mathematics

MDPI

Article

Imbalanced Data Classification Based on Improved Random-SMOTE and Feature Standard Deviation

Ying Zhang [1], Li Deng [1,*] and Bo Wei [2,3]

[1] School of Science, Zhejiang Sci-Tech University, Hangzhou 310018, China; 202120102071@mails.zstu.edu.cn
[2] School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China; weibo@zstu.edu.cn
[3] Longgang Research Institute, Zhejiang Sci-Tech University, Longgang 325000, China
* Correspondence: lideng75@zstu.edu.cn

Abstract: Oversampling techniques are widely used to rebalance imbalanced datasets. However, most of the oversampling methods may introduce noise and fuzzy boundaries for dataset classification, leading to the overfitting phenomenon. To solve this problem, we propose a new method (FSDR-SMOTE) based on Random-SMOTE and Feature Standard Deviation for rebalancing imbalanced datasets. The method first removes noisy samples based on the Tukey criterion and then calculates the feature standard deviation reflecting the degree of data discretization to detect the sample location, and classifies the samples into boundary samples and safety samples. Secondly, the K-means clustering algorithm is employed to partition the minority class samples into several sub-clusters. Within each sub-cluster, new samples are generated based on random samples, boundary samples, and the corresponding sub-cluster center. The experimental results show that the average evaluation value obtained by FSDR-SMOTE is 93.31% (93.16%, and 86.53%) in terms of the F-measure (G-mean, and MCC) on the 20 benchmark datasets selected from the UCI machine learning library.

Keywords: imbalanced data; feature standard deviation; oversampling strategy; Random-SMOTE

MSC: 68T01; 68T07

Citation: Zhang, Y.; Deng, L.; Wei, B. Imbalanced Data Classification Based on Improved Random-SMOTE and Feature Standard Deviation. *Mathematics* **2024**, *12*, 1709. https://doi.org/10.3390/math12111709

Academic Editor: António Lopes

Received: 25 April 2024
Revised: 16 May 2024
Accepted: 27 May 2024
Published: 30 May 2024

1. Introduction

In the field of machine learning, the problem of imbalanced data classification [1] has attracted great interests. In many real-world scenarios, due to a variety of factors such as bias in data collection [2], inherent rarity of categories [3], differences in the frequency of events [4,5], etc., research often needs to deal with a large number of imbalanced datasets, where the number of samples of one class is much larger than that of other classes. However, traditional classification algorithms usually assume that the number of samples in each class is balanced during the training phase, and thus traditional classification algorithms often exhibit degraded performance when dealing with imbalanced data. Specifically, due to the higher number of samples in the majority class, the model is more likely to learn the features of the majority class during the training process, resulting in the classifier being biased towards the majority class during prediction [6]. Bias toward majority class samples makes the classifier less capable of identifying minority class samples, and may even completely ignore the importance of the minority class, which can lead to misclassification in practical applications [7,8]. Therefore, an in-depth study of imbalanced classification algorithms is crucial to improve the performance and generalization ability of classifiers, especially when it comes to minority class samples with practical applications.

Currently, the existing methods for addressing imbalances in datasets are categorized into data-level and algorithm-level methods. Data-centric methods involve resampling techniques, such as augmenting the minority class with oversampling strategies [9], reducing the number of the majority class samples with undersampling strategies [10], and achieving

balance with hybrid sampling methods [11]. In addition, algorithm-level methods dynamically focus on difficult-to-classify samples by adjusting category weights, implementing cost-sensitive learning [12], using ensemble learning [13,14], and so on.

Oversampling is one of the effective means of solving the problem of imbalanced data classification because oversampling can rebalance the dataset while maintaining the data characteristics. Among oversampling techniques, randomly replicating the minority class samples (ROS) [15] is the simplest and most straightforward oversampling method; however, ROS may introduce noisy instances and overfitting risks since the data distribution is not taken into account. In contrast, the oversampling technique of synthetizing minority class samples [16,17] can improve the classification accuracy of the model to some extent, but it still suffers from these problems mentioned above. To solve these problems, researchers have invested a lot of effort in the data preprocessing stage, especially in the noise filtering phase. For example, Nawaf et al. [18] used Tukey's rule to remove outliers and noisy samples from the original dataset. Liang [19] combined support vector machine and nearest neighbor algorithms to deal with noise to reduce its effect on the oversampling phase. Therefore, when dealing with imbalanced data, fully considering the characteristics of data distribution plays a crucial role in improving algorithm performance.

To further enhance the effectiveness of synthesizing new samples, some existing oversampling methods focus on the minority class instances located near the decision boundary after denoising. For instance, based on the confusing information of the samples, Zhang et al. [20] classified the minority class samples into three types: noise samples, boundary samples, and safe samples, where only boundary samples are selected for generating new synthetic samples. In this way, the newly generated samples are closer to the decision boundary, which can improve the performance of the classifier. In addition, Cheng et al. [21] used Natural Neighbors (NaN) to calculate the local density of the samples and filtered the representative samples with maximum local density, aiming at changing the selection strategy for resampling samples. The common goal of all these methods is to improve the quality and effectiveness of synthetic samples and then optimize the performance of the classifier in the imbalanced data classification problem. However, for identifying the minority class samples located near the decision boundary, most of the research tends to focus on the relationships between samples, without considering the different characteristics of them.

In addition to changing the distribution region of the synthetic samples by adjusting their weights, comprehensive strategies for generating new samples are particularly important when improving the performance of oversampling algorithms. For example, stochastic linear interpolation is a common method for generating a new sample, which is based on a stochastic function and two target samples. Although the linear interpolation method is more effective in most cases, it may trigger the problem of redundancy in some regions with high sample density, leading to the ineffectiveness of the synthesized samples. To address this issue, Dong et al. [22] proposed the Random-SMOTE method, which interpolates within a triangular region consisting of the cluster center and two randomly selected neighbor samples. Random-SMOTE can enhance the diversity of synthetic samples and avoid generating redundant samples. Although Random-SMOTE can alleviate the problem of sample redundancy to a certain extent, Random-SMOTE does not fully consider the imbalance within the sample classes, resulting in the samples in the sparse region still carrying limited classification information.

To address the above problems, this paper proposes an imbalanced data classification method (FSDR-SMOTE) based on Random-SMOTE and feature standard deviation. In FSDR-SMOTE, noisy samples are firstly removed based on the Tukey criterion, and the feature standard deviation reflecting the degree of data discretization is utilized to detect the sample locations. Furthermore, the improved Random- SMOTE method is proposed to enhance the stability and accuracy of the algorithm when maintaining the diversity of the synthesized samples. The main contributions of this work are summarized as follows:

- A new strategy based on feature standard deviation is proposed to detect the locations of samples. Feature standard deviation strategy utilizes standard deviation in feature dimensions among different samples to construct a boundary sample set, aiming to optimize the position of the decision boundary.
- An improved three-point interpolation method proposed in Random-SMOTE is presented to synthesize samples, where both inter-class and intra-class attributes are taken into account, with the purpose of avoiding overfitting and optimizing the decision boundary.
- A new method (FSDR-SMOTE) is proposed to deal with imbalanced data. The FSDR-SMOTE can enhance the diversity of synthetic samples in the minority class, reduce the probability of generating noisy samples, and improve the classification performance of the classifier.

2. Related Work

In recent years, many research efforts have been focused on how to improve classification accuracy of algorithms by utilizing useful information carried by the minority class samples effectively [23,24]. The oversampling method is one of the most representative methods for solving the imbalanced data classification problem, which can realize the balance between samples of different classes by synthesizing new minority class samples.

As one of the representative oversampling methods, the SMOTE was proposed by Chawla et al. [9]. In SMOTE, new samples are generated by linearly interpolating between a sample x_i chosen arbitrarily from the minority class and any one x_j of the k-nearest neighbors of that sample. When dealing with an imbalanced data classification task, the SMOTE can mitigate the overfitting problem and improve the classification performance of the classifier. The procedure for synthesizing minority class samples is as follows:

$$x_{new} = x_i + rand(0,1) \times (x_i - x_j) \tag{1}$$

where $rand(0,1)$ denotes a random number chosen from $(0,1)$.

