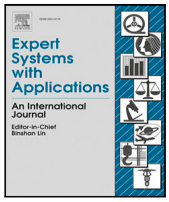
Expert Systems With Applications 237 (2024) 121696

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Generative adversarial minority enlargement—A local linear over-sampling synthetic method

Ke Wang [a,b], Tongqing Zhou [a], Menghua Luo [c], Xionglve Li [a], Zhiping Cai [a,*]

[a] Computer College, National University of Defense Technology, Changsha, China
[b] Information College, GuiZhou University of Finance and Economics, Guiyang, China
[c] Big Data Application and Economics College, GuiZhou University of Finance and Economics, Guiyang, China

ARTICLE INFO

Keywords:
Imbalanced data problem
Generative adversarial model
Synthetic numerical samples
Local linear property of samples
Over-sampling strategy

ABSTRACT

The imbalanced data problem is widely recognized in real-world datasets. To avoid learning bias on imbalanced data, the over-sampling strategy is well studied and adopted for generating synthetic minority samples. Wherein, the Synthetic Minority Over-sampling Technology and its improved algorithms become standard baselines. In recent years, the popular Generative Adversarial Networks and its enhanced variants, introduced from the computer vision community, are believed to generate better samples, by approximating the true data distribution. Yet, we notice that the synthetic samples for the minority category in these existing methods is usually restrained in a limited samples space known in advance, which may mislead the classifiers trained on them to take data in the unsampled region of the minority category as from the majority category. Given such limitations, we propose a Generative Adversarial Minority Enlargement (GAME) method to intentionally extend the sampling margin during data generative and adversarial phases. This is accomplished by adjusting the parameters of a local linear model to approach the majority category. We conduct an extensive evaluation on 28 datasets of different domains, extracted from the UCI real-world datasets. The results show that GAME can achieve more balanced and stable results efficiently than 18 state-of-the-art methods.

1. Introduction

Dealing with imbalanced data is a well-known challenge in conducting data analysis based on machine learning for both academia and industry. In fact, for most of the available datasets obtained from real-world situations, the number of samples for different categories is hardly approximated to be equal. Requiring more real samples for the minority classes usually means enlarging the annotation cost, and is quite hard, if not impossible, for specific applications, such as fraud detection (Zhang et al., 2022), traffic identification (Mehrannia, Bagi, Moshiri, & Al-Basir, 2021), and medical diagnosis (Orooji & Kermani, 2021).

The imbalanced data problem incurs uncertainty and inaccuracy for the training process of the classifier (Zhang et al., 2022). The classification performance for the majority class is always good with sufficient samples, while the machine learning models usually fail to detect the minority class. In order to relieve the pitfall, over-sampling approaches are widely adopted as an efficient technique to generate synthetic samples for the minority class, thus facilitating classification models with balanced inputs in terms of each category.

However, we point out that the existing efforts devoted to over-sampling construct new samples by merely looking at the minority group, which leads to large unsampled space between borders of different categories. On one hand, classic methods, including SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002; Dablain, Krawczyk, & Chawla, 2022) and the improved versions (e.g., SMOTE borderline 1, SMOTE-borderline 2 (Han, Wang, & Mao, 2005), ADASYN (He, Bai, Garcia, & Li, 2008), SMOTE-SVM (Wang, Luo, Huang, Feng, & Liu, 2017),Polynom-fit SMOTE (Gazzah & Amara, 2008), ProWSyn (Barua, Islam, & Murase, 2013), ProWRAS (Bej, Schulz, Srivastava, Wolfien, & Wolkenhauer, 2021; Kovács, 2019), and CURE-SMOTE (Ma & Fan, 2017).), propose to generate new samples by randomly selecting data points on the line segments between known labeled data in the minority class. Obviously, such a generation process cannot get rid of the possible limitation of the original data distribution. On the other hand, the popular Generative Adversarial Networks (GAN) (Goodfellow, 2017; Goodfellow et al., 2014) and its enhanced variants (e.g., DCGAN (Radford, Metz, & Chintala, 2015), CGAN (Mirza & Osindero, 2014), WGAN (Arjovsky, Chintala, & Bottou, 2017), SGAN (Huang,

