Pattern Recognition 140 (2023) 109566

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A novel relation aware wrapper method for feature selection

Zhaogeng Liu[a], Jielong Yang[a,*], Li Wang[c], Yi Chang[a,b,*]

[a] School of Artificial Intelligence, Jilin University, Changchun 130012, China
[b] Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun 130012, China
[c] School of Virtual Reality, Jilin Animation Institute, Changchun 130012, China

ARTICLE INFO

Article history:
Received 15 April 2022
Revised 6 November 2022
Accepted 27 March 2023
Available online 28 March 2023

Keywords:
Feature selection
Sample relation
Feature relation
Classification

ABSTRACT

Feature selection, aiming at eliminating irrelevant and redundant features, is an important data preprocessing technology for downstream tasks, e.g., classification. With the explosive growth of data in various fields, some data are high-dimensional and contain critical and complex hidden relationships, which brings new challenges to feature selection: i) How to find out the underlying available relationships from the data, and ii) how to use the learned relations to better select features? To deal with these challenges, we propose a novel wrapper feature selection method named Relation Aware Feature Selection Method (ERASE), which can learn and use the underlying sample relations and feature relations for feature selection. Different from existing methods, our method jointly learns sample relationships and feature relationships through a graph of samples and trees of features. Furthermore, it uses the relations to select the optimal feature subset according to the new proposed Relation-based Sequence Floating Selection Strategy. Extensive experimental results on nine datasets from different domains demonstrate that our method achieves the best performance in most cases compared with other feature selection methods, including state-of-the-art wrapper methods.

1. Introduction

Feature selection is one of the most important data processing strategies, which has been widely used in pattern recognition [1], machine learning [2], and data mining [3]. By eliminating irrelevant and redundant features, feature selection preserves a subset of the most salient features and improves the performance of learning algorithms [4]. According to different selection strategies, feature selection algorithms are generally categorized into three categories: embedded, filter, and wrapper methods [5]. Various studies have indicated that wrapper methods can obtain better results than filter and embedded methods in most scenarios [6]. Recently, with the explosive growth of data in various fields, some data are not only high-dimensional but also contain some critical and complex hidden relationships. Even though many wrapper methods achieve better performance than embedded and filter methods, they still can not handle these data and their hidden relations well. This brings some new challenges to feature selection. ***Challenge I:****How to find out the underlying available relationships from the data?* ***Challenge II:****How to use the learned relations to better select features?* In this paper, we propose a novel wrapper feature selection method called Relation Aware Feature Selection Method, abbreviated ERASE (rElation awaRe feAture Selection mEthod), to tackle these challenges.

Although several studies have considered the relations between features [7], the underlying relations between samples are often neglected. However, in many applications, the underlying relations between samples also contain very useful information for learning tasks.

As an illustrative toy example (see Fig. 1), for three glass beads denoted as G_1 (Blue, Triangle), G_2 (Cyan, Sphere), and G_3 (Red, Ellipsoid), the relationships between pairwise samples (G_1, G_2), (G_1, G_3), and (G_2, G_3) are reflected in the two attributes of color and shape. If the task is to identify their colors (shapes), then the first (second) feature should be selected and the relationships of G_1, G_2, and G_3 regarding color (shape) similarity are important. Hence, we need to learn different sample relations for different tasks, which is related to selected features and crucial for feature selection. We note that some recent feature selection methods have also begun to focus on learning the underlying sample relations [8,9]. However, these methods can still be improved from following aspects: i) they are filter methods and, in most cases, have worse performance than wrapper methods; ii) the number of features selected by these filter methods must be provided in advance, and thus they cannot dynamically select a feature subset with different lengths in different iterations; and iii) these methods, only con-

* Corresponding authors.
E-mail addresses: zgliu20@mails.jlu.edu.cn (Z. Liu), JYANG022@e.ntu.edu.sg (J. Yang), wangl_yx@foxmail.com (L. Wang), yichang@jlu.edu.cn (Y. Chang).

https://doi.org/10.1016/j.patcog.2023.109566
0031-3203/

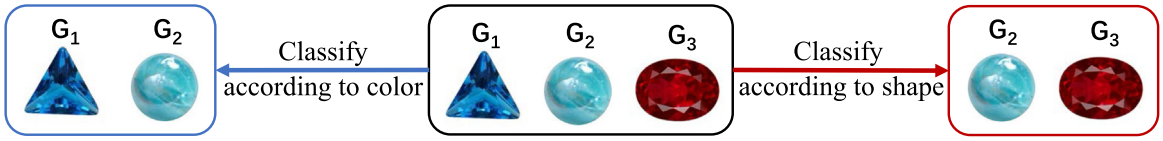

Fig. 1. Toy example of glass bead classification.

sider generating a fixed sample relationship graph based on predefined rules. Consequently, predefined relations of samples will bring a large amount of noise and thus be counterproductive when they are unsuitable for applications.

In view of this, we propose the ERASE that can jointly learn sample and feature relations to solve the feature selection problem better. Given that a graph is a natural way to characterize the similarity between samples, and trees are capable of highlighting the critical hierarchical relations between features, we learn the sample graph and trees of features in ERASE. The contributions are summarized as follows. *i) Our method can simultaneously learn different structures of samples and features. To address* ***Challenge I****, we propose a method that can discover the underlying sample relations using a graph and the underlying feature relations using trees. In other words, our approach can learn the graph of samples and the trees of features together in a unified framework, which is the main difference between our method and the others; ii) We propose a new feature subset selection strategy. To deal with* ***Challenge II****, we first define a new kind of forest-related feature relations based on the learned trees of features. Then, we propose a new Relation-based Sequence Floating Selection Strategy (RSFSS), which can utilize the forest-related feature relations to select the optimal feature subset; iii) Our method achieves better performance than 13 baseline methods on nine benchmark datasets. We use nine datasets with different feature dimensions from multiple areas to validate our proposed method, and it has superior performance than the other methods in terms of both accuracy and dimension reduction.*

The remainder of this paper is organized as follows. Section 2 reviews the related work on feature selection. In Section 3, our proposed method ERASE is described in detail. Section 4 presents the design of the experiments and the analyses of the experimental results. Finally, conclusions and a plan for future work are presented in Section 5.

2. Related works

Feature selection is still a complicated problem, primarily because of the large search space for features [10]. For a dataset with N features, the total number of possible solutions is $2^N - 1$ [11]. Hence, the time complexity of finding an optimal subset of features by exhaustive search is $\mathcal{O}(2^N)$ [12], which is impractical in most situations. In the past two decades, evolutionary computation algorithms have demonstrated powerful search abilities in dealing with complex real-world problems when the search space is huge [13,14]. Evolutionary computation algorithms are inspired by the social behavior of some species in nature, such as breeding [15], hunting [16], and foraging [17]. Recent representative evolutionary computation algorithms include prairie dog optimization algorithm [18], reptile search algorithm [19], butterfly optimization algorithm [20], and forest optimization algorithm [15]. Inspired by evolutionary computation algorithms, many evolutionary computation-based wrapper feature selection methods have been proposed and are at the forefront owing to their strong global search capabilities [21,22]. Alweshah et al. use two new operators with the monarch butterfly optimization algorithm and propose a new feature selection algorithm, which shows effectiveness in terms of classification accuracy and dimension reduction [23]. In [24], a graph model is used to characterize the interactions between features, and the gravitational search algorithm [25] is used to select the optimal feature subset. Based on the hyperbolic transfer function, authors in [26] propose a binary version of the coyote optimization algorithm for classification. However, these methods inevitably retain some limitations of evolutionary computation methods, including uncontrollable random initialization and premature convergence [27,28].

Other methods related to this study include decision-tree-based algorithms. These algorithms have always received extensive attention from researchers in the study of feature selection because of the following two important properties: i) features used to construct a tree are more closely related to each other than to the unused features [29]; ii) the hierarchy of features in a tree reflects their informativeness, and the features closer to the root are more informative [30]. In decision-tree-based algorithms, the importance of a feature can be defined according to the gain values obtained by the feature itself in the node-splitting process [30] for guiding feature selection. In addition, with the extensive research and development of ensemble learning, various ensemble learning algorithms based on decision trees tend to have better feature selection performance than other methods that use only a single decision tree, for example, RF [31], GBDT [32], and XGBoost [33]. When ensemble learning strategies are used to build a unified model with multiple decision trees, the importance of a feature is commonly defined as the average or sum of the gain values caused by the feature across all trees [34]. Currently, there are two main ensemble learning schemes. One is bagging, in which the representative algorithm is random forest [31]. The other is boosting, in which the representative algorithms are GBDT [32] and XGBoost [33]. However, these methods can only organize features into a tree structure, ignoring the use of clear and specific feature relationships to guide feature selection to achieve better results.

3. Methodology

Learning and utilizing sample relations and feature relations are essential in feature selection. Graphs are a natural and appropriate way to represent the complex relational information between samples. Trees are more suitable for characterizing the important hierarchical relations between features. In this section, we present our new feature selection method ERASE, which can simultaneously learn two different structures for selecting the optimal feature subset. Before introducing the details of ERASE, we first give the notations and problem statement as follows:

Notations: We use boldfaced characters to represent sequences, vectors and matrices. Suppose that $\mathbf{M}$ is a matrix, then $\mathbf{M}[m,\cdot]$, $\mathbf{M}[\cdot,n]$, and $\mathbf{M}[m,n]$ denote its mth row, nth column, and (m,n)th element, respectively. The sequence $(x_1,\ldots,x_N)$ is abbreviated as $[x_i]_{i=1}^N$ or $[x_i]$ if the index set that i runs over is clear from the context. The ith element of a sequence $\mathbf{x}$ is $\mathbf{x}[i]$. Consider a dataset with M samples $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_M]^\top$. For each $m\in\{1,2,\ldots,M\}$, $\mathbf{x}_m\in\mathbb{R}^N$ has N features. The labels of these samples are $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,\ldots\mathbf{y}_M]^\top$, where each $\mathbf{y}_m\in\{1,2,\ldots,C\}$ with C being the number of classes. The important notations throughout the paper are summarized in Table 1.

