, and its value corresponds to whether the corresponding feature is selected or not. Therefore, the number of selected features is $\sum x_i$. The node strength of the selected feature node represented by x is the sum of its corresponding row in the weighting matrix B, denoted as $\text{sum}(B_x)$.

Equation (9) is used to calculate the optimization objectives.

$$f(x) = \min(\sum_{i=1}^{N} x_i - k * \text{sum}(B_x)) \tag{9}$$

where k is a coefficient to regulate the relationship between the number of samples selected and the amount of information contained.

The constraint is that the selected set of points must be a dominating set. According to the definition of a dominating set, all points in the network must have a direct connection with at least one point in the selected set S. The adjacency matrix corresponding to the points in set S is represented by the rows of A, denoted as A_{si}. Equation (10) is the merging operation for A_{si}.

$$Q = A_{s1} \cup A_{s2} \cup \cdots \cup A_{st} \tag{10}$$

Let t be the number of the selected features and Q be a $1 \times$ n-dimensional 0–1 type row vector. According to the definition of a dominating set, we can get that all positions should be one except for the position of element x_i in the dominating set S, where the corresponding value can be zero.

Therefore, we can obtain the final optimization objective. Equation (11) is used to calculate the final optimization objective.

$$\begin{aligned} &f(x) = \min(\sum_{i=1}^{N} x_i - k * \text{sum}(B_x)) \\ &x_i = \begin{cases} 1, & v_i \text{ as the dominating point} \\ 0, & v_i \text{ not as the dominating point} \end{cases} \\ &\text{s.t. } Q_{(x_i=0)} \subseteq Z_i \end{aligned} \tag{11}$$

where Z_i denotes the set of connected dominating nodes for a given graph $G(V, E)$, and $Q_{(x_i=0)}$ denotes the set of nodes with element 0 in the row vector Q.

3.3. Algorithm Flow

According to the analysis above, the flow of the proposed algorithm is shown as follows.

Step 1: Information Collection and Processing.

Relevant data is collected through existing network-management systems, and specific network features are obtained through data processing such as data cleaning and normalization;

Step 2: Construct the feature network.

Perform a correlation analysis on the processed set of features, calculate the MIC between the features, and determine whether two nodes (features) are connected based on their MIC values. This process enables the construction of a feature network;

Step 3: Finding the Optimal Dominating Set.

In the constructed feature network, the optimal set of dominating nodes is searched using the BPSO algorithm, with the optimization objective defined by Equation (9). In this paper, if the set of solutions does not correspond to the dominating set of the network, the objective value is set to a very large number (e.g., 10,000);

Step 4: Solve for the optimal feature set.

Map the nodes found in the dominating set back to features to obtain the optimal feature set.

The pseudo-code for Algorithm 1 is shown below.

4. Simulation Verification and Analysis

Because LS-SVM can process massive datasets with fewer parameters and exhibits superior generalization ability and compatibility, this study favors the LS-SVM algorithm for classification learning and utilizes the suggested selection approach founded on symmetric complex network characteristics for validating the UNSW-NB15 dataset.

4.1. Dataset

The UNSW-NB15 dataset was created by the Cyber Range Lab at UNSW Canberra, Australia, using the IXIA Perfect Storm tool, and includes both normal and abnormal network traffic captured from real network operations [24–27]. The dataset contains 100 GB of raw data and includes 49 features, which are described in detail in Algorithm 1. After cleaning the collected data, including script, sport, dtsip, dsport, stime, ltime, and handling abnormal and missing values, a 41-dimensional dataset was obtained. This dataset has the characteristics of high dimensionality and large sample size, which meets the experimental requirements of this study and can be used as experimental data. The selected records are shown in bold in Table 1.

Table 1. UNSW-NB15 intrusion detection dataset.

Feature Category	Feature Name
flow features	srcip, sport, dstip, dsport, **proto**
base features	**state, dur, sbytes, dbytes, sttl, dttl, sloss, dloss, service, sload, dload, spkts, dpkts**
content features	**swin, dwin, stcpb, dtcpb, smeansz, dmeansz, trans_depth, res_bdy_len**
time features	**sjit, djit**, stime, ltime, **sintpkt, dintpkt, tcprtt, synack, ackdat**
additional generated features—general purpose features	**is_sm_ips_ports, ct_state_ttl, ct_flw_http_mthd, is_ftp_login, ct_ftp_cmd**
additional generated features—connection features	**ct_srv_src, ct_srv_dst, ct_dst_ltm, ct_src_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm**
labeled features	attack_cat, Label

The data used in this experiment contains 41 dimensions (attribute categories), such as time, number of bytes, and number of messages, which have various dimensions and greatly differ in data ranges. This situation can affect the accuracy of the classifier. To improve the classification accuracy, the data needs to be normalized. The normalization method is the maximum–minimum method. Equation (12) is used to obtain the normalized results.

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{12}$$

In other words, the constructed feature network is a symmetric complex network with 41 nodes. The illustration of the feature-network model obtained from the correlation analysis of the UNSW-NB15 dataset is shown in Figure 3.

Algorithm 1 Computer the optional dominating set through complex network and BPSO algorithm.

