Computer Science and Information Systems 20(2):831–856 https://doi.org/10.2298/CSIS220817023D

Selective Ensemble Learning Algorithm for Imbalanced Dataset

Hongle Du[1,2,3], Yan Zhang[1,3], Lin Zhang[1,3], and Yeh-Cheng Chen[4]

[1] School of Mathematics and Computer Application, Shangluo University,
726000, Shangluo, China,
{dhl5597,flyingzhang,zhlin002}@163.com
[2] University of the Cordilleras,
Aguio City, Philippines
[3] Shangluo Public Big Data Research Center,
726000, Shangluo, China
[4] Department of computer science, University of California,
Davis, CA, USA
ycch@ucdavis.edu

Abstract. Under the imbalanced dataset, the performance of the base-classifier, the computing method of weight of base-classifier and the selection method of the base-classifier have a great impact on the performance of the ensemble classifier. In order to solve above problem to improve the generalization performance of ensemble classifier, a selective ensemble learning algorithm based on under-sampling for imbalanced dataset is proposed. First, the proposed algorithm calculates the number K of under-sampling samples according to the relationship between class sample density. Then, we use the improved K-means clustering algorithm to under-sample the majority class samples and obtain K cluster centers. Then, all cluster centers (or the sample of the nearest cluster center) are regarded as new majority samples to construct a new balanced training subset combine with the minority class's samples. Repeat those processes to generate multiple training subsets and get multiple base-classifiers. However, with the increasing of iterations, the number of base-classifiers increase, and the similarity among the base-classifiers will also increase. Therefore, it is necessary to select some base-classifier with good classification performance and large difference for ensemble. In the stage of selecting base-classifiers, according to the difference and performance of base-classifiers, we use the idea of maximum correlation and minimum redundancy to select base-classifiers. In the ensemble stage, G-mean or F-mean is selected to evaluate the classification performance of base-classifier for imbalanced dataset. That is to say, it is selected to compute the weight of each base-classifier. And then the weighted voting method is used for ensemble. Finally, the simulation results on the artificial dataset, UCI dataset and KDDCUP dataset show that the algorithm has good generalization performance on imbalanced dataset, especially on the dataset with high imbalance degree.

Keywords: Under Sampling; Imbalanced Dataset; Clustering Algorithm; Selective Ensemble Learning.

1. Introduction

In practical application, due to the difficulty of collecting some samples or the high cost of collecting, the samples of some classes are less, which makes the sample size vary widely. In practical applications, such as network intrusion detection [1, 2], fault detection [3], medical diagnosis [4], etc., these datasets are called imbalanced dataset. The problem of imbalanced dataset in practical application is universal, while the traditional classification algorithms mostly focus on the balanced dataset. Therefore, under imbalanced dataset, the traditional classification algorithm will lead to over-fitting for the majority class and under-fitting for the minority class. That is to say, the classification accuracy of minority class is lower than that of majority class. However, in practical application, the classification accuracy of minority class is more important, such as the recognition rate of network intrusion behaviors, the recognition rate of diseases in medical diagnosis, etc.

Algorithms for imbalanced data are divided into two categories algorithm (algorithm-level methods and data-level methods). The data-level method mainly resamples the training dataset to make the resampled dataset reach balance, and it includes under-sampling for majority class [5-7], over-sampling for minority class [8-10] and combination of the above two methods [11]. Algorithm-level method mainly is to improve the algorithm to reduce the influence of imbalanced data, such as cost sensitive learning [12], single class classification [13], ensemble learning [14, 15], etc. The under-sampling algorithm is to discard some redundant samples from majority class samples according to a certain strategy, so that the number of samples of majority classes after resampling is equal to that of minority class. This makes the resampled dataset be the balanced dataset, such as Bootstrap method, clustering method, etc. Over-sampling is to generate some samples of the minority class according to a certain strategy, so that the number of two classes' samples is imbalanced. SMOTE algorithm [16] is the most representative oversampling algorithm. Because oversampling algorithm will increase the size of the original dataset and lead to over fitting phenomenon, this paper mainly discusses under-sampling algorithm. The purpose of under-sampling method is to make the number of two types of samples nearly equal after resampling, so what to discard and how many samples to discard are the key. The common under-sampling method is to select some majority class samples according to a certain strategy. The number of selected samples is equal to that of minority samples.

However, the essence of imbalanced data is the sample density imbalance of majority class and minority class [17]. Therefore, it is not accurate to resample simply according to the number of the minority class because the resampled dataset cannot be a real balanced dataset. It is more accurate to determine the number of resampled samples according to the relationship between class sample densities [1]. In addition, the classification performance of each base-classifier will also affect the performance of the final ensemble classifier. When sampling, it is necessary to keep the spatial distribution of the original samples well so that each base-classifier has better classification performance. In order to keep the spatial distribution of the original samples space, Liu [14] under-samples the samples from the majority class samples with the K-means clustering algorithm. In this way, the base-classifier has better classification performance, but how to determine K value is the key of this method. In addition, as the number of base classifiers increases, the similarity among them will also increase, which is not conducive to performance of the ensemble classifier. In order to improve

the classification performance of ensemble classifier, Zhou [18] selects better classification performance and larger difference degree of base classifiers. In order to improve the difference degree among base-classifiers, the K-means clustering algorithm is improved. Therefore, the proposed algorithm calculates the number K of under-sampling samples according to the relationship between class sample densities, and under-sampling uses the improved K-means clustering algorithm in this paper.

At the algorithm level method, ensemble learning algorithm is one of the important methods *t* to solve imbalanced data [4, 10]. EasyEnsemble algorithm and BalanceCascade algorithm [14] are the most representative algorithm to solve the classification problem of data imbalance with ensemble learning. In this algorithm, Bootstrap method is used to randomly under-sample from the majority class samples. Then combine the minority class and the under-sampled samples, and get a training subset. Multiple training subsets are trained and multiple base-classifiers can be obtained by repeating the above processes many times. Then ensemble the multiple base-classifiers and obtain the ensemble classifier. The Bootstrap method is under-sampling with playback, which lead to losing of some sample information. In order to let the classifier, learn enough sample information, we may increase the number of resampling times.

However, with the increase of resampling times and the number of base-classifiers, the more similar among the base-classifiers are, the difference degree among multiple base-classifiers will reduce. This will lead to reduce the generalization performance of the final ensemble classifier. To reduce this impact, Zhou [14] prove that select some base-classifiers with good performance and big difference for ensemble has better generalization performance than all base-classifiers, and proposed the selective ensemble algorithm of GASEN. The algorithm and its improved algorithm [4, 19, 20] are to select some base-classifiers with good performance according to certain methods for ensemble. The problem of selection ensemble learning is how to judge the quality of base-classifiers. Potharaju [21] points out that the base-classifier with larger difference degree and better classification performance is a good classifier, which is suitable for ensemble. Subsequently, many scholars [22, 23] improved the selective ensemble learning algorithm, which are mainly focused on how to select the base-classifiers and which base-classifiers to be selected. Generally speaking, it is to select good base-classifiers for integration, so the problem is transformed into how to evaluate the advantages and disadvantages of the base-classifier. In order to describe the generalization performance of the base-classifier, the generalization error is often used to evaluate the base-classifier. It is mostly based on the classification performance and the difference degree among the all base-classifiers.

However, the existing evaluation methods are all aimed at the balanced dataset. For the imbalanced data, the calculation method of the difference degree and the classification performance of the base-classifier are different from those of the balanced dataset. This paper presents an evaluation method of base-classifiers for unbalanced dataset. Then the base-classifier is evaluated and selected according to the evaluation results.

In view of the above problems, an improved selective ensemble learning algorithm for imbalanced data based on the clustering under-sampling (CSEL for short) is proposed in this paper. According to the idea of EasyEnsemble algorithm, the algorithm is improved from the calculation of the number of under-sampling samples, the method of under-sampling, the evaluation method of the base classifier and the calculation

method of base-classifier weight. First, the proposed algorithm calculates the number of under-sampled samples according to the ratio between the two classes' densities. Then, we under-sample samples from majority class with the improved k-means clustering algorithm. Then, all cluster centers (or the sample of the nearest cluster center) are looked as the new majority class samples, and obtain a balanced training subset by combining with the under-sampled samples and the minority class's samples. Repeat above process and generate multiple base-classifiers. However, increasing the number of base classifiers will lead to reduce the difference degree among base-classifiers. Therefore, it is necessary to select some base-classifier with good classification performance and large difference for integration. In the stage of selecting base-classifiers, according to the difference and performance of base-classifiers, we use the idea of maximum correlation and minimum redundancy to select base-classifiers. In the ensemble stage, G-mean or F-mean that is often used to measure the classification performance of classifier for imbalanced data is used to calculate the base-classifier weight in this paper. Then obtain the ensemble classifier with the weighted voting method. The effectiveness of the proposed algorithm is proved by simulation experiment on artificial dataset, UCI dataset and KDDCUP dataset. The experiment mainly compares the experimental results of various algorithms from accuracy, G-mean, F-mean, AUC, Recall, Precision and ROC curve. The experimental results show that the proposed algorithm in this paper has a good classification performance under imbalanced dataset.