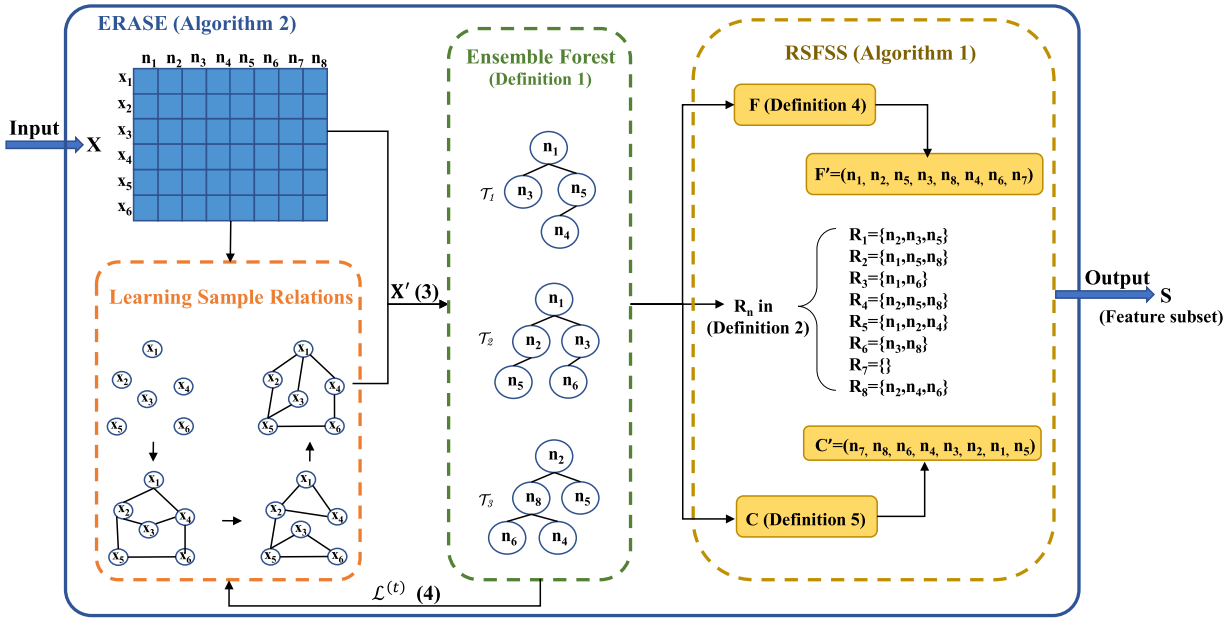

Fig. 2. Toy example of using ERASE for feature selection of input $\mathbf{X} \in \mathbb{R}^{6\times 8}$. We use $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_6$ and $\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_8$ to denote the samples and features, respectively. We learn the sample relations using the method in Section 3.1. We learn three trees, $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$, using the method in Section 3.2 and regard them as the Ensemble Forest. Then, the Ensemble Forest is input into RSFSS (see Section 1) to select the feature subset **S**. In RSFSS, there are three important components, i.e., **F**, **C** and $\mathbf{R}_n \in \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_8\}$, and they are defined in Section 3.2.

Table 1
Summary of important notations.

Notation	Description
$\mathbf{X}$	Input dataset
$\mathbf{Y}$	Labels
$\mathbf{A}$	Adjacency matrix of the sample graph
$\mathbf{E_F}$	Ensemble forest of features
$\mathbf{R}_n$	Set of forest-related features of the nth feature
$\mathbf{F}$	Sequence of feature importance
$\mathbf{C}$	Sequence of feature redundancy
$\mathbf{S}$	Feature subset

Problem Statement: **Feature selection by learning an unknown graph of samples and trees of features.** Given feature matrix $\mathbf{X}$, we aim to solve the following problem: how to learn the unknown graph of samples and unknown trees of features, and to utilize them to select the optimal feature subset $\mathbf{S}$?

In our method, we first jointly learn the graph structure of the sample relations and tree structure of the feature relations. Then, we use the proposed Relation-based Sequence Floating Selection Strategy (RSFSS) to deal with the forest-related feature relations and select the optimal feature subset. We provide a toy example in Fig. 2. For the $\mathbf{X} \in \mathbb{R}^{6\times 8}$ in Fig. 2, we use $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_6$ and $\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_8$ to denote its samples and features, respectively. Then we jointly learn the graph of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_6$ and the trees of $\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_8$ to obtain the Ensemble Forest ($\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$). Finally, the Ensemble Forest is input into RSFSS to select the feature subset $\mathbf{S}$.

3.1. Learn and utilize sample relations

In ERASE, we consider the similarity relationship between samples, and thus construct a graph whose vertices represent samples and edges denote the similarity of vertices. Because the graph of the samples is usually unknown in various applications [24,26], we aim to learn the adjacency matrix $\mathbf{A}$ of the sample graph in this section.

Specifically, we process $\mathbf{X}$ by using (1) and obtain $\tilde{\mathbf{X}}$.

$$\tilde{\mathbf{X}} = \mathbf{G}\mathbf{J} \odot \mathbf{X}, \tag{1}$$

where $\mathbf{G} \in \mathbb{R}^{M\times 1}$ is a trainable matrix, $\mathbf{J} \in \mathbb{R}^{1\times N}$ is an all-one matrix, and $\odot$ represents the element-wise product.

Further, we define the sample relationship similarity matrix $\mathbf{H}$ by

$$\mathbf{H} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\top}. \tag{2}$$

Finally, to highlight the important sample relationships, we set the top η values in each row of $\mathbf{H}$ to one and the others to zero to sparsify $\mathbf{H}$. The sparsified matrix $\mathbf{H}$ is adjacency matrix $\mathbf{A}$, which is used to characterize the sample relationships.

Then we utilize $\mathbf{A}$ in the following formula to incorporate sample relations information into $\mathbf{X}'$:

$$\mathbf{X}' = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X}, \tag{3}$$

where $\mathbf{D}$ denotes an $M \times M$ diagonal matrix with $\mathbf{D}[m,m] \triangleq \sum_{p=1}^{M} \mathbf{A}[m,p] + 1$ for each $m \in \{1, 2, \ldots, M\}$, and $\mathbf{I}$ denotes the identity matrix.

3.2. Learn and utilize feature relations

In this section, we introduce how to use $\mathbf{X}'$ to guide feature relation learning and use feature relations to solve the problem of feature selection.

First, we use XGBoost to generate T trees, which is used to characterize nonlinear interactions between features [33]. Specifically, for $t \in \{1, 2, \ldots, T\}$, we denote the tth tree as $\mathcal{T}_t$. We denote $l\left(y_i, \hat{y}_i^{(t-1)}\right)$ as a differentiable convex loss function which measures the difference between the true label y_i and the $(t-1)$th prediction $\hat{y}_i^{(t-1)}$, and we use $\mathbf{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_M]^{\top}$ to generate the tth tree by minimizing the following objective:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{M}\left[g_i\mathcal{T}_t(\mathbf{x}'_i) + \frac{1}{2}h_i\mathcal{T}_t^2(\mathbf{x}'_i)\right] + \gamma P_t + \frac{1}{2}\lambda\sum_{j=1}^{P_t} w_j^2, \tag{4}$$

where γ and λ are two hyperparameters, P_t is the number of leaves in the tth tree, w_j is the weight of the jth leaf, $g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right)$ and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right)$ are first and second order gradient statistics of the loss function with $\hat{y}_i^{(t-1)} = \sum_{k=1}^{t-1} \mathcal{T}_k(\mathbf{x}_i')$, and for $t \in \{1, 2, \ldots, T\}$, $\mathcal{T}_t(\mathbf{x}_i') = \mathbf{W}_{q(\mathbf{x}_i')}$ with $q : \mathbb{R}^M \to P_t$ and $\mathbf{W} \in \mathbb{R}^{P_t}$.

To jointly learn the graph structure of the samples and tree structure of the features, we choose a relatively simple but effective evolutionary computation-based optimization method, the forest optimization algorithm (FOA) [15]. Specifically, in each iteration of the FOA, the graph guides the generation of trees using (3), and the trees are used to update the sample graphs by optimizing (4). In this manner, both sample graph and feature trees are learned under the same objective (4). For simplicity, this optimization process is denoted as FOA($\mathbf{G}, \mathbf{X}, \eta$) in the following sections.

Furthermore, we propose a new feature subset search algorithm that uses feature relations to select the features. Before introducing our new algorithm, we first provide the required definitions.

Definition 1 (Ensemble Forest)**.** In this paper, we define the sequence of T trees generated by minimizing (4) as an Ensemble Forest, denoted as $\mathbf{E_F} \triangleq (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T)$, where $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T$ represent the trees whose indices are $1, 2, \ldots, T$, respectively.

Definition 2 (Forest-Related Features)**.** For trees in the Ensemble Forest, we believe that edge information can help in feature selection. In this study, we regard the features of two nodes that are directly connected in each tree as related features. We define all related features of a feature obtained by iterating over $[\mathcal{T}_t]_{t=1}^T$ as its forest-related features. For $n \in \{1, 2, \ldots, N\}$, the set of forest-related features of the nth feature is denoted as $\mathbf{R}_n$.

Definition 3 (Leaf Splitting Gains)**.** For $t \in \{1, 2, \ldots, T\}$ and $\mathcal{T}_t = \mathbf{E_F}[t]$, let $L_j^{(t)}$ denote the jth leaf of $\mathcal{T}_t$ and $\mathcal{I}_j^{(t)} \triangleq \{i : q(\mathbf{x}_i') = L_j^{(t)}\}$ be the sample index set of $L_j^{(t)}$. Assume that $\mathcal{I}_j^{(t)} = \mathcal{I}_L \cup \mathcal{I}_R$, where $\mathcal{I}_L$ and $\mathcal{I}_R$ are the sample index sets of the left and right child nodes, respectively, after splitting $L_j^{(t)}$. Then, the gain value of splitting $L_j^{(t)}$ is

$$G_j^{(t)} = \frac{\left(\sum_{i\in\mathcal{I}_L} g_i\right)^2}{\sum_{i\in\mathcal{I}_L} h_i + \lambda} + \frac{\left(\sum_{i\in\mathcal{I}_R} g_i\right)^2}{\sum_{i\in\mathcal{I}_R} h_i + \lambda} - \frac{\left(\sum_{i\in\mathcal{I}_j^{(t)}} g_i\right)^2}{\sum_{i\in\mathcal{I}_j^{(t)}} h_i + \lambda}. \tag{5}$$

Definition 4 (Feature Importance of Ensemble Forest)**.** For $n \in \{1, 2, \ldots, N\}$, if the nth feature is used to split the leaves in the Ensemble Forest, then its importance is

$$f_n = \sum_{t=1}^{T} \sum_{j=1}^{P_t} G_j^{(t)}, \tag{6}$$

where P_t denotes the total number of leaves in $\mathcal{T}_t$. For the features that are not used to split any leaf in the Ensemble Forest, their importance would be zero by default. Then, we denote the feature importance sequence as $\mathbf{F} \triangleq (f_1, f_2, \ldots, f_N)$.

Definition 5 (Redundant Features)**.** Some features in the Ensemble Forest are rarely used to split leaves. We regard the number of feature splitting leaves as an important criterion for judging redundant features, and we define a sequence $\mathbf{C} \triangleq (c_1, c_2, \ldots, c_N)$ to store the redundancy for each feature, where c_n $(n \in \{1, 2, \ldots, N\})$ indicates the number of times that the nth feature is used to split the leaves in the Ensemble Forest. For example, if the nth feature is used 10 times to split the leaves in the Ensemble Forest, then $c_n = 10$. The smaller the value of c_n, the more likely it is that the nth feature is redundant.

