Expert Systems With Applications 164 (2021) 113842

Contents lists available at ScienceDirect

Expert Systems With Applications

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

A feature selection algorithm of decision tree based on feature weight

HongFang Zhou[a,b,*], JiaWei Zhang[a], YueQing Zhou[c], XiaoJie Guo[a], YiMing Ma[a]

[a] *School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China*
[b] *Shaanxi Key Laboratory of Network Computing and Security Technology, Xi'an 710048, China*
[c] *Hisense TransTech Co., Ltd, Qingdao 266071, China*

ARTICLE INFO

Keywords:
Feature selection
Decision tree
ReliefF
Feature weight
Median

ABSTRACT

In order to improve the classification accuracy, a preprocessing step is used to pre-filter some redundant or irrelevant features before decision tree construction. And a new feature selection algorithm FWDT is proposed based on this. Experimental results show that FWDT our proposed method performs better for the measures of accuracy, recall and F1-score. Furthermore, it can reduce the required time in constructing the decision tree.

1. Introduction

As an important step in data preprocessing, feature selection has become a popular research direction (Schiezaro & Pedrini, 2013). And it can delete redundant/irrelevant features and retain some important features in the data. In view of this, we can improve the classification accuracy and accelerate the model construction procedure by means of it (Blum & Langley, 1997; Guyon, 2003; Liu & Motoda, 1998). At present, feature selection has been widely used in text classification, information retrieval, image recognition, medical detection, financial fraud, and so on (Alazab, Hobbs, Abawajy, & Alazab, 2012; Amiri, Rezaei Yousefi, Lucas, Shakery, & Yazdani, 2011; Gao et al., 2013; Huang & Yu, 2013; Jae Young Choi, Yong Man Ro, & Plataniotis, 2011; Khotanzad & Hong, 1990; Lewis, Yang, Rose, & Fan, 2004; Li-Ping Jing, Hou-Kuan Huang, & Hong-Bo Shi, 2002; Qinbao Song, Jingjie Ni, & Guangtao Wang, 2011; Vasconcelos, 2003).

Up to now, there are many feature selection algorithms, including filtering, encapsulation and embedded ones (Ball & Brunner, 2010; Cai, Luo, Wang, & Yang, 2020; Gao, Hu, Zhang, & He, 2018; Gao, Hu, Zhang, & Wang, 2018; Lausch, Schmidt, & Tischendorf, 2015; Tang & Peng, 2017). Decision tree, a typical embedded feature selection algorithm, is widely used in machine learning and data mining (Sun & Hu, 2017). The classic methods to construct decision tree are ID3, C4.5 and CART (Quinlan, 1979, 1986; Salzberg, 1994; Yeh, 1991). Among them, C4.5 is an improvement on ID3 which is liable to select more biased features as partition features. CART can deal with the features with more values as partition ones (Roy, Mondal, Ekbal, & Desarkar, 2016). Besides these, it is qualified for both nominal and continuous features at the same time. In 2017, the heuristic class constraint uncertainty was used as the criteria for selecting features in the decision tree construction (Sun & Hu, 2017). In 2019, DRDT was proposed In DRDT, the discrete rate is used as the standard of selecting features in the decision tree construction (Roy et al., 2016; Roy, Mondal, Ekbal, & Desarkar, 2019b). In 2019, Asma Trabelsi proposed a decision tree classification algorithm for class tag and evidence attribute. In the algorithm, they introduced the evidence theory to deal with the uncertainty attribute value and the class tag (Trabelsi, Elouedi, & Lefevre, 2019). Nour El islem karabadji proposed an improved feature selection method of decision tree in 2019. This method generates an optimized decision tree by selecting the optimal pair of the training samples and the feature subsets. The selected optimization scheme can avoid over fitting (Karabadji et al., 2019). Haidi Rao put forward a feature selection algorithm based on the combination of artificial bee colony and gradient lifting decision tree in 2019. They used bee colony to realize the global optimization of decision tree input, identify informative features and suppress irrelevant features (Rao et al., 2019). All above methods do not consider the importance of each feature and category in the data set. So how to determine the importance of each feature and category is our focus in the paper.

In this paper, the concept of feature weight is introduced and it is used as the standard of feature selection in decision tree construction. When constructing the decision tree, the weight of each feature is

* Corresponding author at: School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China.
E-mail addresses: zhouhf@xaut.edu.cn (H. Zhou), 576852849@qq.com (J. Zhang), 2155261@qq.com (Y. Zhou), 784589380@qq.com (X. Guo), 838475593@qq.com (Y. Ma).

https://doi.org/10.1016/j.eswa.2020.113842
Received 1 February 2020; Received in revised form 4 July 2020; Accepted 2 August 2020
Available online 6 September 2020
0957-4174/

calculated respectively and the feature with the largest weight value is selected as the partition feature of the corresponding layer to construct the decision tree. At the same time, this paper introduces a new algorithm to pre-filter the features before constructing the decision tree and delete the irrelevant features to improve the classification accuracy.

The paper is arranged as follows. The second part introduces the existing feature selection algorithms based on decision tree, their advantages and disadvantages and traditional ReliefF algorithm. The third part introduces the proposed FWDT algorithm and feature pre-filtering method. The fourth part is the experiment and the result analysis. The fifth part is the summary of the paper and the future work.

2. Related works

2.1. Decision tree

In the feature selection method based on decision tree, the process of constructing decision tree is the process of feature selection. When determining the features of partition samples of each layer, each feature is calculated according to some certain standards and the most important feature is selected as the feature of partition samples every time (Quinlan, 1986). The main advantages of the decision tree algorithm are high classification accuracy and strong robustness. The disadvantage is that it is liable to be over-fitting, the decision tree is easily influenced by the samples and the subtree may be repeated many times in the decision tree. For the problem of over-fitting, we can solve it by means of the pruning technology and k-fold cross validation. Through pruning, the redundant branches can be cut beforehand. And furthermore, over-fitting can be avoided. For the second problem, a pre-filtering step can be referred in the data preprocessing stage to remove some irrelevant features. In such way, it can reduce the size of the decision tree and avoid the problem of inaccurate decision trees. As a classic data mining algorithm, decision tree algorithm has become a hot research topic in the field of feature selection (Sun & Hu, 2017). Therefore, many researchers devoted themselves to the study of the decision tree algorithm.

In the decision tree algorithm based on information gain (ID3) proposed by Quinlan, information entropy is used as the measure of importance of features. Entropy is a measure of uncertainty. The more order a system is, the lower the information entropy is (Sun & Hu, 2017). The decision tree is constructed by using information gain, and the feature of maximum entropy reduction is selected to divide the data. The disadvantage of this algorithm is that it only can deal with discrete features and is liable to choose the features with more values.

In view of the disadvantages of ID3, information gain rate is introduced in C4.5 (Salzberg, 1994). But it is required to scan and sort the data set multiple times in constructing the tree. Additionally, it is only suitable for small-size data sets.

In CART, Gini index is used as the standard for selecting features (Yeh, 1991). Gini index was first used in the field of economics. It is mainly used to measure the fairness of income distribution. In the process of decision tree construction, it measures the purity or uncertainty of data. At the same time, Gini index is used to judge the optimal dichotomy of category variables. The advantage of this algorithm is that it can deal with the continuous and discrete features, as well as the classification and regression problems. But it is unsuitable for the continuous features. When there are too many categories, it may take more errors frequently.

After that, CRDT is advanced. It can select features with more values as partition features (Roy et al., 2019b). This method does not favor features with more eigenvalues, and it is suitable for nominal features. This model usually focuses on the correlation between features and categories, and it can get the nonlinear dependence between features and categories. So, it is suitable for quantitative application of classification results. CCDT was proposed by Huaning sun et al in 2017. In CCDT, the heuristic class constraint uncertainty was used as the standard for selecting features in the process of decision tree construction. CCDT avoids the statistical advantages and heuristic bias of multi value features, and improves the classification effect and stability of decision trees (Sun & Hu, 2017). DRDT was proposed in 2019. In DRDT, the discrete rate is used as the standard for selecting features in the process of decision tree construction (Roy, Mondal, Ekbal, & Desarkar, 2019a). DRDT is an improvement on the correlation ratio method. The whole method is divided into two parts. Firstly, the discretization module is used to discretize the continuous features in the data set, and then the decision tree model is constructed using the discretization rate. Because a discretization step is referred, it is better than CRDT (Roy et al., 2016). The main advantage of CRDT is that it does not tend to select features which have too many feature values. Therefore, it is suitable for small-size data sets and fewer feature number. Besides this, it is qualified for the data sets with more than two classes and the unbalanced class distribution (Alazab et al., 2012). But it is difficult to find the exact value of *K* when k-means algorithm is used for discretization.

2.2. ReliefF algorithm

Relief is a classic filtering feature selection method proposed by Kira and Rendall. It is an example-based learning method (Kononenko, 1994). Later, Kononenko improved ReliefF in 1994 (Reyes, Morell, & Ventura, 2015), which was qualified for multiple classification problems and missing data or noise data (Chenwen, Chenyang, Shujin, & Guanghui, 2018). As an independent evaluation filtering feature selection method, this method has high efficiency and low time complexity, and does not use classification accuracy as the evaluation function. Because this algorithm is based on feature weight, it only improves the weight value of features with high correlation with category and deletes features with low weight value Accordingly, it cannot remove redundant features very well.

ReliefF selects relevant features on the basis of their feature weights. It firstly selects one sample *R* in training set. For a sample set *H* which is in the same category with *R*, we calculate the distance between each sample *Hj* and *R*, and then choose *K* nearest samples, At the same time, we need calculate each and *R* different categories of the distance between the sample in the sample sets *M* and *R*, and then choose *K* nearest samples as *R* not same kind of nearest neighbor sample. When calculating the feature weight of a certain feature *A*, there is a difference between two samples *R* and *Hj* in the training set. The feature *A* is given a small feature weight when two samples *R* and *Mj*(c) are different in the training set. Similarly, the feature *A* is given a large feature weight when they are same.