2. Related Work

2.1. EasyEnsemble algorithm

Easyensemble algorithm is the under-sampling algorithm. It randomly selects some samples that the number is equal to the number of the minority class from majority class samples. Put the minority class samples and under-sampled samples together to obtain a balanced data subset [14, 24]. Then, train on the imbalanced training subset with the AdaBoost algorithm [24] and get base-classifier. We can get multiple balanced training subsets and multiple base-classifiers by repeating the above process many times. Finally, ensemble classifier can be getting through by integrating the multiple base-classifiers according to a certain strategy.

Suppose that P presents the minority class and N presents the majority class, the number of minority class samples is $|P|$ presents the number of minority class samples and $|N|$ presents the number of majority class samples. The number of iterations of AdaBoost algorithm is S_i. The detailed describe of EasyEnsemble algorithm can be shown as follows:

Algorithm1: EasyEnsemble

1.for i=1:T

1.1 From majority class samples, randomly take $|P|$ samples with the Bootstrap method.

1.2 Put the minority class samples and selected samples together and get the training subset. With the AdaBoost algorithm, train on training subset to obtain the base classifier. The obtained base-classifier can be represented by the following formula:

$$H_i(x) = \text{sgn}(\sum_{j=1}^{s_i} a_{ij} h_{ij}(x) - \theta_i)$$

2. Then the final classifier is represented by the formula:

$$H(x) = \text{sgn}(\sum_{i=1}^{T} \sum_{j=1}^{s_i} a_{ij} h_{ij}(x) - \sum_{i=1}^{T} \theta_i)$$

From the above algorithm process, we can see that the EasyEnsemble algorithm is a data-level method. During under-sampling, in order to reduce the loss of sample space information, multiple times resampling is used for multiple iterations. The obtained base-classifiers are integrated to obtain the ensemble classifier. There are three shortcomings in this algorithm: (1) the essence of dataset imbalance is the sample density imbalance of majority class and minority class. (2) Increase the number of iteration times will also reduce the difference degree among the base-classifiers, which will affect the classification performance of the ensemble classifier [12]. Therefore, it is better to select some base-classifiers with high accuracy and great difference to integrate than all base-classifiers [12]. That is to say, we should adopt selective ensemble learning algorithm. (3) The generalization errors of the base-classifiers are different, especially in the case of imbalanced dataset, so weighted voting integration is more conducive to improving the performance of the ensemble classifiers. This paper also improves the algorithm from these three aspects.

2.2. The Influence of Class Sample Density on Classifier

Data imbalance means that the number of one class sample is far more than that of other classes' sample. The class that the number of samples is more is called the majority class; on the contrary, it is called minority class. The imbalanced dataset has great influence on the final classification, and detailed information can be found in relevant literature, such as [1, 9, 24]. In this section, we mainly discuss the influence of resampling sample number on ensemble classifier performance. In order to verify the influence of class sample density on classifier, the Hyperplane graph of classifier under linear separable and linear non separable dataset is given in Fig. 1 and Fig. 2 respectively. In the Fig.1 (a), the majority class with 200 samples fellow the $U([0,1]\times[0,1])$ distribution. The minority class with 50 samples fellow the $U([0,1]\times[1,2])$ distribution. In Fig.1 (b), the majority class samples fellow the $U([0.8,1]\times[0,1])$ distribution and the number of samples is 50. The minority class samples fellow $U([0,1]\times[1,2])$ and the number of samples is 20. Fig.1 shows that the decision Hyperplane with the SVM algorithm. Left fig.1 (a) is the classification Hyperplane with the sample number ratio of 100:20. The right fig. (b) is the classification Hyperplane with the sample number ratio of 1:1, but the space area of the two classes is different. From Fig.1 (b), we can see that under-sampled samples still are the imbalanced dataset if the numbers of samples of two classes are equal after resampling. That is to say, the sample density of the two classes is different in fig. 1(a). From the Fig 1, we can see that the classification Hyperplanes drifts towards the class with small class density.

Fig.2 shows uniformly distributed artificially generated samples. The + sign represents ten samples of uniformly generated in a circle with a radius of 1, and the X sign represents one thousand samples of uniformly distributed in a circle with a radius of 1 to 2. The classification Hyperplane constructed by SVM algorithm is shown in Fig.1. Among them, the left figure is a classification Hyperplane based on the original dataset. The right figure is a Hyperplane constructed by support vector machine after K-means clustering of majority class samples (k = 10, i.e., taking samples with the same number of minority class samples). As can be seen from Fig. 1, after under-sampling, the classification Hyperplane is shifted to direction of the original majority class. This is due to a new imbalance in the dataset after under-sampling.

Fig.1 and Fig.2 show that the essence of imbalanced dataset is not the imbalance of samples number, but the imbalance of class sample density. Wu [10] also proves that the essence of imbalanced dataset is the imbalanced density of class samples. Wang [11], *K* is set as the number of minority class samples to carry out *K-means* cluster algorithm, then all cluster centers (or the sample of the nearest cluster center) are looked as the new majority class samples. That is to say, it is under-sampling for majority class samples. This method may lead to new imbalanced dataset after under-sampling. In order to make the resampled data set approximate to the balanced dataset, the number of under-sampled samples is determined according to the ratio between the two class samples densities in this paper.

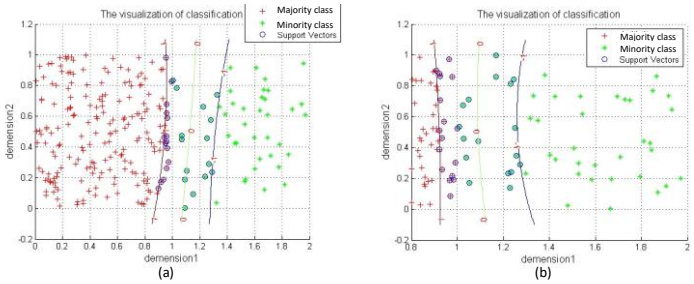


Fig.1 The classification Hyperplane of different density on linear separable dataset

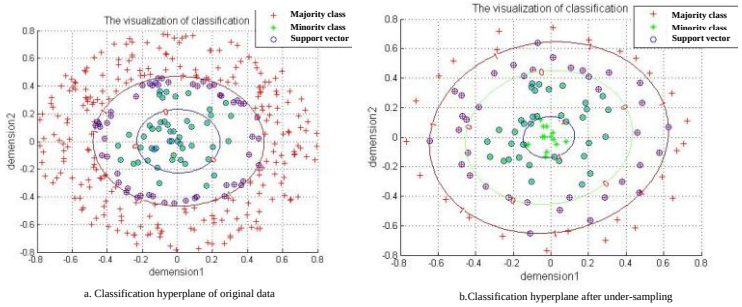


Fig.2 The classification Hyperplane of different density on nonlinear separable dataset

2.3. Under-sampling Method

EasyEnsemble algorithm [14] uses bootstrap method to select the same number of minority class sample from the majority class, and then get one training subset. Random under-sampling can ensure that there is a larger difference between the multiple base-classifiers. At the same time, the performance of some base-classifiers is poor performance because the selected samples cannot keep the spatial distribution of the original sample space. This will reduce the classification performance of the ensemble classifier. In reference [11], in order to ensure that the spatial distribution of the selected samples is similar to that of the original samples, majority class samples are under sampled with the *K-means* clustering algorithm. Then *K* clusters are obtained and the *K* centers of each cluster are looked as the new sample. In order to more accurately maintain the spatial distribution of the original samples after under-sampling, the sample closest to the cluster center is regarded as the new selected sample [20]. Because the cluster center of this method hardly changes at each iteration, the difference of resampled dataset is relatively small, which results in the small difference among the base-classifiers. The improved under-sampling methods [11],[20] are used to keep the spatial distribution characteristics of the original sample space, and make the base-classifier have better classification performance. However, it will also result in have higher similarity among the base-classifiers [25],[26], which is not suitable for ensemble learning. Therefore, this paper improves the *K-means* clustering algorithm with the idea of density cluster method into the *K-means* clustering method to make, and then uses the improved *K-means* clustering algorithm to under-sampling the majority class samples. It can not only keep the original spatial distribution of the samples after under-sampling, but also improve the difference among the datasets obtained by resampling.