Next, we propose a new Relation-based Sequence Floating Selection Strategy (RSFSS) to search the feature subset by minimizing the following fitness function.

$$\text{Fitness}(\mathbf{S}) = \alpha E_r(\mathbf{S}) + \beta \frac{|\mathbf{S}|}{N}, \tag{7}$$

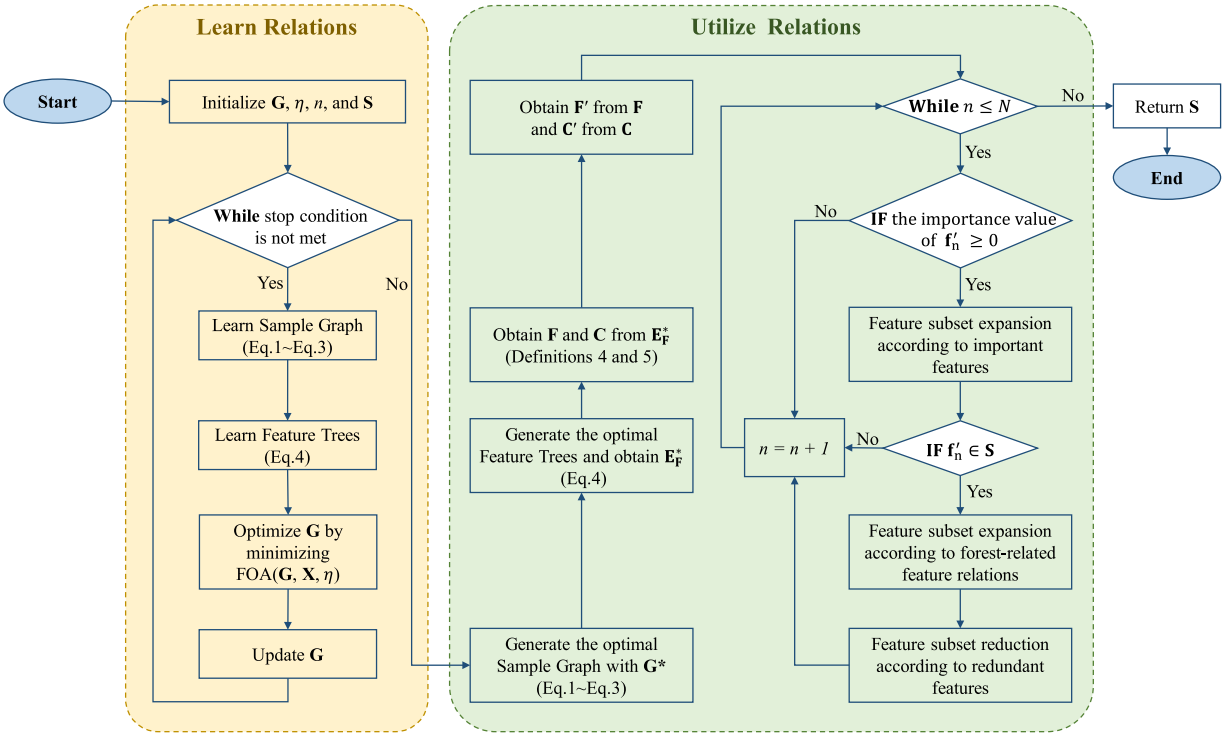

Fig. 3. Flowchart of the proposed ERASE.

Table 2
Brief descriptions of baseline methods.

Algorithm	Description
VCOA	Coyote optimization algorithm using a v-shaped transfer function for feature selection [26]
BMBO	Monarch butterfly optimization algorithm for feature selection [23]
SBOA	Butterfly optimization algorithm for feature selection [35]
HGSA	Gravitational search algorithm for feature selection [24]
Rc-BBFA	Firefly algorithm for feature selection [42]
FSFOA	Forest optimization algorithm for feature selection [43]
BALO	Ant lion algorithm for feature selection [44]
BGWO2	Grey wolf optimization algorithm for feature selection [45]
BPSO	Particle swarm optimization for feature selection [46]
GA	Genetic algorithm for feature selection [47]
GBDT	Gradient boosting decision tree for feature selection [32]
XGBoost	XGBoost for feature selection [33]
RF	Random forest for feature selection [31]

Table 3
Details of the experimental datasets.

Dataset	# Samples	# Features	# Classes	Type
Tic-tac-toe	958	9	2	Game
CongressEW	435	16	2	Social
SpectEW	267	22	2	Medical
Ionosphere	351	34	2	Physical
KrvskpEW	3196	36	2	Game
Pubmed	19,717	500	3	Citation
CNAE-9	1080	856	9	Business
Cora	2708	1433	7	Citation
Citeseer	3327	3703	6	Citation

where $\mathbf{S}$ denotes the feature subset, $|\mathbf{S}|$ denotes the number of features in $\mathbf{S}$, $E_r(\mathbf{S})$ denotes the error rate (i.e., 1 - accuracy) of a given classifier for $\mathbf{S}$, and $\alpha \in [0,1]$ and $\beta = 1-\alpha$ are the two hyperparameters used to control the classification quality and dimension reduction. The fitness function is similar to that in [24,35], and following these works, we use a k-Nearest Neighbors (k-NN) classifier to compute the error rate in (7).

In RSFSS, we first initialize $\mathbf{S} = \varnothing$. Recall that $\mathbf{F}$ and $\mathbf{C}$ are defined in Definitions 4 and 5, respectively. Second, we sort the elements of $\mathbf{F}$ in descending order and obtain a sequence of features $\mathbf{F}' = (\mathbf{f}'_1, \mathbf{f}'_2, \ldots, \mathbf{f}'_N)$. Third, we sort the elements of $\mathbf{C}$ in ascending order and obtain a sequence of features $\mathbf{C}' = (\mathbf{c}'_1, \mathbf{c}'_2, \ldots, \mathbf{c}'_N)$. The feature subset selection process in RSFSS is composed of three parts: feature subset expansion according to important features, feature subset expansion according to forest-related feature relations, and feature subset reduction according to redundant features.

1. **Feature subset expansion according to important features:** In this part, for $n \in \{1,2,\ldots,N\}$, we sequentially select $\mathbf{f}'_n \in \mathbf{F}'$ such that the Fitness$(\mathbf{S} \cup \{\mathbf{f}'_n\})$ iteratively decreases.
2. **Feature subset expansion according to forest-related feature relations:** In this part, for $n \in \{1,2,\ldots,N\}$ and $\mathbf{f}'_n \in \mathbf{F}'$, we first obtain the set of all the forest-related features of $\mathbf{f}'_n$ and denote this set as

$$\mathbf{R}'_n = \mathcal{R}(\mathbf{f}'_n), \tag{8}$$

where $\mathcal{R}: \mathbf{F}' \to \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N\}$ is a bijective function, and $\mathbf{R}_n \in \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_N\}$ is the set of forest-related features of the nth feature (see Definition 2). Next, if $\mathbf{f}'_n \in \mathbf{S}$, then each $\mathbf{r} \in \mathbf{R}'_n$ is sequentially selected for $\mathbf{S}$ such that Fitness$(\mathbf{S} \cup \{\mathbf{r}\})$ iteratively decreases.
3. **Feature subset reduction according to redundant features:** In this part, after feature subset expansion, we remove the inappropriate features from $\mathbf{S}$ according to the order of features in $\mathbf{C}'$ to reduce the number of features in the feature subset and further decrease the fitness function.

The pseudocode of RSFSS is given in Algorithm 1.

Algorithm 1: RSFSS.

Input : $\mathbf{E_F}$ `// Ensemble Forest, which is defined in Definition 1`
Output: The selected feature subset $\mathbf{S}$

Initialize $\mathbf{S} = \varnothing$;
Initialize variable $V = 10$; `//` V `stores the fitness value given in (7)`
Obtain $\mathbf{F}$ and $\mathbf{C}$ using $\mathbf{E_F}$ (see Definition 5 and Definition 5);
Sort the elements of $\mathbf{F}$ in descending order and obtain $\mathbf{F}' = (\mathbf{f}'_1, \mathbf{f}'_2, \ldots, \mathbf{f}'_N)$;
Sort the elements of $\mathbf{C}$ in ascending order and obtain $\mathbf{C}' = (\mathbf{c}'_1, \mathbf{c}'_2, \ldots, \mathbf{c}'_N)$;
for $1 \le n \le N$ **do**
  if *the importance value of* $\mathbf{f}'_n$ *in* $\mathbf{F}$ *is greater than 0* **then**
    `// Feature subset expansion according to important features`
    if Fitness$(\mathbf{S} \cup \{\mathbf{f}'_n\}) < V$ **then**
      $V = $ Fitness$(\mathbf{S} \cup \{\mathbf{f}'_n\})$, $\mathbf{S} = \mathbf{S} \cup \{\mathbf{f}'_n\}$;
    if $\mathbf{f}'_n \in \mathbf{S}$ **then**
      `// Feature subset expansion according to forest-related feature relations`
      Find the set $\mathbf{R}'_n$ containing all the forest-related features of $\mathbf{f}'_n$ according to (8);
      foreach $\mathbf{r} \in \mathbf{R}'_n$ **do**
        if Fitness$(\mathbf{S} \cup \{\mathbf{r}\}) < V$ **then**
          $V = $ Fitness$(\mathbf{S} \cup \{\mathbf{r}\})$, $\mathbf{S} = \mathbf{S} \cup \{\mathbf{r}\}$;
      `// Feature subset reduction according to redundant features`
      for $1 \le v \le N$ **do**
        if *the redundant value of* $\mathbf{c}'_v$ *in* $\mathbf{C}$ *is greater than 0* **then**
          if Fitness$(\mathbf{S} - \{\mathbf{c}'_v\}) < V$ **then**
            $V = $ Fitness$(\mathbf{S} - \{\mathbf{c}'_v\})$, $\mathbf{S} = \mathbf{S} - \{\mathbf{c}'_v\}$;
return $\mathbf{S}$;

Table 4
Values of main hyperparameters.

Hyperparameters	Tic-tac-toe	CongressEW	SpectEW	Ionosphere	KrvskpEW	Pubmed	CNAE-9	Cora	Citeseer
η in Section 3.1	10	3	10	3	2	8	9	9	10
T in Section 3.2	100								
Max tree depth	5								
α in (7)	0.99								
β in (7)	0.01								

Table 5
Acc and Dr results of the comparative algorithms in percentage for using 1NN in (7). Each value is the average result of 20 Monte Carlo experiments. Each number in brackets indicates the ranking and Avg. reports the average ranking of each algorithm. The top three results are in bold.

