IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 September 2022, accepted 26 October 2022, date of publication 4 November 2022, date of current version 9 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219582

RESEARCH ARTICLE

An Ensemble Learning Algorithm Based on Density Peaks Clustering and Fitness for Imbalanced Data

HUI XU AND QICHENG LIU
School of Computer and Control Engineering, Yantai University, Yantai 264005, China
Corresponding author: Qicheng Liu (ytliuqc@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 62172351.

ABSTRACT In view of the low classification accuracy of the minority class in imbalanced data, an algorithm called DPF-EL (density peaks and fitness combined with ensemble learning) based on density peaks clustering and fitness is proposed. Firstly, this method uses the density peaks clustering algorithm to divide the majority class into different sub-clusters, the local density calculated in the clustering process is used to assign weights to each sub-cluster, and the number of under-sampling is determined by the weights. Secondly, the concept of fitness is introduced into the sub-clusters, the selection probability of the samples is calculated according to the size of their fitness, and the majority class is under-sampled based on the selection probability. Finally, combined with boosting algorithm, iterative training is performed on the balanced data set. Experimental tests were conducted with KEEL imbalanced data sets, and the experimental results show that the performance of DPF-EL algorithm is better than other algorithms, which indicates the feasibility of the proposed algorithm.

INDEX TERMS Imbalanced data, density peaks clustering, fitness, under-sampling, classification.

I. INTRODUCTION

Classification is one of the most extensively used machine learning (ML) techniques. Traditional ML classification algorithms usually assume that the sample number of each class in data sets is balanced and treats the samples of different classes equally to improve the overall classification accuracy [1]. However, in real applications, the number of samples in various classes in data sets is often imbalanced. When the number of samples in one or more classes (the majority class) is far more than others (the minority class), the classification algorithm will tilt toward the majority class, causing low classification accuracy for the minority class [2]. The classification accuracy of the minority class is essential in many cases yet. For example, in medical diagnosis [3], [4], [5], people suffering from malignant diseases are the minority class, suppose the traditional ML classification algorithm is used for auxiliary diagnosis of malignant diseases. In that case, the classification accuracy of the malignant diseases are low, which may lead to misdiagnosis and delay the treatment timing of the patients. Today, with the increasing usage of ML technology, the low classification accuracy of the minority class caused by the data imbalance problem has existed in many fields, such as intrusion detection [6], [7], [8], fraud detection [9], [10], [11], and target detection [12], [13], [14], etc.

Many methods have been presented to solve the problem of data imbalanced, and these methods can be categorized into three classes: (a) data resampling, (b) improving the classification algorithms and (c) data resampling combined with ensemble learning [15]. The method of data resampling mainly synthesizes the minority class samples or removes the majority class samples to reduce the data imbalance rate. For example, the minority class synthetic algorithm, synthetic minority over-sampling technique (SMOTE) [16], and the random under-sampling (RUS) method [17]. The method of improving of the classification algorithm is mainly to improve the existing classification algorithm so that it

The associate editor coordinating the review of this manuscript and approving it for publication was Cheng Chin.

can be applied to deal with imbalanced data sets, such as cost-sensitive approach [18], [19], [20], fuzzy support vector machine [21], [22], [23], [24], and improved random forest [25], [26], [27]. Due to the effectiveness of data resampling and the diversity of data brought by ensemble learning, the method of data resampling combined with ensemble learning has become one of the main methods to deal with the problem of data imbalance at present, this paper is also based on this method for research.

The method of data resampling combined with ensemble learning mainly uses different data resampling methods to balance the training data sets at the beginning of ensemble learning training [28]. Reference [29] proposed an algorithm that combines SMOTE with AdaBoost (adaptive boosting algorithm) called SMOTEBoost (synthetic minority over-sampling technique with AdaBoost). This algorithm uses SMOTE to oversample the minority class during the iterative training of AdaBoost, to alleviate the effect caused by data imbalance. However, during the oversampling, SMOTE algorithm has a marginal problem, which makes the classification boundary fuzzy and the accuracy of minority classification worse. Reference [30] proposed an algorithm called RUSBoost (random under-sampling with AdaBoost) that combines RUS with AdaBoost. It was similar to SMOTEBoost, and the difference is that random under-sampling is used to balance the data sets during the iterative training. Though this algorithm can deal with the imbalanced data effectively, but due to the uncertainty of the randomly under-sampling method, the samples carrying important information may be lost during under-sampling. Reference [31] proposed an under-sampling method based on density peaks. First, the majority class of samples in the overlapping areas are identified and removed. Second, the clustering is performed on the majority class of samples with the overlap region removed, and each generated sub-cluster is under-sampling according to its size. Finally, the bagging algorithm is used to integrate the classifier so that better classification performance is obtained. Reference [32] proposed a clustering-based under-sampling method, which takes the centers of the sub-cluster as the representative samples to replace the whole majority class samples, and then combines the AdaBoost for iterative training. This method improves the classification accuracy of imbalanced data to a certain extent. The deficiency of this method is that it only considers the cluster centers as the representative samples and ignores the selection of samples in the boundary area, which leads to the loss of samples near the decision boundary and affects the accuracy of classification.