Input: Dataset, Particle-swarm-size NP, Objective function $f()$, Maximum number of iterations G, Dimension D, Velocity V_{ij}, Random position x_{ij}, Individual best position $p_{ij\text{-}best}$, Global best position g_{best}
Output: the optional dominating set Q

▷Stage 1 Construct feature network

function MIC-MATRIX (*data frame, dataset*)
  compute *mic(I(X:Y))* according to Equation (3)
  if $mic(I(X:Y) < 0.2$ **then**
    $A_{ij} = 0$
  else
    $A_{ij} = 1$
  end if
  return adjacency matrix A_{ij}
end function
edges $\leftarrow A_{ij}$
$G(V, E, W) \leftarrow nx.Graph()$

▷Stage 2 Compute the optional dominating set through BPSO algorithm

Function BPSO (NP, D, G, V_{ij}, x_{ij})
  //Initialization particle swarm and velocity:
  for $i = 0 \rightarrow NP$ **do**
    for $j = 0 \rightarrow D$ **do**
      $v_{ij}, x_{ij} \leftarrow random()$ //Initialize velocity and position with random values
      $p_{ij-best}[ij] \leftarrow x_{ij}[ij]$ //Set initial individual best position as x_{ij}
    if $f(x_{ij}[i]) < f(p_{ij-best}[i])$ **then**
      $p_{ij-best}[i] = x_{ij}[i]$//Update individual best position
    end if
    if $\boldsymbol{f(x_{ij}[i]) < f(g_{best}[i])}$ **then**
      $g_{best}[i] = x_{ij}[i]$//Update global best position
        end if
      end for
  end for
  //Iteration loop
  for $t = 0 \rightarrow G$ **do**
    //Update velocity and position
    for $i = 0 \rightarrow NP$ **do**
      computer $v_{ij}[ij]$ according to Equation (8)
      $x_{ij}[i, j] = sigmoid(v_{ij}[i, j])$//Apply activation function
      if $x_{ij}[i, j] > 1$ **then**
        $x_{ij}[i, j] = 1$
      end if
      if $\boldsymbol{x_{ij}[i, j] < 1}$ **then**
        $x_{ij}[i, j] = 0$
      end if
    end for
    //Update individual and global best positions
    for $i = 0 \rightarrow NP$ **do**
      if $f(x_{ij}[i]) < f(p_{ij-best}[i])$ **then**
        $p_{ij-best}[i] = x_{ij}[i]$//Update individual best position
      end if
      if $\boldsymbol{f(x_{ij}[i]) < f(g_{best}[i])}$ **then**
        $g_{best}[i] = x_{ij}[i]$//Update global best position
      end if
    end for
  end for
  return g_{best}
end function

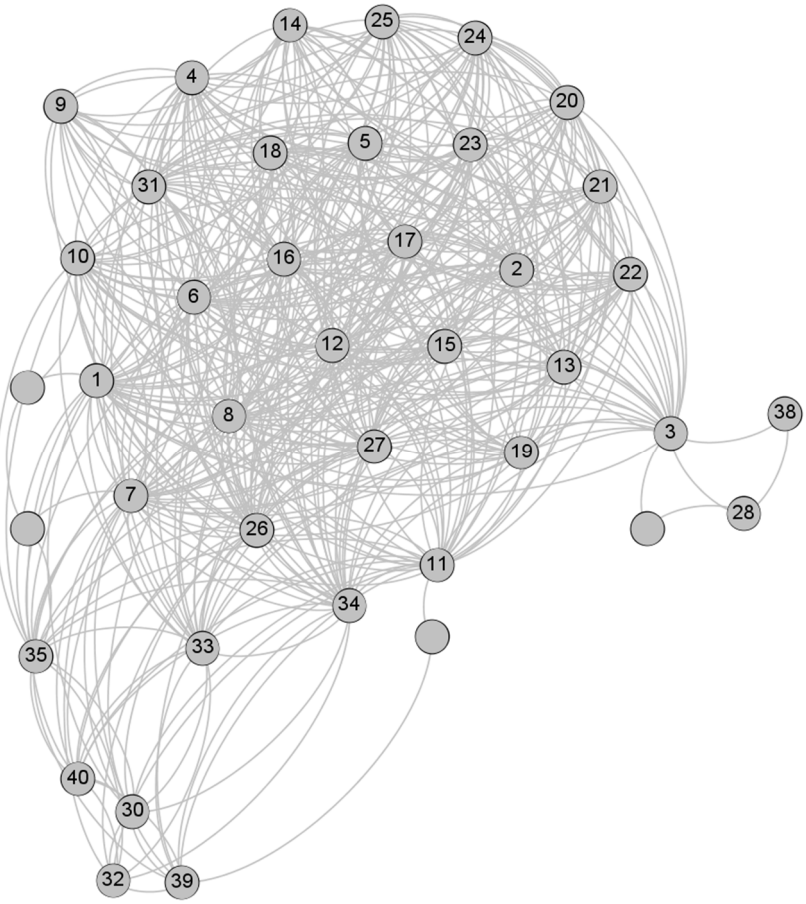


Figure 3. Feature-network topology based on UNSW-NB15 dataset.

4.2. Principles of LS-SVM

LS-SVM is an improved support vector machine algorithm based on statistical theory. It was originally proposed by Suykens. This algorithm can effectively overcome overfitting problems by transforming the learning-optimization problem into a linear equation problem and solving local extremes in the process [28]. At present, the LS-SVM algorithm has been applied in various fields, including data regression and time-series system prediction.

Given m n-dimensional datasets $\{x_i, y_i\}, y_i \in \{-1, 1\}, x_i \in R^n, i = 1, 2, 3 \cdots m$, the mapping of the input space to the feature space is implemented through a nonlinear function $\varphi(x_i)$, which constructs the optimal decision function f(x), as shown in Equation (13).

$$f(x) = b + \langle \varphi(x), w \rangle \tag{13}$$

where w is the weight vector, and b is the bias term.

Equation (14) is used to calculate the objective function and constraints of the algorithm.

$$\begin{aligned} &\min J(w, e) = \min(\tfrac{1}{2}\|w\|^2 + \tfrac{1}{2}\gamma \sum_{i=1}^{N} e_i^2) \\ &\text{s.t. } y_i = \langle w, \varphi(x_i) \rangle + b + e_i \\ &i = 1, 2, \cdots N \\ &\gamma > 0 \end{aligned} \tag{14}$$

where γ is the penalty factor, and e_i denotes the regression error.