	Metrics	Tic-tac-toe	CongressEW	SpectEW	Ionosphere	KrvskpEW	Pubmed	CNAE-9	Cora	Citeseer	Avg.
ERASE	Acc	**92.22 (1)**	**97.02 (2)**	**87.16 (1)**	**95.66 (1)**	**98.12 (1)**	**83.97 (1)**	**93.83 (1)**	**74.85 (1)**	**75.55 (1)**	**1.11**
	Dr	40.0 (4)	78.75 (8)	**79.55 (2)**	**82.94 (2)**	51.39 (12)	**93.04 (2)**	**95.25 (1)**	**94.43 (1)**	**97.56 (1)**	**3.67**
VCOA	Acc	81.08 (7)	96.49 (6)	86.42 (5)	**95.66 (1)**	97.27 (4)	76.47 (6)	**91.48 (3)**	**67.88 (2)**	48.55 (5)	**4.33**
	Dr	37.78 (6)	85.62 (5)	74.09 (4)	73.82 (4)	67.78 (5)	31.48 (13)	29.53 (12)	12.64 (13)	75.75 (7)	7.67
BMBO	Acc	78.75 (11)	96.03 (10)	84.2 (9)	91.6 (11)	95.02 (9)	73.59 (11)	82.9 (12)	45.87 (14)	26.03 (13)	11.11
	Dr	35.56 (7)	68.75 (11)	53.18 (14)	60.0 (13)	54.44 (10)	50.08 (9)	52.51 (7)	53.85 (6)	53.41 (9)	9.56
SBOA	Acc	81.6 (4)	96.79 (4)	84.81 (8)	92.74 (8)	95.88 (7)	72.54 (12)	83.55 (10)	46.13 (12)	26.96 (10)	8.33
	Dr	35.56 (7)	71.25 (10)	65.0 (9)	64.71 (11)	52.5 (11)	50.54 (7)	50.09 (8)	51.62 (8)	51.37 (11)	9.11
HGSA	Acc	81.11 (5)	**97.25 (1)**	84.94 (7)	92.74 (8)	95.66 (8)	72.37 (13)	82.99 (11)	45.9 (13)	26.02 (14)	8.89
	Dr	32.22 (14)	55.62 (14)	57.73 (13)	59.71 (14)	43.89 (14)	49.56 (12)	49.82 (9)	49.83 (11)	50.01 (14)	12.78
Rc-BBFA	Acc	**81.84 (3)**	96.26 (8)	**86.79 (2)**	94.72 (4)	**97.32 (3)**	74.9 (9)	89.75 (5)	53.63 (9)	31.02 (9)	5.78
	Dr	34.44 (10)	86.88 (4)	65.45 (8)	73.82 (4)	62.78 (6)	50.5 (8)	49.36 (10)	50.1 (10)	51.3 (12)	8.0
FSFOA	Acc	**81.87 (2)**	**96.87 (3)**	**86.79 (2)**	**95.47 (3)**	**97.97 (2)**	76.11 (7)	90.46 (4)	**66.83 (3)**	49.1 (4)	**3.33**
	Dr	33.33 (13)	64.38 (13)	70.91 (5)	73.24 (6)	55.56 (8)	31.32 (14)	22.37 (13)	24.92 (12)	75.16 (8)	10.22
BALO	Acc	80.28 (9)	96.11 (9)	82.59 (11)	93.02 (7)	95.02 (9)	74.37 (10)	89.44 (7)	65.47 (4)	**49.36 (3)**	7.67
	Dr	34.44 (10)	80.62 (7)	**84.09 (1)**	70.29 (8)	59.17 (7)	50.02 (10)	21.0 (14)	5.64 (14)	82.16 (5)	8.44
BGWO2	Acc	79.83 (10)	96.03 (10)	86.54 (4)	94.53 (5)	96.78 (5)	**78.62 (3)**	**92.13 (2)**	64.18 (5)	**61.13 (2)**	5.11
	Dr	40.0 (4)	**89.38 (3)**	**74.55 (3)**	**80.88 (3)**	76.67 (4)	85.66 (4)	78.46 (5)	81.13 (4)	81.83 (6)	**4.0**
BPSO	Acc	80.87 (8)	96.56 (5)	86.3 (6)	94.34 (6)	96.38 (6)	75.52 (8)	89.69 (6)	56.77 (8)	33.87 (8)	6.78
	Dr	35.56 (7)	81.25 (6)	69.09 (6)	66.47 (10)	55.28 (9)	50.86 (6)	53.69 (6)	52.5 (7)	52.68 (10)	7.44
GA	Acc	81.11 (5)	96.34 (7)	84.2 (9)	91.79 (10)	94.82 (11)	72.3 (14)	82.5 (13)	46.22 (11)	26.47 (11)	10.11
	Dr	34.44 (10)	67.5 (12)	67.73 (7)	62.94 (12)	47.78 (13)	49.86 (11)	49.35 (11)	50.77 (9)	50.23 (13)	10.89
GBDT	Acc	67.57 (13)	91.76 (14)	75.56 (14)	87.08 (14)	91.89 (12)	**82.01 (2)**	86.6 (9)	58.07 (6)	38.61 (7)	10.11
	Dr	**58.89 (2)**	**93.12 (1)**	65.0 (9)	**86.18 (1)**	**83.33 (2)**	**93.18 (1)**	**89.99 (3)**	**84.3 (3)**	**86.17 (3)**	**2.78**
XGBoost	Acc	72.36 (12)	92.37 (13)	76.54 (12)	88.3 (12)	89.95 (13)	76.92 (4)	87.47 (8)	48.55 (10)	26.08 (12)	10.67
	Dr	**56.67 (3)**	**93.12 (1)**	63.64 (11)	72.94 (7)	**83.33 (2)**	79.28 (5)	88.49 (4)	73.99 (5)	84.13 (4)	4.67
RF	Acc	64.72 (14)	94.2 (12)	76.42 (13)	88.11 (13)	89.05 (14)	76.78 (5)	82.31 (14)	56.95 (7)	40.78 (6)	10.89
	Dr	**71.11 (1)**	73.75 (9)	61.82 (12)	69.71 (9)	**85.28 (1)**	**87.84 (3)**	**90.72 (2)**	**89.24 (2)**	**88.95 (2)**	4.56

Table 6
Acc and Dr results of the comparative algorithms in percentage for using 5NN in (7). Each value is the average result of 20 Monte Carlo experiments. Each number in brackets indicates the ranking and Avg. reports the average ranking of each algorithm. The top three results are in bold.

	Metrics	Tic-tac-toe	CongressEW	SpectEW	Ionosphere	KrvskpEW	Pubmed	CNAE-9	Cora	Citeseer	Avg.
ERASE	Acc	**90.83 (1)**	**97.25 (1)**	**87.9 (1)**	**93.96 (1)**	**97.63 (2)**	**86.19 (1)**	**92.07 (1)**	**74.37 (1)**	**74.17 (1)**	**1.11**
	Dr	48.89 (8)	78.75 (7)	**76.82 (1)**	**84.71 (2)**	65.28 (6)	**94.3 (1)**	**95.86 (1)**	**95.46 (1)**	**98.39 (1)**	**3.11**
VCOA	Acc	80.24 (5)	95.73 (11)	86.42 (7)	92.83 (4)	**97.39 (3)**	78.72 (7)	**91.82 (2)**	**71.84 (2)**	49.69 (4)	5.0
	Dr	47.78 (10)	**92.5 (3)**	58.18 (11)	81.18 (5)	70.83 (5)	53.02 (7)	23.71 (14)	11.65 (13)	83.95 (7)	8.33
BMBO	Acc	77.95 (11)	96.11 (7)	85.68 (10)	89.62 (8)	95.97 (7)	76.38 (11)	79.1 (14)	45.15 (14)	10.63 (10)	10.22
	Dr	37.78 (12)	70.62 (12)	58.18 (11)	68.24 (11)	51.11 (12)	53.76 (6)	53.35 (7)	53.4 (6)	52.82 (10)	9.67
SBOA	Acc	79.97 (7)	96.56 (5)	87.04 (4)	89.25 (10)	95.48 (9)	76.38 (11)	82.44 (11)	51.09 (11)	10.27 (11)	8.78
	Dr	53.33 (4)	71.88 (10)	64.09 (5)	66.18 (13)	55.28 (9)	51.52 (10)	50.32 (8)	50.86 (7)	53.11 (9)	8.33
HGSA	Acc	**81.15 (3)**	**96.79 (2)**	86.67 (6)	89.43 (9)	95.84 (8)	76.01 (14)	82.41 (12)	50.02 (13)	10.26 (12)	8.78
	Dr	33.33 (13)	59.38 (14)	50.45 (14)	61.18 (14)	42.22 (14)	49.3 (12)	49.79 (10)	49.67 (9)	50.34 (14)	12.67
Rc-BBFA	Acc	80.35 (4)	96.64 (4)	86.42 (7)	92.26 (5)	97.23 (5)	78.41 (8)	89.17 (6)	60.01 (7)	11.48 (9)	6.11
	Dr	50.0 (7)	86.25 (5)	67.27 (4)	74.71 (8)	63.61 (7)	50.76 (11)	49.61 (11)	49.41 (10)	51.5 (12)	8.33
FSFOA	Acc	**82.88 (2)**	**96.79 (2)**	**87.9 (1)**	**93.11 (3)**	**97.9 (1)**	78.94 (6)	**90.77 (3)**	**71.78 (3)**	46.98 (5)	**2.89**
	Dr	7.78 (14)	70.62 (12)	60.0 (8)	78.82 (6)	43.89 (13)	31.82 (14)	27.87 (13)	21.94 (12)	81.93 (8)	11.11
BALO	Acc	78.82 (9)	95.8 (10)	82.72 (11)	91.13 (7)	95.41 (10)	77.73 (10)	89.17 (6)	70.2 (4)	**54.86 (3)**	7.78
	Dr	48.89 (8)	77.5 (8)	**75.0 (2)**	82.94 (4)	52.78 (10)	47.2 (13)	33.95 (12)	4.64 (14)	**88.94 (2)**	8.11
BGWO2	Acc	78.44 (10)	96.41 (6)	**87.41 (3)**	**93.77 (2)**	96.67 (6)	**81.15 (3)**	89.38 (5)	64.31 (5)	**63.2 (2)**	**4.67**
	Dr	53.33 (4)	88.75 (4)	**71.36 (3)**	**83.53 (3)**	77.78 (4)	86.06 (4)	79.05 (5)	80.01 (4)	84.85 (5)	**4.0**
BPSO	Acc	79.79 (8)	95.88 (9)	86.79 (5)	91.89 (6)	97.27 (4)	78.39 (9)	90.0 (4)	61.02 (6)	13.83 (8)	6.56
	Dr	52.22 (6)	86.25 (5)	58.64 (10)	76.18 (7)	63.06 (8)	51.72 (9)	53.7 (6)	50.84 (8)	52.78 (11)	7.78
GA	Acc	80.1 (6)	96.11 (7)	86.3 (9)	88.96 (11)	94.56 (11)	76.17 (13)	82.35 (13)	50.48 (12)	10.08 (13)	10.56
	Dr	47.78 (10)	71.88 (10)	59.55 (9)	67.06 (12)	52.22 (11)	51.8 (8)	49.92 (9)	49.16 (11)	51.23 (13)	10.33
GBDT	Acc	70.76 (13)	94.96 (13)	80.25 (12)	86.98 (13)	94.26 (12)	**84.42 (2)**	87.07 (9)	56.84 (10)	17.66 (7)	10.11
	Dr	**58.89 (2)**	**93.12 (1)**	63.64 (6)	**85.88 (1)**	**83.33 (2)**	**93.1 (2)**	**90.01 (3)**	**84.37 (3)**	86.21 (4)	**2.67**
XGBoost	Acc	73.65 (12)	95.57 (12)	77.9 (14)	86.23 (14)	93.55 (13)	80.36 (5)	87.41 (8)	57.38 (9)	9.74 (14)	11.22
	Dr	**56.67 (3)**	**93.12 (1)**	63.64 (6)	72.94 (9)	**83.33 (2)**	79.28 (5)	88.49 (4)	73.99 (5)	84.13 (6)	4.56
RF	Acc	68.85 (14)	94.35 (14)	79.88 (13)	87.17 (12)	90.55 (14)	80.63 (4)	82.5 (10)	59.69 (8)	20.04 (6)	10.56
	Dr	**71.11 (1)**	74.38 (9)	58.18 (11)	70.88 (10)	**83.89 (1)**	**87.8 (3)**	**90.98 (2)**	**88.95 (2)**	**88.88 (3)**	4.67