* Corresponding author.
E-mail addresses: kewang@mail.gufe.edu.cn (K. Wang), zhoutongqing@nudt.edu.cn (T. Zhou), mh_luo@mail.gufe.edu.cn (M. Luo), lixionglve17@nudt.edu.cn (X. Li), zpcai@nudt.edu.cn (Z. Cai).

https://doi.org/10.1016/j.eswa.2023.121696
Received 30 March 2022; Received in revised form 28 April 2023; Accepted 16 September 2023
Available online 21 September 2023

Li, Poursaeed, Hopcroft, & Belongie, 2017), CTGAN (Xu, Skoularidou, Cuesta-Infante, & Veeramachaneni, 2019) and CTABGAN (Zhao, Kunar, Birke and Chen, 2021)) are powerful in generating synthetic images based on the game between the generative and discriminative models. The given minority samples can hardly depict the true distribution comprehensively, so the synthetic data of GAN is restrained to a limited range. Considering the above limitations, existing methods will yield classifiers that show an unexpected preference for the majority category. A data synthetic method that can formally approximate the true distribution of the minority category is still in need.

In this paper, we propose a Generative Adversarial Minority Enlargement model (GAME) to construct synthetic data with an enlarged over-sampling space. A modified Jacobian-based (Papernot et al., 2016) data augmentation method is designed to obtain initial synthetic data in the generative phase. Then we bring forth the local linear property of samples based on the idea that data points falling in a sufficiently small space can be classified with a line. Given this insight, we adopt logistic regression as the building block to classify the updated samples in the adversarial phase and iteratively adjust the synthetic samples to the direction of the majority category by injecting the majority samples into the training data. In this way, our GAME can provide synthetic samples that are distributed in the undistinguishable space of the original dataset, facilitating truly balanced data in over-sampling. Note that the training cost can be shortened and the effect can be better guaranteed as we combine generative adversarial thinking and our local linear classification unit.

Contributions. The main contributions of this work are summarized as follows:

- We propose a novel over-sampling model, GAME, by exploiting the idea of local linear property of samples. GAME integrates logistic regression into a generative and adversarial process to generate synthetic data.
- We leverage the adversarial process to calibrate the generated data for better representing the minority category. By using the mixture of majority and minority samples for training, we can gradually adjust the data with an enlarged sampling space that is expected to better approximate the true minority samples distribution.
- We evaluate the classification performance of the proposal by 4 classifiers on 28 datasets of various imbalanced ratios. Overall, GAME can yield better F1 scores than the 18 baselines, and shows good execution efficiency at the same time.

2. Related work

We summarize the relevant work from two aspects: the SMOTE series algorithms (SMOTEs) and the GAN-based methods (GANs).

The SMOTE series algorithms (SMOTEs). In over-sampling, the enlargement of minority samples is the key task. The duplicated samples method is phased out because of poor accuracy and over-fitting problem. The synthetic minority over-sampling technique (SMOTE) (Chawla et al., 2002) is the cornerstone of the existing methods. This algorithm randomly generates synthetic samples on the selected line segment. On this basis, some improvements have been made in the past years. Borderline-SMOTE (Han et al., 2005) is a significant improvement of SMOTE. This method divides the minority samples into three groups, DANGER, SAFE, and NOISE, where DANGER is the borderline samples set. The method has two specific algorithms, borderline1 generates synthetic samples between DANGER samples and k nearest minority neighbors. In borderline2, the k nearest neighbors include both minority and majority samples. ADASYN (He et al., 2008) is another milestone in the development of SMOTE. It proposes a density distribution of majority samples in k nearest neighbors, according to which assigns the number of the synthetic data samples. SMOTE-SVM (Wang et al., 2017) uses the SMOTE on support vectors to improve the classification effect. Polynom-fit SMOTE (Gazzah & Amara, 2008) oversamples the minority samples using polynomial fitting functions. Proximity Weighted Synthetic algorithm (ProWSyn) (Barua et al., 2013) generates effective weights for the minority samples based on sample's proximity information as distance from boundary. ProWRAS (Bej et al., 2021; Kovács, 2019) integrates the Localized Random Affine Shadowsampling (LoRAS)algorithm and ProWSyn and improves performance by controlling the variance of the synthetic samples, as well as a proximity-weighted clustering system of the minority samples. CURE-SMOTE (Ma & Fan, 2017) uses the combination of Clustering Using Representatives (CURE) to enhance the original SMOTE algorithms. Although different strategies are adopted, the core approach of these methods to generate synthetic samples is not changed. The synthetic samples must be limited within the range of the given minority samples and cannot be further adjusted.