Although the SMOTE can alleviate the overfitting phenomenon effectively, the characteristic of data distribution has not been taken into account. To address this issue, HE et al. [25] proposed an adaptive integrated oversampling method, named ADASYN, which assigns minority class samples with different weights. Borderline-SMOTE [26] proposed by Han et al. can increase the number of borderline minority class samples to alleviate the impact of class overlap; however, it is difficult to guarantee the diversity of synthesized new samples.

The SMOTE-Tomek [27] is a hybrid sampling method used for balancing datasets. Here are the steps involved: First, the SMOTE algorithm is employed to oversample the minority class samples in the dataset. Next, the nearest neighbors of all majority class samples in the dataset are calculated. If the nearest neighbor of a majority class sample is a minority class sample, a Tomek link is established. Finally, all Tomek links are removed, aiming at obtaining a balanced dataset. The schematic diagram of the SMOTE-Tomek method is shown in Figure 1.

SVMSMOTE [28] focuses on generating samples of the minority class along the decision boundary. The process of creating synthetic samples with SVMSMOTE is as follows. First, a Support Vector Machine (SVM) [29] is trained based on the dataset, which can produce multiple support vectors. Then, for each support vector, the k-nearest neighbors from the minority class samples are found. If the number of majority class samples is less than half of its nearest neighbors, the extrapolation method is employed to synthesize minority class samples for extending the area of the minority class. Conversely, the interpolation method is utilized to synthesize minority class samples.

RN-SMOTE [30] is an oversampling method that integrates SMOTE and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [31]. Firstly, RN-SMOTE employs SMOTE for oversampling to achieve a balanced dataset, which may contain some noise samples. Subsequently, DBSCAN clustering is utilized to eliminate the noise within

the minority class samples, with the purpose of obtaining a clean dataset. Finally, SMOTE is reapplied to oversample the now balanced dataset. The main process of RN-SMOTE is shown in Figure 2, where the bars represent the quantity of samples [30].

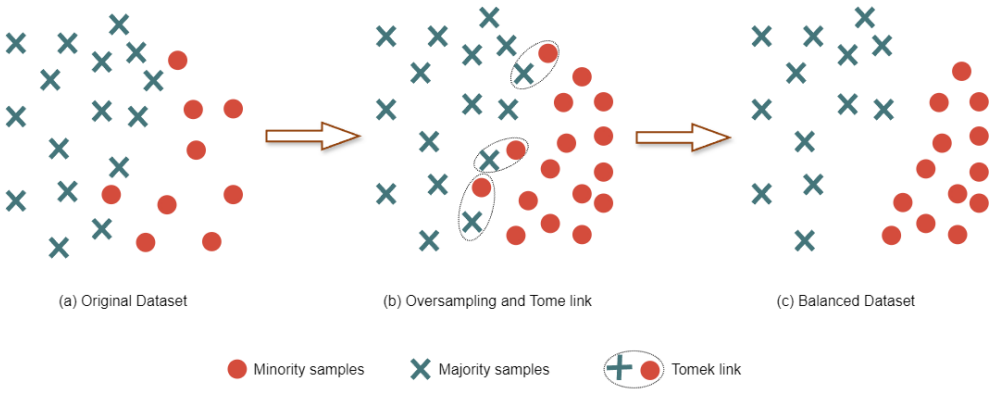


Figure 1. Schematic diagram of SMOTE-Tomek method.

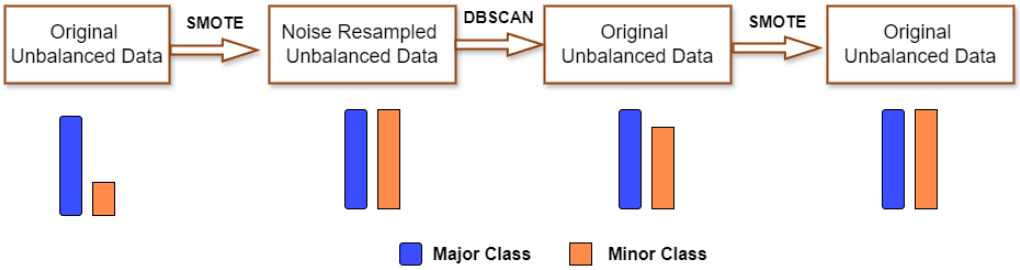


Figure 2. Overview of the RN-SMOTE method [30].

In order to improve the strategy of linear interpolation synthetic samples between two points in SMOTE, Dong et al. [22] proposed Random-SMOTE based on the SMOTE. The main idea of Random-SMOTE is to construct a triangular region, and then generate new samples randomly within the region. New generated samples are closer to the true distribution of samples, which can help to improve the classification performance of the classifier biasing towards the minority class samples. The process of Random-SMOTE for synthesizing minority class samples is shown in Figure 3.

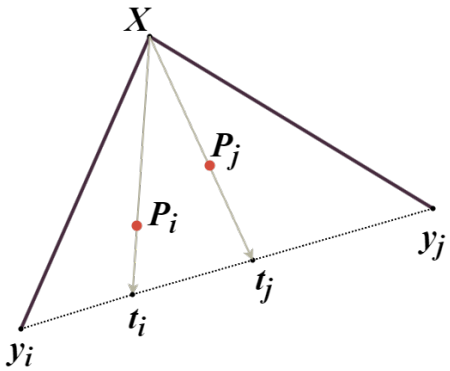


Figure 3. Schematic diagram of Random-SMOTE for synthesizing the new sample.

In the Random-SMOTE algorithm, for each sample x in the minority class, two samples y_i and y_j are selected randomly from them to form a triangular region. First of all, two temporary samples, namely t_i and t_j, are generated by using the following method:

$$t_k = y_i + rand(0,1) \times (y_i - y_j), \quad k = i, j \tag{2}$$

Then, the linear interpolation strategy is utilized based on sample x and temporary samples (t_i and t_j) to generate two new minority class samples (p_i and p_j), shown as Equation (3):

$$p_k = x + rand(0,1) \times (x - t_k), \quad k = i, j \tag{3}$$

Finally, the newly synthesized samples are added to the original dataset. In this way, the process for synthesizing new samples is repeated until the dataset is balanced.

Studies have shown that the Random-SMOTE method can improve the quality of newly synthesized samples to some extent; however, there are still some shortcomings. First, the characteristic of the distribution of samples is not taken into account, which may reduce the diversity of newly synthesized samples. Second, the characteristics of the samples are not taken seriously in the Random-SMOTE method when synthesizing new samples, which may lead to the introduction of noise samples and the reduction in performance of the classifier. Therefore, this paper proposes a new oversampling method, namely FSDR-SMOTE, where different features of samples are taken into account to identify the sample locations when synthesizing new samples.

3. Method

To deal with imbalanced data classification effectively, an improved Random-SMOTE method based on feature standard deviation and adaptive denoising, namely FSDR-SMOTE, is proposed in this section. The main steps of FSDR-SMOTE include three parts. The first part is data pre-processing, where the Tukey rule [32] and K-means [33] algorithms are employed in minority class samples for denoising and clustering, respectively. Based on feature standard deviation, the minority class samples are divided into the boundary samples and safety samples in the second part. Finally, the process of synthesizing new samples in the Random-SMOTE is improved to reduce the influence caused by intra-class imbalance. The flowchart of FSDR-SMOTE is shown in Figure 4, and the pseudo-code for FSDR-SMOTE is given in Algorithm 1.