3.3. Computational complexity analysis

In this section, we analyze the computational complexity of ERASE. The computational complexity of generating an Ensemble Forest is $\mathcal{O}(MNTD)$, where M denotes the number of samples, N represents the number of features, T indicates the number of trees in the Ensemble Forest, and D refers to the maximum depth of trees in the Ensemble Forest. The computational complexity of learning the sample relationship graph is $\mathcal{O}(ZMNTD)$, where Z denotes the total number of individuals generated in FOA$(\mathbf{G}, \mathbf{X}, \eta)$. When selecting a feature subset using RSFSS, we first need to generate an Ensemble Forest using the learned sample relationship graph, whose computational complexity is $\mathcal{O}(MNTD)$. Let $A = \min\{2^D T - T, N\}$ be the maximum number of features whose feature importance scores are greater than zero. Then, we compute the fitness function value using k-NN,

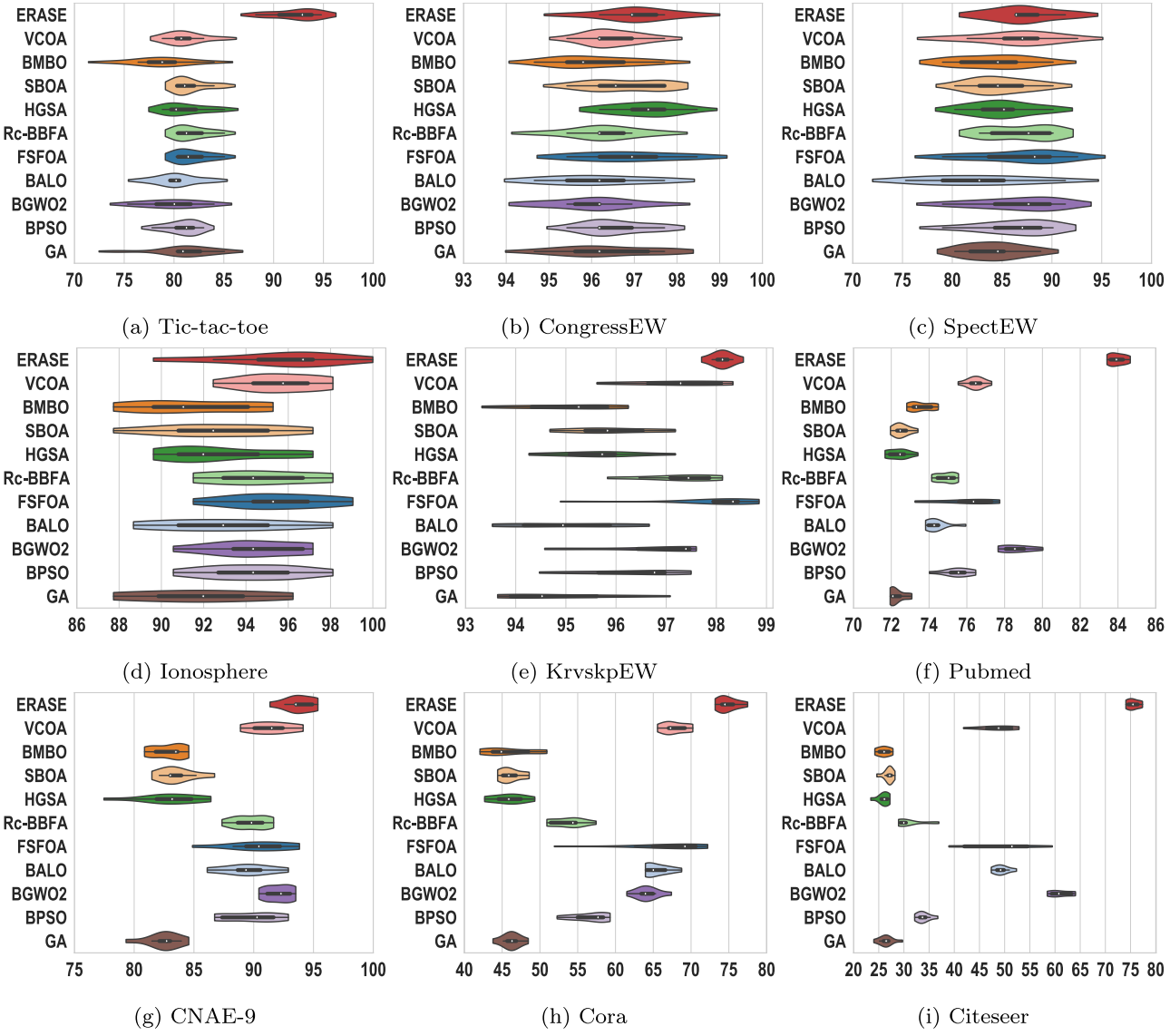

Fig. 4. Violin plot of Acc when 1NN is used in (7). For each sub-plot from (a)∼(i), the horizontal axis indicates the percentage value of Acc, and the vertical axis represents the algorithm. In each violin plot, the white dot denotes the median, and the left and right edges of the black box indicate the 25th and 75th percentiles, respectively.

which can be optimized using a multidimensional binary search tree (i.e., K-D Tree) [36], and the time complexity is reduced to $\mathcal{O}(AMlogM)$ [36,37]. Therefore, the computational complexity of RSFSS is $\mathcal{O}(MNTD + A^3MlogM)$. Combining $\mathcal{O}(ZMNTD)$ and $\mathcal{O}(MNTD + A^3MlogM)$, we conclude that the time complexity of ERASE is $\mathcal{O}(ZMNTD + MNTD + A^3MlogM)$, and is approximately $\max\{\mathcal{O}((Z+1)MNTD), \mathcal{O}(A^3MlogM)\}$.

3.4. Pseudocode of ERASE

We provide the pseudocode of ERASE, shown in Algorithm 2. The intuitive and detailed process of ERASE is shown in Fig. 3.

Algorithm 2: ERASE.

Input : $\mathbf{X}$
Output: The selected feature subset $\mathbf{S}$
Initialize $\mathbf{G}$ and η;
$\mathbf{G}^* \leftarrow \text{FOA}(\mathbf{G}, \mathbf{X}, \eta)$; `// Optimize` $\mathbf{G}$
$\tilde{\mathbf{X}}^* \leftarrow \mathbf{G}^*\mathbf{J} \odot \mathbf{X}$; `// See (1)`
Generate $\mathbf{X}'^*$ using $\tilde{\mathbf{X}}^*$ and η according to (2) and (3);
Generate $\mathbf{E}_\mathbf{F}^*$ using $\mathbf{X}'^*$ according to (4) and Definition 1;
$\mathbf{S} \leftarrow \text{RSFSS}(\mathbf{E}_\mathbf{F}^*)$; `// Input` $\mathbf{E}_\mathbf{F}^*$ `into Algorithm 1`
return $\mathbf{S}$;

4. Experiments

In the experiments, we evaluate the performance of the proposed ERASE and 13 baseline methods on nine datasets.

4.1. Experimental settings

Baselines: We choose 13 methods as baselines and briefly introduce them in Table 2.

Datasets: Nine datasets with different feature dimensions are used to validate the proposed method. These datasets are Pubmed from [38], Cora from [39], Citeseer from [40], and six datasets available in the UCI Machine Learning Repository [41], namely Tic-tac-toe, CongressEW, SpectEW, Ionosphere, KrvskpEW, and CNAE-9. A brief description of the datasets is provided in Table 3. These

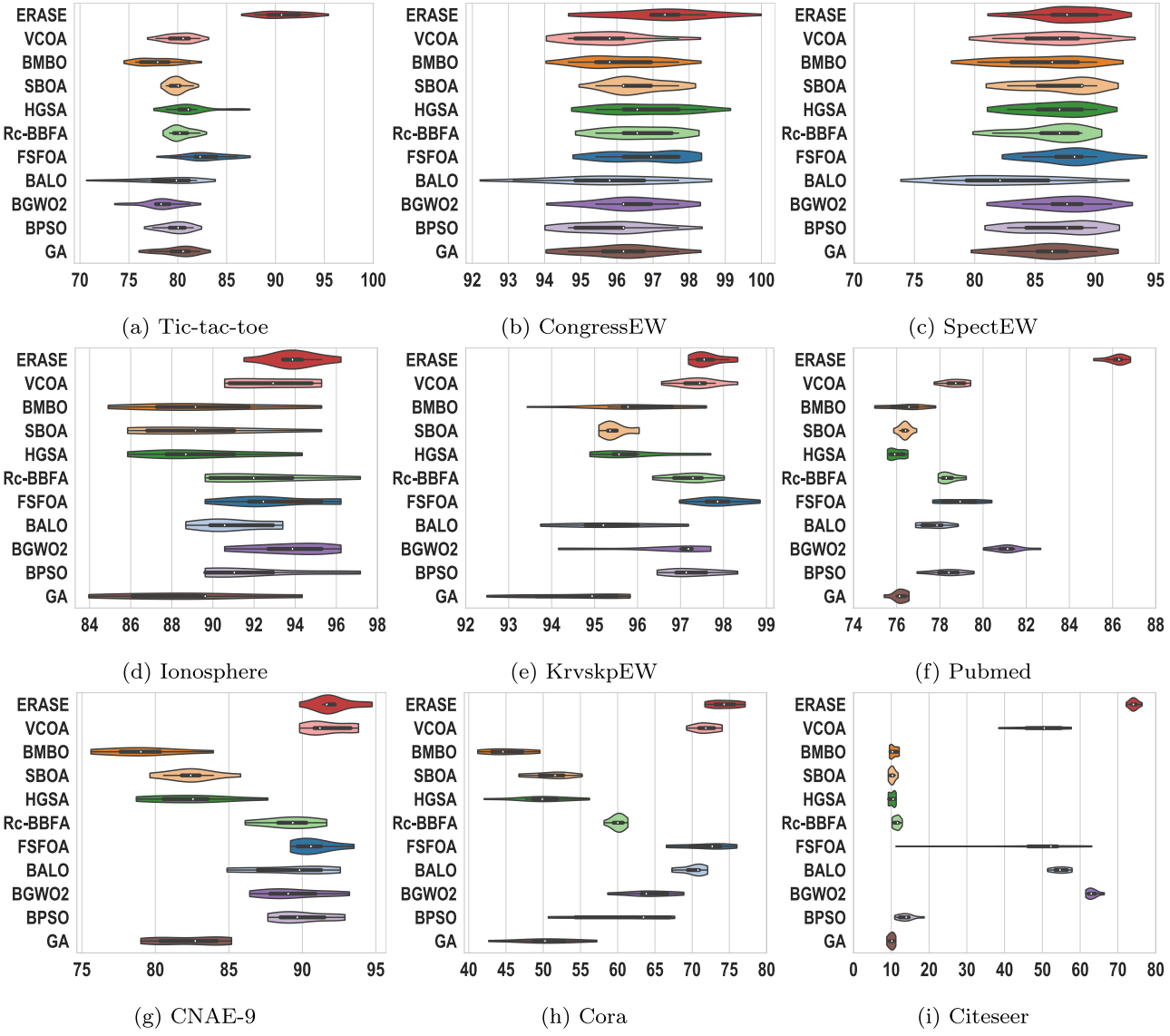

Fig. 5. Violin plot of Acc when 5NN is used in (7). For each sub-plot from (a)∼(i), the horizontal axis indicates the percentage value of Acc, and the vertical axis represents the algorithm. In each violin plot, the white dot denotes the median, and the left and right edges of the black box indicate the 25th and 75th percentiles, respectively.

datasets are chosen from different application domains and are diverse in numbers of features, samples, and classes.