2.4. Selection Method of Base-Classifier

Increasing the number of base-classifiers will lead to reduce the difference degree among the base-classifiers. It will reduce the classification performance of the final ensemble classifier. So, it is a good choice to select the base-classifier with batter classification performance and larger difference degree for ensemble [12]. The key of selective ensemble learning algorithm is how many and which base-classifiers are selected. Therefore, it is necessary to define the evaluation indictor of base-classifier to evaluate the classification performance as the criteria for selecting the base-classifier. It has been proved in reference [15] that the greater difference degree of the base-classifiers, the better the performance of the ensemble classifier. Almost all of the selection methods of base-classifier are based on the generalization errors of base-classifier. And the generalization error of base-classifier is evaluated with the accuracy of base-classifier and the difference degree among base-classifiers [4, 20]. In reference [19], ten methods to measure the difference of base-classifiers are given.

Almost all methods calculate the evaluation indictor according to the relationship between the test results of any two base-classifiers on the same dataset. That is, the inconsistency among the classification results of the same dataset by the different base-classifier. In reference [4], information entropy is used to calculate the difference degree

among the multiple base-classifiers. In reference [20], the difference degree of each base-classifier is calculated by the mutual information and classification error between any two base-classifiers. At present, the commonly used classifier difference measurement methods include Q statistics, correlation coefficient, double error, inconsistent measurement, etc. Suppose there are two classifiers f_i and f_j . The number of samples that are correctly classified by two base-classifiers is presented as N^{11}, and the number of samples that are incorrectly classified by two base-classifiers is expressed as N^{00}. and the number of samples that are correctly classified by f_i and incorrectly classified by f_j is expressed as N^{10}, and the number of samples that are incorrectly classified by f_i and that are correctly classified by f_j is expressed as N^{01}, and the number of all samples is expressed as N. Then the inconsistent measure between base-classifiers can be expressed as:

$$D_{ij} = \frac{N^{01} + N^{10}}{N} \tag{1}$$

Then the inconsistency between the classifier and the classification target can be expressed as:

$$D(f_i, L) = \frac{N^{01} + N^{10}}{N} \tag{2}$$

Where L is the classification target, and N^{10}, N^{01} and N have the same means as above.

3. Algorithm Description

3.1. Determine the number of under-sampled samples

Suppose the majority class is expressed as T_{maj}, and the minority class is expressed as T_{min}. The average distance between every two samples in the majority class is expressed as AD_{maj}, The average distance between every two samples in the minority class is expressed as AD_{min}, The number of the majority class sample is expressed as N_{maj} and the number of the minority class sample is expressed as N_{min}. Then, AD_{maj} and AD_{min} can be calculated by formula (3) and formula (4).

$$AD_{maj} = \frac{1}{N_{maj}} \sum_{i=1}^{N_{maj}} \sum_{j=i+1}^{N_{maj}} d(x_i, x_j) \tag{3}$$

$$AD_{\min} = \frac{1}{N_{\min}} \sum_{i=1}^{N_{\min}} \sum_{j=i+1}^{N_{\min}} d(x_i, x_j) \tag{4}$$

Where, $d(x_i, x_j)$ represents the distance between two samples x_i and x_j. $d(x_i, x_j)$ can be calculated by Manhattan distance, European distance and so on. In this paper, European distance is used.

Class sample density is the number of samples per unit volume. However, the volume of a class is easily affected by noise data. In order to simplify the calculation, the average distance between samples in one class is used to calculate the class sample density. The volume of the class sample can be expressed as:

$$V_{maj} = AD_{maj} * N_{maj} \quad (5)$$

$$V_{\min} = AD_{\min} * N_{\min} \quad (6)$$

The number of selected samples is proportional to the class volume, so the number of under-sampled samples for majority classes can be expressed as:

$$SN = \left\lceil \frac{V_{maj}}{V_{\min}} N_{\min} \right\rceil \quad (7)$$

3.2. How to Select Samples

K-means clustering algorithm determines which cluster to join according to the distance from sample to each cluster center. In order to reflect the distribution of class samples density, this paper presents an improved k-means clustering algorithm. The improved algorithm selects the initial cluster centers according to the density distribution of samples. "Sample density" refers to the number of samples in a specified neighborhood centered on the sample, which is called the "density" of the sample. The algorithm first calculates the "sample density" of each sample, and then sorts it according to the sample density. Then randomly select some samples with high density as the initial cluster centers. On the one hand, it can keep the spatial distribution of samples; on the other hand, it can avoid the influence of noise sample. Then, *K* samples that are not adjacent to each other are randomly selected from the selected samples as the initial cluster center. The detailed pseudo-code of algorithm is shown as follows:

Algorithm 2: improved algorithm of *K*-means clustering algorithm

Input: dataset is T $(x_1, x_2, \mathrm{L}\ , x_n)$ with *n* samples, the neighborhood radius is *R*

Output: the center of each cluster (or the sample closest to the cluster center)

Step 1: The density of each sample x_i in *T* and the samples in its neighborhood are calculated. Then, they were arranged in descending order by the sample density, and select a certain proportion of samples and put them into *S* (In the experiment of this paper, the first 80% of samples are selected);

Step 2: For i=1: *K*

Random sample C_i is selected from *S*, and delete C_i and the samples in its neighborhood from *S*;

Step 3: Calculate the distance from sample to each cluster center, and put the sample into the cluster closest from the sample;

Step 4: Calculate the coordinate mean value of all samples in each cluster, and let this mean value to be the new cluster;

Step 5: Repeat the above process until the cluster center does not change in a large range or the clustering times meet the requirements.

The improvement of *K-means* algorithm is mainly on the selection method of initial class center. The neighborhood radius has a great influence on the algorithm. If the domain radius is too large, it may lead to the selection of not enough *K* initial samples; if the domain radius is too small, each sample has only one neighborhood, and the improved algorithm is the original *K*-means algorithm. In this paper, we choose neighborhood radius by experience, and use two times of the average distance of class samples as the neighborhood radius.

3.3. Selection Method of Base-classifier

The generalization performance and difference of the base-classifier will affect the generalization performance of ensemble classifier [12]. The generalization error is expressed by the formula: $E = \overline{E} - \overline{D}$.

$\overline{E}$ is the average of generalization errors of all base classifiers and $\overline{D}$ is the average of difference degree of the all base-classifiers. According to the above formula, reducing the generalization error or (and) increasing the difference degree can reduce the generalization error of the ensemble classifier. Therefore, we can define the evaluation method of the base-classifier from these two aspects, then evaluate the base-classifier, and select the base-classifier according to the evaluation results in this paper.

Selective ensemble learning is to discard the classifiers that have poor classification accuracy and high redundancy. Maximum correlation minimum redundancy criterion [27] is used to select features according to the correlation between the features of dataset and classification objectives. Selecting the features and selecting classifier have similar objectives and basis. Therefore, the criterion can be applied to select the base-classifiers of high generalization error. Given f_i represent the classification results of the *i*-th base-classifier and *L* represent the classification target. Then similarity degree between the base-classifier and the classification target can be expressed as the formula (8). It is the average mutual information among the classification results and the classification target.

$$E(S, L) = \frac{1}{|S|} \sum_{f_i \in S} I(f_i; L) \tag{8}$$

Redundancy between classifiers can be measured by the average of mutual information among the all base-classifiers. The calculation method can be expressed as the formula (9).

$$D = \frac{1}{|S|^2} \sum_{f_i \in S} \sum_{f_j \in S} I(f_i; f_j) \tag{9}$$

Then the maximum correlation minimum redundancy criterion for classifier selection can be expressed as:

$$GE = \max \phi(E, D) \qquad (10)$$

From the above calculation method, we can see $I(f_i; f_j)$ focus on the probability of occurrence of each class. Then, the similarity between the two classifiers is calculated according to the probability, not whether the classification results are the same for each sample. Therefore, this method is not enough accurate to describe the difference between two classifiers, especially in the case of imbalanced dataset. Therefore, in this paper, the correlation between classifier and classification target is defined as:

$$E_i = 1 - D(f_i, L) \qquad (11)$$

The difference D_{ij} between base-classifiers is calculated by formula (1), then the redundancy between the base-classifier and other base-classifiers is expressed as:

$$D_i = 1 - \frac{1}{|S| - 1} \sum_{f_j \in S} D_{ij} \qquad (12)$$

By formulas (11) and (12), the generalization performance of the base-classifier is defined as:

$$GE_i = E_i - D_i \qquad (13)$$

Or

$$GE_i = E_i / D_i \qquad (14)$$

3.4. Ensemble method of base-classifier

The ensemble method of the base-classifiers mostly uses voting method and average method. Voting method also includes hard voting and weighted voting. Hard voting is to predict which class has more classifiers and which class is the final one. Weighted voting method obtains the ensemble classifier based on the weight of base-classifier that is calculated according to a certain calculation method (for example, accuracy, precision, recall, etc.). For imbalanced dataset, more emphasis is on the classification accuracy of minority class samples. Calculating the weight with accuracy cannot accurately reflect the importance of the base-classifier. For imbalanced dataset, *F-mean* and *G-mean* are often used to evaluate the classification performance of classifier. These two evaluation indications are calculated with the confusion matrix [28]. Table 1 shows the confusion matrix for the binary classification problems.