The feature weight of a feature *A* is defined as Eq. (1).

$$W(A) = W(A) - \sum_{j=1}^{k} diff(A,R,Hj)/m^{*}k + \sum_{C\notin class(R)}\left[\frac{P(c)}{1-P(class(R))}\sum_{j=1}^{k} diff(A,R,Mj(c))\right]/m^{*}k \tag{1}$$

Among them, the initial weight of each feature is 0, $\sum_{j=1}^{k} diff(A,R,Hj)$ $/m^{*}k$ means the weight of the sample set similar to sample *R* and $\sum_{C\notin class(R)}\left[\frac{P(c)}{1-P(class(R))}\sum_{j=1}^{k} diff(A,R,Mj(c))\right]/m^{*}k$ means the weight of the sample set different from sample *R*. *M* is the number of random samples, and *K* is the number of the nearest neighbors. The initial eigenvalue subtracts the weight value of the same category, and the weight value of different categories is taken as the final feature weight value of a feature. The larger the calculated weight value is, the higher the importance of the feature to the category is (Urbanowicz, Meeker, La Cava, Olson, & Moore, 2018).

In the traditional ReliefF, $diff(A,Ix,Iy)$ can be shown in Eq. (2) for the nominal eigenvalue. (Reyes et al., 2015). And for the numerical one, it can be shown as Eq. (3)

$$diff(A, Ix, Iy) = \begin{cases} 0 & value(A, Ix) = value(A, Iy) \\ 1 & otherwise \end{cases} \tag{2}$$

$$diff(A, Ix, Iy) = \frac{|value(A, Ix) - value(A, Iy)|}{max(A) - min(A)} \tag{3}$$

3. Decision tree algorithm based on feature weight with pre-filtering

In the traditional feature selection algorithm based on decision tree, the decision tree is easy to be influenced by the category and the irrelevant features. In such case, it is complex in constructing the decision tree and is liable to be over fitting. Therefore, it is required to select the most relevant feature in building the decision tree. Hence, feature weight, a new concept, is introduced in the paper. The feature weight is used as the standard to select features in constructing the decision tree. The feature weight of each feature is calculated in each layer of the decision tree, and the feature with the largest value of feature weight is selected as the partition feature of this layer.

Before constructing the decision tree, we use the feature selection algorithm to filter the features in advance, remove the features with low correlation with the category, and retain the features with high correlation with the category as the feature subset of the next step of constructing the decision tree. In order to weaken the influence of irrelevant features on the accuracy of the decision tree, this paper introduces the feature selection algorithm ReliefF and uses the improved ReliefF algorithm to filter the initial feature collection.

3.1. Decision tree algorithm based on feature weight

In the traditional decision tree feature selection algorithm, there will be some influence of irrelevant features on the constructed decision tree. At the same time, the decision tree algorithm has strong robustness to redundant features. It can well avoid the problems caused by the inability to remove redundant features in ReliefF algorithm. Therefore, the feature weight is introduced in the decision tree, and the feature weight is used as the criterion of feature selection to construct the decision tree recursively. The decision tree algorithm based on feature weight is described as follows.

Algorithm 1: Decision tree algorithm based on feature weight (FWDT)

```
Input: Training set D, Feature set A
Output: A decision tree
1. Create a root node containing the entire training set D
2. Calculate the feature weight value of each feature (Algorithm 2), and select the
   feature with the largest feature weight value each time
3. The training set is divided based on the features selected in the previous step, and
   the used feature $A_i$ is deleted in feature set A
4. For Every split subset do
5.    If all instances belong to the same class, use the class label to create a leaf node
6.    If the subset is empty, most classes of the parent node are allocated in the
   associated leaf node
7.    If the instance belongs to a different class label, go to step 2.
End For
8.Output a decision tree model
```

The calculation of feature weight in the second step is described in Algorithm 2.

Algorithm 2: Calculating feature weight

```
Input: Training set D, each feature $A_i$ in feature set A, number of random sampling M,
   number of nearest neighbor samples K, Feature set A contains N features
Output: Feature weight value $W(A_i)$ of each feature
1. For i = 1 to N do
2.    For j = 1 to M do
3.       Randomly select a sample R from training set D
4.       Calculate the Euclidean distance between R and each sample in the same
   sample set, and find out k nearest neighbor samples.
5.       Calculate the distance between each sample and R in each sample set different
   from R, and find out nearest neighbor samples in each sample set different from R.
6.       Calculate the feature weight W (A) of each feature: $W(A) = W(A) - \sum_{j=1}^{k} diff(A, R, Hj)/m^*k + \sum_{C \notin class(R)} \left[ \frac{P(c)}{1 - P(class(R))} \sum_{j=1}^{k} diff(A, R, Mj(c)) \right] /m^*k$
    End For
End For
7. Output the feature weight value W(Ai) of each feature
```

(continued on next column)

(continued)

Among them, the features in feature set A used by the above two algorithms are all the feature sets after the previous use of K-means algorithm to discretize the continuous features and use feature pre- filtering. Feature set A contains N features, and the data set is divided into test set and training set D according to 5-fold cross validation.

3.2. Feature pre-filtering strategy

At the same time, there will be irrelevant features in the general decision tree construction method. The decision tree constructed by these features will reduce the classification accuracy when classifying the test data set. Therefore, it is necessary to delete the features with low correlation between the data set and the category before constructing the decision tree. Therefore, we introduce the feature selection method ReliefF to filter the irrelevant features. Because the threshold value of feature weight needs to be set in advance in the traditional ReliefF algorithm, and it is usually determined by the experiments or experiences, there is no good way to determine the threshold value of the feature weight. The disadvantage of this method is that when the data distribution and data characteristics in the data set to be tested are not known or have not been tested many times before, it is difficult to determine the appropriate feature weight threshold and time-consuming.

Therefore, this paper proposes an adaptive method to determine the feature weight threshold. The concept of median is introduced, and the weight of the feature corresponding to the median in the sorted feature weight vector is taken as the feature weight threshold.

Definition of median: It is the number in the middle of a set of data arranged in order, representing a value of a sample, population or probability distribution. It can divide the set of values into two equal parts. For a limited number set, you can find the middle one as the median by sorting all the observations. If there are even observations, the median usually is the average of the two most intermediate values.

Suppose a set of data$x1, x2.....xn$is sorted in ascending order, which is shown as $x(1), x(2)......x(n)$.

When N is odd, $m0.5 = X(N + 1)/2$. And when N is even,$m0.5 = \frac{X(N/2)+X(N/2+1)}{2}$.

The median has the following characteristics

(1) The median is obtained by sorting and is unaffected by the two extremes. Partial data changes have no impact on the median. When individual data in a group of data changes greatly, it is commonly used to describe the central trend of the group of data.
(2) The representation of the median will be affected when the degree distribution of some singular series of discrete variables is skewed.
(3) The median tends to be the middle of a set of sorted data.

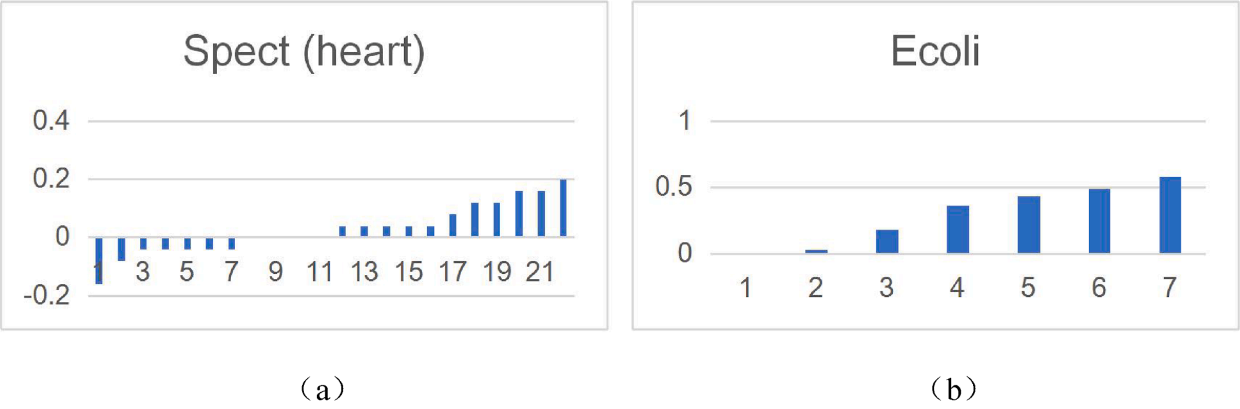
Through testing various types of data sets, we can find that the feature weight of most data sets presents exponential distribution. Therefore, the median of the sorted feature weight vector can be taken as the feature weight threshold, and half of the feature weight vector can be taken as the feature subset to construct the decision tree.

For example, the distribution of feature weights is observed by calculating feature weights. For twelve data sets in the experiments, their details are shown in Table 1.

The distributions of the feature weight values of the data set in above table are shown in Fig. 1.

Table 1
Data set details.

No	Data set	Characteristic number	Sample size	category	Is there a missing value	Area
1	Spect (heart)	23	267	2	no	Life
2	Ecoli	8(7)	336	7	no	Life
3	SPECTF	44	267	2	no	Life
4	Horse-Colic	27	368	2	yes	Life
5	Trial	17	776	2	yes	Life
6	Creadit Approvn	15	690	2	yes	Financial
7	Statlog(heart)	13	270	2	no	Life
8	Immunotherapy	7	90	2	no	Life
9	Cmc	9	1473	3	no	Life
10	Heart	13	270	2	no	Life
11	Breast-Cancer-Wisconsin	9	699	2	yes	Life
12	Bank Marketing	16	4521	2	no	Business



(a). Feature weight distribution for Spect(heart)

(b). Feature weight distribution for Ecoli

Fig. 1. Weight distributions of 12 data sets.

As shown in Fig. 1, we can observe the feature weight distribution of each data set. The abscissa represents the features of the data sets and the ordinate represents the feature weight values of each feature. Furthermore, the larger the feature weight value is, the more important the feature is to the category. When the feature weight value is smaller, it means that the feature is less important for the category. If the weight value of the feature is greater than zero, the feature is beneficial to the classification. Otherwise, when the feature weight value is less than zero, the feature has an adverse effect on the classification. By sorting the calculated feature weight values, the feature weight value corresponding to the median is selected as the threshold value to filter the features, the features with the highest correlation with the category can be screened out as the feature subset, and the features with negative impact on the classification or low correlation with the category can be deleted. If only half of the features with high relevance to the category are used to construct the decision tree, the scale of the decision tree will be simplified and the influence of some features with low relevance to the category will be reduced. It can improve the classification accuracy of decision tree.