The GAN-based methods (GANs). GANs become more and more popular in the field of computer vision for image generation. The game of discriminative and generative models leads to many achievements in image generation. Although varying improved models and techniques (Salimans et al., 2016) are proposed, the non-convergence problem, collapse problem, and uncontrollability have become bottlenecks. Conditional Generative Adversarial Nets (CGAN) (Mirza & Osindero, 2014) introduces condition data on both generator and discriminator. Deep Convolutional Generative Adversarial Networks (DCGAN) (Radford et al., 2015) have certain architectural constraints good for unsupervised learning. Wasserstein GAN (WGAN) (Arjovsky et al., 2017) introduces a stable way of learning dealing with mode collapse problem. Stacked Generative Adversarial Networks (SGAN) (Huang et al., 2017) have a bottom-up discriminative network to invert the hierarchical representations. CTGAN (Xu et al., 2019) uses a conditional generative adversarial network to generate synthetic samples for tabular data. CTABGAN (Zhao, Kunar et al., 2021) is another GAN-based method for tabular data which introduces the information loss, classification loss and generator loss to the conditional GAN and encodes the mixed data type and skewed distribution of data variable by a conditional vector. The approximation of the true data distribution is also a restriction of generating synthetic samples from the limited minority samples. The synthetic samples are essentially limited to the interior of the given minority samples. From the perspective of imbalanced data, it only considers the given minority samples and does not consider the influence of the majority of samples, so there must be a lack of overall data information.

Some Special Approaches. Some effective special approaches focus on adjusting the algorithm itself or integrating domain knowledge to deal with given imbalanced data, without concerning the data distribution. A typical approach here is the cost-sensitive method (Mienye & Sun, 2021). The main point of this method is to design the cost matrix with domain knowledge. To a certain dataset, with the help of domain experts, we may get a satisfactory result, which is both an advantage and a limitation. Ensemble learning methods (Chen, Duan, Kang, & Qiu, 2021; Xu, Yu, Chen, & Liu, 2021) utilize multiple models and some adjustments to the dataset to improve performance. One-class classification method (Hayashi, Fujita, & Hernandez-Matamoros, 2021) is more suitable for the situation of only very few minority samples existing, which tries to shrink the boundary of the majority samples. The classification result is no majority or minority, but whether belonging to the majority. Feature selection (Agrawal, Abutarboush, Ganesh, & Mohamed, 2021) is an effective method to some imbalanced data problems. The imbalanced samples always come with imbalanced features, which means choosing the discriminate features can improve the minority samples' recognition rate. A contrastive variational autoencoder method (Dai et al., 2019) leverages both the majority and minority classes information on 2 clinic datasets with highly imbalanced outcomes. Experiment results show that their algorithm performs better when the number of minority class samples is very small. But the application field of this method is relatively limited. A conditional variational autoencoder-based self-transferred algorithm (Zhao, Hao,

Tang, Chen and Wei, 2021) is proposed to solve the highly imbalanced classification problem, where the training instances of the minority classes are rare. The Variational Autoencoder method is a well-designed data generation model, but there is a certain amount of information loss in its encoding process. As in the image dataset, it is directly calculating the mean square error of the generated image and the original image instead of learning by adversarial like GAN, which makes the generated image a bit blurry. An oversampling framework SWIM (Bellinger, Sharma, Japkowicz, & Zaïane, 2020) uses the density of the well-sampled majority class to guide the generation process, which is designed to address the extremely imbalanced problem without advantages for different imbalanced ratios. The introduction of some mathematical methods (Du, Zhou, Liu, Vong, & Wang, 2021; Xia, Zheng, Wang, Gao, & Wang, 2021) has also produced good results on certain datasets. These methods work well on specific datasets and problems. But they are not universal and difficult to generalize.

The previous approaches have some drawbacks: The generating method of SMOTEs has a limitation of data generating range and is not convenient for adjustment; The GANs methods are difficult to break through the limitations of given data, do not consider the global data information, and consume large computing resources; Some special methods are difficult to widespread use. Our method breaks the limitations mentioned above and achieves better results on 28 real-world datasets of various fields.