To solve the problems existing in the above methods, this paper proposed an algorithm called DPF-EL based on density peaks clustering and fitness. The density peaks clustering algorithm [33] is a density-based clustering algorithm proposed by Rodriguez and Laio. The main advantages of this algorithm are that it does not need iteration, can find cluster centers at one time, and can identify clusters with any shape. Due to its simple implementation and superior clustering performance, the algorithm has been applied to many fields [34]. The algorithm in this paper uses the density peaks clustering to divide the majority class into several different sub-clusters, and the number of under-sampling in each sub-cluster is determined by the weight of the sub-clusters. To select the representative samples in the clusters, the local density of the samples is used as its fitness, and the selection probability of samples is calculated according to the fitness. The experiment was conducted on 13 imbalanced data sets with different actual application backgrounds, and the results show that the DPF-EL algorithm has better classification performance than other contrast algorithms.

The rest of this paper is arranged as follows: Section II introduces the theory of density peaks clustering and the decision tree algorithm. Section III introduces this paper's theory, steps, and algorithm design. Section IV introduces the experimental design and result analysis. Section V concludes the entire paper and points out its limitations.

II. RELATED THEORIES

A. DENSITY PEAKS CLUSTERING ALGORITHM

The basic idea of the density peaks clustering is to form clusters by calculating the local density of sample points and finding density peak points. This algorithm is based on the following assumptions:

1) Samples with high local density may be cluster centers.
2) The distance between cluster centers should be larger.

According to the above assumptions, the local density ρ_i and the minimum distance δ_i to other points with higher density are first needed to calculate to select the clustering centers. The method is given below.

Assume that the data set $D = \{x_1, x_2, \cdots, x_n\}^T$, for the local density ρ_i of any sample x_i, can be calculated using the Gaussian kernel function, as shown in (1).

$$\rho_i = \sum_j \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right) \tag{1}$$

where d_{ij} is the Euclidean distance between any two samples x_i and x_j, d_c is the cutoff distance, generally set to 2% of the Euclidean distance descending sort.

The minimum distance δ_i to other points with higher density is defined as follows:

$$\delta_i = \begin{cases} \min\left(d_{ij}\right), \rho_j > \rho_i \\ \max\left(d_{ij}\right), \rho_j \leqslant \rho_i \end{cases} \tag{2}$$

The density peaks clustering algorithm takes the samples with higher local density and higher minimum distance as cluster centers. After the cluster centers are determined, the remaining samples are assigned to the cluster to which the nearest with higher local density belongs.

B. DECISION TREE ALGORITHM

Decision tree [35] is a common classification model, which has been widely used in ensemble learning due to its simple structure and high classification accuracy. The ensemble

learning algorithm proposed in this paper uses the C4.5 algorithm to generate the base classifier.

C4.5 algorithm is a frequently used method to generate decision tree, and it takes the information gain ratio as the partition metric of the optimal feature. The information gain ratio is calculated as follows:

Assume that the data set D has k categories, where $k = 1, 2, \cdots, K$, p_k denotes the rate of the number of k-type samples to the total number of samples in data set D, data set D is divided into V sub-datasets by the eigenvalues of feature a, $|D^v|$ is the number of samples in the v sub-datasets, the information gain ratio of feature a is calculated as shown in (3).

$$Gain\_ratio(D, a) = \frac{Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent\ (D^v)}{-\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}} \tag{3}$$

where $Ent(D)$ refers to information entropy, which is used to measure the information purity of data set D, and the calculation formula is as shown in (4).

$$Ent(D) = -\sum_{k=1}^{K} p_k \log_2 p_k \tag{4}$$

III. THE PROPOSED ALGORITHM

A. ADAPTIVE UNDER-SAMPLING WEIGHT CALCULATION BASED ON DENSITY

In the existing clustering-based under-sampling methods, the under-sampling number of each cluster is usually determined according to a certain proportion or number, without considering the density of the samples in the cluster. To make the distribution of the data set consistent before and after sampling, the sampling number of the area with dense samples should be larger, and the sampling number of the area with sparse samples can be smaller. Therefore, the sampling weights are assigned to clusters according to it samples local density in this paper. The denser the sub-clusters, the larger their sampling weight, and the sparser the sub-clusters, the smaller their sampling weight.

According to formulas (1) and (2), the local density ρ_i and minimum distance δ_i of the majority class samples in data set D are calculated to generate C different sub-clusters D^k_{maj}, where $k = 1, 2, \cdots, C$, for each the majority sub-cluster formed by clustering, the density Rho^k_{maj} and sampling weight $Weight^k_{maj}$ were calculated by formulas (5) and (6).

$$Rho^k_{maj} = \sum_{i=1}^{\left|D^k_{maj}\right|} \rho_i \tag{5}$$

$$Weight^k_{maj} = \frac{Rho^k_{maj}}{\sum_{j=1}^{C} Rho^k_{maj}} \tag{6}$$

At last, the sampling weight $Weight^k_{maj}$ of the sub-clusters is multiplied by the number of the minority class in date set D to calculate the under-sampling number US^k_{maj} of each sub-cluster, as shown in formula (7).