The description of Improved ReliefF algorithm is shown as follows.

Algorithm 3: Improved ReliefF algorithm

```
Input: Training set $D$, feature set $A$, random sampling number $M$, nearest neighbor sample number $K$, initial feature weight $W$, the number of iterations $N$
Output: Feature subsets after feature pre filtering
1. For i = 1 to $N$ do
```

(continued on next column)

(continued)

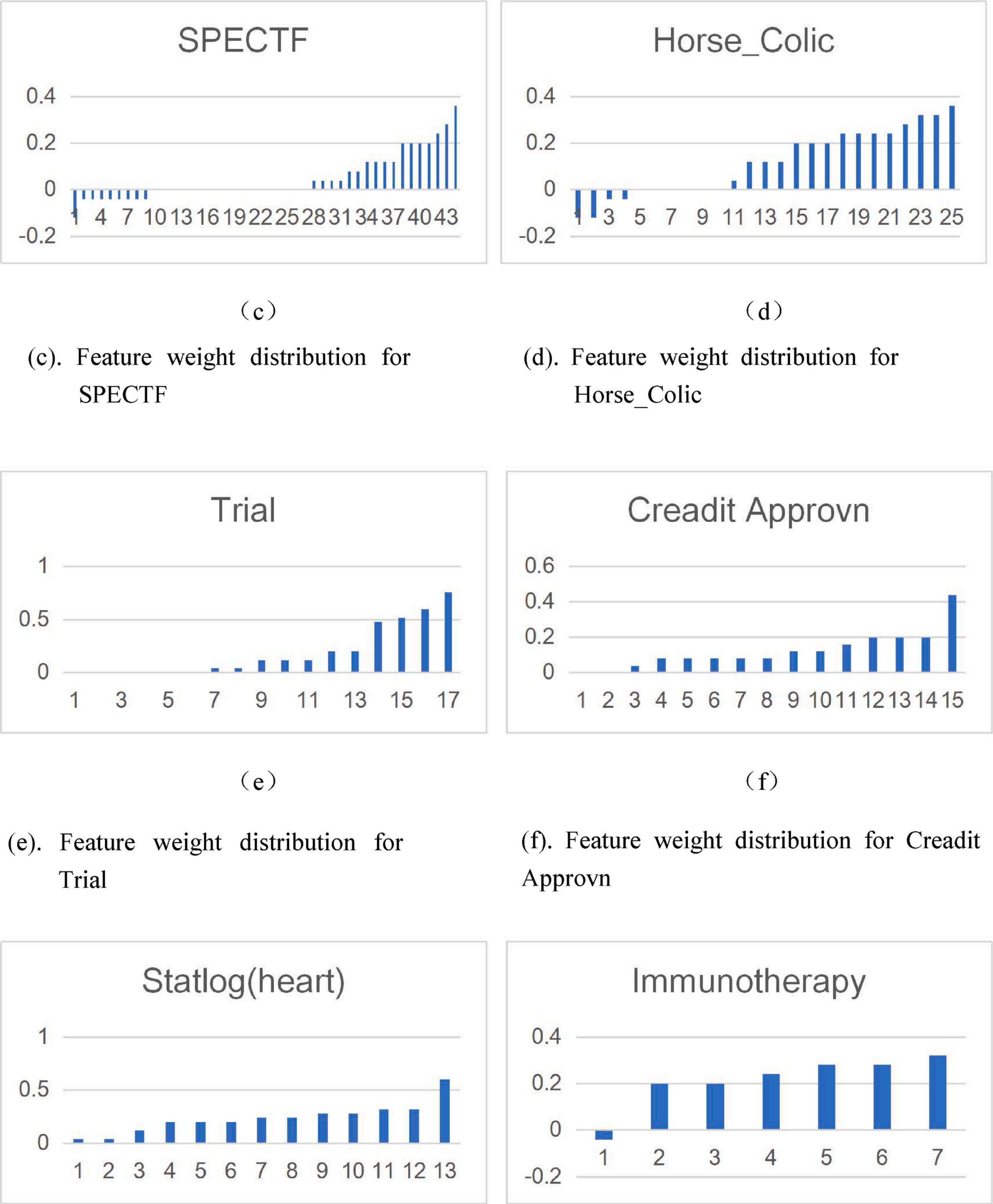
Algorithm 3: Improved ReliefF algorithm

```
2.  For j = 1 to $M$ do
3.      Randomly select a sample $R$ from training set $D$
4       Calculate the Euclidean distance between $R$ and each sample in the same sample set, and find out $K$ nearest neighbor samples
5.      Calculate the distance between each sample and $R$ in each sample set different from $R$, and find out $K$ nearest neighbor samples in each sample set different from $R$
6.      Calculate the feature weight $W(A)$ of each feature: $W(A) = W(A) - \sum_{j=1}^{k} diff(A, R, Hj)/m*k + \sum_{C\notin class(R)} [\frac{P(c)}{1-P(class(R))} \sum_{j=1}^{k} diff(A,R,Mj(c))]/m*k$
    End For
End For
7.  The calculated feature weight values of each feature are sorted from small to large to form the feature weight vector.
8.  If there are odd features in feature set $A$, the median of the feature weight vector is taken as the feature weight threshold $\beta$. Otherwise, if there are even features in feature set $A$, the average value of the middle two numbers of the feature weight vector is taken as the feature weight threshold $\beta$.
9.  The feature weight threshold obtained by calculation is used to filter the feature, and the feature whose feature weight value is less than $\beta$ in feature set $A$ is deleted, while the feature whose feature weight value is greater than or equal to $\beta$ is retained.
10. Output feature subsets after feature pre-filtering.
```

In this paper, the process of continuous feature discretization is described in Algorithm 4.

Algorithm 4: Continuous feature discretization method

(continued on next page)



（c）

(c). Feature weight distribution for SPECTF

（d）

(d). Feature weight distribution for Horse_Colic

（e）

(e). Feature weight distribution for Trial

（f）

(f). Feature weight distribution for Creadit Approvn

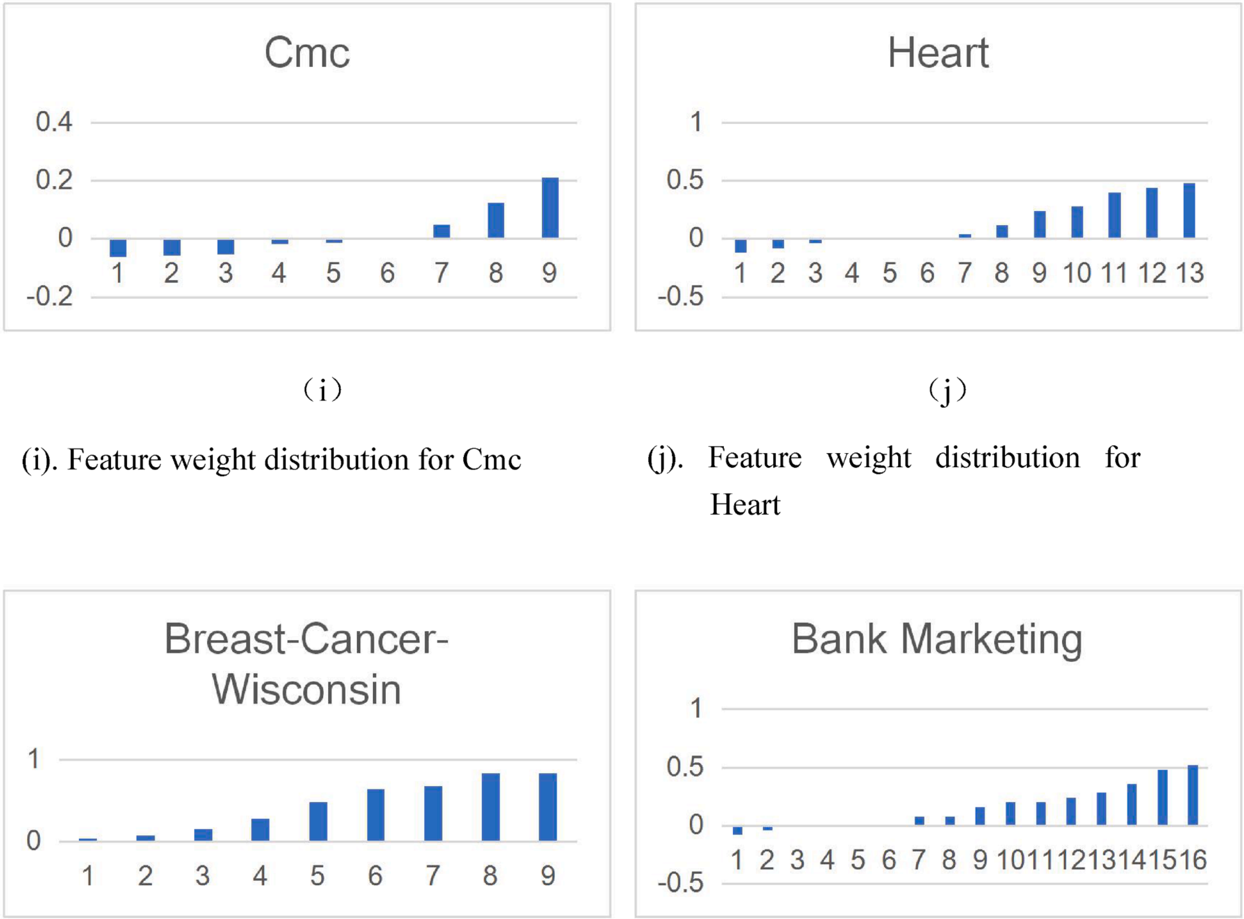
（g）

(g). Feature weight distribution for Statlog(heart)

（h）

(h).Feature weight distribution for Immunotherapy

Fig. 1. (*continued*).