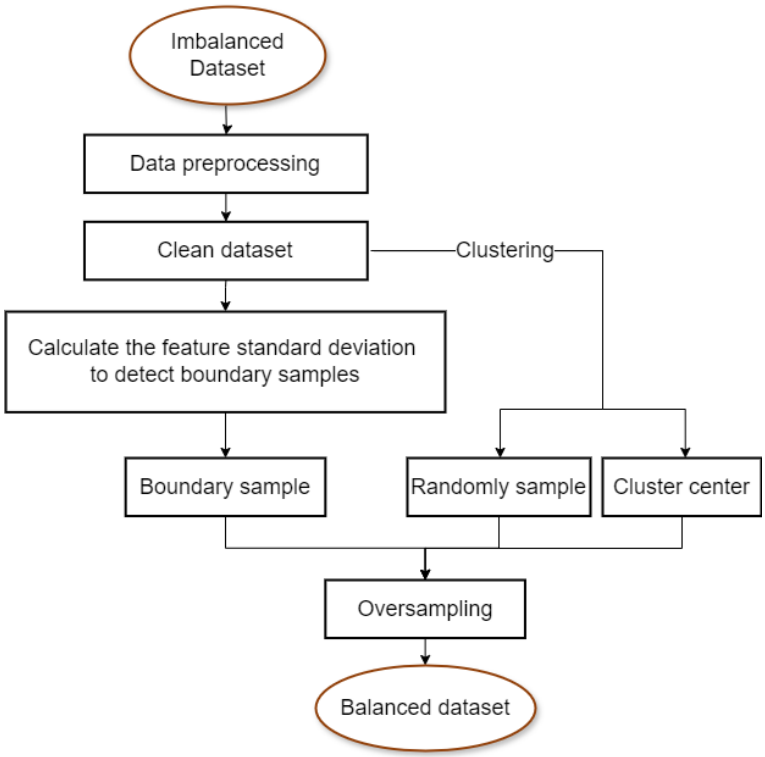


Figure 4. The flowchart of the FSDR-SMOTE.

Algorithm 1 FSDR-SMOTE

Input: Majority Sample Set X_{maj}; Minority Sample Set X_{min}; Number of New Synthetic Samples N; Constant in Tukey rule r; Constant of the characteristic standard deviation discriminant boundary sample k.

Output: Synthetic minority class samples X_{gen}.

$X_{gen} \leftarrow \emptyset$;
for $x_i \in X_{min}$ **do**
  Calculate the distance d between x_i and $\bar{x}_{min}$, sort in ascending order;
  Calculate the ub and lb according to Equations (4)–(8);
  if $d_i > ub$ **then**
    remove x_i from X_{min};
  end if
end for
$X_{minc} \leftarrow X_{min}$;
Use K-means to divide X_{minc} into P clusters, denoted as $L_1, L_2, \cdots, L_P$, and get $C = \{c_i\}, i = 1, 2, 3, \cdots, p$, where c_i is the cluster center of each subcluster;
for $x_i \in X_{minc}$ **do**
  Calculate σ_j according to Equation (10);
  Boundary sample sets B_j are screened according to Equation (11);
end for
$B \leftarrow \cup B_j$;
for n in N **do**
  Randomly select sample x_i in set B and record the cluster L_i to which x_i belongs and the center c_i of that cluster;
  while True **do**
    Randomly select a sample y_i from cluster L_i;
    if $dist(x_i, c_i) > dist(x_i, y_i)$ **then**
      break
    end if
  end while
  New sample s_i is generated from the three points x_i, y_i, and c_i according to the synthesis rule of Equations (13) and (14);
  $X_{gen} \leftarrow s_i$;
end for

3.1. Data Preprocessing

Tukey's rule [32] is an effective method for identifying outliers in multivariate data. When the dataset $X = x_1, x_2, x_3, \cdots, x_n$, where $X \in R$ and $x_1 \leq x_2 \leq x_3 \leq \cdots \leq x_n$. The interquartile range (IQR) can be obtained based on the difference between the first quartile ($Q1$) and the third quartile ($Q3$), as shown in Equations (4) and (5), respectively. Based on IQR, the degree of dispersion of the data distribution can be reflected, as shown in Equation (6). Furthermore, lb and ub mean the lower bound and upper bound of IQR, as shown in Equations (7) and (8), respectively.

$$Q1 = x_i \mid i = round((N+1) \times 0.25) \tag{4}$$

$$Q3 = x_j \mid j = round((N+1) \times 0.75) \tag{5}$$

$$IQR = Q3 - Q1 \tag{6}$$

$$lb = Q3 - r \times IQR \tag{7}$$

$$ub = Q1 + r \times IQR \tag{8}$$

In the above process, the i represents the index of the data point, N represents the total number of data points, and the default value of r proposed by Tukey is 1.5.

In our work, the Tukey criterion is improved to detect the noise effectively in imbalanced data classification task. For a dataset $X_{min} = \{x^1_{min}, x^2_{min}, x^3_{min}, \cdots, x^m_{min}\}$ containing m minority class samples, the mean $\bar{x}_{min}$ at each dimension of the minority samples is taken

as the center of them. The Euclidean distance $d_j = |x_j - \bar{x}_{min}|$ between each minority class sample x_j and the mean $\bar{x}_{min}$ is calculated and then sorted in ascending order. In particular, the upper and lower bounds of Tukey's criterion are calculated using Equations (4)–(8). For a sample x_s, if $d_s < lb$, the sample is retained because d_s is close to the minority class center $\bar{x}_{min}$. For a sample that exceeds the upper bound, i.e., $d_s > lb$, d_s is defined as noise and removed from the dataset X_{min}, aiming at obtaining the noise-reduced minority class sample set (X_{minc}). By employing the k-means clustering algorithm, our work can divide the noise-reduced minority class samples into several clusters, where the center of each cluster is recorded.

3.2. Boundary Sample Screening

Research suggests that the generalization performance of most oversampling methods is affected since they do not pay enough attention to samples in the boundary region. To address this issue, the relative positions of boundary samples in each sub-cluster consisted of the minority class samples are taken into account. In this way, the oversampling method can be optimized effectively by improving the robustness and accuracy of the classifier in our work. For an n-dimensional minority sample set $X = \{x_1, x_2, x_3, \cdots, x_m\}$, sample $x_i = \{a_1^i, a_2^i, a_3^i, \cdots, a_n^i\}$ containing m samples, where $i = 1, 2, \cdots, m$.

$$\bar{a}_j = \frac{\sum_{k=1}^{m} a_j^k}{m} \tag{9}$$

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{m} (a_j^i - \bar{a}_j)^2}{m}}, \quad j = 1,\ 2,\ \cdots,\ n \tag{10}$$

$$B_j = \{x_i \mid |a_j^i - \bar{a}_j| > k \cdot \sigma_j\}, \quad i = 1,\ 2,\ \cdots,\ m \tag{11}$$

$$B = \bigcup_{j=1}^{n} B_j \tag{12}$$

where a_j^i denotes the value of sample i in the jth dimension; $\bar{a}_j$ denotes the mean of the sample in the jth dimension; σ_j denotes the standard deviation of the sample in the jth dimension; and k is a set of parameters.

Furthermore, the mean value and the standard deviation of all samples in each dimension can be calculated by Equations (9) and (10), respectively. Samples that satisfy Equation (11) are put into B_j, and then form the boundary sample set B, which will be used for replacing the rest of the samples in the secure sample set S.

3.3. Synthesis of New Samples

In this section, the K-means clustering algorithm is employed in the minority class samples to obtain P clusters, where the clustering center is denoted as $C = \{c_i\}$, where $i = 1, 2, 3, \cdots, P$, and i denotes the index of the cluster. Firstly, a sample x_l is randomly selected from the boundary sample set B, where the subscript l indicates that the sample belongs to the lth cluster. Secondly, the other sample y_l is randomly selected from the lth cluster (excluding the cluster center c_l), which satisfies $dist(x_l, y_l) < dist(x_l, c_l)$, where $dist(A, B)$ denotes the Euclidean distance between the sample A and B. Next, the temporary sample t_l is generated by using Equation (13). Finally, a new minority class sample s_l is generated by using Equation (14). In this way, a balanced dataset can be obtained by adding the synthetic new sample s_l to the original dataset and repeating the above steps.

$$t_l = x_l + rand(0, 1) \times (x_l - y_l) \tag{13}$$

$$s_l = t_l + rand(0, 1) \times (c_l - t_l) \tag{14}$$

4. Experimental Results and Analysis

4.1. Benchmark Dataset

To validate the effectiveness of the FSDR-SMOTE algorithm, 20 imbalanced datasets were selected from the UCI machine learning dataset. For the multi-class datasets in these 20 datasets, a specific class is regarded as the minority class and the rest of the classes are merged into the majority classes. The details of these 20 datasets are given in Table 1, where the ratio of the number of the majority class samples to that of the minority class samples is known as the imbalance ratio (IR). The number of samples in 20 imbalanced datasets ranges from 132 to 13611, and the IR ranges from 1.25 to 22.69, which can cover a comprehensive evaluation in terms of algorithm performance.