$$US^k_{maj} = Weight^k_{maj} \times |D_{min}| \tag{7}$$

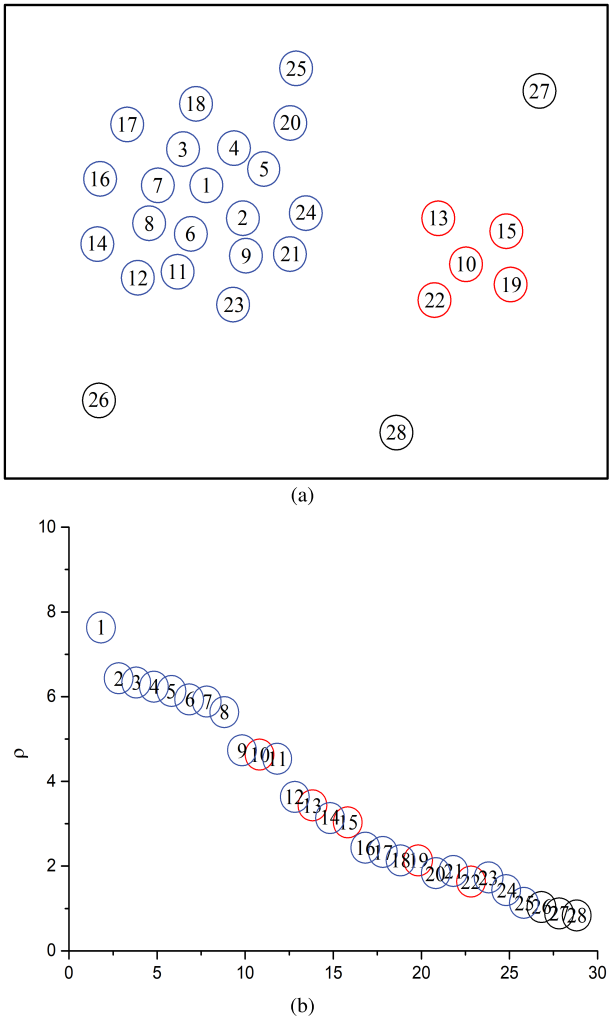


FIGURE 1. Density peaks clustering algorithm in two dimensions. (a) Data distribution. (b) Rank of density for the data.

B. UNDER-SAMPLING METHOD BASED ON FITNESS

This paper used fitness based on samples to choose the samples distributed among the center and periphery of the sub-clusters as much as possible. Because to some extent, the importance of samples can be approximately measured by density, if a sufficient number of high-density instances are selected, the learning models will have better classification performance.

For the sub-clusters generated using the density peaks clustering, the samples with higher local density are the central or peripheral points of the sub-clusters, and the samples with lower local density are boundary or outlier (or noise) points. It can be seen from Figure 1 that the local densities of the sub-cluster center and peripheral points are higher than that of boundary points. Points 26 to 28 are far from most of the

points and have lower local densities, which are regarded as outliers or noise points.

To make the center and peripheral points of sub-clusters have a larger selection chance, the local density of the samples is taken as their fitness. For instance, the fitness of sample point x_i in a sub-cluster is defined as formula (8).

$$f(x_i) = \rho_i \tag{8}$$

In genetic algorithms [36], fitness measures an individual's ability to adapt to the environment, and it is proportional to the selection probability. For point x_i, its fitness is defined as $f(x_i)$, and the selection probability $p(x_i)$ can be calculated using the formula (9).

$$p(x_i) = \frac{f(x_i)}{\sum_{j=1}^{\left|D_{maj}^k\right|} f(x_j)} \tag{9}$$

From the formulas (8) and (9), it can be concluded that the density of the samples in the sub-cluster is proportional to their selection probability. Hence, the central and peripheral points of the sub-clusters have a higher selection probability than the boundary and outlier points. In addition, the samples in the boundary region also have a certain probability of being selected, which will not result in the loss of useful samples related to the decision. For the convenience of description, the steps of the under-sampling method for a single cluster are given below.

Step1: calculate the selection probability $p(x_i)$ of the samples in this cluster according to the formulas (8) and (9).

Step2: calculate the cumulative probability P_i for the samples according to the formula (10).

$$P_i = \sum_{j=1}^{i} p(x_i) \tag{10}$$

Step3: generates a random number r within the interval [0, 1]. If $r < P_1$, select sample x_1; otherwise, select the sample x_i that satisfies condition $P_{i-1} \leq r \leq P_i$.

Step4: repeat Step3 until the number of under-sampling in this cluster is satisfied.

The pseudo code of the above steps can be summarized as in Algorithm 1. Each cluster is under-sampled according to Algorithm 1, and the samples obtained by under-sampling are merged with the minority class samples in data set D to form a balanced data set D'.