Table 1. Confusion matrix of binary classification

class		Prediction class	
		Positive	Negative
Real class	Positive	TP	FN
	Negative	FP	TN

According to the confusion matrix, the evaluation indication given above can be expressed as follow formulas:

$$\mathrm{Re}call = \frac{TP}{TP + FN} \quad (15)$$

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP} \quad (16)$$

$$F - mean = \frac{2 \times \mathrm{Pr}ecision \times \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \quad (17)$$

$$G - mean = \sqrt{\frac{TN}{TN + FP} \times \frac{TP}{TP + FN}} \quad (18)$$

Formula (17) shows that *F-mean* takes into account the precision and recall. Only when the two values are larger, *F-mean* is larger. Therefore, under imbalanced dataset, this indicator can well describe the classification performance of classifier. Formula (18) shows that *G-mean* takes into account the classification accuracy of two classes. Only when the classification accuracy of the two classes is high, the value of *G-mean* will be larger. When the accuracy of any class is low, the value of *G-mean* is low. In this paper, we adopt these two evaluation indicators to calculate the weight of classifier. In order to accurately describe the importance of each base-classifier in the ensemble classifier, the corresponding weights of *F-mean* or *G-mean* are normalized as follows:

$$W_i = \frac{G_i}{\sum_{i=1}^{K} G_i} \quad (19)$$

$$W_i = \frac{F_i}{\sum_{i=1}^{K} F_i} \quad (20)$$

The ensemble classifier can be expressed as the formula (21).

$$H = \arg\max_{y \in Y} \sum_{i=1}^{m} W_i h_i \quad (19)$$

3.5. Algorithm Description

Fig. 3 shows the algorithm flow chart. Combining with the number of under-samplings, and the method of under-sampling, and the selection of base-classifier, and the ensemble method of multiple base-classifiers and so on, an improved selection ensemble learning algorithm based on under-sampling for imbalanced dataset is proposed in this paper. The proposed algorithm is mainly divided into three stages: data processing, construction base-classifier, selecting base-classifier and obtaining the ensemble classifier. First of all, the imbalance degree between any two base-classifiers is calculated according to the class sample density, and K is determined according to the imbalance degree of two class samples. Then, the improved *K-means* clustering method for under-sampling is given, and then it is used to under-sample from majority samples. Combine with the minority class samples and under-sampled samples to get the training

subset. Then obtain multiple base-classifier according to someone machine learning algorithm on training subset.

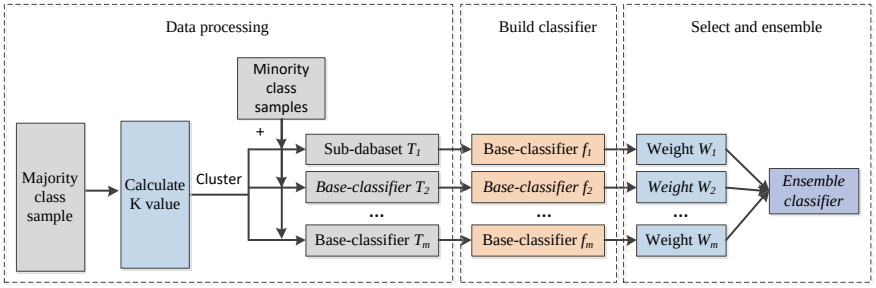


Fig. 3. flow of the algorithm

Then, according to the mutual information, we define the measurement method of the difference degree among the base-classifiers. According to the difference degree and the classification performance, better base-classifiers are selected. Finally, the weight of each base-classifier for the imbalanced data is calculated according to the *G-means*, and then, the multiple base-classifiers are integrated according to the weight. The three stages (data processing, construction base-classifier, selecting base-classifier and obtaining the ensemble classifier) of the algorithm are all designed for imbalance dataset. Each stage is designed to reduce the impact of imbalance dataset on base-classifier and ensemble classifier. The detailed pseudo-code of algorithm is shown as follows:

Algorithm 3: Selective ensemble learning algorithm based on clustering under-sampling for imbalanced dataset

Input: training dataset *TD*, number of algorithm iterations *T*, number of selection base-classifiers *m*, *K* value in *k*-means algorithm

Output: Ensemble classifier

Step 1: Training dataset *TD* is delimited majority class sample set $T_+$and minority class sample set T_-. $|T_+|$and $|T_-|$is the number of samples in the dataset T_+ and T_-;

Step 2: Ccalculate the class sample density and get the number of cluster *SN* using the formula (5);

Step 3: for i=1:*T*

3.1 According to algorithm 2, select the *SN* samples from majority class T_+ as T_+', and then union of two sets T_+' and T_- is the training subsets $SubT_i$;

3.2 Train and obtain the base-classifier h_i, and use h_i to predict the training dataset *TD*. Then the weight W_i of the base-classifier is calculated by formula (18);

End for i

Step 4: Formula (11) and (12) are used to calculate the difference degree and classification performance between base-classifiers, and then the generalization performance GE_i of every base-classifier is calculated with formula (13), and sort by GE_i;

Step 5: Select *M* base-classifiers with small generalization error, and ensemble according to formula (21) to get the weighted ensemble classifier

$$H = \arg\max_{y \in Y} \sum_{i=1}^{m} W_i h_i$$

The purpose of resampling is to make the resampled dataset reach balance, which can improve the performance of base-classifier under imbalanced data. It makes the base-classifier have better classification performance. It is more accurate to calculate the number of under-sampling samples according to the class sample density. Then under sampling adopt improve *K-mean* clustering algorithm. Compared with the direct use the number of minority class samples, classification performance of the base-classifier using this method is better.

According to the class sample density, calculate the selected the number of samples and obtain the *K* clusters using the improved *K-means* cluster algorithm. The obtained clustering center can keep the spatial distribution of the original data as much as possible. In addition, it can avoid the impact of noise samples, and further improve the classification performance of the base-classifier. Compared with calculating the weight with accuracy, calculating the weight with *G-mean* is more accurate to describe the classification performance for the imbalanced dataset.

When increase the number of iterations and obtain the more base-classifiers, the similarity among the obtained base-classifiers will increase and the generalization performance of the final ensemble classifier will reduce. Therefore, it is necessary to select the base-classifiers with high accuracy and large difference from the total base-classifiers to integrate. The accuracy and difference degree are expressed as the generalization error of the base-classifier. Formulas (13) and (14) give the calculation method of generalization error under imbalanced dataset. According to the calculation of generalization error, select several base-classifiers with small generalization error for ensemble. The selection number of the base-classifier will also impact on the generalization performance of the final ensemble classifier. In the experimental part of this paper, the effect of selection proportion on the performance of the ensemble classifier is given through experiment method. The conclusion is that the selected proportion is related to the specific dataset. For the selected base classifier, weight is calculated according to formula (19) or (20) and weighted voting is used to integrate. Such weights can more accurately reflect the classification performance of the base-classifier. Finally, the final ensemble classifier is obtained according to formula (20). From the whole process of the algorithm, we can see that every step of the algorithm is designed to reduce the influence of imbalanced dataset on the performance of ensemble classifier.

4. Experimental Results and Analysis

Because the algorithm has certain randomness, the experimental results of every algorithm in this paper are the standard deviation and mean of 10 experiments. In the experiment part of this paper, all the results are from the simulation experiment on MATLAB 2015b. This section mainly verifies the effectiveness of the proposed algorithm. The experimental results include three parts: the artificial dataset, UCI dataset and KDDCUP dataset. Formulas (13) and (14) have an impact on the performance of the algorithm, as do formulas (19) and (20). In the experiment, we also compare the experimental results of different formulas.

4.1. Experimental Results on Artificial Dataset

In this section, the effectiveness of the algorithm is verified by simulation experiments on artificial dataset. The artificial dataset includes two kinds: linear separable dataset and linear inseparable dataset. Randomly generate two classes' imbalanced samples of uniform distribution to verify the classification performance of the algorithm under imbalanced dataset. The first-class samples (majority class) are $\boldsymbol{U}([0,1.1]\times[0,1])$, and the second-class samples (minority class) are $\boldsymbol{U}([0.9,2]\times[0,1])$. The majority class includes 600 samples, and the minority class includes 30 samples. The test dataset also uses uniform distributed artificial data, but it is balance dataset. The first type of sample is $\boldsymbol{U}([0,1.1]\times[0,1])$, and the second-class sample is $\boldsymbol{U}([0.9,2]\times[0,1])$. There are 500 samples for each class.

