Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 21(3): 4309–4327.
DOI: 10.3934/mbe.2024190
Received: 25 October 2023
Revised: 17 December 2023
Accepted: 22 December 2023
Published: 26 February 2024

Research article

A new imbalanced data oversampling method based on Bootstrap method and Wasserstein Generative Adversarial Network

Binjie Hou and Gang Chen*

Department of Mathematics, Dalian Maritime University, Dalian 116026, China

* **Correspondence:** Email: cgmaritime@163.com.

Abstract: Due to their high bias in favor of the majority class, traditional machine learning classifiers face a great challenge when there is a class imbalance in biological data. More recently, generative adversarial networks (GANs) have been applied to imbalanced data classification. For GANs, the distribution of the minority class data fed into discriminator is unknown. The input to the generator is random noise (z) drawn from a standard normal distribution $N(0, 1)$. This method inevitably increases the training difficulty of the network and reduces the quality of the data generated. In order to solve this problem, we proposed a new oversampling algorithm by combining the Bootstrap method and the Wasserstein GAN Network (BM-WGAN). In our approach, the input to the generator network is the data (z) drawn from the distribution of minority class estimated by the BM. The generator was used to synthesize minority class data when the network training is completed. Through the above steps, the generator model can learn the useful features from the minority class and generate realistic-looking minority class samples. The experimental results indicate that BM-WGAN improves the classification performance greatly compared to other oversampling algorithms. The BM-WGAN implementation is available at: https://github.com/ithbjgit1/BMWGAN.git.

Keywords: imbalanced data; generative adversarial networks (GANs); Bootstrap method (BM); data generation; probability distribution

1. Introduction

In recent years, the research area of imbalanced classification problems has attracted considerable interest. Different from previous traditional classification problems, imbalanced task refers to a situation where the distribution of classes is not equal. This means that one class, typically the majority class, has more instances than another class, which is known as the minority class. In addition, this phenomenon is a very common problem in many aspects of scientific research, such as medical diagnosis [1], abnormal activity recognition [2], fraud detection [3] and so on. However, in these imbalanced tasks, minority

class predictions typically perform worse than majority class predictions, leading to a high proportion of minority class predictions that are incorrect. Therefore, a variety of methods have been proposed to address imbalanced problems. Generally speaking, these techniques can often be divided into two groups: Data level approaches and algorithmic level approaches.

Data level approaches have attracted much attention because of their better performance. Among them, oversampling and undersampling are two major methods to tackle class imbalance in data level approaches. Synthetic minority oversampling technique (SMOTE) [4] is a well-known method for dealing with this problem by going along the line connection with one or more k-nearest neighbors to generate new data. However, this method does not consider the effect of overlap between the minority class and the majority class. Therefore, many variants have been proposed to solve this problem. Borderline-SMOTE [5], for each minority instance, chooses boundary samples to balance data by calculating the number of minority and majority classes in k-nearest neighbors. Kmeans-SMOTE [6] can further improve the classification performance by combining SMOTE and the clustering algorithm. Safe-level-SMOTE (SaSMO) [7], uses information from the k-nearest neighbors to improve the quality of generated data. Adaptive Synthetic Sampling Approach for Imbalanced Learning (ADASYN) [8], an adaptive synthetic sampling method for imbalanced data, makes use of neighboring minority and majority class information. Gaussian Distribution based Oversampling for Imbalanced Data Classification (GDO) [9] is a novel method based on a Gaussian distribution. In GDO, for each minority instance, this method designed a Gaussian distribution to fit the minority class data. Local density-based adaptive sampling for imbalanced data classification (LDAS) [10], provides a local density sampling method, which alleviates the overlapping of majority class instances and assigns a local density to each minority class instance. Based on the samples' selection strategy [11], a method for oversampling that involves identifying the k-nearest minority class neighbors of the minority class is proposed. A new approach for imbalanced data classification based on data gravitation (IDGC) [12] is a novel method for classifying imbalanced data based on data gravity. To pay attention to noise, a radial-based oversampling method is proposed [13], which can identify areas where synthetic items from the minority class should be generated. With the development of artificial intelligence technology, Classification Enhancement Generative Adversarial Networks for unraveling data imbalance problems (CEGAN) [14] is proposed, which is a classification enhancement generative adversarial networks to enhance the quality of generated synthetic minority data. A differential evolution based oversampling approach for highly imbalanced datasets (DEBOHID) [15] is a differential evolution based oversampling approach, which has an effective candidate individual generation mechanism. Density-based [16] weighting is a good way to deal with imbalanced data classification. On the other hand, undersampling is also a popular technique for imbalanced datasets to reduce the skew in class distributions. In epilepsy and Parkinson's disease, the improved overlap based undersampling approach is proposed in this study as a way to make instances of minority class more visible in the overlapped [17] region. However, it is important to note that undersampling may result in the loss of valuable information present in majority samples, so this method is rarely used in the imbalanced research field.

Algorithmic level approaches mainly include two aspects: Cost-sensitive learning and ensemble learning. Cost-sensitive learning is an effective method in the imbalanced data classification field [18]. It assigns higher misclassification cost to the minority class samples [19]. Compared to sampling methods, cost sensitive learning is less common due to the difficulties in accurately assigning misclassification costs. Cost-sensitive learning methods are categorized into three types. The first type involves data

space weighting by modifying the training dataset distribution according to misclassification costs. The second type involves building a cost-sensitive classifier by changing the learning process of a classifier. The third type is based on Bayes risk theory and assigns each instance to the class with the lowest risk [20]. Ensemble learning dependent technique is a type of hybrid model that embeds data-level approaches into general classifiers for better classification and generalization. It can also be divided into three subclasses: Ensemble based on bagging, boosting, and hybrid. Underbagging [21] randomly undersamples the dataset in each bagging iteration, leaving all the minority class instances in each iteration. SMOTEBagging [22] uses SMOTE in each iteration. The new dataset contains twice the number of instances of majority class. SMOTEBoost [23] is another algorithm that combines boosting and data sampling to improve the performance of the model based on unbalanced data training. RUS (random undersampling) Boost [24] uses RUS, a technique for randomly deleting instances from majority classes.

However, researchers often tend to focus on oversampling methods, because too much valuable information would be lost due to undersampling [25]. Additionally, some studies have emphasized the importance of class overlap [26], and this effect has been extensively examined in previous research. To improve the classification performance of minority class, it is advantageous to use the deep learning to synthetic data.

As mentioned in above methods in data level strategies, the oversampling and undersampling approaches have some advantages and limitations. At this time, we regard oversampling approaches as our main research direction. As shown in Figure 1, blue dots represent the majority class and green dots represent the minority class and we can see that the number of minority class is greatly less than that of the majority class.

Additionally, the current oversampling algorithms synthesize new minority class data by interpolation within the available minority class data via the k-nearest neighbors algorithm. It is worth noting that there is a disadvantage: The oversampling algorithms cannot fully utilize the information of minority class samples. It inevitably leads to a reduction in classification performance. Therefore, Wasserstein Generative Adversarial Network (WGAN) is used to extract information from minority class samples and generates realistic-looking minority class samples. Nowdays, WGAN uses a standard normal distribution as random noise since the distribution of the minority class is unknown. It leads to more complicated network and increases difficulty of network training. To alleviate this problem, we use the Bootstrap method (BM) to estimate the distribution of the minority class and improve the quality of the data fed into the generator. After the network training is completed, the saved generator network is used to generate new minority class data. Through the above steps, the classification performance of minority class is further improved.

Specifically, the main contributions of this article include the following:

(1) In order to enhance the diversity of generated data, we change the type of random noise input to the generator. Through this dynamic change, the data generated by the generator will not be concentrated in part of the minority class data.