Algorithm 1 Under-Sampling Method Based on Fitness

Input: the majority class cluster, D_{maj}^k, under-sampling number of the cluster, US_{maj}^k
Output: the new majority class samples, new_majority

```
new_majority=[];
N ← size(D^k_maj);
h ← size(new_majority);
while h<US^k_maj do
    m ← 0; // m is the cumulative probability
    r ← Random(0, 1);
    for i = 1 to N do
        m ← m + p(x_i);
        if r ≤ m then
            return x_i;
        end if
    end for
    new_majority ← x_i; // if x_i is not in new_majority
end while
```

C. BASE CLASSIFIER GENERATION

C4.5 algorithm is used to train the decision tree classifier on the balanced data set D', and the depth of the decision tree is set to d. The steps for generating a decision tree using the C4.5 algorithm are given below:

Step1: for the data set D', the information gain ratio of all features is calculated according to the formulas (3) and (4), and the feature with the maximum information gain ratio is taken as the optimal partition, which is used to establish the root node, and the child nodes are generated according to the different values of the optimal partition feature.

Step2: in the same way as Step1, the feature with the maximum information gain ratio is selected as the optimal partition feature for generated the sub-nodes, and the subsequent branches are recursively established until the samples of nodes all belong to the same class or reach the set depth d.

Step3: the classification rules are extracted to obtain the corresponding base classifier.

D. ALGORITHM DESIGN

The algorithm design of DPF-EL mainly brings the Algorithm 1 into the training framework of ensemble learning, and improves the classification performance for imbalanced data by repeatedly sampling and training corresponding classifiers. The pseudo code of DPF-EL algorithm designed can be summarized as in Algorithm 2.

E. DPF-EL TIME COMPLEXITY ANALYSIS

The time complexity of DPF-EL mainly concentrated in two aspects, the analysis is as follows.

1) The time complexity of clustering the majority class using density peaks clustering. Since the time complexity of the density peaks algorithm is $O(n^2)$, so the time complexity of clustering the majority class using the density peaks clustering is $O(n^2)$.
2) The time complexity of T-round base classifier training. This paper uses the C4.5 algorithm to train and generate the base classifier. The time complexity of the C4.5 algorithm is related to the size of the balanced training set D', which is $O(p|D'|log|D'|)$, where p is the number of features contained in D'. Therefore, the training time complexity of T-round base classifier is $O(Tp|D'|log|D'|)$.

Algorithm 2 DPF-EL Design

Input: imbalanced data set, $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, number of clustering, C, number of iterations, T

Output: classification results

1: initialize the weight of x_i: $W_1(i) = 1/n, i = 1, 2, \cdots, n$;
2: clustering the majority class in data set D using density peaks clustering algorithm;
3: calculating the under-sampling number of each cluster according to formulas (5) to (7);
4: **for** $t = 1$ to T **do**
5: create a balance data set D'_t according to the under-sampling method in Algorithm 1;
6: use D'_t as the training data to train the base classifier h_t;
7: calculate the error rate of h_t: $e_t = \sum_{i=1}^{n} W_t(i) I\left(h_t\left(x_i\right) \neq y_i\right)$, where I is indicator function;
8: calculate the weight of h_t: $\alpha_t = \frac{1}{2} \ln\left(\frac{1-e_t}{e_t}\right)$;
9: update $W_t(i)$: $W_{t+1}(i) = \frac{W_t(i) \exp(-\alpha_t y_i h_t(x_i))}{\sum_{i=1}^{n} W_t(i) \exp(-\alpha_t y_i h_t(x_i))}$;
10: **end for** To use the ensemble classifier to classify sample, x_test:
11: initialize weight of each class to 0;
12: **for** $t = 1$ to T **do**
13: $c = h_t(x\_test)$; //c is the class predicted by h_t
14: add weight α_t to class c;
15: **end for**
16: return the class with the largest weight;

TABLE 1. Confusion matrix.

	Predicted positive	Predicted negative
Actual positive	TP	FN
Actual negative	FP	TN

To sum up, the time complexity of DPF-EL algorithm is $O(n^2) + O(Tp|D'|log|D'|)$.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EVALUATION METHODS

G-mean [37], AUC [38], and Balance [39] are commonly used to assess the classification performance of algorithms for imbalanced data. It can be represented by using a confusion matrix. The method is given below.

According to Table 1, the following evaluation metrics can be obtained.

True Positive Rate, the percentage of positive samples that are correctly classified.

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

False Positive Rate, the percentage of negative samples that are misclassified.

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

TABLE 2. KEEL data sets description.

Dataset	Size	Feature	N_maj, N_min	IR
abalone9-18	731	8	689, 42	16.4
dermatology-6	358	34	338, 20	16.9
glass1	214	9	138, 76	1.82
glass4	214	9	201, 13	15.46
haberman	306	3	225, 81	2.78
new-thyroid1	215	5	180, 35	5.14
new-thyroid2	215	5	180, 35	5.14
pima	768	8	500, 268	1.87
segment0	2308	19	1979, 329	6.02
vowel0	988	13	898, 90	9.98
wine	178	13	130, 48	2.71
winequalityred8vs6	656	11	638, 18	35.44
yeast1	1484	8	1055, 429	2.46

Specificity, the percentage of negative samples that are correctly classified, which is to measure the ability to identify negative classes.

$$Specificity = \frac{TN}{FP + TN} \tag{13}$$

G-mean, the geometric mean of true positive rate and specificity. If an algorithm achieves a higher G-mean value, it means that the algorithm has better classification performance for imbalanced data, and the method of calculation is given in (14).