Therefore, by constructing a Lagrangian function and performing the derivatives, a linear equation system can be obtained. Solving this system of linear equations gives

the classification expression of the algorithm. Equation (15) is used to obtain the final classification result.

$$y(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x) + b \quad (15)$$

where $K(x_i, x)$ is the kernel function, which usually can be a radial basis kernel function, a polynomial kernel function, a linear kernel function, and so on.

There are several advantages of using the radial basis function (RBF):

1. Nonlinear mapping: The radial basis kernel function can map the original feature space to a higher dimensional space through nonlinear mapping, thus enabling support vector machines to handle nonlinear problems. This allows better adaptation to complex data patterns and decision boundaries;
2. Flexibility: The radial basis kernel function is a parameter-free kernel function that does not require a priori determination of the form of the mapping function. It determines the weights of the sample points by calculating the distances between the sample points and the support vectors, so it can be adapted to various irregular data distributions and decision boundary shapes;
3. Efficient computation: The computation of the radial basis kernel function is relatively simple and efficient. It only involves calculating the distance between the sample points and the support vectors without explicit feature mapping. This saves computational resources and allows high-dimensional data to be handled; In summary, the radial basis kernel function is selected in this paper.

4.3. Algorithm Stability Analysis

First, this section will analyze the validity of the proposed method. For feature-selection problems using machine-learning methods, different training samples can affect the set of selected features and, thus, result in different models. In particular, the BPSO algorithm used in this paper for sample selection may affect the final search results due to the initialization of the algorithm and the randomness of the intermediate speed changes. Therefore, assessing the stability of the algorithm's operation is a crucial method to evaluate its effectiveness.

For feature network construction and feature selection, 10,000 samples are randomly selected from the whole dataset, and the classification accuracy without feature extraction and through other methods is compared. The experiment is repeated 200 times, and the classification accuracy of each time is counted. The results are expressed by box-line plotting and shown in Figure 4.

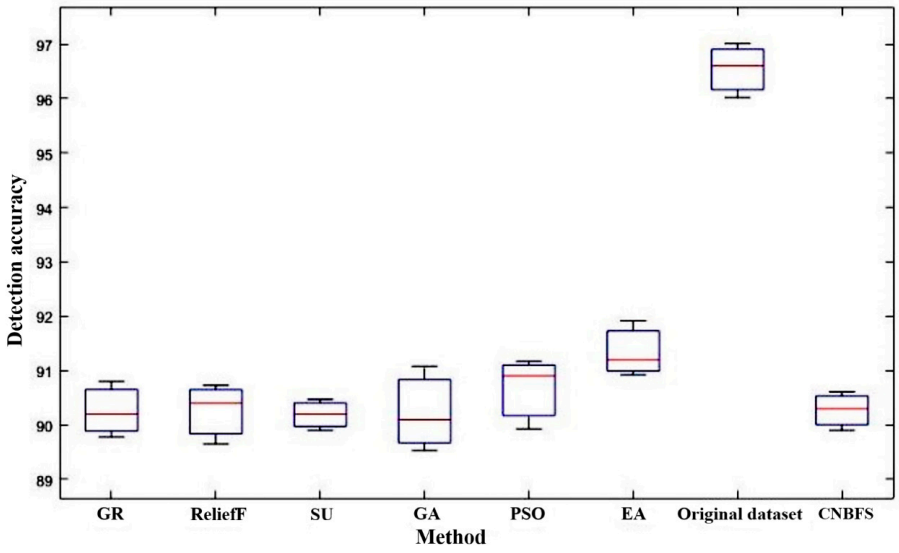


Figure 4. Stability comparison between different methods.

It can be seen that the model boxes established by the feature sets processed by the above methods are compact, which indicates that these methods are stable. Specifically, the average classification accuracy of the proposed method based on the samples of this experiment reaches 90.1%, which is comparable to that based on the complete dataset, regardless of variations during specific tests. Meanwhile, its box is also the most compact with a difference between the upper and lower bounds of only 0.71%, which indicates that integrating symmetric complex-network theory into feature selection, as represented by the proposed method, is stable and effective.

The algorithms are tested for significance, and it is hypothesized that there is no significant difference in performance between the algorithms.

H0: *There is a significant difference in performance between the algorithms.*

H1: *There is no significant difference in performance between the algorithms.*

The level of significance is 0.05. The ANOVA results are shown in the Table 2.

Table 2. The ANOVA results for different algorithms.

Variable Value	Sample Size	Average Value	Variance	F	*p*
GR	1000	90.262	0.0976		
ReliefF	1000	90.533	0.1192		
SU	1000	90.363	0.0174		
GA	1000	90.075	0.1813	99.4489	0.000 ***
PSO	1000	90.801	0.1655		
EA	1000	91.164	0.1487		
CNBFS	1000	90.288	0.0343		
sum	7000	90.498	0.1319		

*** in 0.000 *** is a labeling method used by statistical software to indicate that the *p*-value is very small and meets the significance level standard set by the software.

The ANOVA results show a *p*-value of 0.000 *** $\leq$ 0.05; therefore, the statistical results are significant, indicating that the different algorithms have significant differences in feature selection.

4.4. Feature-Selection Experiment

4.4.1. Feature Selection

During the validation process of our proposed algorithm based on the UNSW-NB15 dataset, the main factor to consider is the number of selected features. Therefore, the coefficient k of the node strength is set to 0.1. After conducting a BPSO search, we obtained four advantageous feature nodes, as shown in Figure 5.