(2) Since the number of minority classes is small, this will increase the difficulty of training in the WGAN to a certain extent. Therefore, we use the BM to estimate the distribution of the minority class, which significantly increases the capacity of the WGAN to learn more sophisticatedly and ultimately improves the quality of the generated data.

(3) Class overlap has been demonstrated to have a significantly negative impact on the classification performance. To reduce the degree of class overlap, the trained generator is used to generate new

minority class data.

The remainder of this paper is organized into four sections, namely, Section 2 introduces preliminary knowledge, Section 3 introduces our new algorithm BM-WGAN, Section 4 introduces the experimental results and Section 5 describes prospects for the future.

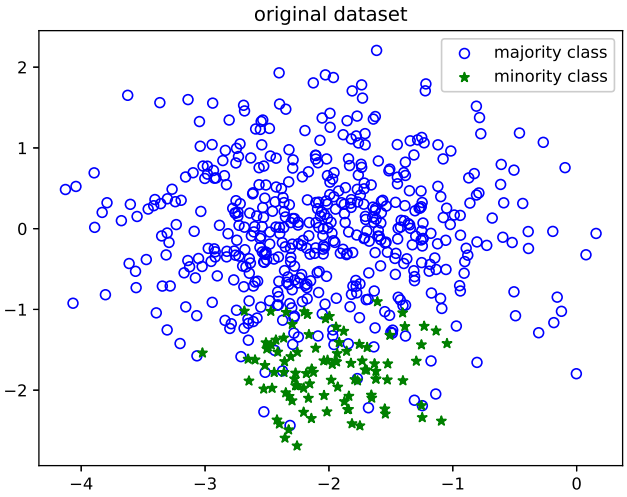


Figure 1. Imbalanced dataset.

2. Related work

2.1. WGAN

GAN [27] is an important class of deep learning models and contains two major components, Generator (G) produces synthetic data when given the noisy data and Discriminator (D) to distinguishes data that is real or fake. A robust generative adversarial networks for time series augmentation (TSA-GAN) [28] is a robust GAN, which designs a newly merged weight to solve the problem of training saturation. Generative Adversarial Nets for Extremely Imbalanced Data Augmentation (EID-GAN) [29] designs a new penalty function by subtracting the outliers from the cropped region of generated instance to guide the generator to learn the features of outliers. Judging from the above situation, neural networks are difficult to train and may suffer from the aforementioned vanishing gradient problem. As shown in Figure 2, we input the minority class data into the discriminator to train the GAN until the discriminator should be able to classify whether the given data is real or fake. In order to increase both generation and discrimination's capacity to locate a Nash equilibrium, the two models in a GAN compete against one another while training. Next, the minimax two-player game depending on G and D is evaluated with a cost function $L(G, D)$ as defined as follows:

$$\min_G \max_D \underset{x\sim p_r}{E} [logD(x)] + \underset{z\sim p_z}{E} [log(1 - D(G(z)))], \tag{1}$$

where x is real data from the real data distribution p_r, z is noise data from the input noise variables p_z and E is the expectation.

It is worth noting that one model is fixed and the other is optimized during the GAN training process. To increase the discrimination accuracy, the discriminator divides the real samples into positive and the

generated samples as negatively feasible after fixing the generator. When the discriminator network is trained to the optimal situation, the discriminator network can be expressed as:

$$D(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}, \tag{2}$$

where $p_g(x)$ is the distribution of generated data. For a fixed D, the generator is trained by minimizing $log(1 - D(G(z)))$. Finally, the optimal value is reached when $p_g(x) = p_r(x)$, but this is only an ideal state and it is difficult to achieve.

However, there are still some significant challenges and unstable behaviors [30] in GAN training. In order to tackle the above problems, the Wasserstein distance [31] is introduced to replace the Jensen-Shannon (JS) and Kullback-Leibler (KL) divergences. The objective function between the generator and the discriminator is expressed as follows:

$$\min_G \max_D \underset{x\sim p_r}{E} [D(x)] - \underset{z\sim p_g}{E} [D(z)], \tag{3}$$

where x is real data sampled from the real data distribution p_r and z is the noise data sampled from a gaussian distribution p_g. The equation demonstrates that by treating the discriminator as a classifier, the Wasserstein distance between the actual data distribution and the distribution assumed by the generator is minimized.

$$W(p_r, p_g) = \inf_{\gamma\sim\prod(p_r,p_g)} E_{(x,y)\sim\gamma}[\|x - y\|], \tag{4}$$

where $W(p_r, p_g)$ is Wasserstein distance, $\prod(p_r, p_g)$ is a collection that contains all joint distributions of p_r and p_g, (x, y) is a sample from p_r and p_g and $E_{(x,y)\sim\gamma}[\|x - y\|]$ is the sample's expectation of distance.

In summary, the WGAN addresses the vanishing gradient problem and outperforms the KL and JS divergence. Therefore, the probability density function of the minority class data can be obtained by the competitive process between the generator and discriminator. Figure 3 shows the evolution of the loss function during the training process in minority class data. By constructing the loss curve, we can see that the loss of the discriminator is close to zero, which indicates that the network has been trained successfully.

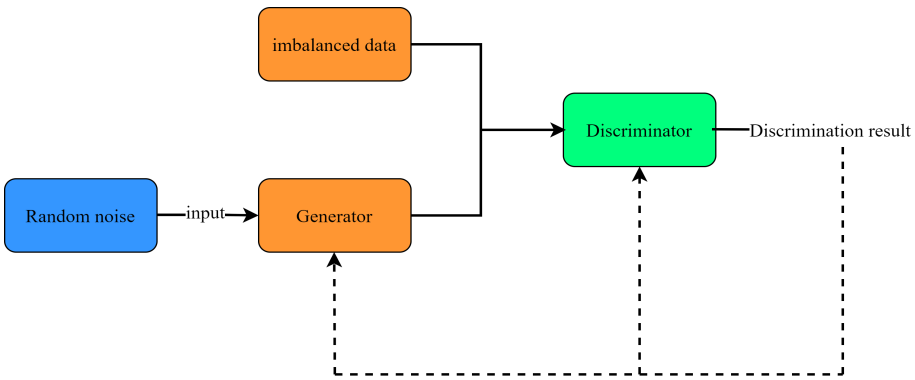


Figure 2. The structure of a basic GAN.

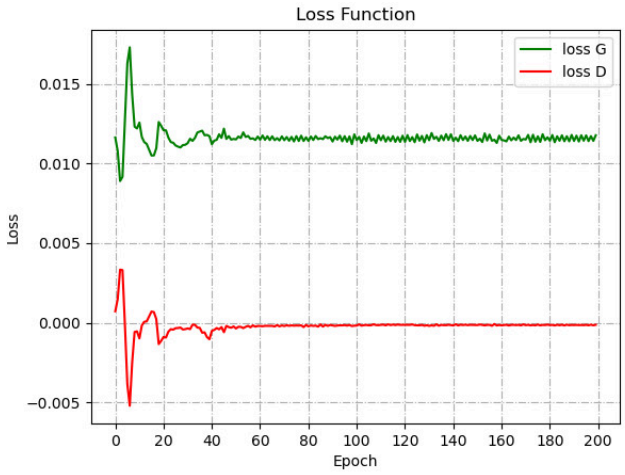


Figure 3. Discriminator loss of BM-WGAN.

3. Proposed method

Many variants of SMOTE differentiate the minority examples by counting the number of majority examples in minority neighbors in data generation. However, most of them not fully utilize the information of the minority class. It will inevitably lead to a waste of data information. Therefore, WGAN is a new and powerful machine learning approach to extract data information and generate new minority class data. Since the distribution of the data is not known, the input z to the generator is sampled from a standard normal distribution $N(0, 1)$, where mean is zero and standard deviation is one. It increases the difficulty of training and reduces the quality of generated data. To alleviate this problem, we propose a new method (BM-WGAN) by leveraging the BM to optimize random noise.