Evaluation Criteria: We choose accuracy and dimension reduction as the main metrics to evaluate the performance of our method. Accuracy (Acc) and dimension reduction (Dr) are two commonly used criteria for feature selection evaluations [26,43]. Accuracy is defined as:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{9}$$

where TP, TN, FP, and FN are the true positives, true negatives, false positives, and false negatives, respectively. Dimension reduction is defined as

$$\text{Dr} = \frac{N - |\mathbf{S}|}{N}, \tag{10}$$

where N and $|\mathbf{S}|$ are the number of features of the dataset and cardinality of the selected feature subset, respectively.

In all the experiments, all the datasets are split 20 times to ensure statistical significance, and the train-test split ratio is 7:3. During the training process, the training data are submitted to the classification task in a 10-fold cross-validation scheme to than the hyperparameters. The settings of the main hyperparameters of the proposed method are listed in Table 4.

The hyperparameters of the baseline methods are set according to the values suggested in the corresponding papers listed in Table 2. The hyperparameter η is tuned in set $\{1, 2, \ldots, 10\}$.

Experimental Environment: Python 3.7 and scikit-learn [48] are used to implement our method[1], and all the codes are run on an ASUS machine with Intel Core i7 CPU (3.80 GHz), NVIDIA GeForce RTX 3060 GPU, 32GB DDR4 RAM, and 2TB hard disks.

4.2. Performance of ERASE

The k-NN classifier has been successfully employed in various feature selection methods [24,42,44]. Following [23,35], we set $k = 1$ and $k = 5$ in all the experiments, and the results are given in Tables 5 and 6, respectively. From Tables 5 and 6, we can observe that ERASE obtains the highest Acc and Dr values in the vast majority of cases. On all nine datasets, ERASE achieves the best performance on eight datasets both when $k = 1$ and $k = 5$. It is worth

[1] Codes is available at https://github.com/Peter7777777/ERASE.

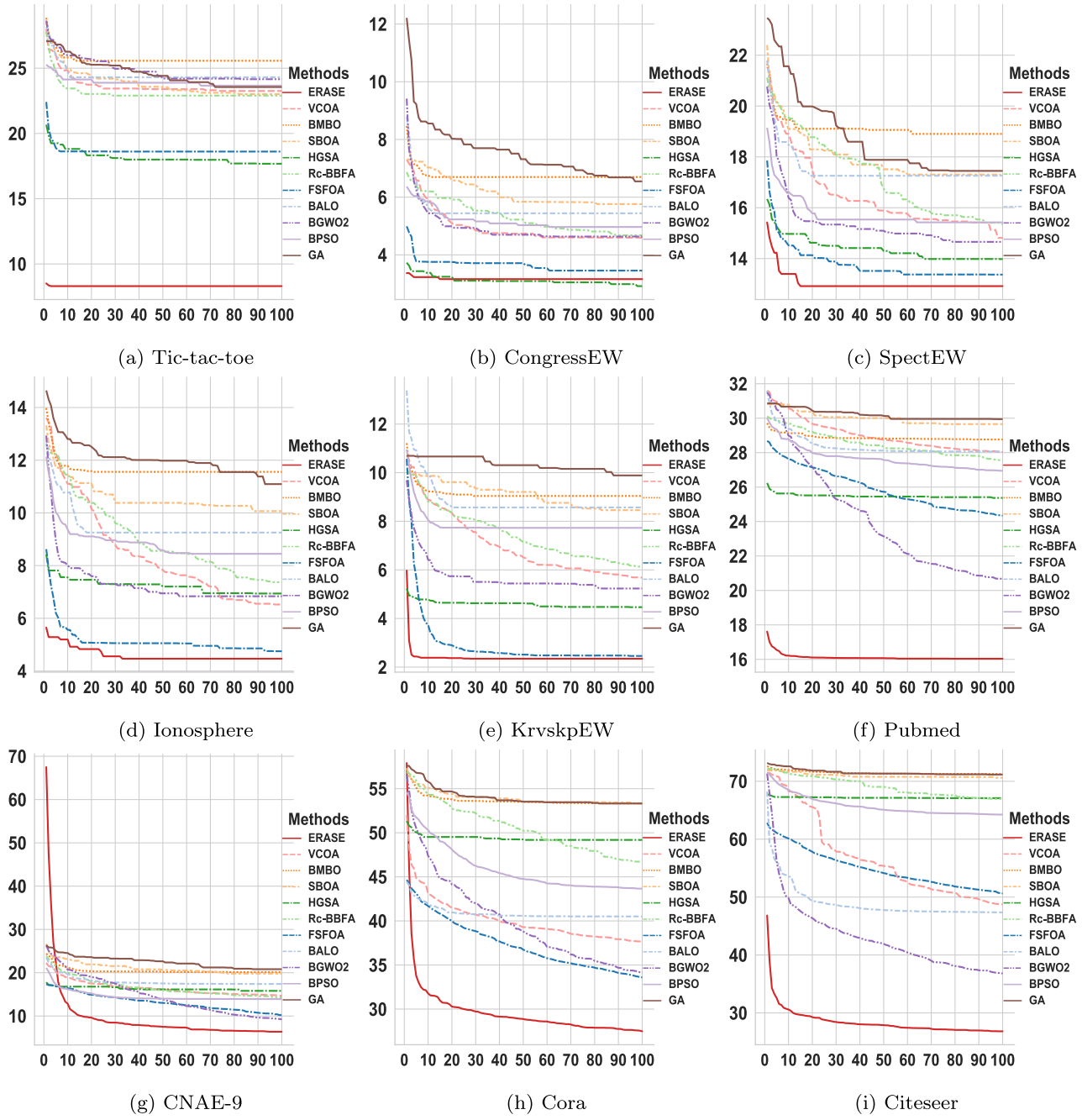

Fig. 6. Convergence curves of the average value of the fitness function of ERASE and other baselines when 1NN is used in (7). We conduct 20 Monte Carlo experiments for each method. For each sub-plot from (a)∼(i), the horizontal axis indicates the iterative steps, and the vertical axis indicates the average value of the fitness function in percentage.

noting that on Tic-tac-toe, ERASE is nearly 10% higher than the best baseline FSFOA, and on Citeseer, ERASE is nearly 12% higher than the best baseline BGWO2. In terms of Dr, the performance of ERASE is in the top two on the six datasets, in both cases where 1NN and 5NN are used. Comparing our method with the ensemble learning algorithms GBDT, XGBoost, and RF, we notice that ERASE ensures both good Dr and Acc in most cases, but GBDT, XGBoost, and RF cannot. We compute the average Acc and average Dr across different datasets for all methods. The results indicate that the average Acc and average Dr of ERASE are in the first and second places, respectively. However, the other baseline algorithms cannot ensure that the average rankings of both Acc and Dr are in the top three simultaneously. Therefore, the average rankings also indicate that ERASE performs better than the baseline methods in terms of Acc and Dr.

4.3. Statistical analysis

Tables 5 and 6 demonstrate that ERASE performs better than the baseline methods. In this section, we conduct some statistical analysis of the results of the proposed algorithm. Figures 4 and 5 depict the violin plots of Acc when 1NN and 5NN are used in (7), respectively. For each sub-plot of Figs. 4 and 5, the horizontal axis indicates Acc in percentage and the vertical axis denotes the algorithm. From Figs. 4 and 5, we observe that the medians of ERASE are higher than those of the other algorithms on most datasets.