$$G - mean = \sqrt{TPR \times Specificity} \tag{14}$$

AUC, the area under the ROC curve. The higher the AUC value, the higher the positive rate, meanwhile, the lower the false positive rate. The calculation formula is shown in (15).

$$AUC = \frac{1 + TPR - FPR}{2} \tag{15}$$

Balance, Balance is a method to measure the classification performance of algorithms for imbalanced data. A higher Balance value means the algorithm gets a better comprehensive classification performance. The calculation formula is shown in (16).

$$Balance = \frac{TPR + Specificity}{2} \tag{16}$$

B. EXPERIMENT DATA SETS

This paper uses 13 groups of imbalanced data sets from KEEL data set repository [40] to train and evaluate the algorithm. Since this paper only studies the two-category problem, the category "3" is selected as the minority class, and the other categories are selected as the majority class on the wine data set. The imbalance ratio distribution of the experimental data sets ranged from 1.82 to 35.44. See Table 2 for detailed information.

C. EXPERIMENTAL DESIGN AND COMPARATIVE RESULTS

In the experiment, this paper compared the proposed algorithm with AdaBoost [41], SMOTEBoost [29], RUSBoost [30], cluster-based under-sampling with boosting

TABLE 3. Average G-mean comparison of different method.

Dataset	AdaBoost	SMOTEBoost	RUSBoost	CUSBoost	NCL	CBU-NN	DPF-EL
abalone9-18	0.562	0.631	0.679	0.506	0.624	0.714	**0.761**
dematology-6	**0.994**	0.981	0.975	0.941	0.969	0.989	0.972
glass1	0.733	0.730	0.715	0.651	0.716	0.715	**0.740**
glass4	0.715	0.728	**0.879**	0.613	0.725	0.821	0.794
haberman	0.519	0.530	0.562	0.511	0.564	0.553	**0.635**
new-thyroid1	0.982	0.984	**0.994**	0.935	0.979	0.983	0.955
new-thyroid2	0.977	0.979	0.981	0.927	0.979	**0.985**	0.974
pima	0.567	0.603	0.582	0.611	0.603	0.593	**0.640**
segment0	0.989	0.987	**0.995**	0.974	0.988	0.983	0.988
vowel0	0.949	0.949	**0.978**	0.894	0.931	0.953	0.960
wine	0.957	0.956	0.961	0.872	0.942	0.943	**0.962**
winequalityred8vs6	0.323	0.404	0.455	0.612	0.344	0.624	**0.676**
yeast1	0.629	0.674	0.637	0.679	0.697	0.658	**0.702**

TABLE 4. Average AUC comparison of different method.

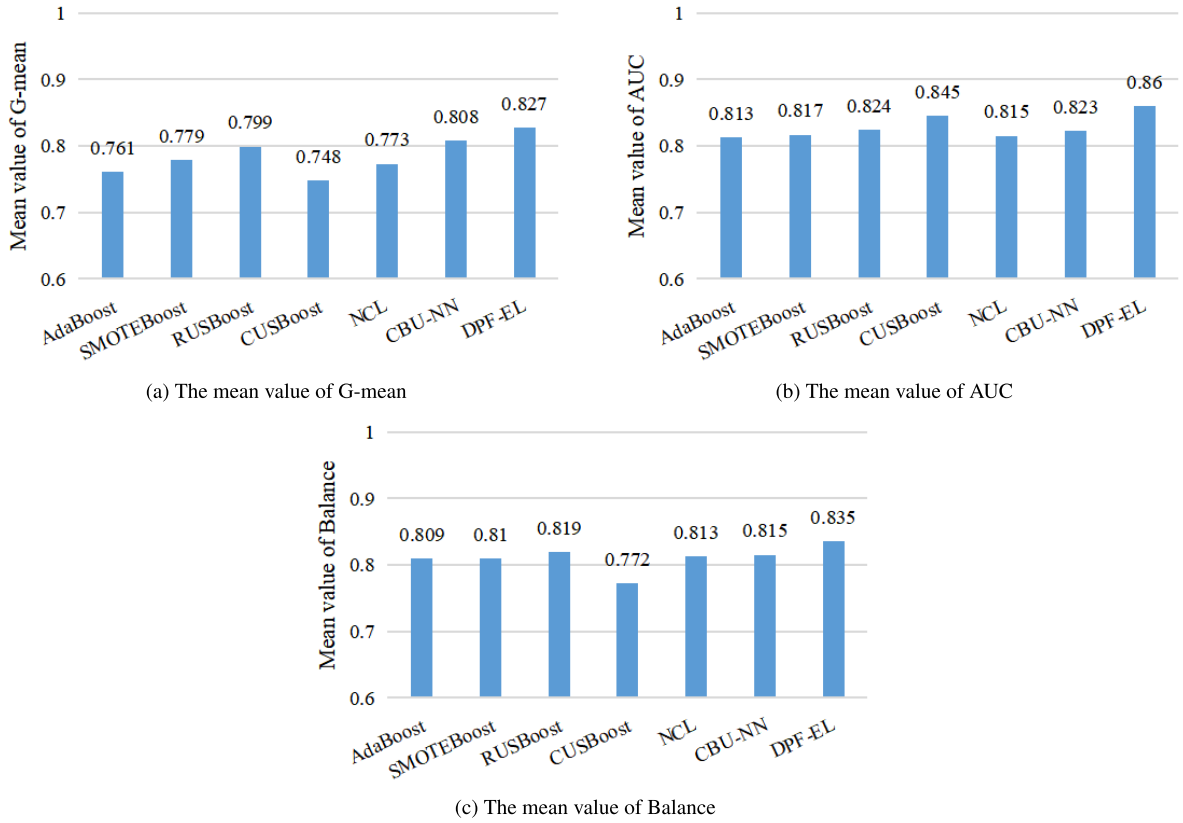
Dataset	AdaBoost	SMOTEBoost	RUSBoost	CUSBoost	NCL	CBU-NN	DPF-EL
abalone9-18	0.674	0.696	0.741	0.724	0.706	0.716	**0.816**
dematology-6	0.990	0.994	0.989	0.995	0.987	0.987	**0.998**
glass1	0.745	0.748	0.721	0.686	0.722	0.732	**0.797**
glass4	0.875	0.829	0.920	**0.934**	0.817	0.861	0.827
haberman	0.551	0.564	0.531	0.607	0.580	0.595	**0.647**
new-thyroid1	0.982	0.980	0.990	0.996	0.982	0.983	**0.998**
new-thyroid2	0.975	0.983	0.992	0.986	0.983	0.984	**0.998**
pima	0.609	0.608	0.602	0.668	0.620	0.581	**0.674**
segment0	0.987	0.988	0.994	0.993	0.989	0.980	**0.995**
vowel0	0.942	0.951	0.983	0.988	0.932	0.958	**0.992**
wine	0.955	0.967	0.978	0.977	0.945	0.954	**0.988**
winequalityred8vs6	0.624	0.634	0.625	0.650	0.626	0.639	**0.670**
yeast1	0.662	0.684	0.658	0.781	0.708	0.738	**0.782**