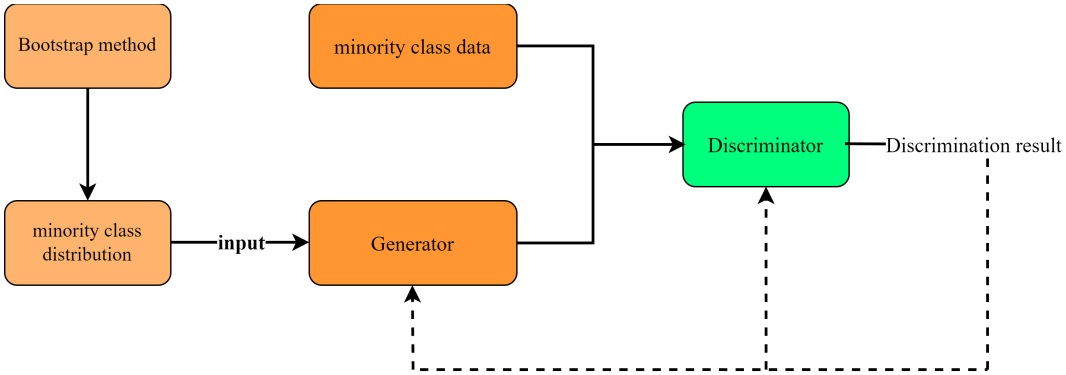


Figure 4. The structure of BM-WGAN.

The Bootstrap is a method for estimating the distribution of an estimator or test statistic by resampling one's data [32]. It is introduced by Efron [33] as an approach to calculate confidence intervals for parameters in circumstances where standard methods cannot be applied. An example of this would be one in which few data points are available, so that approximate large sample methods are inapplicable.

The BM has subsequently been used to solve many other problems that would be too complicated for traditional statistical analysis. Let $X = \{X_1, X_2, \cdots, X_m\}$ be a sample and a collection of m numbers be drawn at random from a completely unspecified distribution F. Let γ be an unknown parameter of the distribution F, such as its mean. The way to obtain the distribution of γ or its characteristics is to repeat the experiment of a sufficient number of time and approximate the distribution of γ by the so obtained empirical distribution. The BM suggests that we resample from a distribution chosen to be close to F in some sense. For example, the sample distribution $\hat{F}$ that approaches F is $n \to \infty$. The pseudocode of the BM algorithm is given in Algorithm 1.

Algorithm 1 Bootstrap method

Input: Minority class D_{min}, Mean=[], Cov=[].

Output: The parameters of minority class distribution.

1: **for** $t = 1 : 1000$ **do**
2: Generate a bootstrap sample $x = \{x_1, x_2, \cdots, x_m\}$ by sampling the $X = \{X_1, X_2, \cdots, X_m\}$ randomly.
3: Calculate mean of the bootstrap sample $x = \{x_1, x_2, \cdots, x_m\}$.
4: Calculate covariance of the bootstrap sample $x = \{x_1, x_2, \cdots, x_m\}$.
5: Mean $\leftarrow$ mean.
6: Cov $\leftarrow$ covariance.
7: Calculate mean $\frac{Mean}{1000}$ of the minority class.
8: Calculate covariance $\frac{Cov}{1000}$ of the minority class.

Let $D = [X_1, X_2, \cdots, X_N]$ be the given training set composed of D_{min} and D_{maj}, where $\mathrm{D}_{min} = [x_1, x_2, \cdots, x_m]$ represents the minority data and $\mathrm{D}_{maj} = [x_1, x_2, \cdots, x_n]$ represents the majority data. N_{min} and N_{maj} is the number of D_{min} and D_{maj}. For each minority instance $x_i \in \mathrm{D}_{min}$, let $k_i = k_i^{min} \cup k_i^{maj}$ be the collection of k-nearest neighbors of x_i, where k_i^{min} and k_i^{maj} is the collection of minority instance and majority instance in k_i.

BM-WGAN mainly primarily comprises two essential components: Bootstrap estimation of minority class distribution and synthesizes new data by WGAN. In the first component, we employ the BM to calculate the probability distribution of the minority class. For the second component, the random noise (z) drawn from the distribution of minority class is fed into the generator to train WGAN. After the training is completed, new minority class data is generated through the generator. By using these two essential steps, the quality of synthetic data can be improved. The pseudocode of the BM-WGAN algorithm is given in Algorithm 2.

GANs are a leading neural network architecture for generative modeling. At this point, the input of the generator model is a Gaussian random noise data. This may be an unreasonable way because it increases the difficulty of training the network when the real distribution is very different from the generated distribution. In BM-WGAN, we use the BM to find generated distribution which is the closest to the real distribution in a new feasible area. Interestingly, the generated distribution is more likely to cross with the real distribution. This additionally reduces the difficulty of getting convergence in network training.

In addition, the new data generated by the trained generator is of higher quality. This means the data point from the original dataset is more similar to the generated data. Once the neural network is

trained with a suitable dataset, the generator network is used to synthesize new minority class data. The generator stops synthesizing minority class data when the sample data size of the minority class is the same as the sample data size of the majority class.

Algorithm 2 BM-WGAN

Input: training set D, D_s=[].
Output: Resampled dataset D_{new}

Calculate the distribution of minority class by Bootstrap method.
for s=1:S **do**
    Train Discriminator.
    for j=1:J **do**
        Discriminator ← minority class data.
        Generator ← data from the distribution of minority class.
    Train Generator.
    Generator ← data from the distribution of minority class.
while $N_{min} < N_{maj}$ **do**
    Synthesize new data x_n by Generator .
    $D_s \leftarrow x_n$.
return result $D_{new} = D \cup D_s$

4. Experiment

4.1. Performance metrics and datasets

Datasets: Table 1 gives the detailed properties of the seven selected KEEL (Knowledge Extraction Evolutionary Learning) datasets. It shows the number of instances (Size), the number of attributes (Attribute), the number of classes (Class), the number of minority and majority instances (N_{min} and N_{maj}) and the imbalance ratio (IR) for each datasets. IR is a very important indicator to measure the degree of imbalanced data, which is defined as follows:

$$\mathrm{IR} = \frac{N_{maj}}{N_{min}} \tag{5}$$

where N_{maj} and N_{min} is the number of majority and minority class instance. According to the KEEL dataset, four of these datasets are low imbalanced data with IR < 9.0, and the other three datasets are high imbalanced data with IR > 9.0. In this study, the 10-folder cross-validation is used.

Performance Measures: The confusion matrix is the basis in measuring a classifier, The rows and columns of the confusion matrix are tabulated in Table 2, corresponding to predicted class and actual class. True positive (TP) is the number of positive samples that are correctly classified, false positive (FP) is the number of negative samples that are incorrectly classified as positive samples, true negative (TN) is the number of negative samples that are correctly classified, and false negative (FN) is the number of positive samples that are incorrectly classified as negative samples.

Table 1. Description of KEEL Datasets.

Datasets	Size	Attribute	Class	N_{min}	N_{maj}	IR
glass0	214	9	2	70	144	2.06
glass1	214	9	2	76	138	1.82
glass2	214	9	2	17	197	10.39
yeast12897	1004	8	2	30	917	30.57
yeast4	947	8	2	51	1433	28.1
pima	768	8	2	268	500	1.90
wisconsin	683	9	2	239	444	1.86

Table 2. Confusion matrix.