Table 1. Dataset information.

Dataset	Size	Attributes	IR	Minority		Majority	
				Class	Number	Class	Number
Australian	690	15	1.25	1	307	2	383
Rice	3810	7	1.34	Cammeo	1630	Osmancik	2180
Hayes-Roth	132	5	1.59	0	51	1	81
Wisconsin	683	10	1.86	4	239	2	444
Qsar	1055	41	1.97	RB	355	NRB	700
Wholesale	440	8	2.1	2	142	1	298
German	1000	20	2.33	2	300	1	700
Yeast1	1484	10	2.46	NUC	429	remainder	1055
Haberman	306	3	2.78	2	81	1	225
Blood	748	5	3.2	1	178	0	570
Ecoli1	336	8	3.31	im	77	remainder	259
Glass6	214	9	6.38	6	29	remainder	185
Dry_bean4	13,611	15	7.35	CALI	1630	remainder	11,981
Abalone11	4177	8	7.58	11	487	remainder	3690
Yeast4	1484	10	8.1	ME3	163	remainder	1321
Ecoli4	336	8	8.6	imU	35	remainder	301
Yeast4vs05679	528	10	9.35	ME2	51	MIT, ME3, EXC, VAC, ERL	477
Climate	540	21	10.74	0	46	1	494
Pageblocks2	5473	10	15.64	2	329	remainder	5144
Pageblocks1	5473	10	22.69	3, 4, 5	231	1, 2	5242

4.2. Evaluation Indicator

When dealing with the two-class imbalanced data classification problem, four evaluation metrics, such as F-measure, G-mean, AUC [34], and MCC [35], are usually calculated based on the confusion matrix for evaluating the classification performance of the oversampling method. The confusion matrix [36] is shown in Table 2, where TP denotes that positive class samples are predicted as positive class samples, FN denotes that positive class samples are predicted as negative class samples, FP denotes that negative class samples are predicted as positive class samples, and TN denotes that negative class samples are predicted as negative class samples.

Table 2. Confusion matrix.

	Predicted Positive	Predicted Negative
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

The formula for calculating the relevant assessment indicators is set out below:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$F - measure = \frac{(1 + \beta^2) \times Recall \times Precision}{\beta^2 \times Recall + Precision} \tag{18}$$

$$G - mean = \sqrt{TPR \times TNR} \tag{19}$$

$$MCC = \frac{TP \times TN - FP \times FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)} \tag{20}$$

Among them, *F-measure* can comprehensively denote the performance of the classifier on recognizing the minority class samples; *G-mean* can reflect the comprehensive performance of the classifier on positive and negative classes; *AUC* is not easily affected by the imbalance ratio and evaluates the overall performance of the classifier objectively. As another effective evaluation index, *MCC* can provide a more comprehensive performance evaluation since *MCC* takes these four classification results into account.

In our work, the Wilcoxon signed-rank test (the significance level $\alpha = 0.05$) is employed to assess whether there is a significant difference in performance between the FSDR-SMOTE method and competitors.

4.3. Experimental Setup and Environment

In this subsection, the proposed FSDR-SMOTE is firstly compared with three classical oversampling methods, including SMOTE [9], ADASYN [25], and Borderline-SMOTE [26]. Furthermore, it is also compared with four other representative oversampling methods, which are SMOTE-Tomek [27], SVMSMOTE [28], RN-SMOTE [30], and Random-SMOTE [23]. At the same time, the Random Forest classifier based on a grid search algorithm is employed to optimize its parameters. The 5-fold cross-validation strategy is adopted to ensure the reliability and statistical validity of the experimental results in our work.

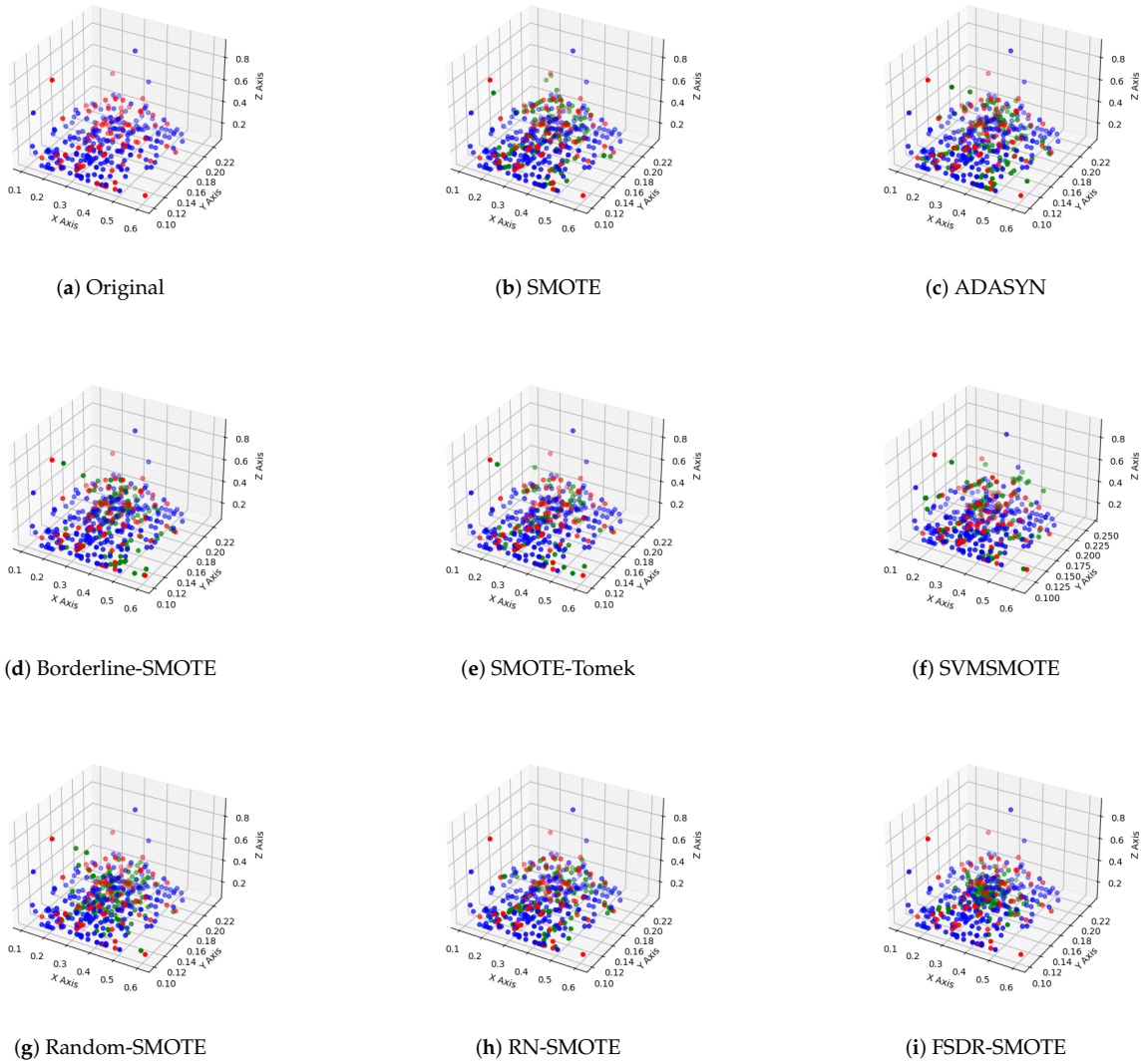
All experiments are conducted on a computer with Windows 10, AMD Ryzen 5-3600 processor, NVIDIA GTX 3060 graphics card, 16 GB of RAM, and the programming language is Python v3.9, which is implemented using the PyCharm platform.