（i）

(i). Feature weight distribution for Cmc

（j）

(j). Feature weight distribution for Heart

（k）

(k). Feature weight distribution for Breast-Cancer-Wisconsin

（l）

(l).Feature weight distribution for Bank Marketing

Fig. 1. (*continued*).

(*continued*)

```
Algorithm 4: Continuous feature discretization method
Input: Training set $D$, continuous feature set $F$, wherein feature set $F$ contains $S$ features
Output: Feature set $A$ after discretization
For i = 1 to $S$ do
    Calculate the $K$ value of each continuous feature:
        For each continuous feature, set the $k$ value from 2 to 10
        For $k = 2$ to 10 do
        Firstly, $K$ cluster centers are determined randomly, and training set $D$ is clustered into $K$ clusters
        Select a sample $i$ randomly in training set $D$, and calculate the Euclidean distance between $i$ and other samples in the same cluster average $a(i)$.At the same time, calculate the average value $b(i)$ of the distance between sample $i$ and each sample in other clusters.
Calculation$s(i) = \frac{b(i) - a(i)}{max(b(i), a(i))}$
        End For
        The $k$ value corresponding to the maximum $s(i)$ value is taken as the $k$ value of the continuous feature
        The training set $D$ is clustered into $K$ clusters, and the continuous characteristic $Si$ is discretized
End For
Put the discretized feature back into feature set $A$
Output feature set $A$ after discretization
```

3.3. Feature selection framework and classifier

At present, there are three types of feature selection methods which are filter, wrapper and embedded ones (Kira, 1992). Our proposed method combines the filtering and embedded ones. Because the main advantages of the filtering feature selection algorithm are fast computation speed and low complexity, it is difficult to determine whether the features selected can maximize the classification accuracy of a specific classifier. Therefore, we use this method to delete the irrelevant features and retain the more important features to construct the decision tree. In the process of constructing decision tree, feature weight is selected as the standard of selecting and dividing sample features to construct decision tree recursively. At the same time, decision tree based on information gain (ID3), decision tree based on information gain rate (C4.5), decision tree based on Gini index (CART) and decision tree based on discrete rate (DRDT) are used as the comparison algorithm. The decision tree constructed by different standards is used for predict the test set, calculate the classification accuracy, recall, F1-score and the time required to construct a decision tree. The feature selection framework in this paper is shown in Fig. 2.

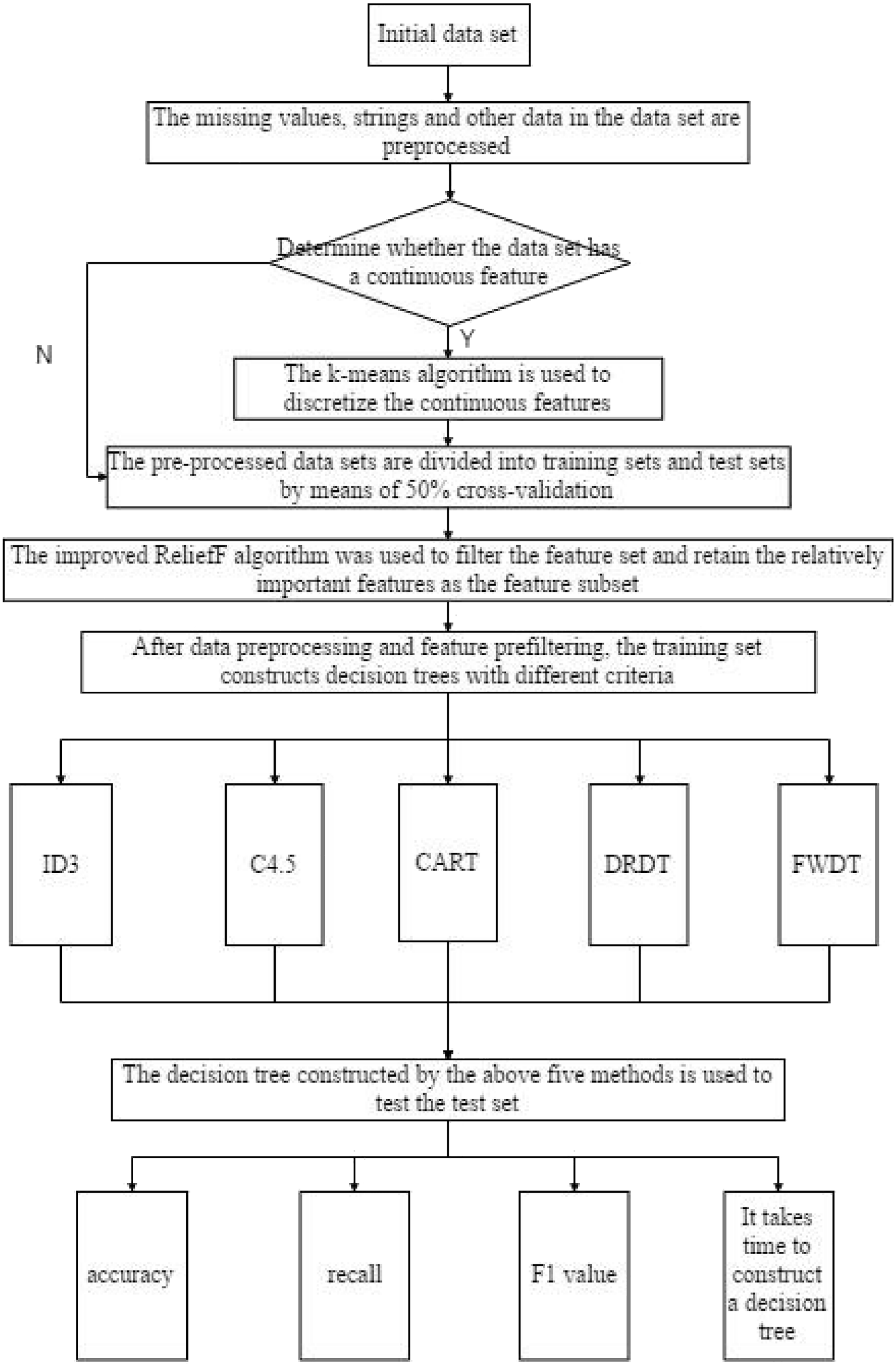

Initial data set
The missing values, strings and other data in the data set are preprocessed
Determine whether the data set has a continuous feature
N
Y
The k-means algorithm is used to discretize the continuous features
The pre-processed data sets are divided into training sets and test sets by means of 50% cross-validation
The improved ReliefF algorithm was used to filter the feature set and retain the relatively important features as the feature subset
After data preprocessing and feature prefiltering, the training set constructs decision trees with different criteria
ID3
C4.5
CART
DRDT
FWDT
The decision tree constructed by the above five methods is used to test the test set
accuracy
recall
F1 value
It takes time to construct a decision tree

Fig. 2. Feature selection framework.

4. Experimental evaluation

4.1. Experiment setup

In the experiment, our platform is Lenovo ThinkPad T520 notebook. The hardware condition is Intel (R) core i5-2520 m CPU, 2.50 GHz processing speed and 8 GB memory size. The operating system is 64-bit Windows 7, the experimental environment is Spyder, and the experimental language is Python.

In the experiment, classification accuracy, recall, F1-score and the time taken in constructing decision tree are all used as the evaluation indexes. The higher the classification accuracy, recall and F1-score, the better the feature selection algorithm is. And the less time in constructing the decision tree indicates the better performance of the algorithm. In the experiment, IG, GI, GR and DR are used as contrast algorithms. In the experiment, in order to verify the effectiveness and generalization of the pre-filtering method, five kinds of decision tree algorithms were used to conduct a comparison experiment under the two conditions of using the pre-filtering method and not using the pre-filtering method.

Experimental parameter settings are shown as follows. In the improved ReliefF algorithm, the random sample number M and the nearest neighbor sample number K are set to be 5, and the initial feature weight value W is set to be 0. The experiment was carried out by using the 5-fold cross validation method.

4.2. Data set

The data sets used in the experiment are all real data sets, all of which are from UCI machine learning database (Roy et al., 2019b). They are from medical, commercial, engineering, biological and other fields. The number of samples in these data sets ranges from 90 to 4521, and the number of features ranges from 7 to 44. These data sets contain continuous and discrete features, and some contain missing values. The data set details are shown in Table 2.

4.3. Experimental results and discussions

Through the experiments on 12 data sets, compared with pre-filtering strategy and unused pre filtering strategy in both cases, five different decision tree algorithm in classification accuracy rate and recall rate, F1-score, to construct the decision tree takes time the four aspects of the performance, analysis in two different cases, the performance of five different decision tree.

4.3.1. Accuracy assessment

Firstly, the performance of five different decision tree algorithms in classification accuracy was compared under the two conditions of using the pre-filtering method and not using the pre- filtering strategy. Table 3 shows the classification accuracy of five decision trees under the premise of using and not using the pre-filtering method on 12 data sets. The first line in each data set shows the classification accuracy without the pre-filtering method, and the second line shows the classification accuracy with the pre-filtering method. If the classification accuracy after using the filtering method is greater than that without using the filtering method, the data will be shown in red. The first row of each data set in the table represents the classification accuracy without the pre-filtering method, and the second row represents the classification accuracy with the pre-filtering method. The maximum value of each row is shown in bold font.

Table 3
Classification accuracy with and without pre-filtering.