Actual	Predicted	
	Minority class	Majority class
Minority class	TP	FN
Majority class	FP	TN

On this basis, three indicators, Geometric Mean (G-mean), F1-score (F1-measure) and Area Under the ROC Curve (AUC), are used to evaluate the classification performance, and detailed definitions are shown in Eqs (9)–(11). It can be seen from Eqs (9) and (10) that G-mean considers the proportion of correctly classified instances in both minority and majority classes, while F-measure focuses more on the average performance of precision and recall.

$$Recall = \frac{TP}{TP + FN}, \tag{6}$$

$$Precision = \frac{TP}{TP + FP}, \tag{7}$$

$$Specificity = \frac{TN}{TN + FP}, \tag{8}$$

$$G - mean = \sqrt{Recall * Specificity}, \tag{9}$$

$$F1 - measure = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{10}$$

AUC, which is another significant metric obtained from the receiver operating characteristic (ROC) curve, can also be defined using *Recall* and $Specificity$ as follows:

$$AUC = \frac{Recall + Specificity}{2}. \tag{11}$$

4.2. Comparison of the BM with normally distributed $N(0, 1)$ random noise

In this section, to better verify the effectiveness of the BM, the following experiments are conducted. When network training is completed, we record the discriminator loss between the BM with random noise. For WGAN, the closer the loss function of discriminator is to zero, the more successful the network training is. Table 3 shows the discriminator loss of different methods. At this point, we can find that the loss function of our proposed algorithm is smaller, and the results show that the BM is better than the random noise. Therefore, the quality of the data attainable by the BM is very high.

Table 3. Discriminator loss of the Random noise and the BM.

Datasets	Random noise	Bootstrap method
glass0	-4.00×10^{-4}	-6.83×10^{-6}
glass1	-2.00×10^{-4}	-2.35×10^{-5}
glass2	-5.00×10^{-4}	-2.09×10^{-5}
yeast12897	-2.12×10^{-6}	-4.57×10^{-7}
yeast4	-4.45×10^{-6}	-4.35×10^{-7}
pima	-7.00×10^{-3}	-1.00×10^{-3}
wisconsin	-4.96×10^{-5}	-1.54×10^{-5}

4.3. Results and discussion

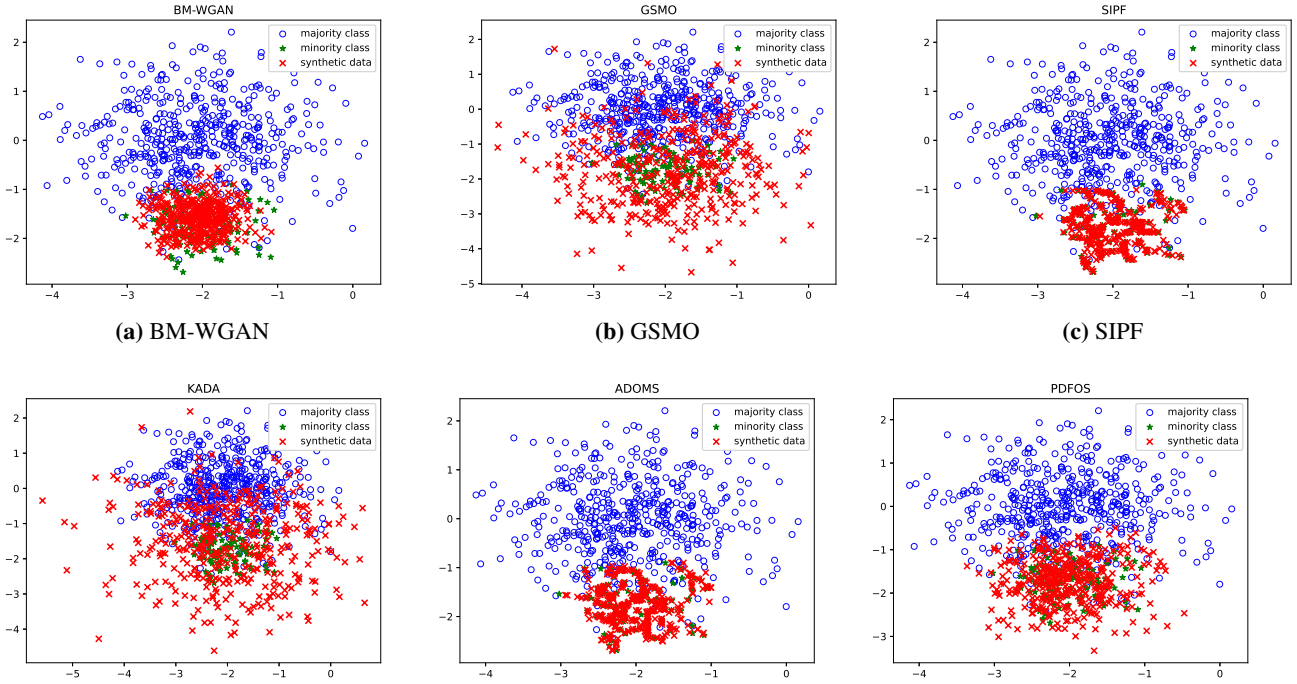
In addition, we compared our proposal with eight sampling algorithms, namely, SaSMO, Kernel-ADASYN (KADA) [34], Gaussian-SMOTE (GSMO) [35], SMOTE-IPF (SIPF) [36], PDFOS [37], ADOMS [38], ADASYN (ADA), LoRAS [39] and GANs [40]. To better verify the effectiveness of the BM-WGAN, the data with two dimensions is simulated in Figure 5, one for minority class and the other for majority class.

Figure 5 shows the significance of BM-WGAN by comparing GSMO, SIPF, KADA, PDFOS and ADOMS in two folds. In Figure 5(b) and (d), most of the instances synthesized by KADA and GSMO may be more fit for the distribution of minority class. However, the synthetic data has class overlap with the majority class, which affects the classification performance of the classifier. In Figure 5(c) and (e), more new synthetic data pays more attention to the impact of high-density areas on minority class data. As can be seen, BM-WGAN not only takes into account the location of generated data, but also further reduces the intersection and overlap with majority class. When compared with KADA and GSMO, the data generated by the BM-WGAN is more consistent with the distribution of the initial minority class. By doing so, the class information of data samples is fully utilized. Therefore, more minority class data located among the majority class can be distinguished to improve the classification performance of the classifier.

Tables 4–6 presents the specific parameter configurations of the compared algorithms. Decision Tree is used as the base learning model for BM-WGAN and the compared algorithms. Table 7 shows the distribution parameters of the Gaussian distribution estimated by the BM. Tables 8–10 presents the AUC, G-mean and F-measure results of the experimental study and the best result of each datasets is highlighted in boldface. It can be seen from Tables 8 that BM-WGAN has the best performances on AUC. Moreover, our method has been validated in seven independent datasets and performs well, such as 'glass0'. Second, ADOMS has also demonstrated a good classification performance in these experiments, such as 'glass2'. In 'glass1', we found that the classification performance of the new

algorithm is greatly improved compared with other algorithms. For ADOMS, the AUC value of ADOMS is not much different from that of SIPF.

Table 9 presents the G-mean results of the experimental study, and BM-WGAN performs best for five of the seven dataset. However, LoRAS is slightly worse than the new algorithm and performs better on the one datasets. It is worth noting that ADOMS achieves good results on 'glass2', which may be BM-WGAN cannot fully utilize the information of the minority class and generate high-quality data. In addition, because GSMO and SaSMO algorithms worse on all datasets when compared to SIPF and ADOMS. In short, compared with other oversampling algorithms, the results of the new algorithm are better.



(a) BM-WGAN **(b)** GSMO **(c)** SIPF

(d) KADA **(e)** ADOMS **(f)** PDFOS

Figure 5. Data visualization of different sampling methods.