4.4. Comparison of Sampling Effects of Different Methods

Figure 5 illustrates the sampling effects of the proposed FSDR-SMOTE algorithm and seven competitors on the Haberman dataset. In Figure 5, blue points represent original majority class samples; red points denote original minority class samples; green points symbolize newly synthesized samples.

It can be seen from Figure 5a that the original dataset contains some noise samples that are located at a significant spatial distance from the minority class samples. Figure 5b–g represents the sampling effects obtained by SMOTE, ADASYN, Borderline-SMOTE, SMOTE-Tomek, SVMSMOTE, and Random-SMOTE, respectively. The ADASYN and Borderline-SMOTE algorithms, as shown in Figure 5, have synthesized more samples in the border area. In contrast, the SMOTE, SMOTE-Tomek, SVMSMOTE, and Random-SMOTE algorithms have generated more samples in the non-border area.

Figure 5h,i displays the sampling renderings of RN-SMOTE and the FSDR-SMOTE proposed in this paper, respectively. Unlike the first six comparison algorithms, both RN-SMOTE and FSDR-SMOTE have taken noise into account in data preprocessing, which can effectively filter the noise in the minority class. Furthermore, the FSDR-SMOTE algorithm can effectively synthesize samples in the minority class boundary area, thereby mitigating the imbalance problem within the sample class.



(**a**) Original (**b**) SMOTE (**c**) ADASYN

(**d**) Borderline-SMOTE (**e**) SMOTE-Tomek (**f**) SVMSMOTE

(**g**) Random-SMOTE (**h**) RN-SMOTE (**i**) FSDR-SMOTE

Figure 5. Sampling effects of different algorithms on the dataset Haberman. Red dots denote minority samples, blue dots denote majority samples, and green dots are new synthetic samples.

4.5. Experimental Results

Table 3 shows the comparison of F-measure, G-mean, and MCC indicators of eight methods combined with the Random Forest [37] classifier on 20 imbalanced datasets. The optimal values obtained by different methods on each evaluation metric are shown in bold. As can be seen in Table 3, FSDR-SMOTE outperforms the comparison algorithms in most cases. Specifically, under the F-measure evaluation value, FSDR-SMOTE achieved the best results on 14 out of 20 datasets. Under the G-mean evaluation index, FSDR-SMOTE has the best results on 11 out of 20 datasets. FSDR-SMOTE also achieved 12 best results under the MCC evaluation index. Compared to the optimal results of the other seven comparative algorithms, the F-measure (G-mean and MCC) evaluation metric obtained by the FSDR-SMOTE method has improved by 6.95% (6.23% and 12.35%) on the Hayes-Roth dataset. Meanwhile, on the Wholesale, Haberman, and Blood datasets, the performance of FSDR-SMOTE by improved more than 1% compared with the other seven methods in terms of the F-measure, G-mean, and MCC evaluation.

Table 3. Experimental results obtained by the FSDR-SMOTE and seven comparison methods.

Dataset	EI	SMOTE	ADASYN	BS	STK	SVMS	RNS	RS	FSDS
Australian	F1	0.8730	0.8640	0.8621	0.9118	0.8775	0.8899	0.8805	**0.9162**
	G	0.8755	0.8606	0.8664	0.9130	0.8807	0.8927	0.8857	**0.9180**
	MCC	0.7512	0.7226	0.7352	0.8267	0.7631	0.7868	0.7729	**0.8367**
Rice	F1	0.9293	0.9092	0.9190	**0.9519**	0.9207	0.9301	0.9267	0.9463
	G	0.9297	0.9132	0.9213	**0.9518**	0.9220	0.9304	0.9305	0.9465
	MCC	0.8595	0.8288	0.8458	**0.9036**	0.8451	0.8608	0.8612	0.8932
Hayes-Roth	F1	0.8470	0.8470	0.8406	0.8294	0.8433	0.8348	0.8189	**0.9165**
	G	0.8500	0.8278	0.8346	0.8365	0.8455	0.8398	0.8330	**0.9123**
	MCC	0.7116	0.6741	0.6783	0.6823	0.6985	0.6903	0.6773	**0.8351**
Wisconsin	F1	0.9774	0.9779	0.9802	0.9795	0.9776	0.9815	0.9765	**0.9818**
	G	0.9775	0.9784	0.9802	0.9798	0.9778	0.9813	0.9785	**0.9818**
	MCC	0.9553	0.9584	0.9613	0.9603	0.9565	0.9632	0.9578	**0.9636**
Qsar	F1	0.9010	0.9051	0.9068	0.9149	0.9031	0.9174	0.8958	**0.9182**
	G	0.9014	0.9090	0.9084	0.9147	0.9055	**0.9177**	0.9044	0.9164
	MCC	0.8028	0.8198	0.8182	0.8299	0.8124	**0.8351**	0.8088	0.8340
Wholesale	F1	0.9351	0.9221	0.9339	0.9436	0.9268	0.9372	0.9333	**0.9566**
	G	0.9373	0.9239	0.9368	0.9452	0.9278	0.9383	0.9396	**0.9566**
	MCC	0.8760	0.8500	0.8762	0.8914	0.8579	0.8784	0.8803	**0.9147**
German	F1	0.8400	0.8380	0.8392	0.8482	0.8307	0.8445	0.8326	**0.8591**
	G	0.8381	0.8373	0.8397	**0.8481**	0.8303	0.8428	0.8440	0.8424
	MCC	0.6770	0.6753	0.6800	0.6965	0.6616	0.6869	0.6861	**0.7049**
Yeast1	F1	0.8414	0.8280	0.8334	0.8509	0.8360	0.8561	0.8224	**0.8561**
	G	0.8444	0.8365	0.8378	0.8538	0.8376	**0.8582**	0.8381	0.8526
	MCC	0.6905	0.6776	0.6784	0.7097	0.6760	**0.7169**	0.6799	0.7014
Haberman	F1	0.7772	0.7542	0.7666	0.7936	0.7675	0.7629	0.7676	**0.8092**
	G	0.7826	0.7634	0.7718	0.8031	0.7680	0.7681	0.7915	**0.8073**
	MCC	0.5718	0.5322	0.5494	0.6131	0.5393	0.5450	0.5941	**0.6162**
Blood	F1	0.7542	0.7575	0.7770	0.7950	0.7884	0.7841	0.8001	**0.8290**
	G	0.7565	0.7606	0.7747	0.7864	0.7826	0.7821	0.8196	**0.8245**
	MCC	0.5162	0.5255	0.5514	0.5771	0.5695	0.5659	0.6356	**0.6524**
Ecoli1	F1	0.9397	0.9232	0.9363	0.9409	0.9229	0.9315	0.9192	**0.9453**
	G	0.9414	0.9263	0.9383	0.9431	0.9269	0.9332	0.9266	**0.9472**
	MCC	0.8880	0.8580	0.8826	0.8898	0.8586	0.8736	0.8622	**0.8956**
Glass6	F1	0.9972	0.9970	0.9973	0.9963	0.9968	0.9978	0.9951	**0.9985**
	G	0.9972	0.9970	0.9973	0.9964	0.9966	0.9978	0.9951	**0.9985**
	MCC	0.9947	0.9942	0.9946	0.9927	0.9915	0.9951	0.9910	**0.9971**
Dry_bean4	F1	0.9897	0.9879	0.9908	0.9906	0.9898	0.9904	0.9883	**0.9941**
	G	0.9898	0.9880	0.9909	0.9906	0.9899	0.9904	0.9895	**0.9942**
	MCC	0.9796	0.9762	0.9819	0.9812	0.9800	0.9809	0.9796	**0.9883**
Abalone11	F1	0.9202	0.9220	**0.9263**	0.9200	0.9250	0.9196	0.8444	0.8916
	G	0.9219	0.9244	**0.9276**	0.9219	0.9252	0.9215	0.8592	0.8896
	MCC	0.8458	0.8521	**0.8568**	0.8458	0.8504	0.8454	0.7358	0.7802
Yeast4	F1	0.9713	0.9704	0.9744	**0.9753**	0.9739	0.9730	0.9698	0.9750
	G	0.9714	0.9708	0.9746	**0.9755**	0.9740	0.9732	0.9723	0.9750
	MCC	0.9430	0.9428	0.9502	**0.9515**	0.9482	0.9471	0.9469	0.9502
Ecoli4	F1	0.9510	0.9558	0.9545	0.9544	0.9558	0.9559	0.9486	**0.9623**
	G	0.9522	0.9568	0.9554	0.9555	0.9572	0.9556	0.9498	**0.9628**
	MCC	0.9064	0.9165	0.9141	0.9128	0.9191	0.9142	0.9142	**0.9265**
Yeast4vs05679	F1	0.9589	0.9530	0.9587	0.9618	**0.9649**	0.9561	0.9149	0.9499
	G	0.9590	0.9538	0.9596	0.9624	**0.9654**	0.9578	0.9216	0.9495
	MCC	0.9188	0.9093	0.9210	0.9256	**0.9318**	0.9156	0.8541	0.9009
Climate	F1	0.9806	0.9763	0.9724	0.9782	**0.9810**	0.9804	0.9670	0.9753
	G	**0.9809**	0.9765	0.9725	0.9782	0.9729	0.9799	0.9699	0.9756
	MCC	**0.9625**	0.9535	0.9449	0.9570	0.9486	0.9609	0.9411	0.9508
Pageblocks2	F1	0.9925	0.9923	0.9939	0.9923	0.9940	0.9919	0.9919	**0.9948**
	G	0.9925	0.9922	0.9938	0.9923	0.9940	0.9919	0.9930	**0.9947**
	MCC	0.9849	0.9845	0.9877	0.9847	0.9881	0.9837	0.9858	**0.9895**
Pageblocks1	F1	0.9881	0.9873	0.9885	**0.9887**	0.9882	0.9882	0.9792	0.9870
	G	0.9882	0.9874	0.9885	**0.9888**	0.9882	0.9882	0.9808	0.9870
	MCC	0.9766	0.9751	0.9772	**0.9778**	0.9766	0.9767	0.9641	0.9740