From the above process of building dataset, we can see that the two classes of samples overlap in space, which increases the difficulty of classification. The mean and standard deviation of ten experiments of the artificial linear separable dataset is shown in table 2. The ROC curve of four algorithms is shown in Fig. 3, Moreover, the ROC curve of four algorithms is the one random experiment result. Table 2 and fig. 4 mainly compare the experimental results of BalanceCascade algorithm, EasyEnsemble algorithm, AdaBoost algorithm and CSEL algorithm. Table 2 shows the detailed experimental results. The change curve of each evaluate indicator of ten experiments is shown in Fig. 5. From the Fig. 5, we can see that *AUC* and *G-means* vary greatly and the *F-means*, *Precision* and *Recall* changes smoothly from Fig.5.

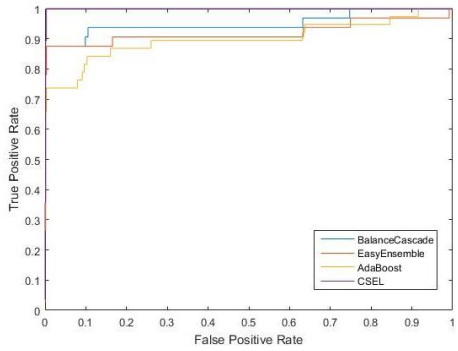


Fig. 4. ROC curve on linear separable dataset

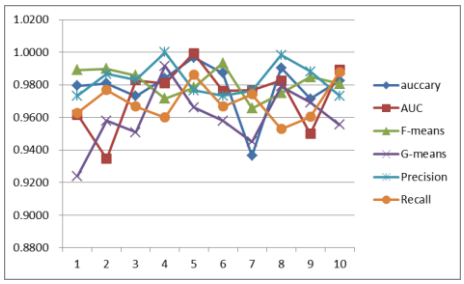


Fig. 5. Change curve of ten experiments on linear separable dataset

Table 2. Comparison of four algorithms on linear separable dataset

Algorithm	Evaluation Indicators (mean ±standard deviation)					
	Accuracy	AUC	F_mean	G_mean	Recall	Precision
CSEL	97.68±2.03	97.64±1.91	97.51±1.87	96.85±2.33	98.74±0.96	97.64±1.28
BalanceCascade	96.73±3.11	94.94±4.06	96.89±2.86	95.47±3.27	96.49±2.83	95.48±3.16
EasyEnsemble	92.45±2.45	92.95±3.44	95.41±2.14	93.79±2.88	94.13±3.08	93.97±3.67
Adaboost	92.38	92.42	93.11	92.86	92.58	92.84

In order to verify the difference between formula (13) and formula (14), ten experiments were carried out with formula (13) and formula (14) respectively. The standard deviation and mean of ten experiments about four algorithms (BalanceCascade algorithm, EasyEnsemble algorithm, AdaBoost algorithm and CSEL algorithm) is shown in table 3. From experimental results, we can see that the classification performance with formula (13) is better than formula (14).

Table 3. Experimental results with formula (13) and formula (14)

Formula	Evaluation Indicators (mean±standard deviation)					
	Accuracy	AUC	F_mean	G_mean	Recall	Precision
Formula (13)	97.68±2.03	97.64±1.91	97.51±1.87	96.85±2.33	98.74±0.96	97.64±1.28
Formula (14)	96.14±3.11	95.83±2.54	95.61±2.86	94.97±2.85	96.88±1.47	95.51±1.79

This part is an experiment on the linear inseparable data, which uses two classes of concentric circle samples of $\begin{cases} x = \rho g \cos\theta \\ y = \rho g \sin\theta \end{cases}, \theta \in U[0, 2\pi]$. The first-class sample (the majority class) is a uniform distribution with radius $\rho \in [0,1.1]$, and the second-class sample (the minority class) is a uniform distribution with radius $\rho \in [0.9, 2]$. The majority class includes 1000 samples, and the minority class includes 100 samples. The two classes of test dataset all include 500 samples, and the distribution is the same as the training dataset. The detailed results of ten random experiments are shown in table 4 about evaluate dictators of *Accuracy*, *AUC*, *F-means*, *G-means*, *Recall* and *Precision*, and the ROC curves about four algorithms are random one experiment is shown in Fig. 6. The change curve of each evaluate indicator of ten experiments is shown in Fig.7. From the fig.7, we can see that *AUC* and *G-means* vary greatly and the *F-means* from Fig.7.

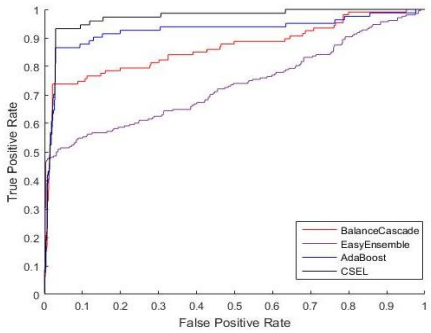


Fig.6. ROC curve on linear inseparable dataset

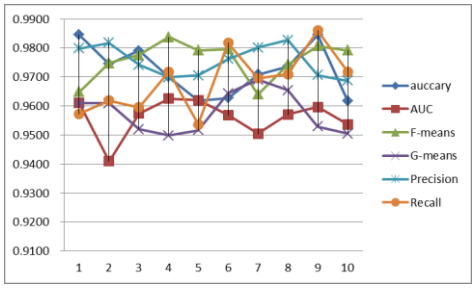


Fig. 7. Change curve of ten experiments on linear inseparable dataset

Table 4. Experimental results of four algorithms on linear inseparable dataset

Algorithm	Evaluation Indicators (mean±standard deviation)					
	Accuracy	AUC	F_mean	G_mean	Recall	Precision
CSEL	97.27±1.81	96.87±1.58	97.52±1.83	92.86±2.44	97.16±0.31	98.43±0.24
BalanceCascade	96.73±2.72	92.68±5.84	96.22±2.76	88.86±4.85	96.89±1.33	97.51±0.31
EasyEnsemble	95.55±2.06	90.52±4.39	95.54±2.81	87.63±3.96	96.27±1.27	97.28±1.56
Adaboost	89.64	83.43	93.97	84.77	96.98	88.99

In order to verify the difference between formula (19) and formula (20), ten experiments were carried out with formula (19) and formula (20) respectively. Table 5 is the mean and standard deviation of ten experiments with formula (19) and formula (20). We can see that the results of formula (19) are better than formula (20).

Table 5. Experimental results with formula (19) and formula (20)

formula	Evaluation Indicators (mean ±standard deviation)					
	Accuracy	AUC	F_mean	G_mean	Recall	Precision
Formula (19)	97.27±1.81	96.87±1.58	97.52±1.83	92.86±2.44	97.16±0.31	98.43±0.24
Formula (20)	96.85±2.01	96.04±1.77	96.93±2.18	92.41±2.36	96.98±0.59	97.89±0.56

4.2. Experimental Results on UCI Dataset

This section carries out simulation experiments on 7 UCI datasets (Car, Breast cancer, Haberman's survival, Blood transfusion, Contraceptive, Teaching and Tic-Tac-To). The experimental results of the proposed algorithm and three others algorithm (BalanceCascade, EasyEnsemble and AdaBoost) show in table 6. Every UCI dataset is divided into test dataset and training dataset. The samples in two datasets are randomly selected from the corresponding UCI dataset. Table 6 show the detailed sample distribution of the experimental dataset. We can see that there is a high imbalance degree in the training datasets and the test datasets are all close to the balance datasets. In addition, the algorithm proposed in this paper mainly aim at binary classification, so all experimental datasets are transformed into dataset only include two classes.

Table 6. Experiment results on UCI data set

Dataset	Attribute	Training dataset			Test dataset		
		Majority	Minority	Proportion	Majority	Minority	Proportion
Car	9	400	100	4.0	226	232	1
breast	9	100	25	4	101	60	1.7
haberman	3	150	30	5	75	51	1.5
blood	4	450	50	9	120	128	1
Contraceptive	9	644	100	6.44	200	200	1
Teaching	5	70	20	3.5	32	28	1
Tic-Tac-To	9	400	100	4	226	228	1

Table 7 shows the mean and standard deviation of ten experiments results on the above seven UCI datasets using CSEL algorithm, BalanCecascade algorithm, EasyEnsemble and AdaBoost algorithm. Table 5 shows the detailed experimental results about evaluate dictators of *Accuracy*, *AUC*, *F-means*, *G-means*, *Recall* and *Precision*. Fig.5, Fig.6, Fig.7 and Fig.8 show the ROC curve of CSEL algorithm, BalanceCascade algorithm, EasyEnsemble algorithm and AdaBoost algorithm on above seven datasets.