Table 4. Parameter settings of the BM-WGAN.

Parameter	Value
Optimizer of Discriminator	RMSprop(l r= 1×10^{-5})
Optimizer of Generator	RMSprop(lr = 1×10^{-3})
Epochs	500
Batch Size	N_{min}
Generator layers sizes	{(Attribute,(Attribute,64))}
Discriminator layers sizes	{(Attribute,(Attribute,128))}
Activation function of hidden layers	Leaky ReLU
Number of Generator crosslayers	3
Number of Discriminator crosslayers	3

Table 5. Parameter settings of the GANs.

Parameter	Value
Optimizer of Discriminator	Adamprop(lr = 1×10^{-8})
Optimizer of Generator	Adamprop(lr = 1×10^{-3})
Epochs	500
Batch Size	N_{min}
Generator layers sizes	{(32,(32,64))}
Discriminator layers sizes	{(Attribute,(Attribute,128))}
Activation function of hidden layers	Leaky ReLU
Number of Generator crosslayers	3
Number of Discriminator crosslayers	3

Table 6. Parameter settings of the compared algorithms.

Oversampling method	Parameter
Safe-level-SMOTE	n_neighbors = 5, n_jobs = 1
Gaussian-SMOTE	n_neighbors = 5, sigma = 1.0, n_jobs = 1
SMOTE-IPF	n_neighbors = 5, k = 3, p = 0.01, n_jobs = 1
Kernel-ADASYN	k = 5, h = 1.0, n_jobs = 1
PDFOS	n_jobs = 1
ADOMS	n_neighbors = 5, n_jobs = 1,
ADASYN	n_neighbors = 5, sampling_strategy = "auto"
LoRAS	std = 0.005, sampling_strategy = "auto"
Decision tree	criterion = "gini", splitter= "best"

Figure 6 demonstrates the Precision-Recall Curve (PR Curve) for each model to more clearly demonstrate how the six models' different performances are further displayed by their ROC curves. ROC curves for all six advanced methodologies are represented in Figure 6: (a) PR Curve of BM-WGAN; (b) PR Curve of GSMO; (c) PR Curve of SIPF; (d) PR Curve of KADA; (e) PR Curve of ADOMS; (f) PR Curve of FDFOS. For BM-WGAN, the closer the AUC value to one demonstrates that the overall performance is better. In Figure 6, we show only six oversampling methods and BM-WGAN ranked first for six oversampling methods. In addition, we can also find that the AUC value of SIPF is slightly worse than the value of BM-WGAN.

Table 10 shows the F1-measure results of the nine oversampling algorithms in this paper. For F-measure, BM-WGAN performs best in seven datasets. In this case, ADOMS and LoRAS still achieved good classification results, such as 'glass2' and 'yeast12897'. Meanwhile, we note that the F-measure result for BM-WGAN in 'glass1' is greatly improved compared to other oversampling algorithms. Additionally, compared to SIPF and ADOMS, there is not much difference between two algorithms. On the contrary, the classification performance of GSMO and SaSMO has not greatly improved.

Table 7. Parameter settings on KEEL datasets by the BM.

Datasets	Parameter
glass0	$\mu = (1.51, 13.25, 3.55, 1.16, 72.61, 0.44, 8.80, 0.01, 0.03)$ $\sigma^2 = (5.05 \times 10^{-6},\ 2.60 \times 10^{-1},\ 6.06 \times 10^{-2},\ 7.36 \times 10^{-2},\ 3.20 \times 10^{-1},\ 4.57 \times 10^{-2},$ $3.56 \times 10^{-1},\ 7.13 \times 10^{-3},\ 4.52 \times 10^{-3})$
glass1	$\mu = (1.51,\ 13.11,\ 3.00,\ 1.40,\ 72.60,\ 0.52,\ 9.07,\ 0.05,\ 0.05)$ $\sigma^2 = (1.41 \times 10^{-5},\ 4.34 \times 10^{-1},\ 1.45, 1.00 \times 10^{-1},\ 5.23 \times 10^{-1},\ 4.44 \times 10^{-2},\ 3.63,\ 1.33 \times 10^{-1},\ 1.02 \times 10^{-2})$
glass2	$\mu = (1.51,\ 13.42,\ 3.54,\ 1.20,\ 72.39,\ 0.40,\ 8.78,\ 0.008,\ 0.05)$ $\sigma^2 = (3.43 \times 10^{-6},\ 2.48 \times 10^{-1},\ 2.51 \times 10^{-2},\ 1.12 \times 10^{-1},\ 2.49 \times 10^{-1},\ 4.98 \times 10^{-2},\ 1.36 \times 10^{-1},$ $1.23 \times 10^{-3},\ 1.08 \times 10^{-2})$
yeast12897	$\mu = (0.54,\ 0.52,\ 0.46,\ 0.20,\ 0.50,\ 0,\ 0.52,\ 0.25)$ $\sigma^2 = (0.019,\ 0.017,\ 0.008,\ 0.008,\ 0,\ 0,\ 0.002,\ 0.002)$
yeast4	$\mu = (0.72,\ 0.60,\ 0.41,\ 0.28,\ 0.51,\ 0,\ 0.51,\ 0.24)$ $\sigma^2 = (0.025,\ 0.014,\ 0.005,\ 0.015,\ 0.004,\ 0,\ 0.003,\ 0.002)$
pima	$\mu = (4.85,\ 141.21,\ 70.79,\ 22.15,\ 100.73,\ 35.16,\ 0.55,\ 37.11)$ $\sigma^2 = (1.39 \times 10^{1},\ 1.00 \times 10^{3},\ 4.61 \times 10^{2},\ 3.12 \times 10^{2},\ 1.91 \times 10^{4},\ 5.24 \times 10^{1},\ 1.38 \times 10^{-1},\ 1.19 \times 10^{2})$
wisconsin	$\mu = (7.19,\ 6.57,\ 6.55,\ 5.60,\ 5.32,\ 7.61,\ 5.98,\ 5.85,\ 2.59)$ $\sigma^2 = (5.92,\ 7.38,\ 6.58,\ 10.16,\ 5.91,\ 9.70,\ 5.20,\ 11.\ 17,\ 6.52)$

Table 8. AUC results on KEEL datasets obtained by Decision Tree.