TABLE 5. Average Balance comparison of different method.

Dataset	AdaBoost	SMOTEBoost	RUSBoost	CUSBoost	NCL	CBU-NN	DPF-EL
abalone9-18	0.676	0.696	0.695	0.659	0.690	0.713	**0.764**
dematology-6	0.989	**0.990**	0.974	0.945	0.984	0.989	0.976
glass1	0.735	0.758	**0.794**	0.678	0.733	0.721	0.771
glass4	0.871	0.810	**0.926**	0.734	0.823	0.862	0.824
haberman	0.527	0.540	0.561	0.571	0.571	0.546	**0.641**
new-thyroid1	0.984	0.982	0.983	0.951	0.981	**0.989**	0.960
new-thyroid2	0.981	0.983	**0.990**	0.932	0.982	0.986	0.970
pima	0.593	0.611	0.608	0.617	0.618	0.589	**0.643**
segment0	0.988	0.986	**0.995**	0.974	0.988	0.983	0.989
vowel0	0.936	0.953	0.966	0.906	0.939	0.950	**0.969**
wine	0.961	0.947	0.968	0.870	0.940	0.949	**0.968**
winequalityred8vs6	0.616	0.619	0.550	0.524	0.629	0.654	**0.677**
yeast1	0.665	0.668	0.643	0.686	0.700	0.665	**0.712**

(CUSBoost) [42], neighborhood cleaning rule (NCL) [43], and clustering-based under-sampling (CBU) [32]. Among them, the reference [32] uses two strategies for under-sampling, this paper chooses the second strategy called CBU-NN (clustering-based under-sampling with nearest neighbors of the cluster centers) with better classification performance as comparison algorithm, and the number of clusters is set to the quantity of the minority class samples. All algorithms use the C4.5 algorithm training the base classifier, G-mean, AUC, and Balance as the method of evaluation. To make the experimental results fair and objective, the algorithm in this paper is run ten times with ten-fold cross-validation, and their mean evaluation metrics values are shown in Table 3 to Table 5. The bold value is the highest under this evaluation metrics.

From Table 3 to Table 5, it can be seen that the DPF-EL algorithm has achieved high G-mean and Balance evaluation values on 7 data sets and high AUC evaluation values on 12 data sets, compared with G-mean and Balance, the effect of DPF-EL algorithm is more obvious when using AUC for evaluation. On the data sets abalone9-18, haberman, pima, wine, winequalityred8vs6 and yeast1, the comprehensive



(a) The mean value of G-mean

(b) The mean value of AUC

(c) The mean value of Balance

FIGURE 2. The mean value bars of the different methods on the metrics.

classification performance of DPF-EL algorithm is better, their G-mean, AUC, and Balance values are all the highest, especially on the winequalityred8vs6 data set with imbalance rate as high as 35.44, the performance of this algorithm is well, compared with CBU-NN algorithm, the G-mean value is increased by 5.2%, the AUC value is increased by 3.1%, and the Balance value is increased by 2.3%. It shows that the classification performance of the proposed algorithm is still better on the data set with high imbalance rate, at the same time, it is proved that under-sampling combined with ensemble learning is a better method to solve the imbalanced data classification problem.