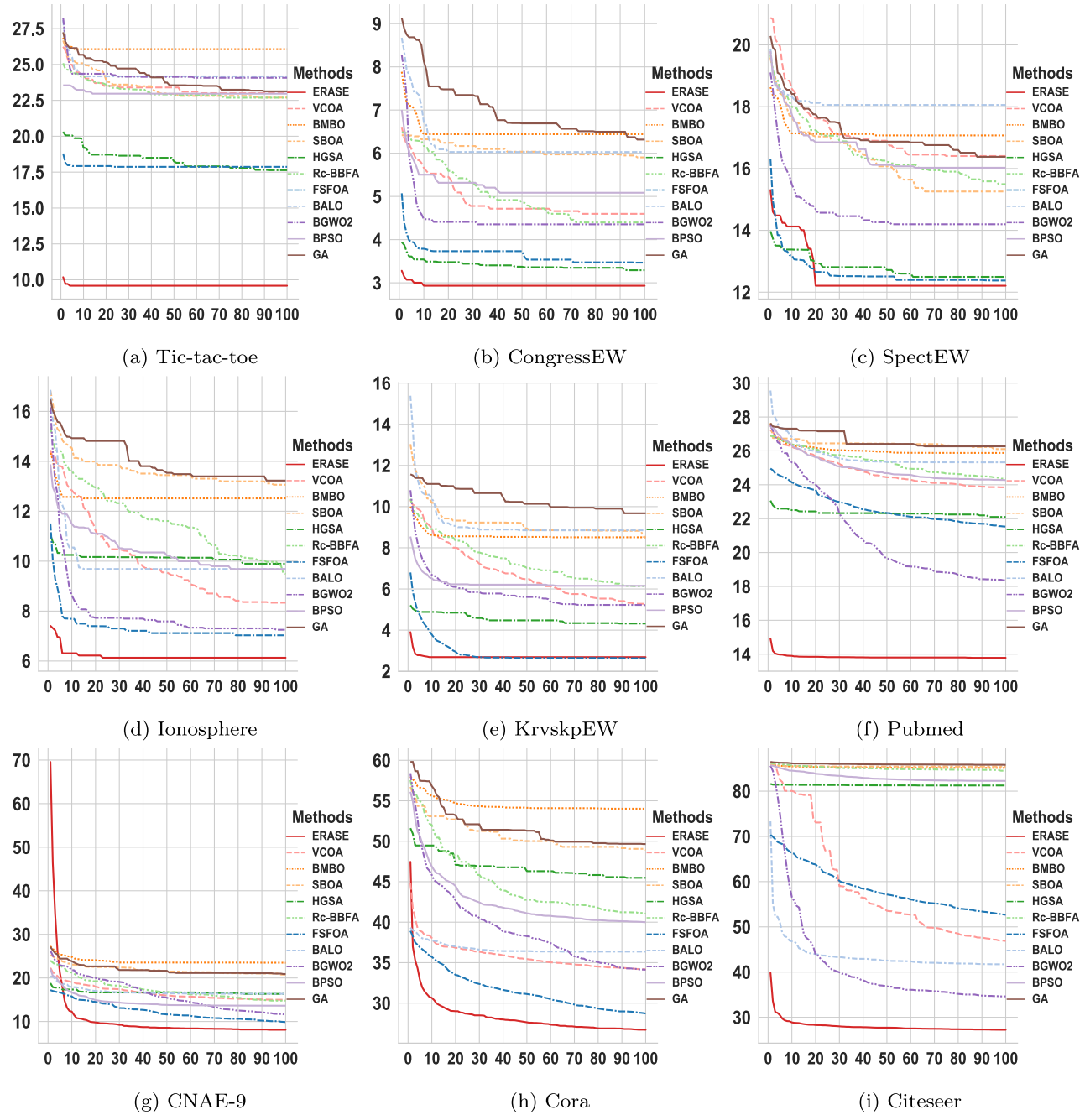

Fig. 7. Convergence curves of the average value of the fitness function of ERASE and other baselines when 5NN is used in (7). We conduct 20 Monte Carlo experiments for each method. For each sub-plot from (a)∼(i), the horizontal axis indicates the iterative steps, and the vertical axis indicates the average value of the fitness function in percentage.

Next, we use the Wilcoxon rank sum test to measure the difference between our algorithm and other algorithms. The Wilcoxon rank sum test, proposed by Frank Wilcoxon, is a well-known non-parametric test to check whether the results of the proposed methods are statistically different from those of the comparative methods. In various studies related to feature selection, a parameter called the p-value is utilized to verify the significance level of the two algorithms [35,44]. The null hypothesis of the test is that the results from two different methods are from continuous distributions with equal medians, and the test returns the p-value to determine whether the null hypothesis is accepted or rejected. Because a p-value less than 0.05 indicates strong evidence that the null hypothesis is rejected, there is less than a 5% probability that the null hypothesis is correct. We then compute the p-values of the Wilcoxon rank sum test of ERASE versus other algorithms when using 1NN and 5NN, and present the results in Tables 7 and 8, respectively. From Tables 7 and 8, we observe that, in most cases, the p-values are less than 0.05, indicating that the results of ERASE are significantly different from the results of the baseline methods on the majority of the datasets.

Finally, to analyze the convergence of the proposed ERASE, the fitness function convergence curves of ERASE versus the other approaches when using 1NN and 5NN are shown in Figs. 6 and 7, respectively. From Figs. 6 and 7, we can see that the curve of our algorithm decreases significantly within the first 10 steps, and the optimization process can be basically completed within 20 steps.

Table 7
p-values of the Wilcoxon rank sum test of the proposed ERASE vs. other algorithms when 1NN is used in (7). The *p*-values greater than 0.05 are underlined.

	Tic-tac-toe	CongressEW	SpectEW	Ionosphere	KrvskpEW	Pubmed	CNAE-9	Cora	Citeseer
VCOA	8.88E-5	5.62E-2	3.95E-1	4.09E-1	2.88E-2	9.13E-5	4.89E-3	8.88E-5	9.08E-5
BMBO	8.98E-5	1.59E-2	4.56E-2	3.85E-3	8.98E-5	9.03E-5	8.29E-5	9.13E-5	9.03E-5
SBOA	8.78E-5	3.05E-1	6.76E-2	1.96E-2	8.73E-5	9.13E-5	8.06E-5	9.08E-5	9.03E-5
HGSA	8.78E-5	2.76E-1	5.42E-2	1.95E-2	8.83E-5	9.08E-5	8.58E-5	9.08E-5	8.98E-5
Rc-BBFA	8.78E-5	2.73E-2	4.85E-1	2.22E-1	2.18E-3	9.13E-5	1.32E-4	9.03E-5	9.03E-5
FSFOA	8.78E-5	3.47E-1	4.54E-1	3.51E-1	2.23E-1	9.13E-5	2.13E-3	9.13E-5	8.98E-5
BALO	8.93E-5	3.45E-2	1.83E-2	3.40E-2	8.93E-5	9.08E-5	2.08E-4	9.08E-5	9.03E-5
BGWO2	9.03E-5	1.20E-2	4.39E-1	1.48E-1	8.73E-5	9.08E-5	3.22E-3	9.13E-5	9.03E-5
BPSO	8.93E-5	9.71E-2	4.09E-1	1.04E-1	8.83E-5	9.08E-5	2.74E-4	9.13E-5	9.03E-5
GA	8.78E-5	7.59E-2	2.17E-2	5.38E-3	8.93E-5	9.13E-5	8.44E-5	9.13E-5	9.08E-5
GBDT	9.03E-5	4.69E-4	1.47E-4	2.38E-4	8.88E-5	9.13E-5	8.54E-5	8.93E-5	9.08E-5
XGBoost	8.98E-5	3.51E-3	8.25E-5	5.44E-4	8.88E-5	9.08E-5	8.58E-5	9.13E-5	9.03E-5
RF	9.03E-5	1.13E-4	1.13E-4	1.69E-3	8.93E-5	9.08E-5	8.49E-5	9.08E-5	9.03E-5

Table 8
p-values of the Wilcoxon rank sum test of the proposed ERASE vs. other algorithms when 5NN is used in (7). The *p*-values greater than 0.05 are underlined.

	Tic-tac-toe	CongressEW	SpectEW	Ionosphere	KrvskpEW	Pubmed	CNAE-9	Cora	Citeseer
VCOA	8.83E-5	5.69E-3	1.60E-1	1.01E-1	1.35E-1	9.13E-5	2.12E-1	4.52E-3	9.08E-5
BMBO	8.98E-5	1.94E-2	9.08E-2	2.35E-3	1.54E-3	9.13E-5	8.63E-5	9.13E-5	9.08E-5
SBOA	8.83E-5	7.59E-2	2.45E-1	1.32E-3	8.29E-5	9.03E-5	8.49E-5	9.13E-5	9.03E-5
HGSA	8.83E-5	2.20E-1	1.51E-1	7.48E-4	6.41E-4	9.13E-5	8.58E-5	9.13E-5	9.03E-5
Rc-BBFA	8.83E-5	1.22E-1	1.25E-1	4.65E-2	3.72E-2	9.13E-5	5.93E-4	8.98E-5	9.08E-5
FSFOA	8.83E-5	2.39E-1	4.85E-1	1.10E-1	8.53E-2	9.08E-5	1.52E-2	4.45E-2	9.08E-5
BALO	8.98E-5	1.90E-2	3.29E-3	9.62E-4	1.02E-4	9.13E-5	4.79E-3	1.64E-4	9.03E-5
BGWO2	8.83E-5	5.68E-2	3.36E-1	4.85E-1	1.65E-3	9.13E-5	5.48E-3	9.13E-5	9.08E-5
BPSO	8.98E-5	9.66E-3	1.99E-1	2.30E-2	6.90E-2	9.13E-5	2.41E-2	9.03E-5	9.03E-5
GA	9.03E-5	1.89E-2	9.49E-2	2.53E-3	8.78E-5	9.08E-5	8.63E-5	9.13E-5	9.03E-5
GBDT	9.03E-5	9.94E-4	2.67E-4	8.34E-5	8.73E-5	9.13E-5	1.12E-4	9.13E-5	9.08E-5
XGBoost	8.98E-5	4.64E-3	1.32E-4	8.15E-5	8.78E-5	9.13E-5	1.34E-4	9.13E-5	8.98E-5
RF	8.98E-5	2.61E-4	2.28E-4	1.91E-4	8.88E-5	9.13E-5	8.54E-5	9.03E-5	9.08E-5

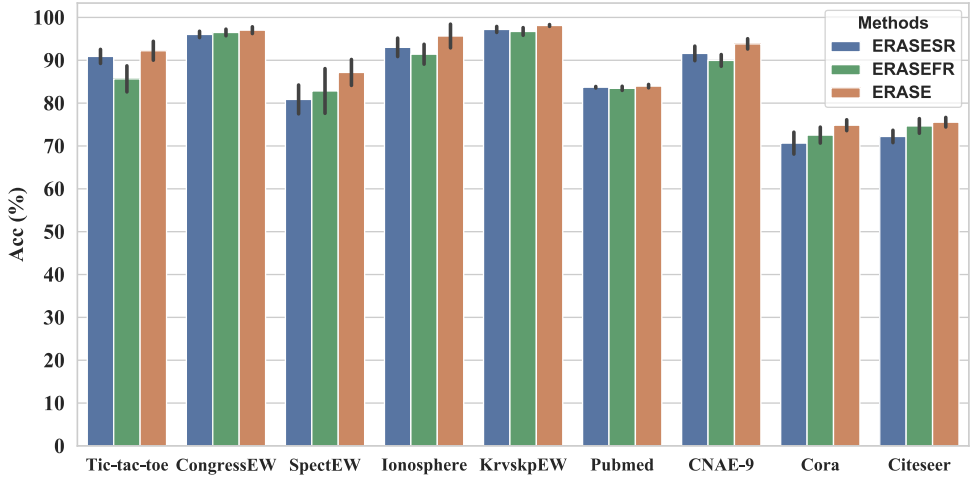

Fig. 8. Histogram for comparing average Acc when 1NN is used in (7). ERASESR denotes primarily utilizing sample relations for feature selection. ERASEFR indicates only utilizing feature relations for feature selection. We conduct 20 Monte Carlo experiments for each method. The black line in each bar indicates standard deviation.

Comparing the convergence curves of ERASE and the other methods, we observe that the convergence speed of ERASE is competitive with that of the others. It is worth noting that the worst convergence curve of our method is better than the optimal ones of other algorithms on the Tic-tac-toe, Pubmed, and Citeseer.

4.4. Ablation study

To evaluate the effectiveness of sample and feature relations, we conduct an ablation study to compare ERASE with its two variants, ERASESR and ERASEFR. In ERASESR, we primarily consider utilizing sample relations to select feature subsets, whereas in ERASEFR, we only consider utilizing feature relations to select feature subsets. For ERASESR, we use the graph of samples learned in the last iteration of the FOA, and use the randomly initialized trees of features. For ERASEFR, we only use the trees of features learned in the last iteration of the FOA. The results of the ablation study are shown in Figs. 8 and 9.