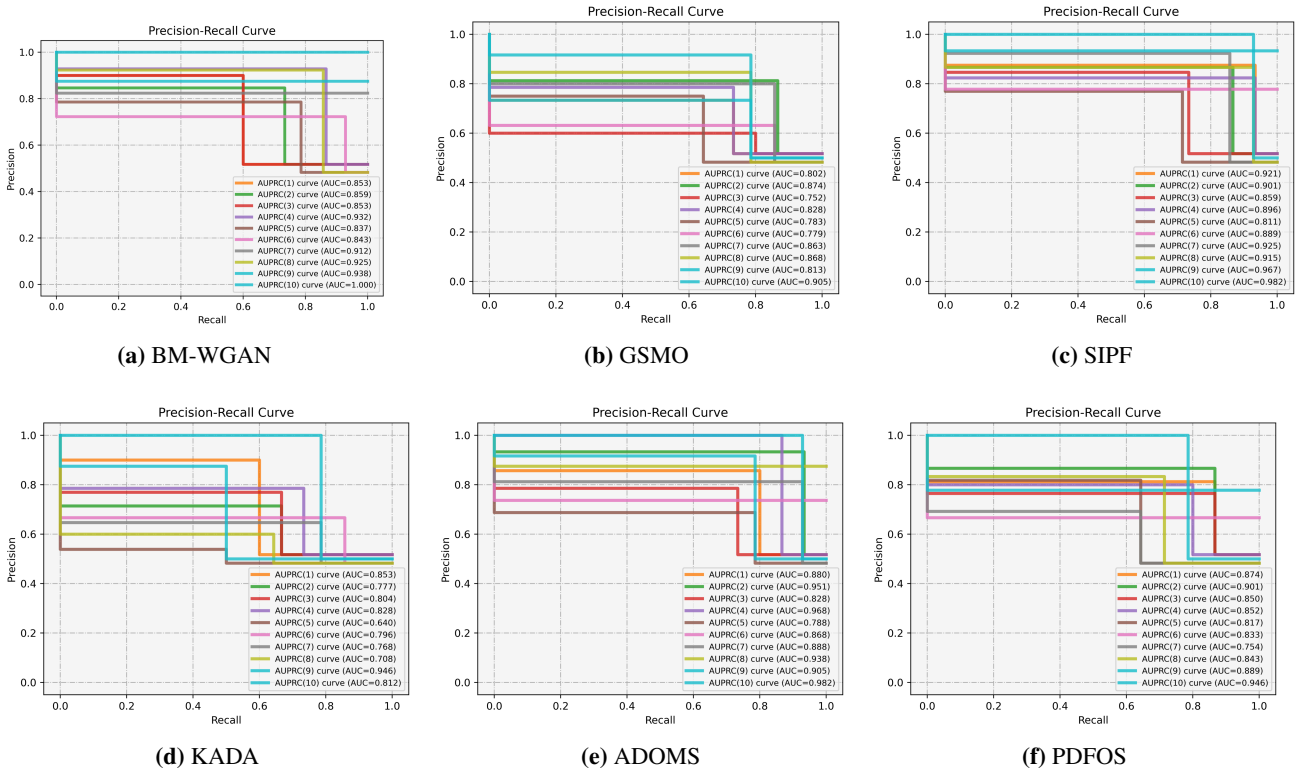
	BM-WGAN	SaSMO	GSMO	SIPF	KADA	PDFOS	ADOMS	ADA	LoRAS	GANs
glass0	**0.865 ± 0.081**	0.841 ± 0.090	0.757 ± 0.054	0.844 ± 0.078	0.677 ± 0.119	0.830 ± 0.091	0.844 ± 0.065	0.818 ± 0.086	0.792 ± 0.105	0.847 ± 0.100
glass1	**0.795 ± 0.068**	0.630 ± 0.074	0.743 ± 0.085	0.771 ± 0.097	0.742 ± 0.101	0.783 ± 0.105	0.759 ± 0.041	0.759 ± 0.077	0.695 ± 0.085	0.792 ± 0.076
glass2	0.920 ± 0.018	0.759 ± 0.102	0.868 ± 0.103	0.928 ± 0.036	0.770 ± 0.094	0.894 ± 0.114	**0.931 ± 0.058**	0.924 ± 0.065	0.851 ± 0.116	0.920 ± 0.118
yeast12897	0.961 ± 0.043	0.876 ± 0.116	0.959 ± 0.042	0.945 ± 0.013	0.951 ± 0.048	0.930 ± 0.043	0.951 ± 0.038	0.909 ± 0.039	**0.966 ± 0.044**	0.961 ± 0.043
yeast4	**0.969 ± 0.050**	0.897 ± 0.100	0.966 ± 0.049	0.960 ± 0.017	0.951 ± 0.045	0.966 ± 0.049	0.954 ± 0.019	0.912 ± 0.024	0.954 ± 0.036	0.967 ± 0.049
pima	**0.779 ± 0.108**	0.619 ± 0.038	0.708 ± 0.069	0.753 ± 0.068	0.735 ± 0.117	0.696 ± 0.064	0.746 ± 0.055	0.732 ± 0.047	0.746 ± 0.101	0.792 ± 0.105
wisconsin	**0.961 ± 0.037**	0.949 ± 0.038	0.948 ± 0.037	0.957 ± 0.042	0.939 ± 0.026	0.938 ± 0.022	0.952 ± 0.029	0.953 ± 0.022	0.955 ± 0.041	0.960 ± 0.040

Table 9. G-mean results on KEEL datasets obtained by Decision Tree.

	BM-WGAN	SaSMO	GSMO	SIPF	KADA	PDFOS	ADOMS	ADA	LoRAS	GANs
glass0	**0.858 ± 0.086**	0.834 ± 0.099	0.753 ± 0.055	0.838 ± 0.083	0.662 ± 0.124	0.822 ± 0.098	0.840 ± 0.068	0.813 ± 0.094	0.787 ± 0.112	0.839 ± 0.108
glass1	**0.788 ± 0.066**	0.608 ± 0.085	0.734 ± 0.091	0.758 ± 0.114	0.733 ± 0.113	0.771 ± 0.120	0.753 ± 0.042	0.752 ± 0.082	0.683 ± 0.092	0.781 ± 0.081
glass2	0.900 ± 0.173	0.738 ± 0.127	0.851 ± 0.144	0.928 ± 0.036	0.752 ± 0.125	0.874 ± 0.167	**0.930 ± 0.059**	0.921 ± 0.070	0.833 ± 0.161	0.900 ± 0.173
yeast12897	0.959 ± 0.048	0.876 ± 0.160	0.957 ± 0.046	0.945 ± 0.013	0.949 ± 0.052	0.928 ± 0.047	0.950 ± 0.040	0.908 ± 0.040	**0.964 ± 0.049**	0.959 ± 0.048
yeast4	**0.966 ± 0.05**	0.888 ± 0.119	0.964 ± 0.055	0.959 ± 0.017	0.950 ± 0.050	0.964 ± 0.055	0.954 ± 0.020	0.911 ± 0.025	0.953 ± 0.040	0.965 ± 0.055
pima	**0.774 ± 0.110**	0.608 ± 0.043	0.707 ± 0.070	0.751 ± 0.069	0.721 ± 0.122	0.692 ± 0.068	0.744 ± 0.055	0.731 ± 0.046	0.743 ± 0.103	0.774 ± 0.112
wisconsin	**0.960 ± 0.037**	0.948 ± 0.038	0.948 ± 0.037	0.957 ± 0.042	0.938 ± 0.027	0.937 ± 0.023	0.951 ± 0.030	0.952 ± 0.022	0.954 ± 0.043	0.960 ± 0.041

Table 10. F-measure results on KEEL datasets obtained by Decision Tree.