Data Set	CART	ID3	C4.5	DRDT	FWDT
Spect (heart)	43.480%	**61.824%**	60.713%	61.502%	51.356%
	61.461%	64.808%	**67.442%**	62.572%	63.669%
Ecoli	**76.760%**	35.435%	61.277%	44.925%	75.268%
	67.256%	64.583%	64.289%	65.171%	**68.450%**
SPECTF	51.698%	61.062%	**71.139%**	65.164%	60.273%
	60.650%	65.164%	**69.266%**	68.875%	65.220%
Horse-Colic	40.333%	74.333%	**77.667%**	67.000%	69.333%
	63.667%	63.333%	64.000%	59.667%	**69.667%**
Trial	89.176%	100%	100%	100%	100%
	99.743%	100%	100%	100%	100%
Creadit Approvn	59.420%	72.754%	74.203%	62.464%	**74.348%**
	83.043%	65.362%	68.261%	64.058%	**83.913%**
Statlog(heart)	57.037%	68.148%	**70.741%**	64.815%	65.556%
	63.333%	70.000%	**72.741%**	67.037%	67.778%
Immunotherapy	54.444%	58.889%	58.889%	53.333%	**62.222%**
	75.556%	70.000%	70.000%	71.111%	**77.778%**
Cmc	29.192%	36.321%	35.982%	21.928%	**38.762%**
	41.169%	39.381%	**41.755%**	37.277%	41.619%
Heart	57.037%	68.148%	**70.741%**	65.926%	63.333%
	64.074%	68.889%	**71.370%**	67.296%	69.630%
Breast-Cancer-Wisconsin	89.704%	**90.135%**	88.700%	84.695%	90.125%
	87.842%	89.845%	89.558%	89.696%	**90.129%**
Bank Marketing	72.373%	81.730%	**82.305%**	78.501%	80.513%
	82.902%	74.849%	82.598%	76.819%	**83.986%**
Average	60.055%	67.398%	**71.030%**	64.188%	69.257%
	70.891%	69.685%	71.773%	69.132%	**73.487%**

It can be seen in Table 3 that the five decision tree algorithms using the pre-filtering method have improved the classification accuracy. In 12 data sets, when no pre-filtering method is used, CART algorithm has the highest classification accuracy for 1 data set, ID3 algorithm has the highest classification accuracy for 2 data sets, C4.5 algorithm has the highest classification accuracy for 5 data sets, DRDT algorithm has the highest classification accuracy without data sets, and FWDT algorithm has the highest classification accuracy for 3 data sets. After using the pre-filtering method, 5 data sets have the highest classification accuracy for C4.5 algorithm, 6 data sets have the highest classification accuracy for FWDT algorithm. Fig. 3 shows the average classification accuracy of different decision tree algorithms on 12 data sets by using and not using pre-filtering methods.

As shown in Table 3 and Fig. 3, the classification accuracy of the five decision tree algorithms has been improved after using the pre-filtering

Table 2
Details of data set.

No	Data set	Characteristic number	Sample size	Category	Missing value	Area
1	Spect (heart)	23	267	2	no	Life
2	Ecoli	8(7)	336	7	no	Life
3	SPECTF	44	267	2	no	Life
4	Horse-Colic	27	368	2	yes	Life
5	Trial	17	776	2	yes	Life
6	Creadit Approvn	15	690	2	yes	Financial
7	Statlog(heart)	13	270	2	no	Life
8	Immunotherapy	7	90	2	no	Life
9	Cmc	9	1473	3	no	Life
10	Heart	13	270	2	no	Life
11	Breast-Cancer-Wisconsin	9	699	2	yes	Life
12	Bank Marketing	16	4521	2	no	Business

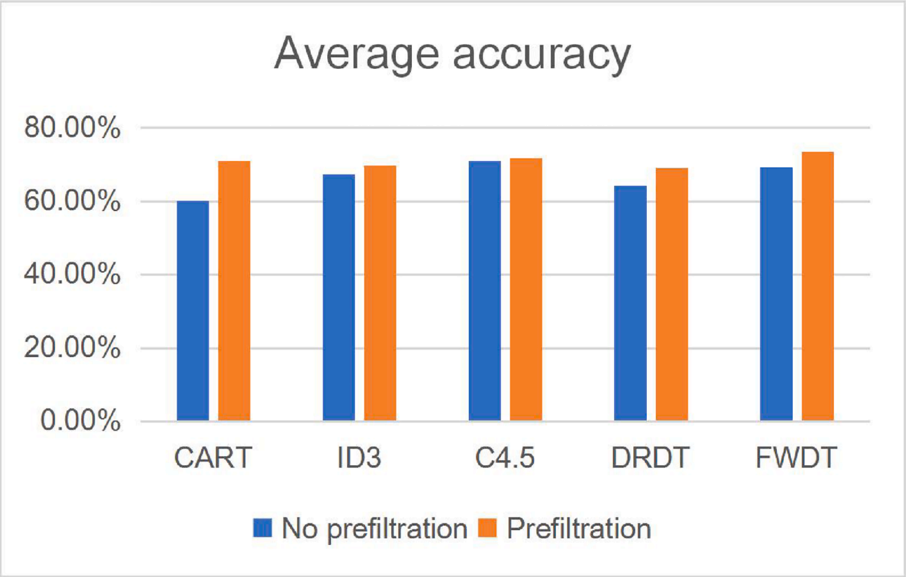


Fig. 3. Comparison of average classification accuracy.

method. Among them, the average classification accuracy of CART algorithm is improved by 10.83%, ID3 algorithm by 2.29%, C4.5 algorithm by 7.43%, DRDT algorithm by 4.944%, and FWDT algorithm by 4.23%.

As shown in Table 3, after using the pre-filtering method, FWDT algorithm has the highest average classification accuracy of 73.487% on the 12 data sets.

As shown in Table 3, FWDT algorithm has the average classification accuracy of 9.2% higher than CART algorithm, 1.86% higher than ID3 algorithm, and 5.07% higher than DRDT algorithm. Compared with C4.5 algorithm, it has a slight decline. After using pre-filtering method, FWDT algorithm has the average classification accuracy of 2.6% higher than CART algorithm, 1.86% higher than ID3 algorithm, and 1.71% higher than C4.5 algorithm, 4.35% higher than DRDT algorithm. It shows that the effectiveness of the pre-filtering strategy is extensive in FWDT.

In 12 data sets, the performance of five different decision tree algorithms increased, leveled, and decreased after using the pre-filtering method and before using the pre-filtering method, respectively. Table 4 shows the number of data sets whose classification accuracy increase with pre-filtering.

As shown in Table 4, after using the pre-filtering method, the classification accuracy of the five decision tree algorithms in most data sets was improved. Among them, FWDT algorithm has the obvious superiority in 11 data sets.

Table 5 shows the number of data sets with the highest, lowest or intermediate classification accuracy of each decision tree algorithm compared with the other four algorithms in 12 data sets under two conditions. The left column of each algorithm in the table is the case where the pre-filtering method is not used, and the right column is the case where the pre-filtering method is used.

As can be seen from Table 5, when no pre-filtering method is used, C4.5 algorithm has the best classification accuracy in 6 of the 12 data sets, and the overall effect of C4.5 algorithm is the best. After using the pre-filtering method, FWDT algorithm in this paper has the best classification accuracy in 7 of the 12 data sets. On the whole, the classification effect of FWDT algorithm is better.

Table 4
Number of data sets with the improved\flat\falling classification accuracy with pre-filtering.

	CART	ID3	C4.5	DRDT	FWDT
Rising	10	7	8	9	11
Flat	0	1	1	1	1
Falling	2	4	3	2	0

Table 5
Number of data sets with the best\middle\worst classification accuracy.

	CART		ID3		C4.5		DRDT		FWDT	
Best	1	2	3	1	6	6	1	1	4	7
Middle	2	6	8	9	6	4	9	8	8	5
Worst	9	4	1	2	0	2	2	3	0	0

It can be seen in Table 3 that, the average classification accuracy of C4.5 algorithm is the highest (71.030%) which is 1.773% higher than that of FWDT algorithm before pre-filtering. C4.5 use information gain rate to select the splitting attributes and it avoids the problem that other algorithms tend to select attributes with more attribute values as splitting attributes. Additionally, it can handle missing attribute values, All of these make the classification accuracy be the highest. After pre-filtering, the average classification accuracy of the five decision tree algorithms on 12 data sets are increased by 10.836%, 2.287%, 0.743%, 4.944% and 4.23%, respectively. It can be seen that the feature pre-filtering method proposed in this paper is effective in improving the classification accuracy of the five algorithms. Although the average classification accuracy of C4.5 is the highest before prefiltering, the average classification accuracy is also improved after pre-filtering. This indicates the effectiveness and stability of the prefiltering method in this paper. In addition, the average classification accuracy of FWDT algorithm after pre-filtering is higher than that of C4.5 algorithm, reaching the best (73.487%).

4.3.2. Recall assessment

Table 6 shows the recall rates of five different decision tree algorithms with and without pre-filtering methods. If the recall rate with the pre-filter method is greater than that without the pre-filtering method, it is indicated in red. The first row of each data set in the table represents the recall rates without the pre-filtering method, and the second row represents the recall rates with the pre-filtering method. The maximum value of each row is shown in bold.

As shown in Table 6, when the five decision tree algorithms did not use the pre-filtering method, in 12 data sets, the data set with the highest

Table 6
Recall rates with and without pre-filtering.

Data Set	CART	ID3	C4.5	DRDT	FWDT
Spect (heart)	25.107%	**33.397%**	33.124%	31.691%	28.281%
	32.505%	37.380%	37.635%	**38.004%**	35.905%
Ecoli	**45.246%**	15.638%	40.427%	29.330%	44.571%
	43.907%	**45.285%**	**45.285%**	**45.285%**	44.506%
SPECTF	27.857%	30.872%	**36.427%**	34.208%	31.433%
	31.901%	34.630%	36.883%	**37.259%**	31.984%
Horse-Colic	36.385%	**38.200%**	37.334%	35.733%	35.191%
	34.539%	30.463%	29.884%	28.765%	**37.800%**
Trial	38.152%	44.059%	44.059%	44.059%	44.059%
	43.826%	44.059%	44.059%	44.059%	44.059%
Creadit Approvn	30.010%	35.476%	**37.004%**	29.538%	35.523%
	39.838%	20.119%	21.562%	19.827%	39.513%
Statlog(heart)	27.833%	**33.778%**	33.212%	28.677%	33.737%
	33.980%	35.494%	35.798%	34.513%	**35.562%**
Immunotherapy	29.904%	33.349%	33.023%	28.054%	**33.350%**
	40.065%	33.956%	33.956%	35.450%	**40.755%**
Cmc	18.107%	20.919%	20.559%	13.841%	**22.642%**
	26.849%	23.466%	**26.866%**	21.468%	26.795%
Heart	27.833%	**34.155%**	33.212%	28.677%	30.031%
	34.773%	**36.232%**	**36.232%**	33.541%	35.477%
Breast-Cancer-Wisconsin	42.057%	**42.733%**	42.507%	40.787%	42.547%
	42.360%	**43.142%**	42.733%	42.872%	42.711%
Bank Marketing	37.010%	**40.498%**	40.471%	39.103%	40.142%
	41.577%	38.088%	40.656%	39.014%	**41.996%**
Average	32.13%	33.59%	**35.95%**	31.98%	35.13%
	37.18%	35.19%	35.96%	35.01%	**38.09%**

recall rate of CART algorithm was 1, ID3 algorithm had the highest recall rate of 6 data sets, C4.5 algorithm had the highest recall rate of 2 data sets, and FWDT algorithm had the highest recall rate of 2 data sets. In the Trial data set, there are four algorithms with equal recall rates. After using the pre-filtering method, CART algorithm has the highest recall rate for 1 data set, ID3 algorithm has the highest recall rate for 3 data sets, C4.5 algorithm has the highest recall rate for 3 data sets, DRDT algorithm has the highest recall rate for 3 data sets, and FWDT algorithm has the highest recall rate for 4 data sets. There are four algorithms in the Trial data set with equal recall rates.