3. Local linear property of samples in imbalanced data problem

In the data space, the classification hyperplane is always an irregular surface. In machine learning, varying models and algorithms are proposed to deal with hyperplane fitting problems. In over-sampling problems, insufficient data makes it impossible to obtain real global data distribution of the minority category. So we choose a local perspective in our research. In the samples space which is a subspace of the data space, we just focus on the local top k points on both sides of the hyperplane, as shown in Fig. 1(a). The blue triangles stand for the minority samples, while the red points represent the majority samples. We select a tiny space around the hyperplane, where k is small enough, as the yellow space indicated by the cyan arrow in Fig. 1(a). Let us zoom the yellow space, as shown in Fig. 1(b). The several samples in this tiny space can always be linear separable. And a tiny classification line can be calculated as the blue line in Fig. 1(b). All these tiny local classification lines make up the accurate classification multiple dimensional curves, without the requirement of composing a global classification hyperplane. “Local linear property” means that in a very small part of the whole data, where are only several, or only one, minority sample(s), making the minority samples and the majority samples in this small region linear separable.

The traditional deep learning method is to construct the hyperplane globally and then adjust and optimize it through iterations. The local linear view is to optimize the local area. For data synthesis, the local samples’ generation is the focus point, but not the global hyperplane. From the view of our local linear property of samples, local samples’ characteristic is the decisive factor for the synthetic data generation, while the data points far away are of little significance for local data synthesis. Therefore, using the properties of local data points to adjust and optimize the generated samples is the core of our method. In addition, the local linear property of samples makes use of local computing to save computing resources and improve efficiency. This property brings a lot of convenience to machine learning processing. The linear method is simple and mature. For a small number of data, it can deal with linear and nonlinear classification problems in a reasonable range of errors and reduce the probability of the over-fitting problem. From the point of view of the model training process, the linear models are much faster than the DNN and perform as well as them in small training samples.

4. Generative adversarial minority enlargement model

According to our local linear property of samples, the whole dataset is made up of many local groups. There are only two kinds of groups, the inside group consists of only samples of one category, and the outside group is from both the majority and the minority samples. For the inside groups of majority samples, we maintain the existing state. For the inside groups of minority samples, we generate some synthetic samples by SMOTE to keep the data distribution stable. For the outside groups, we design and use the GAME to generate the synthetic samples.

In this section, we will first introduce the local linear function, then go through the generative part and discriminative part of our GAME model, and finally depict the whole data generation process.

4.1. The local linear function

This function selects the local linear region data and prepares it for the follow-up work, where the details are shown in Algorithm 1. First, it calculates the initial classification line $line_0$ for the given minority samples S_{min} and majority samples S_{maj} (Line Input), even if they are nonlinear separable. Then it makes S_{maj}, S_{min}, and $line_0$ as inputs for the local linear function. For each sample x_j in S_{maj}, it calculates D_{maj}, the set of distances between x_j and $line_0$ (Line 1–3). Next, the top n samples with shortest distances in D_{maj} are chosen as $Near_{maj}$, which contains the nearest n majority samples to the minority samples (Line 5). After that, for each x_i in S_{min}, it selects local k samples from the minority samples and $Near_{maj}$, requiring at least one of them is from $Near_{maj}$. The $local_i$ here is the data set of x_i and its k-nearest neighbors points. $Local_k$ is the set of all the $local_i$. D is the data set of distances from $Local_k$ (Line 6–11). At last, it returns $Local_k$ and D for the generative process.

Algorithm 1 Local_Linear_F

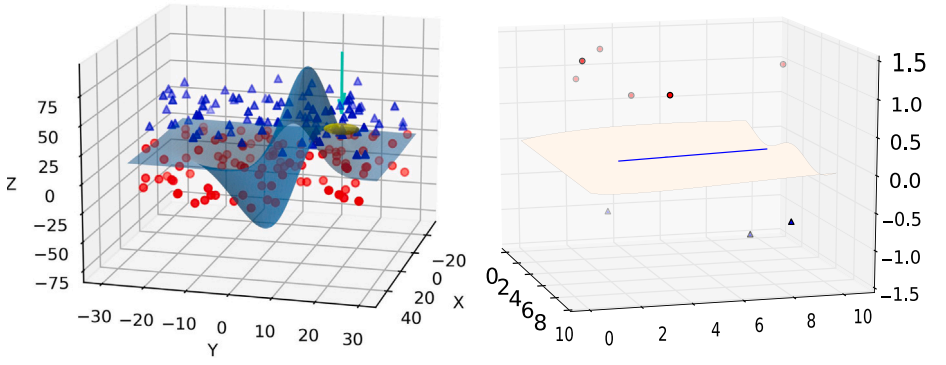
Input: S_{maj}, S_{min}, $line_0$
Output: $Local_k$, D