Table 7. Experimental result of four algorithms on UCI dataset

Dataset	Alogrithm	Evaluation Indicators (mean±standard deviation)					
		Accuracy	AUC	F-mean	G-mean	Recall	Precision
teaching	CSEL	**70.28±1.03**	**71.55±0.54**	**73.62±0.87**	**68.44±1.35**	**65.74±0.74**	**72.15±0.91**
	BalanceCascade	64.05±2.64	67.84±0.78	48.48±1.92	56.57±3.15	49.02±3.47	64.18±2.18
	EasyEnsemble	68.43±3.00	64.52±0.34	68.30±1.01	60.47±2.80	63.18±0.68	65.37±1.85
	AbaBoost	68.39	61.75	63.26	64.51	63.75	66.81
car	CSEL	**82.24±1.73**	82.11±1.21	**77.42±0.98**	**80.07±1.19**	**75.04±1.38**	**81.73±2.15**
	BalanceCascade	81.70±2.71	**82.87±2.84**	75.08±3.17	76.46±2.86	70.24±2.09	75.71±3.61
	EasyEnsemble	72.23±1.94	77.28±1.91	65.54±2.54	70.42±2.05	64.47±1.84	62.87±3.49
	AbaBoost	70.83	75.33	60.50	66.62	70.87	49.32
breast	CSEL	**68.53±1.57**	**72.37±1.53**	**60.55±1.72**	**68.34±1.97**	78.95±1.74	**72.79±2.39**
	BalanceCascade	65.73±2.84	70.18±2.06	56.38±3.35	67.82±3.01	**48.08±2.85**	75.04±4.21
	EasyEnsemble	62.36±2.44	68.50±1.83	51.36±3.24	63.63±2.90	81.99±2.88	68.09±6.91
	AbaBoost	60.23	65.41	35.29	49.61	74.36	48.57
haberman	CSEL	**73.86±1.08**	**90.80±2.31**	**68.25±0.81**	**79.77±1.75**	83.64±1.56	**78.01±1.08**
	BalanceCascade	74.05±1.74	87.03±1.66	64.44±0.97	71.86±1.04	82.67±1.39	77.89±1.67
	EasyEnsemble	67.78±1.96	76.15±0.96	53.90±1.34	70.20±1.28	**84.77±0.98**	75.79±3.07
	AbaBoost	65.82	67.70	27.45	41.79	77.86	28.42
blood	CSEL	**70.18±0.61**	**82.56±0.83**	50.84±1.04	**71.08±0.98**	82.58±1.15	**70.35±2.04**
	BalanceCascade	67.37±2.34	78.21±1.52	45.85±1.97	63.98±2.16	**83.37±1.96**	65.17±3.51
	EasyEnsemble	66.20±1.80	72.20±0.77	49.44±1.49	67.22±1.38	83.26±1.05	69.44±3.47
	AbaBoost	69.95	63.18	**50.98**	63.23	80.87	43.82
Contraceptive	CSEL	**69.24±0.76**	**75.85±0.81**	**60.33±0.95**	**70.19±0.83**	**81.72±0.57**	**75.41±1.66**
	BalanceCascade	66.45±1.26	71.63±1.32	51.32±1.83	67.38±1.64	80.05±1.21	73.03±2.76
	EasyEnsemble	66.77±0.84	74.23±0.93	53.57±1.21	68.59±1.05	81.02±0.95	72.87±2.35
	AbaBoost	68.32	66.39	46.55	57.98	80.41	36.00
Tic-Tac-Toe	CSEL	**80.58±0.97**	**79.86±0.83**	**66.67±0.92**	**72.42±0.84**	82.30±0.89	**63.52±1.18**
	BalanceCascade	78.08±2.18	78.56±1.51	65.32±1.75	69.32±1.62	80.42±1.73	56.02±2.37
	EasyEnsemble	78.81±1.33	76.87±1.09	65.28±1.48	70.85±0.99	80.89±1.42	60.96±1.29
	AbaBoost	79.75	48.38	63.53	69.56	**84.50**	50.90

From table 7, in most of the experimental results of most indicators (*Accuracy*, *AUC*, *F-means*, *G-means*, *Recall* and *Precision)*, the experimental results of proposed algorithm are relatively better than that of AdaBoost algorithm, EasyEnsemble algorithm and BalanceCascade algorithm except on the indicator of Recall. The average classification accuracy of the propose algorithm increases by 2.17%. The average AUC of the proposed algorithm increases by 2.84%. The average *F-mean* and *G-mean* of the

propose algorithm increases by 7.18% and 5.27% respectively. The classification accuracy of minority class is greatly improved. The average *Recall* and *Precision* of the proposed algorithm increase by 2.87% and 3.84% respectively.