As can be seen from Table 6, when no pre-filtering method is used, the average recall rate of FWDT algorithm on the 12 data sets is 3% higher than CART algorithm, 1.54% higher than ID3 algorithm, and 3.15% higher than DRDT algorithm. Compared with C4.5 algorithm, it is basically flat. When the pre-filtering method is used, the average recall rate of FWDT algorithm in this paper is 0.91% higher than CART algorithm, 2.9% higher than ID3 algorithm, 2.13% higher than C4.5 algorithm, and 3.08% higher than DRDT algorithm under the same conditions. It shows that the pre-filtering method in this paper has a good effect on improving the recall rate.

The average recall rates of the five decision tree algorithms were compared over 12 data sets with and without the pre-filtering method.

As shown in Fig. 4, the average recall rate of the five decision tree algorithms on the 12 data sets was improved to varying degrees after the pre-filtering method was used, indicating that the pre-filtering method has an effect on different algorithms. According to the analysis of Table 6 and Fig. 4, the average recall rate of FWDT algorithm proposed in this paper is the highest on the 12 data sets, which is 38.09%, after using the pre-filtering method.

Table 6 shows that the average recall rate of CART algorithm on the 12 data sets increased by 5.05%, ID3 algorithm by 1.60%, C4.5 algorithm by 0.015%, DRDT algorithm by 3.03%, and FWDT algorithm by 2.96% after using the pre-filtering method for the five algorithms.

In 12 data sets, the number of data sets whose performance was improved, flat, or reduced by the five decision tree algorithms in terms of recall rate after the use of the pre-filtering method was calculated, compared with that before the use of the pre-filtering method. Table 7 shows the number of data sets whose recall rates increase with pre-filtering.

According to Table 7, the improved pre-filtering method proposed in this paper has a certain improvement effect on the recall rate of the five algorithms. Recall rates increased for most of the 12 data sets. FWDT algorithm proposed in this paper has the best performance.

Table 8 shows the number of data sets with the highest, lowest or intermediate classification accuracy of each decision tree algorithm compared with the other four algorithms in 12 data sets under two conditions. The left column of each algorithm in the table is the case where the pre-filtering method is not used, and the right column is the

Table 7
Number of data sets with the improved\flat\falling recall rate with pre-filtering.

	CART	ID3	C4.5	DRDT	FWDT
Rising	10	8	9	8	10
Flat	0	1	1	1	1
Falling	2	3	2	3	1

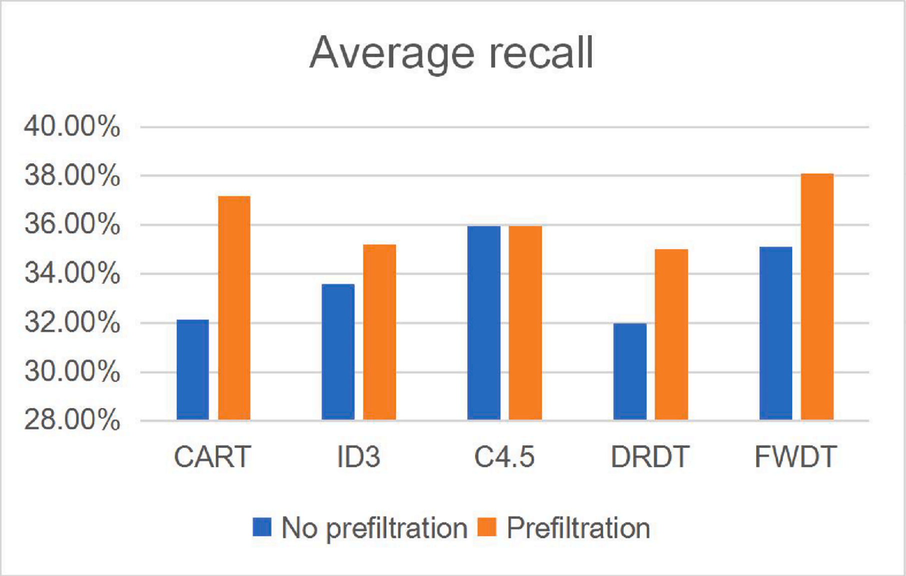


Fig. 4. Comparison of average recall rates.

Table 8
Number of data sets with the best\middle\worst recall rate.

	CART		ID3		C4.5		DRDT		FWDT	
Best	1	1	8	4	3	5	1	4	3	4
Middle	5	5	3	7	9	7	7	5	8	8
Worst	6	6	1	1	0	0	4	3	1	0

case where the pre-filtering method is used.

As can be seen from Table 8, when no pre-filtering method is used, ID3 algorithm has the best recall rate in 8 of the 12 data sets, indicating that ID3 algorithm has a better recall rate effect in 12 data sets. When the pre-filtering method is used, C4.5 algorithm has the best recall rate in 5 of the 12 data sets, while FWDT algorithm in this paper has the best recall rate in 4 data sets and no data set with the worst recall rate. In general, the effect of C4.5 algorithm is basically the same as that of FWDT algorithm. As shown in Fig. 4 and Table 7, FWDT has the best performance.

As can be seen from Table 6, the average recall rate of C4.5 algorithm is the highest, reaching 35.95% before pre-filtering. Because C4.5 algorithm use information gain rate as the criterion for choosing split attributes, it can avoid the problem of preference to select attributes with multiple attribute values as classification attributes, and it can handle missing values, so the recall rate is the highest among the five decision trees. After pre-filtering, the average recall rate of the five decision tree algorithms on 12 data sets are increased by 5.05%, 1.6%, 0.01%, 3.03% and 2.96%, respectively. It can be seen that the feature pre-filtering method in this paper improves the recall rate of the five algorithms to different degrees, indicating that the feature pre-filtering method is effective and stable. After pre-filtering, the average recall rate of FWDT algorithm exceeded that of C4.5 algorithm, reaching the best, is 38.09%.

4.3.3. F1-score assessment

F1-score of the five decision tree algorithms was calculated respectively with and without the pre-filtering method. When the F1-score calculated using the pre-filtering method is greater than the F1-score calculated without the pre-filtering method, the calculated result is shown in red. Table 9 shows the F1-score comparison of five decision tree algorithms in two cases. The first row of each data set in the table represents the F1-score without the pre-filtering method, and the second row represents the F1-score with the pre-filtering method. The maximum value of each row is shown in bold.

Table 9
F1-score with and without pre-filtering.

Data set	CART	ID3	C4.5	DRDT	FWDT
Spect (heart)	32.230%	**39.993%**	39.735%	39.041%	34.952%
	41.067%	43.350%	**44.201%**	43.449%	42.893%
Ecoli	**56.072%**	24.337%	49.574%	38.605%	56.003%
	49.938%	50.225%	50.225%	49.746%	**50.904%**
SPECTF	38.732%	41.801%	**46.557%**	44.889%	41.856%
	43.334%	45.295%	47.127%	**47.751%**	43.343%
Horse-Colic	45.229%	**48.015%**	47.566%	44.774%	44.877%
	41.378%	40.745%	40.585%	39.802%	**47.879%**
Trial	50.432%	56.286%	56.286%	56.286%	56.286%
	56.126%	56.286%	56.286%	56.286%	56.286%
Creadit Approvn	39.646%	45.576%	46.215%	37.928%	**46.302%**
	50.094%	31.240%	33.498%	30.429%	**50.337%**
Statlog(heart)	37.995%	43.041%	42.985%	41.235%	**43.109%**
	44.923%	**45.339%**	45.168%	45.110%	43.562%
Immunotherapy	39.956%	42.498%	42.843%	38.276%	**44.297%**
	49.800%	44.984%	44.984%	46.304%	**50.721%**
Cmc	25.801%	29.050%	28.636%	21.591%	**29.965%**
	30.565%	29.451%	30.544%	27.593%	30.496%
Heart	37.995%	**43.391%**	42.985%	41.275%	39.857%
	44.950%	**45.107%**	44.841%	44.088%	44.824%
Breast-Cancer-Wisconsin	54.330%	**54.845%**	54.637%	53.281%	54.544%
	54.685%	**55.198%**	54.877%	55.029%	54.882%
Bank Marketing	48.627%	52.082%	**52.184%**	51.810%	50.847%
	53.035%	49.662%	52.606%	50.738%	**53.408%**
Average	42.25%	43.41%	**45.85%**	42.42%	45.24%
	46.66%	44.74%	45.41%	44.69%	**47.46%**

Through the analysis of Table 9, when the pre-filtering method is not used, among the five decision tree algorithms, CART algorithm has the highest F1-score of 1 data set in 12 data sets, ID3 algorithm has the highest F1-score of 4 data sets, C4.5 algorithm has the highest F1-score of 2 data sets, and FWDT algorithm has the highest F1-score of 4 data sets. In the Trial data set, F1-score of four algorithms is equal. After the pre-filtering method was used, the performance of five decision tree algorithms in 12 data sets was highest in F1-score for 1 data set of CART algorithm, F1-score for 2 data sets of ID3 algorithm, F1-score for 1 data set of C4.5 algorithm, F1-score for 1 data set of DRDT algorithm and F1-socre for 4 data sets of F1-socre algorithm. There are four algorithms whose F1-scores are equal on 1 data set.