EI: Evaluation Indicators; F1: F-measure; G: G-mean; BS: Borderline-SMOTE; STK: SMOTE-Tomek; SVMS: SVMSMOTE; RNS: RN-SMOTE; RS: Random-SMOTE; FSDS: FSDR-SMOTE. The best results are in bold.

In addition, in datasets that did not achieve optimal evaluation results, such as Rice, Qsar, Yeast1, etc., the evaluation results obtained by FSDR-SMOTE differed from the optimal results by less than 0.5%. It can be seen from the comprehensive results obtained by the FSDR-SMOTE algorithm that it has certain advantages compared with the seven comparison algorithms.

To further analyze the significant difference between the FSDR-SMOTE and these competitors, the Wilcoxon signed rank test ($\alpha = 0.05$) is performed on the results of F-measure and MCC, with the null hypothesis that all algorithms have the same effect. Obtained test results are shown in Table 4.

Table 4. The results obtained by FSDR-SMOTE and competitors on Wilcoxon cosigned-rank test.

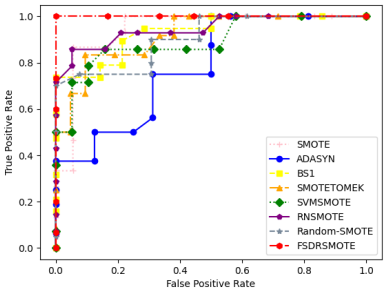
Methods	SMOTE	ADASYN	BS	STK	SVMS	RNS	RS
EI	F1						
R+	177	188	181	156	175	169	210
R−	33	22	29	54	35	41	0
Assuming	reject	reject	reject	accept	reject	reject	reject
Select	FSDS	FSDS	FSDS	Both	FSDS	FSDS	FSDS
EI	MCC						
R+	172	184	176	136	179	158	210
R−	38	26	34	74	31	52	0
Assuming	reject	reject	reject	accept	reject	reject	reject
Select	FSDS	FSDS	FSDS	Both	FSDS	FSDS	FSDS

EI: Evaluation Indicators; F1: F-measure; BS: Borderline-SMOTE; STK: SMOTE-Tomek; SVMS: SVMSMOTE; RNS: RN-SMOTE; RS: Random-SMOTE; FSDS: FSDR-SMOTE.

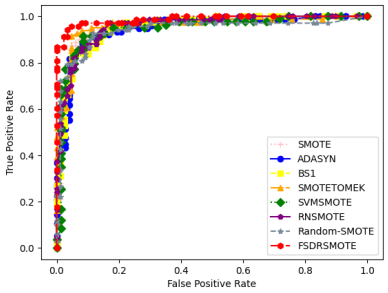
It can be seen from Table 4 that the null hypothesis is accepted by FSDR-SMOTE and SMOTE-Tomek, whereas it is rejected by the remaining six algorithms. This means that FSDR-SMOTE significantly outperforms these six algorithms in a statistically significant way. Although FSDR-SMOTE does not obtain significant superiority compared with SMOTE-Tomek, it achieves a more positive rank R+ on both key metrics, i.e., F-measure and MCC, which suggests that FSDR-SMOTE outperforms SMOTE-Tomek more often on these 20 datasets in the comparison experiments.

To visualize the advantages of the FSDR-SMOTE method, the ROC curves of the eight algorithms on six different datasets were plotted. In Figure 6, on the Hayes-Roth, Australian, Haberman, and Yeast1 datasets, the ROC curve clearly shows that the FSDR-SMOTE method consistently outperforms other comparison algorithms. This indicates that the FSDR-SMOTE algorithm has achieved higher AUC values on these datasets, demonstrating its superior classification effects compared to the other seven algorithms. In conclusion, the combination of the FSDR-SMOTE algorithm and the Random Forest classifier exhibits better performance on most datasets, thereby proving the algorithm's stability.

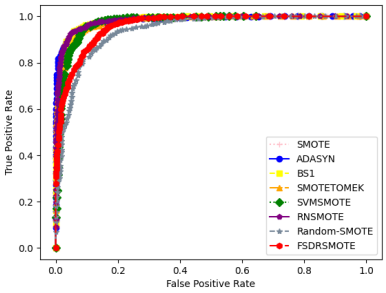
Furthermore, the proposed FSDR-SMOTE is combined with different classifiers to validate the model performance. These classifiers include Random Forest [37], Adaboost [38], KNN [39], Gaussian Naive Bayes (BC) [40], and Logistic Regression (LR) [41]. Figure 7 shows the experimental curves obtained by FSDR-SMOTE with five different classifiers on 20 datasets in terms of four metrics, i.e., F-measure, G-mean, AUC, and MCC. In Figure 7, the abscissa represents 20 datasets, which are numbered according to the order in Table 1. It can be seen that FSDR-SMOTE combined with Random Forest achieves optimal or near-optimal classification accuracies on most of these 20 datasets, which can also show the compatibility of FSDR-SMOTE with different classifiers.



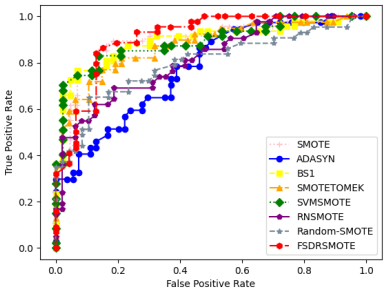
(**a**) ROC curve for the dataset Hayes-Roth



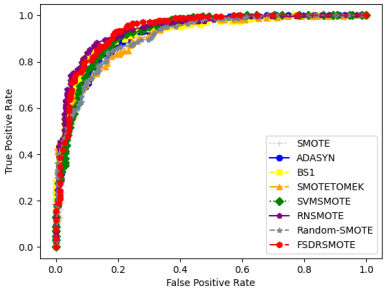
(**b**) ROC curve for the dataset Australian



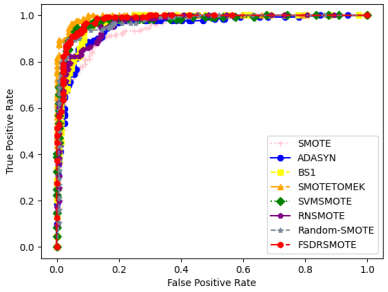
(**c**) ROC curve for the dataset Abalone11



(**d**) ROC curve for the dataset Haberman

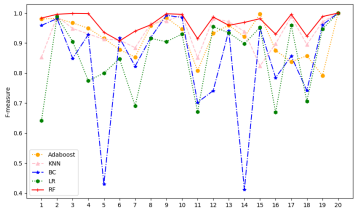


(**e**) ROC curve for the dataset Yeast1

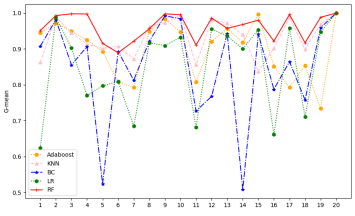


(**f**) ROC curve for the dataset Qsar

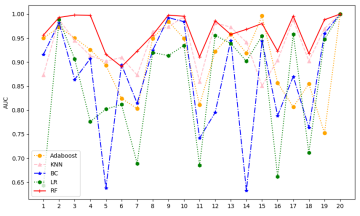
Figure 6. ROC curves of eight algorithms on six datasets.