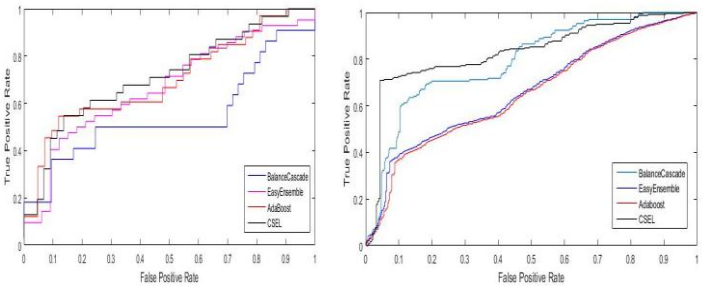


Fig.8. ROC curve of Teaching and Car dataset

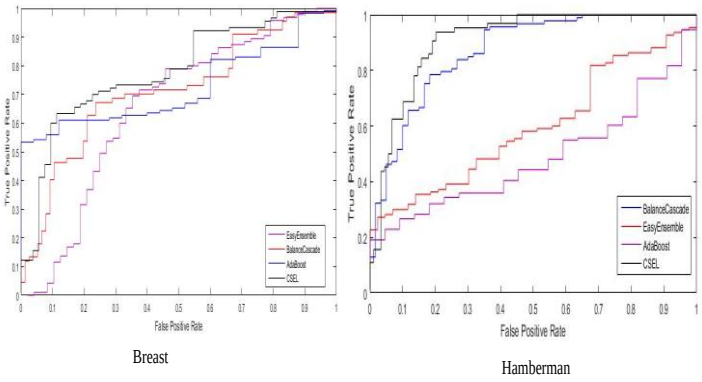


Fig.9. ROC curve of Breast and Hamberman dataset

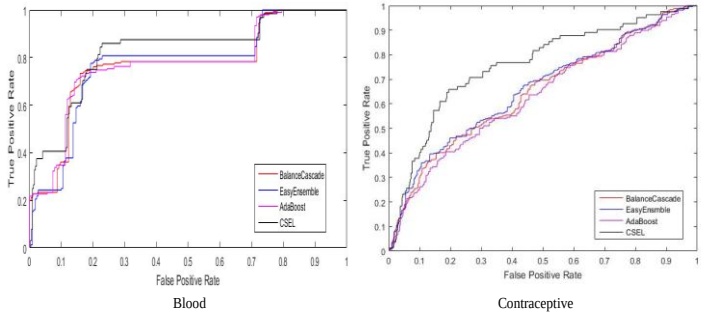


Fig.10. ROC curve of Blood and Contraceptive dataset

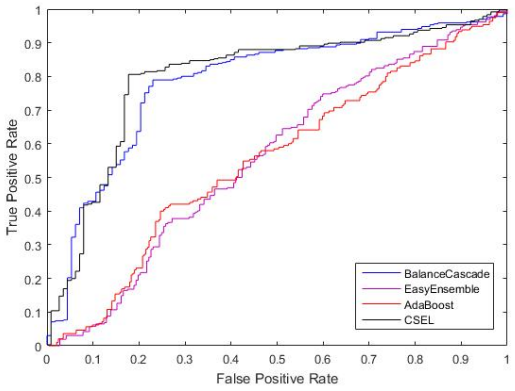


Fig.11. ROC curve of Tic-Tac-Toc dataset

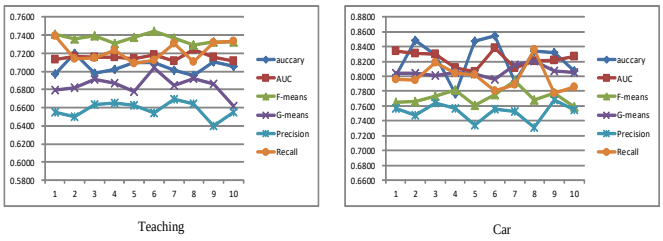


Fig.12. Change curve of ten experiments of Teaching and Car dataset

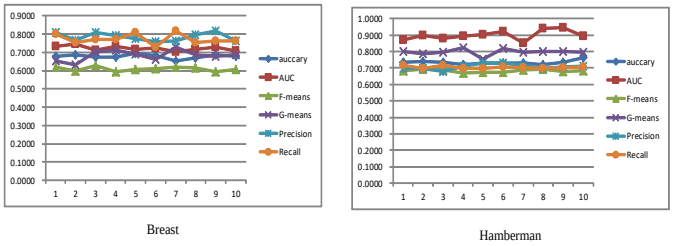


Fig.13. Change curve of ten experiments of Breast and Hamberman dataset

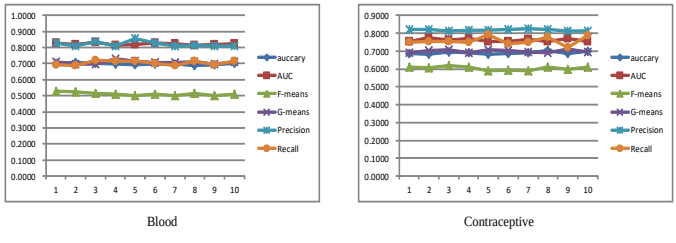
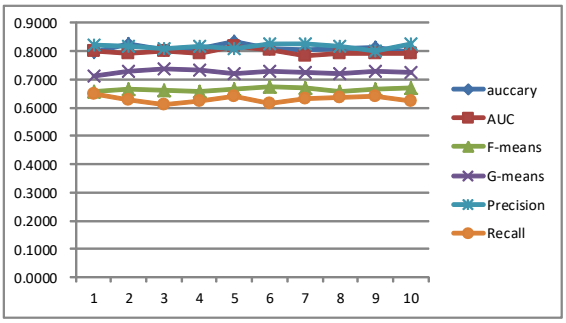


Fig.14. Change curve of ten experiments of Blood and Contraceptive dataset



Tic-Tac-Toc

Fig.15. Change curve of ten experiments of Tic-Tac-Toc dataset

Fig. 8-11 show the ROC curve of seven UCI datasets. Compared with EasyEnsemble algorithm, BalanceCascade algorithm and AdaBoost algorithm, CSEL algorithm has better classification performance on evaluate indicator AUC. From the experimental data in Table 7 and seven ROC curve, CSEL algorithm has better experimental results for imbalanced dataset. And it has a high precision of minority class. Because we pay more attention to the recognition rate and false alarm rate of minority class samples in practical application, so it meets the needs of practical application.

The change curve of each evaluate indicator of ten experiments on sever UCI datasets is shown in Fig.12, Fig.13, Fig.14 and Fig.15. From Fig.12, it can be seen from the figure that every evaluate indicator vary greatly on Teaching and Car dataset. However, in other UCI datasets, the change of all indicators is relatively gentle, which shows that the algorithm is relatively stable from Fig.13, Fig.14 and Fig.15.

4.3. Experimental results on KDDCUP dataset

KDDCUP99 dataset is an imbalanced dataset, because there are a large number of normal behaviour data and a small amount of attack behavior data in practical application. It includes training dataset and test dataset. Training set includes 494 022 samples and test set includes 31000 samples. There are 24 kinds of attack records in training dataset and 38 kinds of attack records in test dataset. The algorithm proposed in this paper is oriented to class binary-classification problem. Therefore, all attack samples are looked as abnormal class and all normal samples are looked as normal class. The experimental data in this paper were selected at equal intervals to obtain 5176 samples as test data and 2674 samples as training data to maintain the original dataset space distribution. The detailed distribution of experimental data is shown in Table 6. From the training dataset, we can see that the experimental dataset is imbalanced.

This section still compares the performance with AdaBoost algorithm, BalanCecascad algorithm and EasyEnsemble algorithm on KDDCUP dataset [29]. As can be seen from table 6, test dataset includes some attack type samples that do not include in the training data. The purpose is to verify the detection of new attacks. The experimental performance has been compared with the above seven indicators.Table 8 shows the detailed data of the experiment about evaluate dictators of *Accuracy*, *AUC*, *F-*

means, *G-means*, *Recall* and *Precision*. The ROC curve of a random experiment is shown in Fig. 9 of four algorithms (CSEL algorithm, BalanceCascade algorithm, EasyEnsemble algorithm and AdaBoost algorithm).

From table 9, we can see that the test data is an imbalanced dataset and the training data is an imbalanced dataset that the imbalanced ratio between the number of majority class and it of minority class is 10.3. The attack behaviour samples are looked as minority class and the normal behaviour samples are looked as majority class in the training data set. Test dataset includes 12 kinds of attack samples and 24 kinds of unknown attack samples. In addition, the test data is an approximately balanced dataset. Therefore, it will have a high classification error rate of minority class if the classification Hyperplane offset to the minority class. The proposed selective ensemble learning algorithm for imbalanced data is suitable for anomaly detection of network intrusion. It is important for anomaly detection of network intrusion to reduce the classification error rate of minority class, meanwhile, to reduce the missing alarm rate and false alarm rate.

Table 8. Experiment dataset of KDDCUP

Data type	dataset	
	Test dataset	Train dataset
Total samples	5176	2674
Normal samples	2146	2438
Attack samples	2582	236
Unknown samples	448	0
Attack types	28	17

Table 9. classification accuracy of four algorithms on KDDCUP dataset

algorithm	Detailed accuracy of every type of data (mean± standard deviation)				
	All records	Normal record	Intrusion record	Known intrusion	Unknown intrusion
CSEL	85.58±0.31	91.21±0.37	81.78±0.31	83.04±0. 23	74.55±0.45
BalanceCascade	82.08±0.97	91.99±0.44	75.05±0.48	75.91±0.37	70.09±0.89
EasyEnsemble	79.14±0. 8	91.38±0.21	70.43±0.32	72.11±0.34	62.72±0.24
Adaboost	80.12	90.86	72.48	73.74	65.18

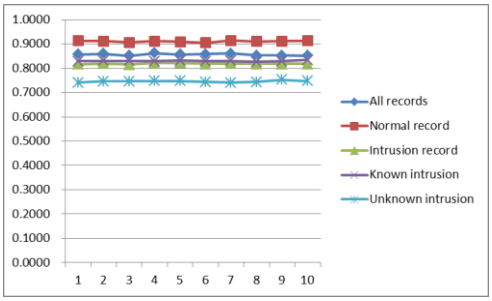


Fig.16. Accuracy change curve of ten experiments of KDDCUP dataset

Table 9 is classification accuracy of all kinds of data (include overall accuracy, normal type, intrusion type, unknown intrusion type and known intrusion type). It can be seen that classification accuracy of intrusion samples is greatly increased from table

9, especially classification accuracy of unknown intrusion samples. The detection rate of unknown intrusion is increased by 4.46% compare with BalanceCascade algorithm. Classification accuracy of known intrusion is increased by 7.13%. It can be seen that the proposed algorithm may increase the classification accuracy of the minority class and total samples from table 9. In other words, it may reduce false alarm rate and missed alarm rate. Fig.16 shows the change curve of classification accuracy of various data (includes: All records, Normal records, Intrusion records, Known intrusion, Unknown records, intrusion records) in ten experiments. From the change curve, we can see that in ten experiments, the indicators are relatively stable, indicating that the algorithm is relatively stable.

Table 10. Experimental results of KDDCUP dataset

Algorithm	Evaluation Indicators (mean ±standard deviation)					
	Accuracy	AUC	F_mean	G_mean	Recall	Precision
CSEL	85.58±0.31	87.52±0.26	80.57±0.18	83.54±0.23	81.97±0.12	68.94±0.29
BalanceCascade	82.08±0.85	84.26±1.16	80.15±0.95	82.69±0.74	80.76±0.97	65.82±1.24
EasyEnsemble	79.14±0.18	83.39±0.35	78.72±0.24	82.18±0.17	81.92±0.23	63.96±0.43
Adaboost	80.12	84.18	79.19	83.04	83.76	63.89

The detailed experimental results about accuracy rate, *F-mean*, *G-mean*, *AUC*, *Recall* and *Precision* is shown in table 10. It can be seen that the experimental results of the most evaluate indicators are better compared with BalanceCascade algorithm, EasyEnsemble algorithm and Adaboost algorithm from table 10, especially the *recall* and *precision* of intrusion behavior (minority class). This is due to the proposed algorithm pay more attention on sample distribution.

The ROC curves about four algorithms are random one experiment on KDDCUP dataset is shown in Fig. 18. The change curve of each evaluate indicator of ten experiments on KDDCUP dataset is shown in Fig.18. We can see that the change of all indicators is relatively gentle from Fig.18, which shows that the algorithm is relatively stable.