In the two cases, the average F1-score of the five decision tree algorithms on 12 data sets was compared, and the F1-score of the five algorithms was observed after the pre-filtering method was used. Fig. 5 shows the changes of the five algorithms in F1-score before and after the pre-filtering method is used.

As can be seen from Fig. 5, after using the pre-filtering method, the average F1-score of four of the five decision tree algorithms in the 12 data sets was improved significantly, and the average F1-score of one algorithm showed a small decrease. It shows that the improved pre-filtering method proposed in this paper can improve the F1-score of most decision tree algorithms. As can be seen from Table 9, the average F1-score of CART algorithm improved by 4.404%, ID3 algorithm improved by 1.33%, DRDT algorithm improved by 2.278%, FWDT algorithm improved by 2.22%, and only C4.5 algorithm decreased by 0.438%.

The average F1-score of the proposed FWDT algorithm on 12 data sets is the best among the five algorithms after using the pre-filtering method, which is 47.76%. It is 0.803%, 2.72%, 2.05% and 2.77% higher than the other four algorithms under the same condition. Without the pre-filtering method, the average F1-score of FWDT algorithm proposed in this paper is 2.99% higher than CART algorithm, 1.83% higher than ID3 algorithm, and 2.83% higher than DRDT algorithm on the 12 data sets under the same conditions, which is basically the same as C4.5 algorithm.

The F1-score of the five decision tree algorithms was improved after the pre-filtering method was used. Table 10 shows the number of data sets whose F1-score increase with pre-filtering.

As can be seen from Table 10, after using the pre-filtering method, the five decision tree algorithms were improved in most F1-scores of the 12 data sets, indicating that the improved pre-filtering method proposed in this paper has extensive effectiveness. In general, the proposed FWDT algorithm is better among the five decision tree algorithms.

Each of the five decision tree algorithms was compared for the number of F1-score best, worst, or intermediate data sets on the 12 data sets without and without pre-filtering, respectively. The left column of each algorithm represents the case where the pre-filtering method is not used, and the right column represents the case where the pre-filtering method is used.

As can be seen from Table 11, when no pre-filtering method is used, FWDT algorithm has the best f1-score in 5 of the 12 data sets, and the worst F1-score without data sets. F1-score of FWDT algorithm with 6 data sets is the best after using the pre-filtering method. Compared with other algorithms, FWDT algorithm has the best effect in F1-score, and the pre-filtering method used at the same time plays a role in improving F1-score.

As can be seen from Table 9, the average F1-score of C4.5 algorithm is the highest, reaching 45.85%, which is 0.61% higher than that of FWDT algorithm before pre-filtering. Because C4.5 algorithm takes information gain rate as the standard for attribute selection, it avoids the problem that other algorithms tend to choose attributes with many

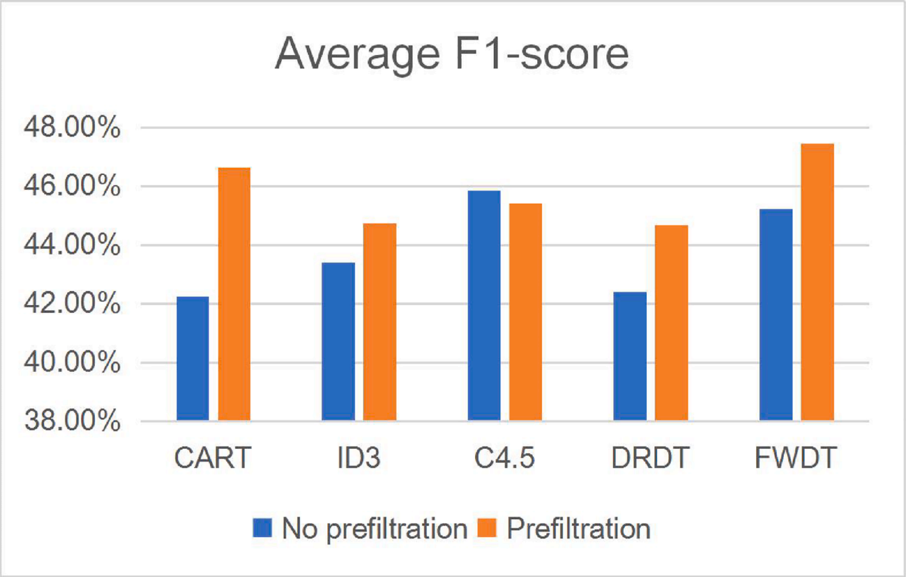


Fig. 5. Comparison of average F1-score.

Table 10
Number of data sets with the improved\flat\falling F1-score with pre-filtering.

	CART	ID3	C4.5	DRDT	FWDT
Rising	10	8	9	8	10
Flat	0	1	1	1	1
Falling	2	3	2	3	1

Table 11
Number of data sets with the best\middle\worst F1-score.

	CART		ID3		C4.5		DRDT		FWDT	
Best	1	1	5	4	3	2	2	2	5	6
Middle	5	7	6	7	9	9	6	5	7	5
Worst	6	4	1	1	0	1	4	5	0	1

attribute values, and it can handle missing attribute values and continuous attribute values, so it has the highest F1-score among the five algorithms. After pre-filtering, the average F1-score of CART, ID3 and DRDT algorithms on 12 data sets are increased by 4.41%, 1.33%, 2.27% and 2.22%, respectively. Although the average F1-score of C4.5 algorithm after pre-filtering decreased by 0.44%, it can be seen from Table 10 that after pre-filtering, F1-score of C4.5 algorithm in 9 data sets of 12 data sets is improved, and F1-score of C4.5 algorithm is also improved in general. The effectiveness of the feature pre-filtering method in this paper is illustrated. In addition, after pre-filtering, the average F1-score of FWDT algorithm exceeds that of C4.5 algorithm and reaches the highest (47.46%).

4.3.4. Time assessment

The time taken by the five decision tree algorithms to construct the decision tree is calculated when the pre-filtering method is used and the pre-filtering method is not used. Table 12 shows the construction times of different decision trees in both cases. The red color in the table indicates that the construction time of the decision tree after using the pre-filtering method is less than the construction time without the pre-filtering method. The first row of each data set in the table represents the time without the pre-filtering method, and the second row represents the time with the pre-filtering method. The maximum value of each row is shown in bold.

Table 12
Time assessment with and without pre-filtering.

Data set	CART	ID3	C4.5	DRDT	FWDT
Spect (heart)	0.141	0.113	**0.104**	29.204	29.155
	0.056	0.046	**0.043**	8.865	10.257
Ecoli	0.039	0.038	**0.028**	6.456	23.435
	0.010	0.012	0.011	1.543	6.663
SPECTF	0.379	**0.181**	0.314	47.932	38.989
	0.114	**0.086**	0.122	19.977	20.172
Horse-Colic	0.184	**0.111**	0.117	23.585	46.480
	0.069	0.049	**0.048**	7.849	22.049
Trial	0.202	0.101	**0.094**	0.315	15.725
	0.078	0.040	**0.039**	0.164	7.096
Creadit Approvn	0.246	**0.143**	0.158	31.193	29.924
	0.061	**0.059**	0.069	6.012	12.555
Statlog(heart)	0.141	0.113	**0.104**	29.204	29.155
	0.056	0.046	**0.043**	8.865	10.257
Immunotherapy	**0.007**	0.009	0.008	1.625	0.817
	0.003	0.007	**0.003**	0.266	0.305
Cmc	0.231	**0.189**	0.219	37.801	72.876
	0.064	0.074	**0.061**	2.002	20.152
Heart	0.078	0.056	**0.049**	12.426	9.160
	0.020	0.025	0.025	3.374	3.341
Breast-Cancer-Wisconsin	**0.082**	0.083	0.092	5.584	10.626
	0.033	0.040	0.042	1.477	4.222
Bank Marketing	1.479	**1.073**	1.379	102.016	599.353
	0.471	0.514	0.497	20.785	179.227
Average	0.267	**0.184**	0.222	27.278	75.475
	0.086	**0.083**	0.084	6.765	24.691

Table 12 shows that when no pre-filtering method is used, five decision tree algorithms construct the average time of a decision tree on 12 data sets. The decision tree construction time of CART algorithm with 2 data sets is the minimum, ID3 algorithm with 5 data sets is the minimum, and C4.5 algorithm with 5 data sets is the minimum. The decision tree construction time of DRDT algorithm and FWDT algorithm is two orders of magnitude higher than the first three algorithms. After using the pre-filtering method, among the five decision tree algorithms, CART algorithm has the shortest time to construct the decision tree with 5 data sets, ID3 algorithm has the shortest time to construct 2 data sets, and C4.5 algorithm has the shortest time to construct 5 data sets. The construction time of the remaining two DRDT and FWDT algorithms is two orders of magnitude higher than that of the first three.

It can be seen from Table 12, the time spent in constructing the decision tree by the five decision tree algorithms with the pre-filtering strategy has been greatly reduced. That is, it improves the speed of the training model. The average time taken by the five algorithms to construct the decision tree after using the pre-filtering method on 12 data sets is 1/3, 1/2, 1/3, 1/4 and 1/3 of that before using the pre-filtering method, indicating that the improved pre-filtering method proposed in this paper is effective.

Recurring to pre-filtering strategy, the time spent in constructing the decision tree by the five decision tree algorithms is reduced. Table 13 shows the number of data sets whose time is Rising\Flat\Falling after pre-filtering.

Table 13
Number of data sets with the improved\flat\falling time with pre-filtering.

	CART	ID3	C4.5	DRDT	FWDT
Rising	0	0	0	0	0
Flat	0	0	0	0	0
Falling	12	12	12	12	12

As can be seen in Table 13, when the pre-filtering method is used, the time taken by five decision tree algorithms to construct a decision tree on 12 data sets is reduced, and there are no equal or increased data sets. It shows that the pre-filtering method proposed in this paper has a good effect on reducing the construction time of decision tree. It can reduce the inaccuracy of the training model caused by the introduction of the features with low category correlation into the decision tree model, simplify the scale of the training model, and improve the construction speed of the training model.