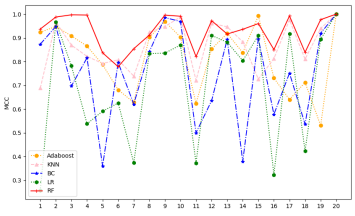
(**a**) F-measure



(**b**) G-mean



(**c**) AUC



(**d**) MCC

Figure 7. The different evaluation index results obtained by FSDR-SMOTE with different classifiers.

4.6. Ablation Study

To further investigate the effectiveness of data preprocessing techniques in the FSDR-SMOTE method, we conducted ablation studies on eight imbalanced datasets, namely, Haberman, Blood, Wisconsin, German, Australian, Qsar, Yeast1, and Abalone11.

Figure 8 intuitively illustrates the changes in the number of minority class samples across these eight different datasets before and after data preprocessing. As can be inferred from the figure, the application of the improved Tukey rule for data preprocessing effectively reduces the number of minority class samples, primarily by removing noise samples. Notably, after preprocessing, the proportion of minority class samples within the original minority class samples remains between 84.85% and 89.54%. This indicates that the FSDR-SMOTE method successfully eliminates samples above the threshold while effectively retaining most of the genuine and significant data, thereby minimizing the impact on minority class diversity.

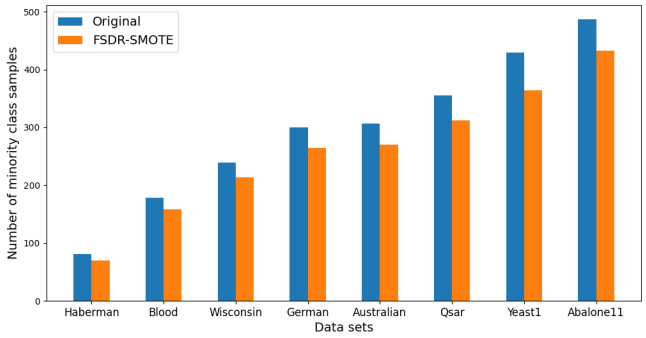
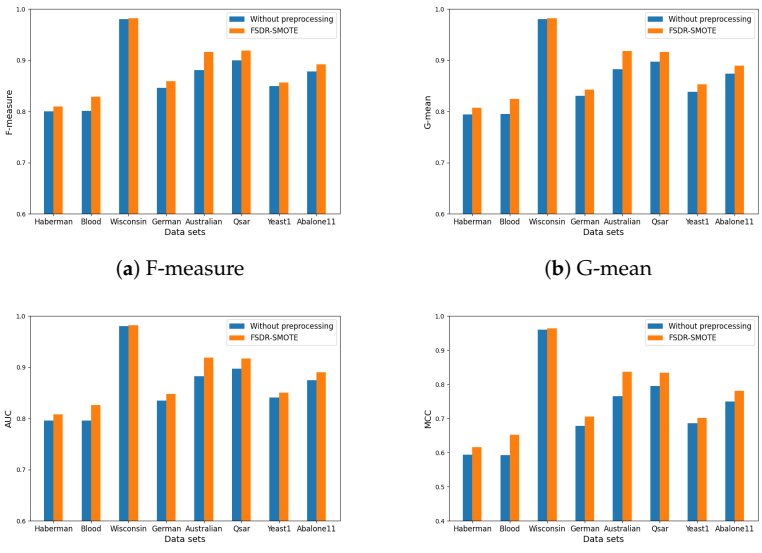


Figure 8. Sample size of different datasets before and after preprocessing.

Figure 9 presents the evaluation results using F-measure, G-mean, AUC, and MCC. It is evident from the figure that all evaluation metrics of the FSDR-SMOTE method outperform those of the competitors on the eight imbalanced datasets without preprocessing. These results demonstrate that for most datasets, the improvement in evaluation metrics exceeds 1%, particularly on the Blood and Australian datasets, where the improvement in MCC reaches as high as 6.05% and 7.19%, respectively. These substantial improvements underscore the effectiveness of employing the improved Tukey rule for data preprocessing in the FSDR-SMOTE method.



(**a**) F-measure (**b**) G-mean

(**c**) AUC (**d**) MCC

Figure 9. Comparison of evaluation indicators for dataset ablation study.

5. Discussion

The classic oversampling method, SMOTE, can effectively enhance the classification accuracy of minority class samples in imbalanced datasets. However, SMOTE is not sensitive to noise and is prone to intra-class imbalance of samples, potentially introducing noise and generating redundant samples. Effective data preprocessing is a crucial step in addressing the noise problem, but there is a risk of inadvertently deleting real data in this process. Thus, our research focuses on effectively eliminating noise data while preserving most of the real data. In the FSDR-SMOTE method, we utilize the improved Tukey rule [32] for data preprocessing. The experimental results showed that the preprocessed dataset retained at least 84.85% of minority class samples. After using an improved Tukey rule for data preprocessing on eight datasets of ablation studies, the F-measure (G-mean, AUC, and MCC) evaluation values increased by 1.59% (1.78%, 1.70%, and 3.38%), respectively.

Additionally, our proposed FSDR-SMOTE method takes into account the imbalance within sample classes. During the sample synthesis process, we combined K-means [33] clustering with an improved three-point interpolation strategy in Random-SMOTE [23]. We conducted comparative experiments to evaluate the FSDR-SMOTE method against seven other existing sampling methods (SMOTE [9], ADASYN [25], Borderline-SMOTE [26], SMOTE-Tomek [27], SVMSMOTE [28], RN-SMOTE [30], and Random-SMOTE [23]). FSDR-SMOTE achieved an average improvement of 0.72% (0.48% and 0.98%) over the second-best results across the 20 datasets in terms of F-measure (G-mean and MCC), respectively. Thereby, fully considering the distribution characteristics of samples and adopting a more efficient sample synthesis strategy are keys to enhancing model classification performance.

6. Conclusions

In this paper, a new oversampling method (FSDR-SMOTE) is proposed to deal with the imbalanced data classification problem. In FSDR-SMOTE, the noisy samples are firstly removed based on the improved Tukey criterion in the data preprocessing stage. Secondly, a new feature standard deviation method is proposed to divide these samples into boundary samples and safety ones in view of the impact of different distributions of synthetic samples on the model performance. Finally, the strategy of synthesizing new samples in Random-SMOTE is improved. The experimental results show that the average evaluation values of F-measure, G-mean, and MCC obtain by FSDR-SMOTE on 20 imbalanced datasets reach 93.31%, 93.16%, and 86.53%, respectively. The average values of all evaluations are superior to the other seven comparison methods. Furthermore, FSDR-SMOTE was combined with Random Forest and the other four classifiers to validate the compatibility of FSDR-SMOTE.

However, FSDR-SMOTE performs poorly in dealing with datasets with a higher imbalance ratio (IR > 8). Therefore, our future work will devote to constructing new hybrid strategies for optimizing the FSDR-SMOTE method proposed in this paper.