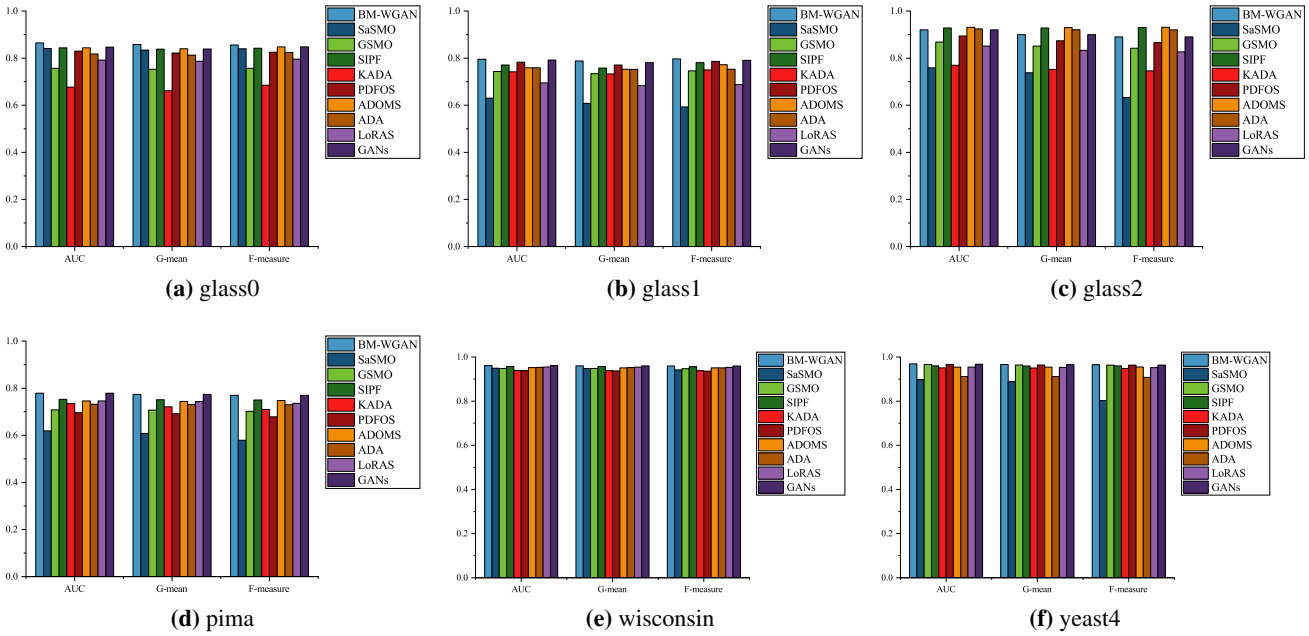
	BM-WGAN	SaSMO	GSMO	SIPF	KADA	PDFOS	ADOMS	ADA	LoRAS	GANs
glass0	**0.856 ± 0.095**	0.840 ± 0.092	0.757 ± 0.046	0.842 ± 0.082	0.685 ± 0.106	0.825 ± 0.102	0.848 ± 0.065	0.824 ± 0.070	0.796 ± 0.098	0.848 ± 0.099
glass1	**0.797 ± 0.078**	0.593 ± 0.098	0.746 ± 0.091	0.781 ± 0.095	0.750 ± 0.073	0.786 ± 0.100	0.772 ± 0.051	0.753 ± 0.076	0.688 ± 0.107	0.791 ± 0.078
glass2	0.890 ± 0.211	0.633 ± 0.151	0.842 ± 0.177	0.930 ± 0.036	0.746 ± 0.157	0.866 ± 0.205	**0.931 ± 0.060**	0.920 ± 0.077	0.827 ± 0.199	0.890 ± 0.211
yeast12897	0.958 ± 0.052	0.778 ± 0.116	0.956 ± 0.051	0.946 ± 0.012	0.950 ± 0.055	0.928 ± 0.053	0.950 ± 0.043	0.905 ± 0.043	0.963 ± 0.053	0.958 ± 0.052
yeast4	**0.965 ± 0.061**	0.802 ± 0.139	0.963 ± 0.061	0.960 ± 0.017	0.948 ± 0.056	0.963 ± 0.061	0.955 ± 0.019	0.908 ± 0.026	0.952 ± 0.043	0.963 ± 0.055
pima	**0.770 ± 0.132**	0.579 ± 0.062	0.702 ± 0.072	0.750 ± 0.075	0.710 ± 0.155	0.679 ± 0.082	0.748 ± 0.059	0.730 ± 0.053	0.736 ± 0.119	0.769 ± 0.112
wisconsin	**0.959 ± 0.039**	0.942 ± 0.044	0.947 ± 0.039	0.956 ± 0.044	0.938 ± 0.029	0.936 ± 0.025	0.951 ± 0.031	0.951 ± 0.023	0.953 ± 0.046	0.959 ± 0.043



(a) BM-WGAN **(b)** GSMO **(c)** SIPF

(d) KADA **(e)** ADOMS **(f)** PDFOS

Figure 6. Precision-Recall Curve of different sampling methods in glass0.



(a) glass0 **(b)** glass1 **(c)** glass2

(d) pima **(e)** wisconsin **(f)** yeast4

Figure 7. Comparison of different oversampling algorithms in different datasets.

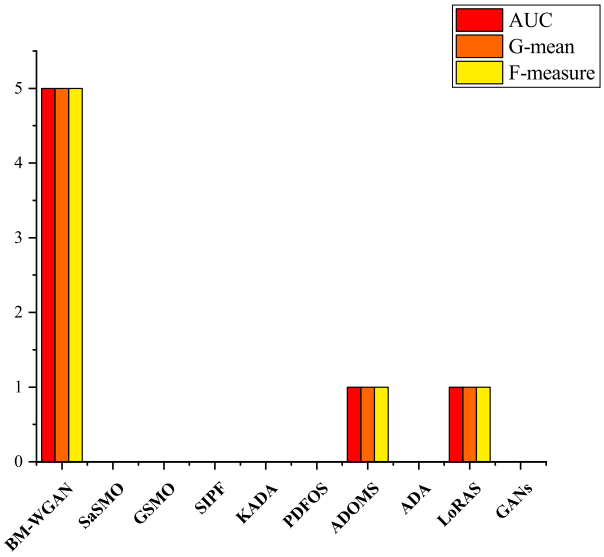


Figure 8. Comparison of winning times.

As shown in Figure 7–8, it describes the comparison of different sampling methods for different KEEL datasets and the winning times of nine different oversampling methods. In summary, BM-WGAN has the most winning times than other oversampling techniques. The winning times of BM-WGAN has reached 15 times. The second place is ADOMS and LoRAS, which also has good performance and its winning times has reached three times. The rest of the oversampling algorithm achieved a lower number of winning times compared to the new algorithm. These results indicate that our new method is more effective in imbalanced data classification.

5. Conclusions

This paper proposed a new oversampling algorithm based on BM and WGAN. First, the distribution of minority class data is unknown. The random noise from a standard normal distribution increases the complexity of the WGAN when fed into the generator. Therefore, we used the BM to estimate the distribution of minority class. The data from the distribution of minority class as random noise was fed into the generator to train WGAN. It increased the training accuracy of the WGAN. In addition, the generated data lacked diversity, which is a common problem that occurs when using the SMOTE family. However, the data generated by the WGAN will not be concentrated, enhancing the diversity of the data. Finally, BM-WGAN reduced the degree of class overlap and tried to make use of as many instances that contain further information as possible. Combining the above methods, our method ensures the stability of minority class distribution and improves the quality of the generated data.

Although the method proposed in this paper performed well, it still has some limitations. For example, the information about the neighbors for each minority sample is not sufficient. In addition, the information of majority class is not fully utilized. In future research, we will pay more attention to these limitations.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. N. V. Chawla, Data mining for imbalanced datasets: An overview, in *Data mining and knowledge discovery handbook*, Springer, (2010), 875–886. https://doi.org/10.1007/978-0-387-09823-4_45
2. X. Gao, Z. Chen, S. Tang, Y. Zhang, J. Li, Adaptive weighted imbalance learning with application to abnormal activity recognition, *Neurocomputing*, **173** (2016), 1927–1935. https://doi.org/10.1016/j.neucom.2015.09.064
3. J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P. E. Portier, L. He-Guelton, et al., Sequence classification for credit-card fraud detection, *Expert Syst. Appl.*, **100** (2018), 234–245. https://doi.org/10.1016/j.eswa.2018.01.037
4. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *J. Artif. Intell. Res.*, **16** (2002), 321–357. https://doi.org/10.1613/jair.953
5. H. Han, W. Y. Wang, B. H. Mao, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, in *Advances in Intelligent Computing*, Springer, (2005), 878–887. https://doi.org/10.1007/11538059_91
6. G. Douzas, F. Bacao, F. Last, Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE, *Inform. Sci.*, **465** (2018), 1–20. https://doi.org/10.1016/j.ins.2018.06.056
7. C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, *Advances in Knowledge Discovery and Data Mining*, Springer, (2009), 475–482. https://doi.org/10.1007/978-3-642-01307-2_43
8. H. B. He, Y. Bai, E. A. Garcia, S. T. Li, ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, (2006), 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969
9. Y. Xie, M. Qiu, H. Zhang, L. Peng, Z. X. Chen, Gaussian Distribution based Oversampling for Imbalanced Data Classification, *IEEE Trans. Knowl. Data Eng.*, **34** (2020), 667–669. https://doi.org/10.1109/TKDE.2020.2985965
10. Y. T. Yan, Y. F. Jiang, Z. Zheng, C. J. Yu, Y. W. Zhang, Y. P. Zhang, LDAS: Local density-based adaptive sampling for imbalanced data classification, *Expert Syst. Appl.*, **191** (2022), 13. https://doi.org/10.1016/j.eswa.2021.116213