1: **for** $each\ x_j \in S_{maj}$ **do**
2: $d_j = distance(x_j, line_0)$
3: $D_{maj} = D_{maj} + d_j$
4: **end for**
5: select the minimal n of D_{maj} as $Near_{maj}$
6: **for** $each\ x_i \in S_{min}$ **do**
7: $[x_1, x_2, \dots, x_k] = KNN(x_i, (S_{min} + Near_{maj}))$
8: $local_i = [x_i, x_1, x_2, \dots, x_k]$
9: $Local_k = Local_k + local_i$
10: $D = distance(KNN(Local_k))$
11: **end for**
12: **return** $Local_k$, D

4.2. The generative part

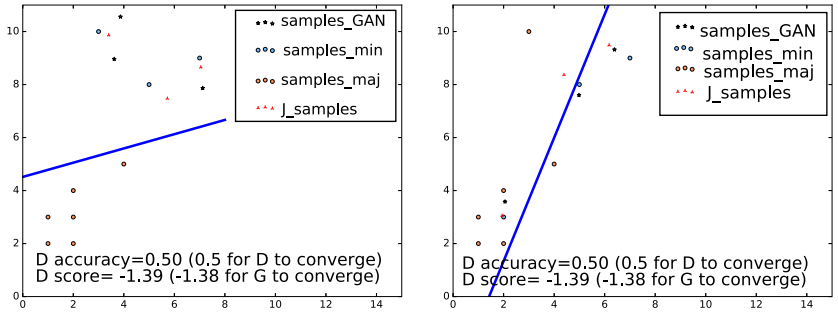
The generating method is the core of the over-sampling synthetic technology. The purpose of our method is to generate synthetic samples of minority samples, not to calculate a global piecewise classification boundary. The local repetitions generate the required samples, but do not consider whether the local classification line can form an overall classification boundary or hyperplane. In this case, the required local synthetic samples can be iterative generated. When the local generation of each minority sample is completed, the goal of balancing data is achieved, regardless of the distribution of the known samples or the global classification boundary. We compare the present methods and adjust them as the basis of our generative method.

In SMOTE and its improved algorithms, the generation process almost uses Eq. (1). It randomly picks up x_j, one of the k nearest neighbors of the selected minority sample x_i, and then generates synthetic sample x_g with Eq. (1), where $\lambda \in (0, 1)$. In simple words, x_g is



(a) Local hyperplane (b) Approximation of line on local hyperplane

Fig. 1. Local linear property of samples in imbalanced data problem.



(a) linear separable samples (b) nonlinear separable samples

Fig. 2. Jacobian method VS. GAN in the local generating process.

selected randomly on the line segment (x_i, x_j). In SMOTE-borderline2, the parameter λ is adjusted ($\lambda \in (0, 0.5)$) to make the generated samples close to the minority sample x_i.

$$\vec{x_g} = \vec{x_i} + \lambda * (\vec{x_j} - \vec{x_i}) \tag{1}$$

There is a space range between the majority and minority samples. The SMOTEs and GANs generate data within the given data distribution. These generating methods neglect the most space. The direction of our study is to enlarge the generating space and to make the synthetic samples adjustable in the required direction. In order to generalize the method of synthetic data in a more general way, we propose Eq. (2), where the $\Delta\vec{noise}$ here is the key point of the algorithm design. Obviously, Eq. (1) is a specific form of Eq. (2).

$$\vec{x_g} = \vec{x_i} + \Delta\vec{noise} \tag{2}$$

$$\vec{x_g} = \vec{x_i} + \lambda_i * sgn(J_F(O(\vec{x_i}))) \tag{3}$$

A Jacobian-based data augmentation technology (Papernot et al., 2016) is proposed as Eq. (3), where λ_i is the step size of the learning, J_F is the Jacobian Matrix, and $O(\vec{x_i})$ is the label of x_i. To the demand of our work, we adjust the Jacobian method as Eq. (4). We change $O(\vec{x_i})$ to $\vec{x_i}$ as for the Jacobian Matrix dimension requirement. We replace λ_i with $distance_i$ which is the distance between two of the top k points.