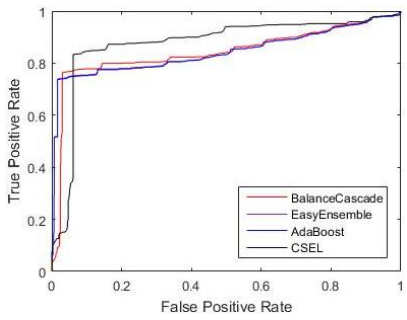


Fig.17. ROC curve of KDDCUP dataset

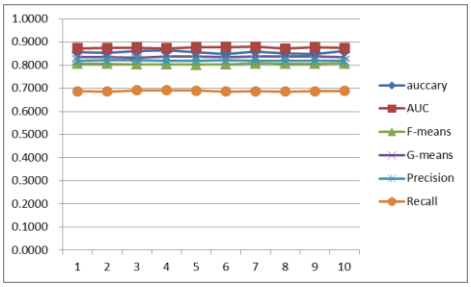


Fig.18. Change curve of ten experiments of KDDCUP dataset

5. Conclusion and Discussion

Ensemble learning algorithm can also be used to solve the problem of imbalanced data. It mostly uses some methods to re-sample the majority samples and obtains a dataset equivalent to the number of samples in a minority class. Then, a training subset is formed with the minority class samples. A certain classification algorithm is used to train the base-classifier. Repeat many times to get multiple base-classifiers, and then ensemble multiple base-classifiers. A selective ensemble learning algorithm for imbalanced data is proposed to reduce the influence of data imbalance in this paper. The simulation results on artificial dataset, UCI dataset and KDDCUP dataset show that the proposed algorithm can reduce the effect of imbalanced data.

Acknowledgment. This work is supported by Key Research and Development Program of Shaanxi of China (Program No.2022GY-073); Science and Technology Plan Project of Shangluo City of China (No. 2021-J-0002); Science and Technology Innovation Team Building Project of Shangluo University of China (No.18SCX002).

References

1. Du H.L., Zhang Y., Ke G., et al. (2021) A selective ensemble learning algorithm for imbalanced dataset, Journal of Ambient Intelligence and Humanized Computing, DOI.10.1007/s12652-021-03453-w.
2. Wan J.W., Yang M., Chen Y.J (2012) Const Sensitive Semi-Supervised Laplacian Support Vector machine [J]. ACTA ELECTRONICA SINICA of China, 40(7):1410-1415
3. Duan L.X., Guo H.,Wang J.J (2016) A mechanical fault severity identification method under unbalanced datasets [J]. JOURNAL OF VIBRATION AND SHOCK of China, 35(20):178-182
4. Liu L., Wang B., Zhong Q., et al. (2015) A selective ensemble method based on K-means method[C]// International Conference on Computer Science & Network Technology. IEEE, 2015.
5. Zhou Y.H., Zhou Z.H (2016) Large margin distribution learning with cost interval and unlabeled data [J]. IEEE Transactions on Knowledge and Data Engineering,2016,28(7):1749-1763

6. Li Y.J., Guo H.X., Li Y.N., etc. (2016) A Boosting based Ensemble Learning Algorithm in Imbalanced Data Classification[J]. Systems Engineering-Theory & Practice of China,2016,36(1):189-199
7. Xing S.,Wang X.H., Wang X.L (2016) Extreme Learning Machine Ensemble Learning based on Multi Class Resampling for imbalanced Data[J].JOURNAL OF NANJING UNIVERSITY (NATURAL SCIENCES) of China,2016,52(1):203-211
8. Jian, C., Gao, J., Ao, Y (2016) A new sampling method for classifying imbalanced data based on support vector machine ensemble [J]. Neurocomputing, 2016,193(1), 115–122
9. Wang Q., Luo Z.H., Huang J.C, et al. (2017) A Novel Ensemble Method for Imbalanced Data Learning: Bagging of Extrapolation-SMOTE SVM [J]. Computational intelligence and neuroscience, 2017, 2017(3):1827016.
10. Wu S., Liu L., Lu D (2017) Imbalanced Data Ensemble Classification based on Cluster-based Under-sampling Algorithm[J]. Chinese Journal of Engineering, 2017, 39(08):1244-1253.
11. Wang S., Minku L.L., Yao X (2015) Resampling-based ensemble methods for online class imbalance learning[J]. IEEE Transactions on Knowledge and Data Engineering,2015,27(5):1356-1368
12. Zhang, H., Li, J. L,, Liu, X. M., et al. (2021) Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection[J]. Future Generation Computer Systems, 122: 130-143.
13. Chen, N. N., Gong, X. T., Wang, Y. M., et al. (2021) Random clustering forest for extended belief rule-based system [J]. Soft Computing, 2021, 25(6): 4609-4619.
14. Liu X.Y., Wu J., Zhou Z.H (2009) Exploratory undersampling for class-imbalance learning [J]. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, 2009, 39(2): 539-550.
15. Guo H., Li Y., Li Y., et al. (2016) BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification[J]. Engineering Applications of Artificial Intelligence, 2016, 49(C):176-193
16. Shipp C.A., Kuncheva L.I (2002) Relationships between combination methods and measures of diversity in combining classifiers [J]. Information Fusion,2002,3(2):135-148
17. Brown G (2009) An information theoretic perspective on multiple classifier systems[C]. //Proc of the 8th International Workshop on Multiple Classifier Systems,2009:344-353
18. Zhou Y.H., Zhou Z.H (2016) Large margin distribution learning with cost interval and unlabelled data [J]. IEEE Transactions on Knowledge and Data Engineering,2016,28(7):1749-1763
19. Zhang C.X., Zhang J.S (2011) A survey of selective ensemble learning algorithms [J]. Jisuanji Xuebao/chinese Journal of Computers, 2011, 34(8):1399-1410.
20. Zhang Y., Liu B., Yu J (2017) A selective ensemble learning approach based on evolutionary algorithm[J]. Journal of Intelligent & Fuzzy Systems, 2017, 32(3):2365-2373.
21. Potharaju S.P., Sreedevi M (2017) Ensembled Rule Based Classification Algorithms for predicting Imbalanced Kidney Disease Data [J]. Journal of Engineering Science & Technology Review, 2017, 9(5):201-207.
22. Zhai J., Zhang S., Wang C (2017) The classification of imbalanced large data sets based on MapReduce and ensemble of ELM classifiers [J]. International Journal of Machine Learning & Cybernetics, 2017, 8(3):1009-1017.
23. Haque M.N., Noman N., Berretta R., et al. (2016) Heterogeneous Ensemble Combination Search Using Genetic Algorithm for Class Imbalanced Data Classification[J]. PLOS one, 2016,11(1): e0146116
24. Wan J.W., Yang M., Chen Y.J (2012) Const Sensitive Semi-Supervised Laplacian Support Vector machine [J]. ACTA ELECTRONICA SINICA of China, 40(7):1410-1415
25. Zhong S, Chen T, He F, et al. (2014) Fast Gaussian kernel learning for classification tasks based on specially structured global optimization [J]. Neural Networks, 2014, 57: 51-62.

26. Zhang Y., Du H.L (2019) Imbalanced Heterogeneous Data Ensemble Classification based on HVDM-KNN [J]. CAAI Transactions on Intelligent Systems of China, 2019,14(4):733-742
27. KUNCHEVA L.I., WHITAKER C. J (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy[J]. Machine Learning,2003,51(2):181-207
28. Yu H. and Ni J (2014) An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data [J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2014,11(4):1339-1347
29. Du H.L., Zhang Y., Ke G., et al. (2021) Online ensemble learning algorithm for imbalanced data stream, Applied Soft Computing,Volume 107,2021, https://doi.org/10.1016/j.asoc.2021.107378.

Hongle Du received the B.S. degree in computer science and technology from Henan University, Kaifeng, China, in 2004 and the M.S. degree in computer application technology from Guangdong University of Technology, Guangzhou, China, in 2010. His current research interests include machine learning, pattern recognition and network security.

Yan Zhang received the B.S. degree in computer science and technology from Xidian University, Xi'an, China, in 2004 and the M.S. degree in computer application technology from Northwest A&F University, Xi'an, China, in 2011. His current research interests include machine learning, pattern recognition and network security.

Lin Zhang is a professor in School of Mathematics and Computer Application, Shangluo University. He received the B.S. degree in computer science and technology from Shaanxi Normal University, Xi'an, China, in 1991. His current research interests include machine learning, Online learning and network security.

Yeh-Cheng Chen is a PhD at the Department of Computer Science, University of California, Davis, CA, USA. His research interests are radio-frequency identification (RFID), data mining, social network, information systems, wireless network artificial intelligence, IoT and security.

Received: August 17, 2022; Accepted: January 15, 2022.