Author Contributions: Conceptualization, Y.Z. and L.D.; methodology, Y.Z. and B.W.; software, Y.Z.; validation, Y.Z.; formal analysis, Y.Z. and L.D.; investigation, Y.Z.; resources, Y.Z.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, L.D. and B.W.; visualization, Y.Z.; supervision, L.D. and B.W.; project administration, L.D. and B.W.; funding acquisition, L.D. and B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by grants from the National Natural Science Foundation of China (61806204), the Basic Public Welfare Research Project of Zhejiang Province (LGF22F020020), and the Research Fund Project of Zhejiang Sci-Tech University Longgang Research Institute (LGYJY2023003).

Data Availability Statement: The data used in this article come from the UCI Machine Learning Repository, http://archive.ics.uci.edu/ (accessed on 7 March 2024).

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

References

1. Tomašev, N.; Mladenić, D. Class imbalance and the curse of minority hubs. *Knowl. Based Syst.* **2013**, *53*, 157–172. [CrossRef]
2. Vasighizaker, A.; Jalili, S. C-PUGP: A cluster-based positive unlabeled learning method for disease gene prediction and prioritization. *Comput. Biol. Chem.* **2018**, *76*, 23–31. [CrossRef] [PubMed]
3. Jurgovsky, J.; Granitzer, M.; Ziegler, K.; Calabretto, S.; Portier, P.E.; He-Guelton, L.; Caelen, O. Sequence classification for credit-card fraud detection. *Expert Syst. Appl.* **2018**, *100*, 234–245. [CrossRef]
4. Malhotra, R.; Kamal, S. An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. *Neurocomputing* **2019**, *343*, 120–140. [CrossRef]
5. Zhou, X.; Hu, Y.; Liang, W.; Ma, J.; Jin, Q. Variational LSTM enhanced anomaly detection for industrial big data. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3469–3477. [CrossRef]
6. Tao, X.; Li, Q.; Guo, W.; Ren, C.; Li, C.; Liu, R.; Zou, J. Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification. *Inf. Sci.* **2019**, *487*, 31–56. [CrossRef]
7. Daneshfar, F.; Aghajani, M.J. Enhanced text classification through an improved discrete laying chicken algorithm. *Expert Syst.* **2024**, e13553. [CrossRef]
8. Revathy, V.; Pillai, A.S.; Daneshfar, F. LyEmoBERT: Classification of lyrics' emotion and recommendation using a pre-trained model. *Procedia Comput. Sci.* **2023**, *218*, 1196–1208. [CrossRef]
9. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
10. Nadi, A.A.; Gildeh, B.S.; Kazempoor, J.; Tran, K.D.; Tran, K.P. Cost-effective optimization strategies and sampling plan for Weibull quantiles under type-II censoring. *Appl. Math. Model.* **2023**, *116*, 16–31. [CrossRef]
11. Tao, X.; Chen, W.; Li, X.; Zhang, X.; Li, Y.; Guo, J. The ensemble of density-sensitive SVDD classifier based on maximum soft margin for imbalanced datasets. *Knowl. Based Syst.* **2021**, *219*, 106897. [CrossRef]
12. Li, W.; Sun, S.; Zhang, S.; Zhang, H.; Shi, Y. Cost-Sensitive Approach to Improve the HTTP Traffic Detection Performance on Imbalanced Data. *Secur. Commun. Netw.* **2021**, *2021*, 6674325. [CrossRef]
13. Li, J.; Zhu, Q. A boosting self-training framework based on instance generation with natural neighbors for K nearest neighbor. *Appl. Intell.* **2020**, *50*, 3535–3553. [CrossRef]
14. Xia, S.; Wang, G.; Chen, Z.; Duan, Y. Complete random forest based class noise filtering learning for improving the generalizability of classifiers. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 2063–2078. [CrossRef]
15. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [CrossRef]
16. Wei, G.; Mu, W.; Song, Y.; Dou, J. An improved and random synthetic minority oversampling technique for imbalanced data. *Knowl. Based Syst.* **2022**, *248*, 108839. [CrossRef]
17. Meng, D.; Li, Y. An imbalanced learning method by combining SMOTE with Center Offset Factor. *Appl. Soft Comput.* **2022**, *120*, 108618. [CrossRef]
18. Shrifan, N.H.; Akbar, M.F.; Isa, N.A.M. An adaptive outlier removal aided k-means clustering algorithm. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6365–6376. [CrossRef]
19. Liang, X.; Jiang, A.; Li, T.; Xue, Y.; Wang, G. LR-SMOTE—An improved unbalanced data set oversampling based on K-means and SVM. *Knowl. Based Syst.* **2020**, *196*, 105845. [CrossRef]
20. Zhang, A.; Yu, H.; Zhou, S.; Huan, Z.; Yang, X. Instance weighted SMOTE by indirectly exploring the data distribution. *Knowl. Based Syst.* **2022**, *249*, 108919. [CrossRef]
21. Cheng, D.; Zhu, Q.; Huang, J.; Yang, L.; Wu, Q. Natural neighbor-based clustering algorithm with local representatives. *Knowl. Based Syst.* **2017**, *123*, 238–253. [CrossRef]
22. Dong, Y.; Wang, X. A new over-sampling approach: Random-SMOTE for learning from imbalanced data sets. In *Proceedings of the Knowledge Science, Engineering and Management: 5th International Conference, KSEM 2011, Irvine, CA, USA, 12–14 December 2011*; Proceedings 5; Springer: Berlin/Heidelberg, Germany, 2011; pp. 343–352. [CrossRef]
23. Bader-El-Den, M.; Teitei, E.; Perry, T. Biased random forest for dealing with the class imbalance problem. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 2163–2172. [CrossRef] [PubMed]
24. Rekha, G.; Tyagi, A.K.; Sreenath, N.; Mishra, S. Class imbalanced data: Open issues and future research directions. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021; pp. 1–6. [CrossRef]
25. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328. [CrossRef]
26. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proceedings of the International Conference on Intelligent Computing, Hefei, China, during 23–26 August 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 878–887. [CrossRef]
27. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
28. Nguyen, H.M.; Cooper, E.W.; Kamei, K. Borderline over-sampling for imbalanced data classification. *Int. J. Knowl. Eng. Soft Data Paradig.* **2011**, *3*, 4–21. [CrossRef]

29. Huang, S.; Cai, N.; Pacheco, P.P.; Narrandes, S.; Wang, Y.; Xu, W. Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genom. Proteom.* **2018**, *15*, 41–51.
30. Arafa, A.; El-Fishawy, N.; Badawy, M.; Radad, M. RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced data classification. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 5059–5074. [CrossRef]
31. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Trans. Database Syst. (TODS)* **2017**, *42*, 1–21. [CrossRef]
32. Huyghues-Beaufond, N.; Tindemans, S.; Falugi, P.; Sun, M.; Strbac, G. Robust and automatic data cleansing method for short-term load forecasting of distribution feeders. *Appl. Energy* **2020**, *261*, 114405. [CrossRef]
33. Ahmed, M.; Seraj, R.; Islam, S.M.S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics* **2020**, *9*, 1295. [CrossRef]
34. Su, C.T.; Chen, L.S.; Yih, Y. Knowledge acquisition through information granulation for imbalanced data. *Expert Syst. Appl.* **2006**, *31*, 531–541. [CrossRef]
35. Zhu, Q. On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset. *Pattern Recognit. Lett.* **2020**, *136*, 71–80. [CrossRef]
36. Visa, S.; Ramsay, B.; Ralescu, A.L.; Van Der Knaap, E. Confusion matrix-based feature selection. *Maics* **2011**, *710*, 120–127. https://api.semanticscholar.org/CorpusID:3026044.
37. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
38. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
39. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
40. Venkata, P.; Pandya, V. Data mining model and Gaussian Naive Bayes based fault diagnostic analysis of modern power system networks. *Mater. Today Proc.* **2022**, *62*, 7156–7161. [CrossRef]
41. Hosmer, D.W., Jr.; Lemeshow, S.; Sturdivant, R.X. *Applied Logistic Regression*; John Wiley & Sons: Hoboken, NJ, USA, 2013.