In order to more visually compare the classification performance between different methods, Figure 2 shows the mean values of the different evaluation metrics of 7 methods on 13 data sets. It can be seen in Figure 2 that compared with other methods, the mean value of the different evaluation metrics of the proposed method has been improved to a certain extent, which shows that the classification performance of the proposed method is better than other methods.

On the whole, compared with other methods, the G-mean, AUC and Balance evaluation values of the proposed method are higher, which indicates that this method has a higher classification accuracy and better classification performance for imbalanced data.

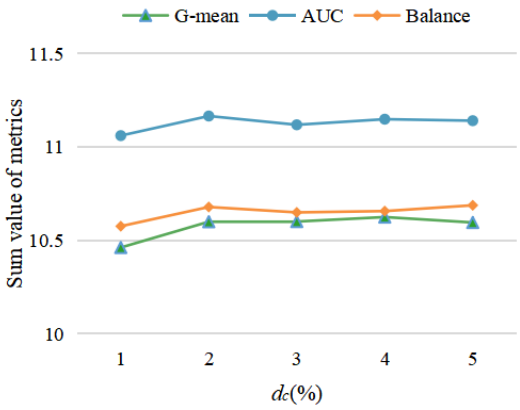


FIGURE 3. Effect of different d_c on evaluation metrics.

D. THE IMPACT OF CUTOFF DISTANCE

In the under-sampling phase, the selection probability of the samples is a positive correlation with their local density. To observe the influence of the parameter d_c value on the algorithm performance, the parameter was set with different values. Figure 3 shows the changes of evaluation metrics values of the DPF-EL algorithm under different d_c (1%, 2%, 3%, 4%, 5%). The evaluation metrics values in Figure 3 are the sum of the different metrics values on 13 groups data sets.

As can be seen from Figure 3, when $d_c = 1\%$, the sum value of all metrics are lowest, and when d_c changes from 2% to 5%, the line chart changes gently, and the evaluation metrics values that increase or decrease are small. On the whole, when the d_c changes little, it has a limited effect on the performance of the algorithm.

V. CONCLUSION

Under-sampling combined with ensemble learning can effectively solve the problems of imbalanced data learning and bring about the diversity of data. However, the existing algorithms usually have two problems: the size of clusters after clustering is different, how to reasonably allocate the number of under-sampling and select representative samples.

This paper proposed an algorithm called DPF-EL. This algorithm calculates the number of under-sampling for each sub-cluster according to the density of samples in the cluster, which keeps the consistency of data distribution before and after sampling. The fitness concept of genetic algorithm is used to model the samples of the sub-clusters, so that the central and the surrounding samples of the sub-clusters have a larger selection probability, and the representative samples in the cluster are reserved as much as possible. At last, the feasibility of this method is verified through the experiments.

In real-life applications, imbalanced data may have multiple classification circumstances. The following work will use the DPF-EL algorithm to study the classification of multi-class imbalanced data sets. In addition, since the method in this paper uses the density peaks clustering algorithm, the running time of the algorithm in this paper is slightly longer in the data set with a large amount of data. It is also worth studying how to shorten the running time in a parallel way.

REFERENCES

[1] S. Dhar and V. Cherkassky, "Development and evaluation of cost-sensitive universum-SVM," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 806–818, Apr. 2015.

[2] P. Liu, M. Hong, D. Huang, Y. Luo, and S. Wang, "Joint ADASYN and AdaBoost SVM for imbalanced learning," *J. Beijing Univ. Technol.*, vol. 43, no. 3, pp. 368–375, Mar. 2017.

[3] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance," *Neural Netw.*, vol. 21, nos. 2–3, pp. 427–436, Mar. 2008.

[4] M. Sharifmoghadam and H. Jazayeriy, "Breast cancer classification using AdaBoost-extreme learning machine," in *Proc. 5th Iranian Conf. Signal Process. Intell. Syst. (ICSPIS)*, Dec. 2019, pp. 1–5.

[5] S. Saxena, S. Shukla, and M. Gyanchandani, "Breast cancer histopathology image classification using kernelized weighted extreme learning machine," *Int. J. Imag. Syst. Technol.*, vol. 31, no. 1, pp. 168–179, Mar. 2021.

[6] V. Engen, J. Vincent, and K. Phalp, "Enhancing network based intrusion detection for imbalanced data," *Int. J. Knowl.-Based Intell. Eng. Syst.*, vol. 12, nos. 5–6, pp. 357–367, 2008.

[7] J. Liu, J. He, W. Zhang, T. Ma, Z. Tang, J. P. Niyoyita, and W. Gui, "ANID-SEoKELM: Adaptive network intrusion detection based on selective ensemble of kernel ELMs with random features," *Knowl.-Based Syst.*, vol. 177, pp. 104–116, Aug. 2019.

[8] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," *IEEE Sensors Lett.*, vol. 3, no. 1, pp. 1–4, Jan. 2019.

[9] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Inf. Sci.*, vol. 479, pp. 448–455, Apr. 2019.