The red dots in Figure 5 are the dominant nodes. It can be seen from the figure that the selected feature nodes have connections with other non-dominant nodes, indicating the effectiveness of the proposed algorithm. Moreover, the selected feature set contains only four features, which greatly reduces the burden on the information-gathering system. The specific features are listed in Table 3.

Table 3. Results of the feature selection.

Feature Name	Feature Description	Feature Name	Feature Description
dur	Record total duration	service	http, ftp, smtp, ssh, dns, ftp-data, irc, and (-) if not much used service
sbytes	Source to destination transaction bytes	sload	Source bits per second

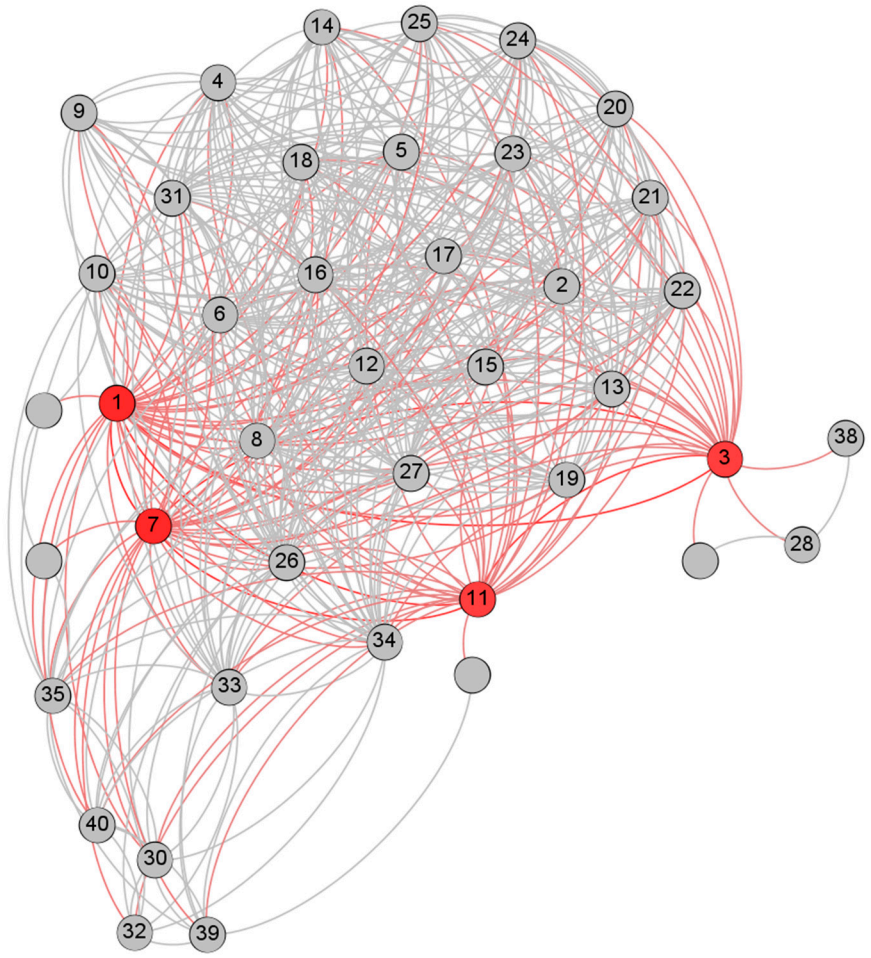


Figure 5. Optimal dominant nodes.

The UNSW-NB15 dataset is very large. We randomly selected 10,000 normal data and 3000 abnormal data for training, and an additional 4000 samples were randomly selected for testing. Some of the training results are shown in Figure 6.

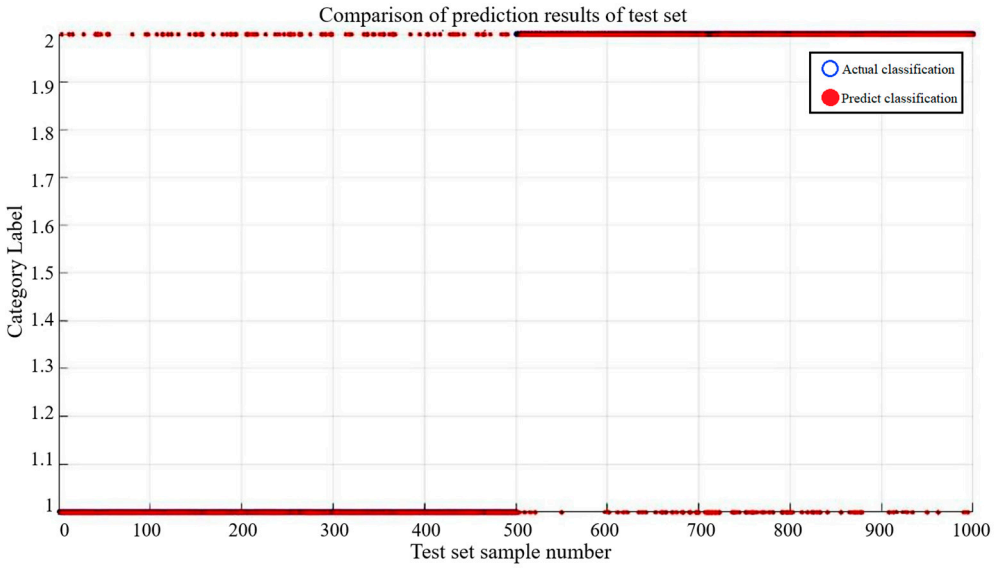


Figure 6. Comparison chart of some training results.