$$\vec{x_g} = \vec{x_i} + distance_i * sgn(J_F(\vec{x_i})) \tag{4}$$

On the basis of full study of traditional over-sampling methods and GANs, in the GAME, the generating technology is the Jacobian method, as Eq. (4). The Jacobian Matrix represents the optimal linear approximation of a differentiable equation to a given point. Here is the sign of the Jacobian Matrix value of input $\vec{x_i}$, and a mapping direction of the change from $\vec{x_i}$ to the straight line is obtained, so that the added noise can be near $\vec{x_i}$ and the classification line without being

Algorithm 2 Generator

Input:G_n, $Local_k$, D, Num_{Max_iter}
Output:GneratedSamples:X_{gen}

for *each* $x_i \in S_{min}$ **do**
  $local_i = Local_k[i]$
  $d_i = D[i]$
  $Maj_i, Min_i = Local_i$
  $Xg_0 = Min_i$
  $Num_{iter} = [G_n/Num(S_{min})]/Num(Min_i)$
  if $Num_{iter} > Num_{Max\_iter}$ **then**
    $Num_{iter} = Num_{Max\_iter}$
  end if
  for $j = 1; j \leq Num_{iter}; j++$ **do**
    if $Discriminator_{flag}$ **then**
      break
    end if
    line(w, b)=Logistic_Regression(Maj_i, Min_i)
    $J_F(x)$ = Jacobian(line$(w, b)(x)$)
    $Xg_j = Min_i + d_i \times sgn(J_F(Xg_{j-1}))$
    $X_{gen} = X_{gen} + Xg_j$
    $Min_i = Min_i + X_{gen}$
  end for
  $Num_{inside} = Gn - Num_{X_{gen}}$
  $Xg_{inside} = SMOTE(Num_{inside})$
  $X_{gen} = X_{gen} + Xg_{inside}$
end for
return X_{gen}

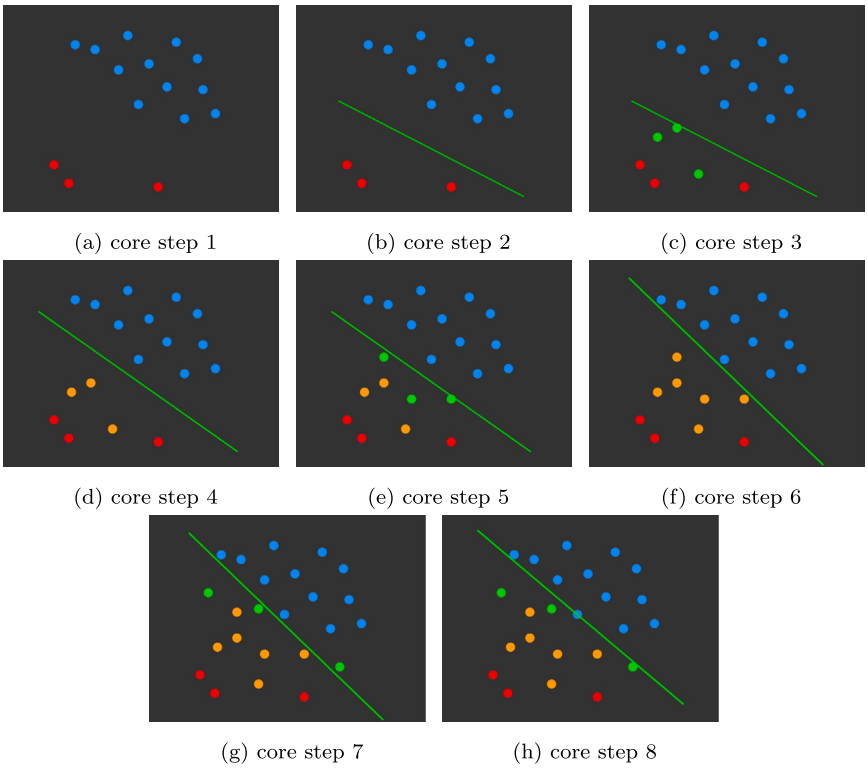


Fig. 3. The demonstration of core steps.

too random. And it is faster than simple random initialization and the GAN model.