It can be seen in Table 12 that the time spent by CART, ID3 and C4.5 algorithm in constructing decision tree are basically equal. DRDT and FWDT algorithms take a long time to construct the decision tree, and FWDT algorithm in this paper takes the longest time. This is related to the iteration number and nearest neighbor sample number when calculating feature weight. This problem will be improved in the following research to improve the speed of FWDT algorithm to construct the decision tree and reduce the time spent in constructing the decision tree while keeping other evaluation indexes unchanged.

5. Conclusion

First, this paper proposes an improved ReliefF algorithm as the data preprocessing stage feature pre-filtering method. Median was taken as the threshold value standard of ReliefF algorithm, which improved the problem that ReliefF algorithm needed to set the threshold value of ReliefF algorithm. At the same time, the feature weight is used as the feature selection criterion to construct the decision tree. In 12 UCI data sets, the accuracy, recall and F1-score calculated by five algorithms using pre-filtering method were improved to different degrees. The time to construct the decision tree is greatly reduced. In this paper, FWDT algorithm without filtering, the calculated accuracy, recall, F1-score and the optimal value are basically equal. After pre-filtering, the calculated accuracy, recall and F1-score are the best among the five algorithms. Through comparison, it was found that the pre-filtering method and FWDT algorithm in this paper were more suitable for the data sets with discrete attribute value type, sample number less than 1000, number of attributes within 100, and category dichotomy.

The experimental results show that FWDT algorithm in this paper takes a long time to construct the decision tree. This is mainly related to the number of iterations and the determination of the nearest neighbor sample number. Therefore, this problem will be improved in the following research to reduce the time required by the algorithm to build the decision tree while other evaluation indexes remain unchanged. At the same time, the classification of unbalanced data sets is becoming more and more important. In the following work, we will focus on the use of decision tree algorithm to classify unbalanced data sets, and improve the existing unbalanced data set processing algorithm.

CRediT authorship contribution statement

HongFang Zhou: Conceptualization, Methodology, Writing - original draft. **JiaWei Zhang:** Software, Validation, Formal analysis, Writing - original draft. **YueQing Zhou:** Investigation. **XiaoJie Guo:** Writing - original draft. **YiMing Ma:** Writing - original draft.

Declaration of Competing Interest

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

Acknowledgements

The corresponding author would like to thank the support from the National Key Research and Development Plan under the Grant of 2017YFB1402103, the National Natural Science Foundation of China under the Grant of 61402363 and 61771387, the Education Department of Shaanxi Province Key Laboratory Project under the Grant of 15JS079, Xi'an Science Program Project under the Grant of 2020KJRC0094, the Ministry of Education of Shaanxi Province Research Project under the Grant of 17JK0534, and Beilin district of Xi'an Science and Technology Project under the Grant of GX1625.

References

Alazab, A., Hobbs, M., Abawajy, J., & Alazab, M. (2012). Using feature selection for intrusion detection system. 2012 International Symposium on Communications and Information Technologies (ISCIT), 296–301. https://doi.org/10.1109/ISCIT.2012.6380910.

Amiri, F., Rezaei Yousefi, M., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications, 34*(4), 1184–1199. https://doi.org/10.1016/j.jnca.2011.01.002.

Ball, N. M., & Brunner, R. J. (2010). Data mining and machine learning in astronomy. *International Journal of Modern Physics D, 19*(07), 1049–1106. https://doi.org/10.1142/S0218271810017160.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*(1–2), 245–271. https://doi.org/10.1016/S0004-3702(97)00063-5.

Cai, J., Luo, J., Wang, S., & Yang, S. (n.d.). Feature selection in machine learning: a new perspective. Neurocomputing, S0925231218302911.

Chenwen, W., Chenyang, L., Shujin, G., & Guanghui, Y. (2018). Feature gene selection method based on ReliefF and ant colony optimization. *Application Research of Computers*, 2610–2613.

Gao, W., Hu, L., Zhang, P., & He, J. (2018). Feature selection considering the composition of feature relevancy. *Pattern Recognition Letters, 112*, 70–74. https://doi.org/10.1016/j.patrec.2018.06.005.

Gao, W., Hu, L., Zhang, P., & Wang, F. (2018). Feature selection by integrating two groups of feature evaluation criteria. *Expert Systems with Applications, 110*, 11–19. https://doi.org/10.1016/j.eswa.2018.05.029.

Gao, Y.-F., Li, B.-Q., Cai, Y.-D., Feng, K.-Y., Li, Z.-D., & Jiang, Y. (2013). Prediction of active sites of enzymes by maximum relevance minimum redundancy (mRMR) feature selection. *Molecular BioSystems, 9*(1), 61–69. https://doi.org/10.1039/C2MB25327E.

Guyon, I. (2003). An introduction to variable and feature selection.

Huang, D.-S., & Yu, H.-J. (2013). Normalized Feature Vectors: A Novel Alignment-Free Sequence Comparison Method Based on the Numbers of Adjacent Amino Acids. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 10*(2), 457–467. https://doi.org/10.1109/TCBB.2013.10.

Choi, Jae Young, Ro, Yong Man, & Plataniotis, K. N. (2011). Boosting Color Feature Selection for Color Face Recognition. *IEEE Transactions on Image Processing, 20*(5), 1425–1434. https://doi.org/10.1109/TIP.2010.2093906.

Karabadji, N. E. I., Khelf, I., Seridi, H., Aridhi, S., Remond, D., & Dhifli, W. (2019). A data sampling and attribute selection strategy for improving decision tree construction. *Expert Systems with Applications, 129*, 84–96. https://doi.org/10.1016/j.eswa.2019.03.052.

Khotanzad, A., & Hong, Y. H. (1990). Rotation invariant image recognition using features selected via a systematic method. *Pattern Recognition, 23*(10), 1089–1101. https://doi.org/10.1016/0031-3203(90)90005-6.

Kira, K., & 1992., L. A. B. T.-P. of the 10th N. C. on A. I. S. J. R. C. J. 12-16. (1992). The Feature Selection Problem: Traditional Methods and a New Algorithm. Proceedings of the 10th National Conference on Artificial Intelligence. San Jose, CA, July 12-16, 1992. 1992/01/01.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. In Machine Learning: ECML-94 (pp. 171–182). https://doi.org/10.1007/3-540-57868-4_57.

Lausch, A., Schmidt, A., & Tischendorf, L. (2015). Data mining and linked open data – New perspectives for data analysis in environmental research. *Ecological Modelling, 295*(Sp. Iss. SI), 5–17. https://doi.org/10.1016/j.ecolmodel.2014.09.018.

Lewis, D. D., Yang, Y., Rose, T. G., & Fan, L. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research, 5*(2), 361–397.

Jing, Li-Ping, Huang, Hou-Kuan, & Shi, Hong-Bo (2002). Improved feature selection approach TFIDF in text mining. *Proceedings International Conference on Machine Learning and Cybernetics, 2*, 944–946. https://doi.org/10.1109/ICMLC.2002.1174522.

Liu, H., & Motoda, H. (1998). Feature Selection for Knowledge Discovery and Data Mining. https://doi.org/10.1007/978-1-4615-5689-3.

Song, Qinbao, Ni, Jingjie, & Wang, Guangtao (2011). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering, 25*(1), 1–14. https://doi.org/10.1109/TKDE.2011.181.

Quinlan, J. R. (1979). *Induction over large data bases.* Stanford University.

Quinlan, J Ross (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. https://doi.org/10.1007/BF00116251.

Rao, H., Shi, X., Rodrigue, A. K., Feng, J., Xia, Y., Elhoseny, M., & Gu, L. (2019). Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing, 74*, 634–642. https://doi.org/10.1016/j.asoc.2018.10.036.

Reyes, O., Morell, C., & Ventura, S. (2015). Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context. Neurocomputing, 161, 168–182. https://doi.org/10.1016/j.neucom.2015.02.045.

Roy, S., Mondal, S., Ekbal, A., & Desarkar, M. S. (2016). CRDT: Correlation Ratio Based Decision Tree Model for Healthcare Data Mining. 2016 IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE), 36–43. https://doi.org/10.1109/BIBE.2016.21.

Roy, S., Mondal, S., Ekbal, A., & Desarkar, M. S. (2019). Dispersion ratio based decision tree model for classification. *Expert Systems with Applications, 116*, 1–9. https://doi.org/10.1016/j.eswa.2018.08.039.

Roy, S., Mondal, S., Ekbal, A., & Desarkar, M. S. (2019b). UCI machine learning repository. Retrieved from http://archive.ics.uci.edu/ml/.

Salzberg, S. L. (1994). C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Machine Learning, 16(3), 235–240. https://doi.org/10.1023/A:1022645310020.

Schiezaro, M., & Pedrini, H. (2013). Data feature selection based on Artificial Bee Colony algorithm. *EURASIP Journal on Image and Video Processing, 2013*(1), 47. https://doi.org/10.1186/1687-5281-2013-47.

Sun, H., & Hu, X. (2017). Attribute selection for decision tree learning with class constraint. *Chemometrics and Intelligent Laboratory Systems, 163*, 16–23. https://doi.org/10.1016/j.chemolab.2017.02.004.

Tang, P., & Peng, Y. (2017). Exploiting distinctive topological constraint of local feature matching for logo image recognition. *Neurocomputing, 236*, 113–122. https://doi.org/10.1016/j.neucom.2016.08.110.

Trabelsi, A., Elouedi, Z., & Lefevre, E. (2019). Decision tree classifiers for evidential attribute values and class labels. *Fuzzy Sets and Systems, 366*, 46–62. https://doi.org/10.1016/j.fss.2018.11.006.

Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S., & Moore, J. H. (2018). Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics, 85*, 189–203. https://doi.org/10.1016/j.jbi.2018.07.014.

Vasconcelos, N. (2003). Feature selection by maximum marginal diversity: optimality and implications for visual recognition. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., I-762-I–769. https://doi.org/10.1109/CVPR.2003.1211430.

Yeh, C.-H. (1991). Classification and regression trees (CART). Chemometrics and Intelligent Laboratory Systems, 12(1), 95–96. https://doi.org/10.1016/0169-7439(91)80113-5.