4.4.2. Comparison of the Classification Prediction Based on the Original Dataset and the Selected Features

To demonstrate the effectiveness of the CNBFS method, we first compare the classification prediction results based on the selected features by using it with those based on the original dataset. As shown in Table 4, we compare the number of features, classification accuracy, and algorithm running time.

Table 4. Comparison of the classification prediction based on the original and the selected data.

Dataset	Original Dataset	CNBFS Filtered Dataset
Number of features (pcs)	41	4
Classification accuracy (%)	96.62	90.3
Algorithm time (s)	29.8	6.3

The table shows that, after feature selection using CNBFS, the number of features in the data is reduced to only four. Although the classification accuracy after data filtering is slightly lower compared to using the LS-SVM algorithm to classify the original dataset, the decrease is only 6.32%, which is within an acceptable range. From the perspective of algorithm running time, since feature selection greatly reduces the amount of data, the operating speed is improved significantly, and the overall running time (feature selection and model training) has been reduced from 29.8 s to 6.3 s. The above results indicate that, compared to traditional classification algorithms, the proposed method can obtain the most concise features, with only a small gap in classification accuracy compared to the original data, while significantly shortening the running time of the classification. For scenarios that do not require very high target accuracy, it is already sufficient to meet the requirements of target classification. For scenarios that demand high accuracy, it can help improve the screening efficiency as the initial screening step (reducing time from 29.8 s to 6.3 s), making it easier to conduct more precise screening in subsequent steps.

4.4.3. Comparison between the Proposed Method and Some Typical Feature-Selection Algorithms

To validate the superiority of the proposed method, some state-of-the-art wrapper algorithms, including CMIM [29], JMI [30], DISR [31], mRMR [32], Relax-mRMR [33], IBSCA3 [34], and EOSSA [35] and some traditional filter algorithms, including Gini index, Fisher score, FS-OLS [36], ReliefF [37], and Kruskal–Wallis [38] are used for comparison. The LS-SVM algorithm is used to test the dataset after feature selection. The experimental results are shown in Tables 5 and 6.

Table 5. Comparison between the performance of the proposed method and some wrapper methods.

Methods	CMIM	JMI	DISR	mRMR	Relax-mRMR	CNBFS	IBSCA3	EOSSA
Number of features (pcs)	20	22	18	23	25	4	6	5
Classification accuracy (%)	90.2	89.6	88.3	89.1	88.8	90.3	90.1	89.9
Algorithm time (s)	7.9	8.7	10.4	8.5	19.1	6.3	6.7	7.1

CMIM = conditional mutual information maximization; JMI = joint mutual information; DISR = double input symmetrical relevance; mRMR = max-relevance and min-redundancy; CNBFS = conditional mutual breadth first search; IBSCA = binary improved sine cosine algorithm; EOSSA = elite opposite sparrow search algorithm.

Table 6. Comparison between the performance of the proposed method and some filter methods.

Methods	Gini Index	Fisher Score	Kruskal–Wallis	ReliefF	FS-OLS	CNBFS
Number of features (pcs)	19	20	21	18	18	4
Classification accuracy (%)	89.3	90.0	88.7	89.9	90.5	90.3
Algorithm time (s)	17.9	10.4	16.3	21.5	13.1	6.3

FS-OLS = feature selection–orthogonal least squares; CNBFS = complex-network-based feature selection.

The results show that the CNBFS method outperforms all selected wrapper-based and filter-based feature-selection methods in every aspect of the comparison experiments. First of all, the average number of the selected features through the selected methods for comparison is 20.4, the smallest number is 18, and the largest is 25. In contrast, the proposed method results in a number of four. Furthermore, the classification accuracy of the proposed method is 90.3%, while the average accuracy of the selected methods for comparison is only 81.3%. The highest value is 90.5%, and the lowest is 88.3%. In terms of the operation speed, the running time of the proposed method is only 6.3 s, the average running time of the selected methods for comparison is 12.16 s, and the shortest one is 7.9 s. Based on the comparison with the limited sample of these algorithms, the proposed method is arguably the optimal.

4.4.4. Comparison of the Performance of the Proposed Method Based on Different Datasets

In order to better validate the effectiveness of the algorithm, several datasets from the UCI machine-learning repository are selected for repeated validation [39,40]. The selected datasets are listed in Table 6. In each dataset, 90% of the samples are randomly selected for feature-network construction and feature selection, so as to compare the classification accuracy without feature extraction and through the proposed method. This represents the average number of samples each feature has and is used to represent the dimensionality of a dataset. The selected filter algorithm is the ReliefF algorithm, and the selected wrapper algorithm is the EA algorithm, since they outperform other algorithms within their own categories in the previous comparison experiments. The experiments are to be repeated 200 times, and the classification accuracy and algorithm running time are counted each time. The results are shown in Tables 7 and 8, and Figure 6.

Table 7. UCI datasets for comparison.

Dataset	Number of Original Features (N)	Number of Samples (D)	Number of Categories (C)	D/N
Parkinson's	22	195	2	8.9
SPECT Heart	22	267	2	12.1
Statlog (German Credit Data)	24	1000	2	41.7
Breast Cancer Wisconsin	30	569	2	19.0
WDBC	31	569	2	18.4
Ionosphere	33	351	2	10.6
Dermatology	34	358	6	10.5
Soybean (small)	35	47	4	1.3
Chess (KR vs. KP)	36	3196	2	88.8
Connectionist Bench	60	208	2	3.5
Libras Movement	90	360	15	4.0
Semeion Handwritten Digit	256	1593	10	6.2
Arrhythmia	279	452	13	1.6
UJIIndoorLoc	520	21,048	3	40.5
Gisette	5000	6000	2	1.2

BCW = breast cancer wisconsin; Chess = chess (KR vs. KP); CB = connectionist bench; LM = Libras movement; SHD = Semeion handwritten digit.