From the results of Figs. 8 and 9, we see that on some datasets, the performance of ERASESR is better than that of ERASEFR, and on other datasets, the performance of ERASEFR is better than that of ERASESR. However, the performances of both ERASESR and

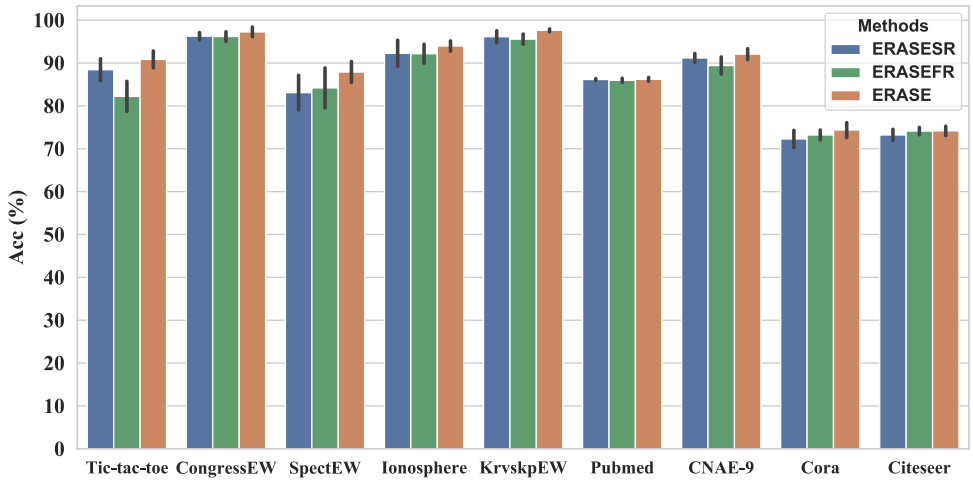

Fig. 9. Histogram for comparing average Acc when 5NN is used in (7). ERASESR denotes primarily utilizing sample relations for feature selection. ERASEFR indicates only utilizing feature relations for feature selection. We conduct 20 Monte Carlo experiments for each method. The black line in each bar indicates standard deviation.

ERASEFR on all datasets are inferior to that of ERASE. Therefore, our proposed methods of utilizing sample relations and feature relations are conducive to ERASE for feature selection.

5. Conclusion

Feature selection is an important preprocessing step in machine learning and pattern recognition. Recent studies have demonstrated that learning underlying sample and feature relations is beneficial for feature selection. In this study, we propose a novel wrapper method named ERASE. To the best of our knowledge, ERASE is the first wrapper method that can simultaneously learn the graph structure of sample relations and the tree structure of feature relations. An ablation study is also conducted to show that the learned sample relations and feature relations are conducive to ERASE for feature selection. Experimental and statistical analysis results on multiple datasets from various fields demonstrate that the proposed ERASE has apparent advantages in feature selection compared with other baseline methods.

There are still some limitations on ERASE, which deserve our further investigation. In ERASE, the feature subset is selected sequentially, which is disadvantageous to parallel computation. An improved ERASE algorithm that is suitable for parallel computation deserves further exploration. Besides, although our algorithm considers the underlying relationships between samples, the relation learning in small samples is not considered. Our future work will extend the proposed method into a new feature selection method by considering small sample relation learning.

Declaration of Competing Interest

We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China (No.62106082) and the National Natural Science Foundation of China (No.61976102, No.U19A2065).

References

[1] J. Ircio, A. Lojo, U. Mori, J.A. Lozano, Mutual information based feature subset selection in multivariate time series classification, Pattern Recognit. 108 (2020) 107525.
[2] B.S.C. Wade, S.H. Joshi, B.A. Gutman, P.M. Thompson, Machine learning on high dimensional shape data from subcortical brain surfaces: a comparison of feature selection and classification methods, Pattern Recognit. 63 (2017) 731–739.
[3] L. Zhang, Q. Zhang, B. Du, X. Huang, Y.Y. Tang, D. Tao, Simultaneous spectral-spatial feature selection and extraction for hyperspectral images, IEEE Trans. Cybern. 48 (1) (2018) 16–28.
[4] G. Ditzler, R. Polikar, G. Rosen, A sequential learning approach for scaling up filter-based feature subset selection, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2018) 2530–2544.
[5] H. Liu, H. Motoda, R. Setiono, Z. Zhao, Feature selection: an ever evolving frontier in data mining, in: Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, FSDM, held at PAKDD 2010, in: JMLR Proceedings, Vol. 10, JMLR.org, 2010, pp. 4–13.
[6] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.
[7] M. Bennasar, Y. Hicks, R. Setchi, Feature selection using joint mutual information maximisation, Expert Syst. Appl. 42 (22) (2015) 8520–8532.
[8] L. Cui, L. Bai, Z. Zhang, Y. Wang, E.R. Hancock, Identifying the most informative features using a structurally interacting elastic net, Neurocomputing 336 (2019) 13–26.
[9] L. Cui, L. Bai, Y. Wang, P.S. Yu, E.R. Hancock, Fused lasso for feature selection using structural information, Pattern Recognit. 119 (2021) 108058.
[10] N. García-Pedrajas, J.A.R. del Castillo, G.C. García, SI(FS)2: fast simultaneous instance and feature selection for datasets with many features, Pattern Recognit. 111 (2021) 107723.
[11] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (Mar) (2003) 1157–1182.
[12] K.C. Tan, E.J. Teoh, Q. Yu, K. Goh, A hybrid evolutionary algorithm for attribute selection in data mining, Expert Syst. Appl. 36 (4) (2009) 8616–8630.
[13] O.N. Oyelade, A.E. Ezugwu, T.I.A. Mohamed, L.M. Abualigah, Ebola optimization search algorithm: a new nature-inspired metaheuristic optimization algorithm, IEEE Access 10 (2022) 16150–16177.
[14] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Eng. 376 (2021) 113609.
[15] M. Ghaemi, M.-R. Feizi-Derakhshi, Forest optimization algorithm, Expert Syst. Appl. 41 (15) (2014) 6676–6687.
[16] L.M. Abualigah, D. Yousri, M.A. El-Aziz, A.A. Ewees, M.A.A. Al-qaness, A.H. Gandomi, Aquila optimizer: a novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250.
[17] J.O. Agushaka, A.E. Ezugwu, L. Abualigah, Dwarf mongoose optimization algorithm, Comput. Methods Appl. Mech. Eng. 391 (2022) 114570.
[18] A.E. Ezugwu, J.O. Agushaka, L. Abualigah, S. Mirjalili, A.H. Gandomi, Prairie dog optimization algorithm, Neural Comput. Appl. 34 (22) (2022) 20017–20065.
[19] L.M. Abualigah, M.A. Elaziz, P. Sumari, Z.W. Geem, A.H. Gandomi, Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer, Expert Syst. Appl. 191 (2022) 116158.
[20] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Comput. 23 (3) (2019) 715–734.
[21] A. Unler, A. Murat, R.B. Chinnam, mr^2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, Inf. Sci. 181 (20) (2011) 4625–4641.

[22] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626.
[23] M. Alweshah, S. Al Khalaileh, B.B. Gupta, A. Almomani, A.I. Hammouri, M.A. Al–Betar, The monarch butterfly optimization algorithm for solving feature selection problems, Neural Comput. Appl. (2020) 1–15.
[24] M. Taradeh, M.M. Mafarja, A.A. Heidari, H. Faris, I. Aljarah, S. Mirjalili, H. Fujita, An evolutionary gravitational search-based feature selection, Inf. Sci. 497 (2019) 219–239.
[25] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[26] R.C.T. de Souza, C.A. de Macedo, L. dos Santos Coelho, J. Pierezan, V.C. Mariani, Binary coyote optimization algorithm for feature selection, Pattern Recognit. 107 (2020) 107470.
[27] T. Bäck, D.B. Fogel, Z. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators, CRC press, 2018.
[28] H.-P. Schwefel, T. Bäck, D. Fogel, Z. Michalewicz, Advantages (and disadvantages) of evolutionary computation over other approaches, Evol. Comput. 1 (2000) 20–22.
[29] F. Provost, T. Fawcett, Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking, O'Reilly Media, Inc., 2013.
[30] X.M. Zhou, T.S. Dillon, A statistical-heuristic feature selection criterion for decision tree induction, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 834–841.
[31] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[32] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (5) (2001) 1189–1232.
[33] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: Proceedings of SIGKDD 2016, 2016, pp. 785–794.
[34] G. Louppe, L. Wehenkel, A. Sutera, P. Geurts, Understanding variable importances in forests of randomized trees, in: Proceedings of NeurIPS 2013, 2013, pp. 431–439.
[35] S. Arora, P. Anand, Binary butterfly optimization approaches for feature selection, Expert Syst. Appl. 116 (2019) 147–160.
[36] J.L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (9) (1975) 509–517.
[37] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.
[38] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008) 93–106.
[39] A. McCallum, K. Nigam, J. Rennie, K. Seymore, Automating the construction of internet portals with machine learning, Inf. Retr. Boston 3 (2) (2000) 127–163.
[40] C.L. Giles, K.D. Bollacker, S. Lawrence, CiteSeer: an automatic citation indexing system, in: Proceedings of the 3rd ACM International Conference on Digital Libraries, ACM, 1998, pp. 89–98.
[41] D. Dua, C. Graff, UCI machine learning repository, 2017.
[42] Y. Zhang, X. Song, D. Gong, A return-cost-based binary firefly algorithm for feature selection, Inf. Sci. 418 (2017) 561–574.
[43] M. Ghaemi, M.-R. Feizi-Derakhshi, Feature selection using forest optimization algorithm, Pattern Recognit. 60 (2016) 121–129.
[44] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary ant lion approaches for feature selection, Neurocomputing 213 (2016) 54–65.
[45] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing 172 (2016) 371–381.
[46] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms, Appl. Soft Comput. 18 (2014) 261–276.
[47] A.E. Eiben, P. Raué, Z. Ruttkay, Genetic algorithms with multi-parent recombination, in: Proceedings of PPSN 1994, in: Lecture Notes in Computer Science, Vol. 866, Springer, 1994, pp. 78–87.
[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

Zhaogeng Liu is currently pursuing his PhD degree in the School of Artificial Intelligence at Jilin University, supervised by Prof. Yi Chang. His research focuses on feature selection.

Jielong Yang received the PhD degree in Electrical and Electronic Engineering from Nanyang Technological University, Singapore, in 2020. He is currently an Assistant Professor in the School of Artificial Intelligence at Jilin University, PR China. He received the BEng and MSc degrees in Mechanical Engineering from Xi'an Jiaotong University in 2012 and 2014, respectively. His research interests include semi-supervised and unsupervised learning with bayesian networks and graph neural networks.

Li Wang received her master's degree from the Changchun University of Technology. She is now a teaching assistant in the School of Virtual Reality at Jilin Animation Institute. Her research interests include feature selection and information theory.

Yi Chang is the Dean of School of Artificial Intelligence, Jilin University, China. He was a Technical Vice President at Huawei Research America, and a research director at Yahoo Research before that. His research interests include information retrieval, data mining, machine learning, natural language processing, and artificial intelligence. He has published more than 100 papers on premium conferences or journals, and he has been severed as the conference general chair for ACM WSDM'2018 and ACM SIGIR'2020. He was elected as an ACM Distinguished Scientist in 2018, for his contributions to intelligent algorithms for search engines.