[10] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Cost sensitive credit card fraud detection using Bayes minimum risk," in *Proc. 12th Int. Conf. Mach. Learn. Appl. (ICMLA)*, vol. 1, Dec. 2013, pp. 333–338.

[11] M. Di Martino, F. Decia, J. Molinelli, and A. Fernández, "Improving electric fraud detection using class imbalance strategies," in *Proc. Int. Conf. Pattern Recognit. Appl. Methods*, vol. 2, 2012, pp. 135–141.

[12] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9268–9277.

[13] K. Rujirakul and C. So-In, "Histogram equalized deep PCA with ELM classification for expressive face recognition," in *Proc. Int. Workshop Adv. Image Technol. (IWAIT)*, Jan. 2018, pp. 1–4.

[14] X. Ximeng, Y. Rennong, and Y. Yang, "Threat assessment in air combat based on ELM neural network," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Mar. 2019, pp. 114–120.

[15] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, Jan. 2002.

[17] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, p. 20–29, Jun. 2004.

[18] F. Feng, K.-C. Li, J. Shen, Q. Zhou, and X. Yang, "Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification," *IEEE Access*, vol. 8, pp. 69979–69996, 2020.

[19] C. L. Castro and A. P. Braga, "Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 888–899, Jun. 2013.

[20] H. Yu, C. Sun, X. Yang, S. Zheng, Q. Wang, and X. Xi, "LW-ELM: A fast and flexible cost-sensitive learning framework for classifying imbalanced data," *IEEE Access*, vol. 6, pp. 28488–28500, 2018.

[21] S. Datta and S. Das, "Near-Bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs," *Neural Netw.*, vol. 70, pp. 39–52, Oct. 2015.

[22] R. Batuwita and V. Palade, "FSVM-CIL: Fuzzy support vector machines for class imbalance learning," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 558–571, Jun. 2010.

[23] H. Yu, C. Sun, X. Yang, S. Zheng, and H. Zou, "Fuzzy support vector machine with relative density information for classifying imbalanced data," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 12, pp. 2353–2367, Dec. 2019.

[24] X. Fan and Z. He, "A fuzzy support vector machine for imbalanced data classification," in *Proc. Int. Conf. Optoelectron. Image Process. (ICOIP)*, Nov. 2010, pp. 11–14.

[25] C. Su, S. Ju, Y. Liu, and Z. Yu, "Improving random forest and rotation forest for highly imbalanced datasets," *Intell. Data Anal.*, vol. 19, no. 6, pp. 1409–1432, Jan. 2015.

[26] S. Bo, "Research on the classification of high dimensional imbalanced data based on the optimizational random forest algorithm," in *Proc. 9th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Jan. 2017, pp. 228–231.

[27] M. P. Paing and S. Choomchuay, "Improved random forest (RF) classifier for imbalanced classification of lung nodules," in *Proc. Int. Conf. Eng., Appl. Sci., Technol. (ICEAST)*, Jul. 2018, pp. 1–4.

[28] L. Nanni, C. Fantozzi, and N. Lazzarani, "Coupling different methods for overcoming the class imbalance problem," *Neurocomputing*, vol. 158, pp. 48–61, Jun. 2015.

[29] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Proc. PKDD*, vol. 2838, 2003, pp. 107–119.

[30] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 1, pp. 185–197, Jan. 2010.

[31] C. Y. Cui, F. Y. Cao, and J. Y. Liang, "Adaptive under-sampling based on density peak clustering," *Pattern Recognit. Artif. Intell.*, vol. 33, no. 9, pp. 811–819, Sep. 2020.

[32] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci.*, vols. 409–410, pp. 17–26, Oct. 2017.
[33] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[34] M. Parmar, D. Wang, X. Zhang, A. H. Tan, C. Miao, J. Jiang, and Y. Zhou, "REDPC: A residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, Jul. 2019.
[35] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[36] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: Concepts and applications [in engineering design]," *IEEE Trans. Ind. Electron.*, vol. 43, no. 5, pp. 519–534, Oct. 1996.
[37] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[38] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, pp. 1145–1159, Jul. 1997.
[39] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 3121–3124.
[40] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Log. Soft Comput.*, vol. 17, no. 23, pp. 255–287, 2011.
[41] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, vol. 96, 1996, pp. 148–156.
[42] F. Rayhan, S. Ahmed, A. Mahbub, R. Jani, S. Shatabda, and D. M. Farid, "CUSBoost: Cluster-based under-sampling with boosting for imbalanced classification," in *Proc. CSITSS*, Dec. 2017, pp. 1–5.
[43] J. Laurikkala, "Improving identification of difficult small classes by balancing class distribution," in *Proc. Conf. AI Med. Eur., Artif. Intell. Med.*, 2001, pp. 63–66.

HUI XU is currently pursuing the master's degree with Yantai University. His research interests include machine learning and data mining.

QICHENG LIU received the Ph.D. degree in engineering from the China University of Petroleum, Beijing. He is currently a Professor with the School of Computer and Control Engineering, Yantai University. His research interests include big data, multi-agent systems, and data mining.