The proposed method outperforms all other selected methods for comparison. As shown in Figure 7, the classification accuracy of all tested algorithms for the same dataset is ranked to obtain the average prediction accuracy ranking for the 15 datasets. It can be seen that the average prediction-accuracy ranking of the CNBFS algorithm on the LS-SVM classifier using the MIC is 3.13, which is the best. The second-ranked algorithm is Relax-mRMR with an average prediction accuracy ranking of 4.13, and the lowest-ranked algorithm is Kruskal–Wallis, with an average prediction accuracy ranking of 9.8.

The maximum information coefficient is the most important factor that influences the betterment of the results algorithm because an established feature network by MIC can effectively search for nodes rich in information, thus significantly improving the search efficiency and accuracy of the algorithm.

Table 8. Sizes of the candidate feature subset of different feature-selection methods based on different datasets.

Dataset	CMIM	JMI	DISR	mRMR	Relax-mRMR	IBSCA3	EOSSA	Gini Index	Fischer Score	Kruskal–Wallis	ReliefF	FS-OLS	CNBFS
Parkinson's	16	14	11	15	13	14	10	13	14	13	12	16	4
SPECT Heart	14	15	12	13	12	8	5	10	13	15	12	15	3
Statlog	15	14	15	14	10	12	4	12	15	12	11	18	4
BCW	16	15	16	15	11	9	13	13	16	14	15	17	4
WDBC	16	18	17	16	12	13	15	15	18	13	17	19	5
Ionosphere	15	19	16	14	11	12	11	17	20	15	16	18	6
Dermatology	17	18	18	15	13	16	17	18	18	16	17	19	4
Soybean	16	19	17	18	14	17	16	17	19	17	18	18	6
Chess	18	20	19	17	16	17	18	17	19	17	18	20	5
CB	23	21	20	18	17	19	20	21	21	19	20	22	7
LM	25	23	22	25	22	23	25	20	26	23	22	24	9
SHD	32	34	31	40	38	31	32	31	29	35	37	38	88
Arrhythmia	42	48	45	52	47	41	42	43	38	41	45	44	93
UJIIndoorLoc	283	293	254	299	279	254	261	260	253	256	254	267	103
Gisette	3328	3103	2994	3127	3047	3106	2856	2951	2876	2954	2763	2845	1839

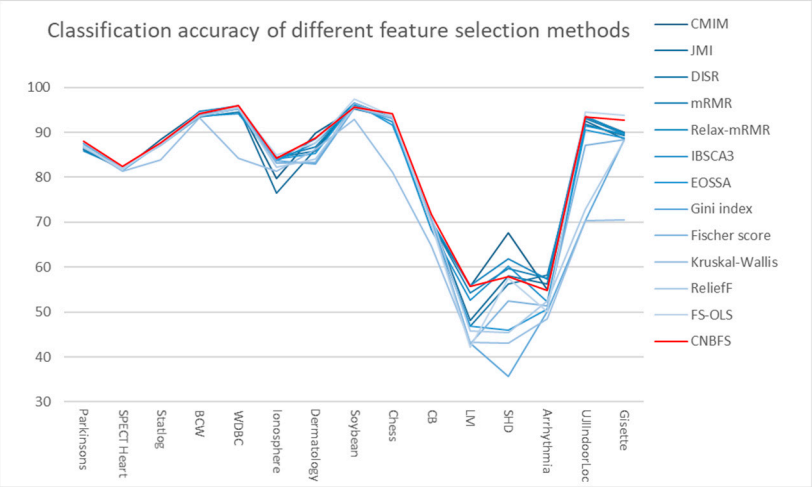


Figure 7. Classification accuracy of different feature-selection methods based on different datasets (%).

There are variations between distinct subsets and diverse data distributions. The dataset comprises several subsets, including binary classification, multiclass classification, large-scale multiclass classification, regression, and other subsets. These subsets present disparities in data volume, feature dimensions, data distribution, number of categories, data quality, and more, consequently causing the same algorithm to produce different outcomes. In particular, the CNBFS method achieves better performance with high D/N data (e.g., "Chess (KR vs. KP)", "UJIIndoorLoc"), and mundane performance with low D/N data [38]. The results are not as good as before (e.g., "Soybean (small)", "Arrhythmia"). Similarly, the CNBFS method also performs better with large volume data (D > 1000), even though their D/N ratios may be low (e.g., "Gisette") because the MIC among the data can be adequately processed due to the large sample size to obtain a more accurate dominant set.

The average running time of all the tested methods for all the tested datasets is shown in Figure 8. All operations are done using Matlab version 2018b. The results show that the CNBFS method runs faster than most of the tested algorithms, and only the Fisher score is faster than the CNBFS method; none of the selected state-of-the-art feature-selection algorithms outperforms the CNBFS method. In particular, the average running time of

Relax-mRMR is seven times that of the CNBFS method, and that of ReliefF is five times that of the CNBFS method.

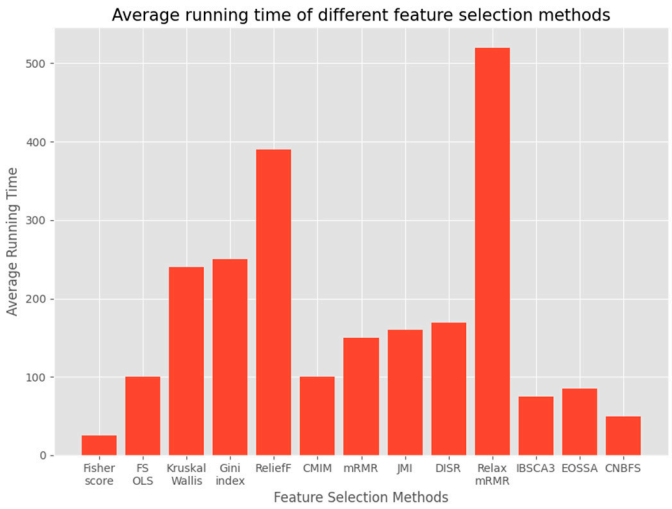


Figure 8. Average running time of different feature selection methods.