11. W. H. Xie, G. Q. Liang, Z. H. Dong, B. Y. Tan, B. S. Zhang, An Improved Oversampling Algorithm Based on the Samples' Selection Strategy for Classifying Imbalanced Data, *Math. Probl. Eng.*, **2019** (2019), 526–539. https://doi.org/10.1155/2019/3526539

12. L. Z. Peng, H. L. Zhang, B. Yang, Y. H. Chen, A new approach for imbalanced data classification based on data gravitation, *Information Sciences*, **288** (2014), 347-373. https://doi.org/10.1016/j.ins.2014.04.046

13. M. Koziarski, B. Krawczyk, M. Wozniak, Radial-Based oversampling for noisy imbalanced data classification, *Neurocomputing*, **343** (2019), 19–33. https://doi.org/10.1016/j.neucom.2018.04.089

14. S. Suh, H. Lee, P. Lukowicz, Y. O. Lee, CEGAN: Classification Enhancement Generative Adversarial Networks for unraveling data imbalance problems, *Neural Netw.*, **133** (2021), 69–86. https://doi.org/10.1016/j.neunet.2020.10.004

15. E. Kaya, S. Korkmaz, M. A. Sahman, A. C. Cinar, DEBOHID: A differential evolution based oversampling approach for highly imbalanced datasets, *Expert Syst. Appl.*, **169** (2021), 794–801. https://doi.org/10.1016/j.eswa.2020.114482

16. F. Rahmati, H. Nezamabadi-Pour, B. Nikpour, A gravitational density-based mass sharing method for imbalanced data classification, *SN Appl. Sci.*, **2** (2020), 50–59. https://doi.org/10.1007/s42452-020-2039-2

17. H. K. Lee, S. B. Kim, An overlap-sensitive margin classifier for imbalanced and overlapping data, *Expert Syst. Appl.*, **98** (2018), 72–83. https://doi.org/10.1016/j.eswa.2018.01.008

18. V. López, A. Fernández, J. G. Moreno-Torres, F. Herrera, Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics, *Expert Syst. Appl.*, **38** (2012), 6585–6608. https://doi.org/10.1016/j.eswa.2011.12.043

19. C. Elkan, The foundations of cost-sensitive learning, *Acm SIGKDD Explor. Newsl.*, **6** (2004), 50–59. https://doi.org/10.1145/1007730.1007738

20. B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by cost-proportionate example weighting, in *Third IEEE International Conference on Data Mining*, (2003), 435–442. https://doi.org/10.1145/1007730.1007738

21. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, *IEEE Trans. Syst. Man Cybern.-C*, **42** (2011), 463–484. https://doi.org/10.1109/TSMCC.2011.2161285

22. Wang. S, Yao. X, Diversity analysis on imbalanced data sets by using ensemble models, in *2009 IEEE symposium on computational intelligence and data mining*, (2009), 324-331. https://doi.org/10.1109/CIDM.2009.4938667

23. N. V. Chawla, A. Lazarevic, L. O. Hall, K. W. Bowyer, SMOTEBoost: Improving prediction of the minority class in boosting, in *Knowledge Discovery in Databases: PKDD 2003*, Springer, (2003), 107–119. https://doi.org/10.1007/978-3-540-39804-2_12

24. C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, A. Napolitano, RUSBoost: A hybrid approach to alleviating class imbalance, in *IEEE Trans. Syst. Man Cybern.-A*, **40** (2009), 185–197. https://doi.org/10.1109/TSMCA.2009.2029559

25. L. Cao, H. Shen, Combining re-sampling with twin support vector machine for imbalanced data classification, in *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, (2016), 325–329. https://doi.org/10.1109/PDCAT.2016.076
26. G. E. Batista, R. C. Prati, M. C. Monard, Balancing strategies and class overlapping, in *Advances in Intelligent Data Analysis VI*, (2005), 24–35. https://doi.org/10.1007/11552253_3
27. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, Generative adversarial networks: An overview, *IEEE Signal Proc. Mag.*, **35** (2018), 53–65. https://doi.org/10.1109/MSP.2017.2765202
28. Z. Li, C. Ma, X. Shi, D. Zhang, W. Li, L. Wu, Tsa-gan: A robust generative adversarial networks for time series augmentation, in *2021 International Joint Conference on Neural Networks (IJCNN)*, (2021), 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534001
29. W. Li, J. Chen, J. Cao, C. Ma, J. Wang, X. Cui, et al., EID-GAN: Generative Adversarial Nets for Extremely Imbalanced Data Augmentation, *IEEE Trans. Ind. Inform.*, **19** (2022), 3208–3218. https://doi.org/10.1109/TII.2022.3182781
30. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, et al., Improved techniques for training GANs, in *Advances in neural information processing systems*, (2016), 29.
31. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in *Proceedings of the 34th International Conference on Machine Learning*, (2017), 214–223.
32. A. M. Zoubir, B. Boashash, The bootstrap and its application in signal processing, *IEEE Signal Proc. Mag.*, **15** (1998), 56–76. https://doi.org/10.1109/79.647043
33. B. Efron, Bootstrap methods: another look at the jackknife, in *Breakthroughs in Statistics*, (1992), 569–593. https://doi.org/10.1007/978-1-4612-4380-9_41
34. B. Tang, H. He, KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning, in *2015 IEEE Congress on Evolutionary Computation (CEC)*, (2015), 664–671. https://doi.org/10.1109/CEC.2015.7256954
35. H. Lee, J. Kim, S. Kim, Gaussian-based SMOTE algorithm for solving skewed class distributions, *Int. J. Fuzzy Log. Intell. Syst.*, **17** (2017), 229–234. https://doi.org/10.5391/IJFIS.2017.17.4.229
36. J. A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, *Inform. Sci.*, **291** (2015), 184–203. https://doi.org/10.1016/j.ins.2014.08.051
37. M. Gao, X. Hong, S. Chen, C. J. Harris, E. Khalaf, PDFOS: PDF estimation based over-sampling for imbalanced two-class problems, *Neurocomputing*, **138** (2014), 248–259. https://doi.org/10.1016/j.neucom.2014.02.006
38. S. Tang, S. P. Chen, The generation mechanism of synthetic minority class examples, in *2008 International Conference on Information Technology and Applications in Biomedicine*, (2008), 444–447. https://doi.org/10.1109/ITAB.2008.4570642
39. S. Bej, N. Davtyan, M. Wolfien, M. Nassar, O. Wolkenhauer, LoRAS: An oversampling approach for imbalanced datasets, *Mach. Learn.*, **110** (2021), 279–301. https://doi.org/10.1007/s10994-020-05913-4

40. H. Bhagwani, S. Agarwal, A. Kodipalli, R. J. Martis, Targeting class imbalance problem using GAN, in *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, (2021), 318–322. https://doi.org/10.1109/ICEECCOT52851.2021.9708011

AIMS Press