The experiments on different datasets show that although the results vary on different datasets, the proposed method achieves relatively optimal performance, since both its prediction accuracy and feature reduction efficiency are better than those of the tested conventional approaches. The three sets of experiments prove that the proposed method can achieve better performance when applied in both the binary and multi-classification·fields.

Moreover, the conclusion of this paper is verified by these comparison experiments, and the proposed algorithm is able to find out almost all the classification information by means of dominating sets. This paper proposes the construction of a feature network utilizing symmetric complex networks, primarily by utilizing pre-existing network features and weights. To iterate on the network, simple addition and subtraction operations are typically used and the computational space required is small. In contrast, traditional machine-learning algorithms, including logistic regression, decision trees, support vector machines, plain Bayes, and others rely on fitting parameters to classifiers based on provided data. This process requires constant training and testing of the classifiers, resulting in high operation costs. Overall, the CNBFS algorithm can effectively reduce the computational effort of feature selection and can obtain a streamlined dataset and high classification accuracy.

4.5. Algorithm Limitations and Future Improvements

The method of feature selection proposed in this paper based on symmetric complex networks requires high-quality data and has the following algorithm limitations:

1. Data Size: Symmetric complex networks usually require a large amount of data to construct accurate network models for better feature selection. This requirement may be limited for problems with small data size;
2. Data quality requirement: Symmetric complex networks usually require high data quality, and special processing methods may be needed for problems with more noise and missing data.

Considering issues such as the high demand for raw data and future research hotspots, this paper suggests that the next step could be to further optimize the following aspects:

1. Data Size: Complex networks usually require a large amount of data to construct accurate network models for better feature selection. This requirement may be limited for problems with small data size;
2. Optimize algorithm efficiency: Develop faster and more efficient algorithms to reduce the computation time of the feature-selection algorithm and improve its efficiency;

3. Introduction of feedback mechanism: Through the introduction of the feedback mechanism, the feature-selection algorithm can adaptively adjust the network structure and parameters so that its performance on different data sets is more universal;
4. Integrating deep-learning methods: by combining deep-learning methods, the characterization ability of neural network models can be utilized to further improve the accuracy and efficiency of feature selection.

5. Conclusions

In this paper, a methodology, called CNBFS, is proposed to solve feature-selection problems with high-dimensional data based on the combined use of a maximum information coefficient and a symmetric complex network.

The use of BPSO in CNBFS allows feature-selection problems to be solved in a more accurate and efficient way compared to the traditional methods in the UNSW-NB15 dataset and state-of-the-art feature-selection algorithms. The methodology is focused on the degree of correlation of information between the nodes and the topology of the overall network. The experimental results show that the model's running speed is significantly reduced, decreasing from 29.8 s to 6.3 s. Furthermore, when benchmarked against advanced feature-selection algorithms, the model achieves an impressive average accuracy of 90.3% with an average time consumption of 6.3 s. These results prove that the proposed method is stable and reliable, can accurately identify relevant features, has a greatly reduced running time, and provides new insight into solving feature-selection problems.

Author Contributions: Conceptualization, X.W. and M.W.; investigation, Y.H. and L.Y.; Writing, W.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: https://research.unsw.edu.au/projects/unsw-nb15-dataset (access date: 21 February 2024).

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

References

1. Kozodoi, N.; Lessmann, S.; Papakonstantinou, K.; Gatsoulis, Y.; Baesens, B. A multi-objective approach for profit-driven feature selection in credit scoring. *Decis. Support Syst.* **2019**, *120*, 106–117. [CrossRef]
2. Labbé, M.; Martínez-Merino, L.I.; Rodríguez-Chía, A.M. Mixed Integer Linear Programming for Feature Selection in Support Vector Machine. *Discret. Appl. Math.* **2019**, *261*, 276–304. [CrossRef]
3. Jayaprakash, A.; Keziselvavijila, C. Feature selection using Ant Colony Optimization (ACO) and Road Sign Detection and Recognition (RSDR) system. *Cogn. Syst. Res.* **2019**, *58*, 123–133. [CrossRef]
4. Blum, A.L.; Langley, P. Selection of relevant features and examples in machine learning. *Artif. Intell.* **1997**, *97*, 245–271. [CrossRef]
5. Jolliffe, I.T. Principal component analysis. *J. Mark. Res.* **2002**, *87*, 513.
6. Liu, C. Gabor-based kernel PCA with fractional power polynomial models for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 572–581.
7. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [CrossRef]
8. Baudat, G.; Anouar, F. Generalized discriminant analysis using a kernel approach. *Neural Comput.* **2000**, *12*, 2385–2404. [CrossRef]
9. Hyvarinen, A.; Oja, E.; Karhunen, J. *Independent Component Analysis*; Wiley: New York, NY, USA, 2001.
10. Bach, F.R.; Jordan, M.I. Kernel independent component analysis. *J. Mach. Learn. Res.* **2002**, *3*, 1–48.
11. Cox, T.; Cox, M. *Multidimensional Scaling*; Chapman & Hall: London, UK, 1994.
12. Tasci, E.; Jagasia, S.; Zhuge, Y.; Camphausen, K.; Krauze, A.V. GradWise: A Novel Application of a Rank-Based Weighted Hybrid Filter and Embedded Feature Selection Method for Glioma Grading with Clinical and Molecular Characteristics. *Cancers* **2023**, *15*, 4628. [CrossRef]
13. Langley, P. Selection of relevant features in machine learning. In Proceedings of the AAAI Fall Symposium on Relevance, New Orleans, LA, USA, 4–6 November 1994; pp. 1–5.
14. Hahs-Vaughn, D.L.; Lomax, R.G. *Statistical Concepts—A Second Course*; Routledge: London, UK, 2020.
15. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene Selection for Cancer Classification using Support Vector Machines. *Mach. Learn.* **2002**, *46*, 389–422. [CrossRef]
16. Cao, F.; Zhu, Y.Z. Lasso method based on multicollinearity. *Nat. Sci.* **2012**, *11*, 87–90.

17. Zhang, J. Identification and Analysis of Glass Components by Fusing K-Means Clustering and Ridge Regression. *Acad. J. Comput. Inf. Sci.* **2022**, *5*, 30–37.
18. Mawuena, B.; Braeden, K.; Georgine, C.; Wong, A.W.; Fibke, C.; García, H.A.V.; Adu, P.; Levin, A.; Mishra, S.; Sander, B.; et al. An Elastic Net Regression Model for Identifying Long COVID Patients Using Health Administrative Data: A Population-Based Study. *Open Forum Infect. Dis.* **2022**, *9*, ofac640.
19. Reshef, D.N.; Reshef, Y.A.; Finucane, H.K.; Grossman, S.R.; Mcvean, G.; Turnbaugh, P.J.; Lander, E.S.; Mitzenmacher, M.; Sabeti, P.C. Detecting novel associations in large data sets. *Science* **2011**, *334*, 1518–1524. [CrossRef]
20. Zhang, X.; Deng, H.; Xiong, Z.; Liu, Y.; Rao, Y.; Lyu, Y.; Li, Y.; Hou, D.; Li, Y. Secure Routing Strategy Based on Attribute-Based Trust Access Control in Social-Aware Networks. *J. Signal Process. Syst.* **2024**, *96*, 1–16. [CrossRef]
21. Jiang, X.R.; Wen, X.X.; Wu, M.G.; Song, M.; Tu, C. A complex network analysis approach for identifying air traffic congestion based on independent component analysis. *Phys. A Stat. Mech. Its Appl.* **2019**, *523*, 1665–1672. [CrossRef]
22. Pearson, K. Notes on the history of correlation. *Biometrika* **1920**, *13*, 25–45. [CrossRef]
23. Son, S.W.; Bizhani, G.; Christensen, C.; Grassberger, P.; Paczuski, M. Percolation theory on interdependent networks based on epidemic spreading. *Europhys. Lett.* **2012**, *97*, 16006. [CrossRef]
24. Andrzej, B.; Andrzej, J. *Life Time of Correlations and Its Applications*; Wydawnictwo Niezależne: Warsaw, Poland, 2010; pp. 5–21.
25. ACCS. The UNSW-NB15 Dataset [EB/OL]. Available online: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 23 April 2022).
26. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6.
27. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset. *Inf. Secur. J. Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
28. Botes, F.H.; Leenen, L.; Harpe, R. Ant colony induced decision trees for intrusion detection. In Proceedings of the European Conference on Cyber Warfare & Security, Dublin, Ireland, 29–30 June 2017.
29. Suykens, J.; Vandewalle, J.; Moor, B.D. Optimal control by least squares support vector machines. *Neural Netw.* **2001**, *14*, 23–35. [CrossRef] [PubMed]
30. Fleuret, F. Fast binary feature selection with conditional mutual information. *J. Mach. Learn. Res.* **2004**, *5*, 1531–1555.
31. Yang, H.H.; Moody, J. Feature selection based on joint mutual information. In Proceedings of the International ICSC Symposium on Advances in Intelligent Data Analysis, Rochester, NY, USA, 22–25 June 1999; pp. 22–25.
32. Meyer, P.E.; Schretter, C.; Bontempi, G. Information-theoretic feature selection in microarray data using variable complementarity. *IEEE J. Sel. Top. Signal Process.* **2008**, *2*, 261–274. [CrossRef]
33. Peng, H.; Long, F.; Ding, C. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *8*, 1226–1238. [CrossRef] [PubMed]
34. Vinh, N.X.; Zhou, S.; Chan, J.; Bailey, J. Can high-order dependencies improve mutual information based feature selection? *Pattern Recognit.* **2016**, *53*, 46–58. [CrossRef]
35. Abed-Alguni, B.H.; Alawad, N.A.; Al-Betar, M.A.; Paul, D. Opposition-based sine cosine optimizer utilizing refraction learning and variable neighborhood search for feature selection. *Appl. Intell.* **2023**, *53*, 13224–13260. [CrossRef]
36. Fang, Q.; Shen, B.; Xue, J. A new elite opposite sparrow search algorithm-based optimized LightGBM approach for fault diagnosis. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 10473–10491. [CrossRef] [PubMed]
37. Liu, H.; Motoda, H. *Computational Methods of Feature Selection*; Chapman & Hall: London, UK, 2008.
38. Zhang, S.; Lang, Z.Q. Orthogonal least squares based fast feature selection for linear classification. *Pattern Recognit.* **2022**, *123*, 108419. [CrossRef]
39. Wei, L.J. Asymptotic conservativeness and efficiency of Kruskal-Wallis test for k dependent samples. *J. Am. Stat. Assoc.* **1981**, *76*, 1006–1009. [CrossRef]
40. Lichman, M. UCI Machine Learning Repository. Available online: http://archive.ics.uci.edu/ml (accessed on 1 July 2022).