In order to get efficient and fast initialized generated samples, we compare the Jacobian method with GAN, as shown in Figs. 2(a) and 2(b). After thousands of iterations of training, GAN gives the convergent results, while the Jacobian method directly generates the final results without training. From the results shown in Fig. 2, there are no obvious differences between the two approaches in both linear and nonlinear separable samples. So in the generative part, we choose the Jacobian method as the initialized generating approach. First, we get coefficients of the required generating number G_n, the maximum number of iterations Num_{Max_iter}, and $Local_k, D$ from the local linear function (Line Input). The next step is the core generation process. For each $x_i \in S_{min}$ with the $Local_k[i]$ and $D[i]$, the local majority samples Maj_i and minority samples Min_i, an initialized input data Xg_0 and the required number of iterations Num_{iter} are prepared for the iterative generation process (Line 2–6). If the required number of iterations is greater than the maximum value, the maximum value is taken as the actual number of iterations. During the loop, it determines whether the discriminator condition is met first (Line 11). After that, a new classification line is calculated by Maj_i and Min_i (Line 14). Then a Jacobian Matrix $J_F(x)$ is calculated by which the synthetic samples' matrix in this iteration Xg_j is generated (Line 15–16). Next, Xg_j is integrated into the generated dataset X_{gen}. And X_{gen} is added to Min_i for the next iteration (Line 17–18). At last, if there is an insufficient amount of synthetic samples, the rest are generated by the inside group with the SMOTE algorithm. This generative process is shown as Algorithm 2.

The demonstration of core steps is shown as Fig. 3. For each minority sample (the red point), its k-nearest neighbors $local_i$ (shown in the red and blue points) and the corresponding distances set d_i are loaded as Fig. 3(a). The classification line (green line) is calculated as Fig. 3(b). The synthetic samples (green points) are generated by Eq. (4) as Fig. 3(c). In the iterative process, the new minority samples (the red and orange points) and majority samples (the blue points) recalculate the classification line as Figs. 3(d) to 3(h). This process proceeds until the classification line in the discriminator cannot tell the majority points (blue points) and the new synthetic samples (green points) as Fig. 3(h).

Comparing the classic generating method (Eq. (1)) in SMOTE and its improved algorithms, our generative approach is more efficient and controllable. The SMOTE method randomly generates synthetic samples in both inside group and outside group, which reflects the characteristics of the minority samples distribution in some degree but is uncontrollable. GANs care about the overall optimization process without focusing on local details, whose optimization effect does not have advantages for several local points. In our way, the generated synthetic samples are optimized locally and close to the real data distribution within an enlarged given data range. In the next section, we discuss the condition for the discriminator.

4.3. The discriminative part

In the training process of GANs, the minority samples are treated as real data, but the majority samples are not involved at all. The synthetic samples are trained to follow the real data distribution. In our method, the core target is to refine and make full use of data distribution information. We introduce the majority samples in the discriminator.

As an input parameter in the discriminative part, X_{gen} from the Generator iteration is integrated with S_{min}, of which the result and S_{maj} determine a new classification line $line_{new}$ (Line 1). Then it selects samples from S_{maj} with k nearest distances to $line_{new}$ as S'_{maj}, where k equals the number of X_{gen} (Line 2–4). We compute and record the classification results of $prob_{maj}$ and $prob_{min}$ for Loss, as Eq. (5) (Line 5–7). In detail, first, we make sure that the synthetic samples and the given minority samples are on the same side of the classification line. Then, each sample in X_{gen} is moved towards $line_{new}$ by a value which is the smallest value in D_{new} multiplied by a random number between 0 and 1. After that, it updates X_{gen} from the next round of iteration in the Generator (Line 8). So that the new classification line gradually

Algorithm 3 Discriminator

Input: $X_{gen}, S_{maj}, S_{min}$
Output: Sythetic Samples: X_{syn}

$line_{new} = Logistic\_Regression((X_{gen}, S_{min}), S_{maj})$
$D_{new} = distance(S_{maj}, line_{new})$
$k = Num_{X_{gen}}$
$S'_{maj} = KNN(S_{maj})$
$prob_{min} = line_{new}(X_{gen})$
$prob_{maj} = line_{new}(S'_{maj})$
$Loss = \bigtriangledown \frac{1}{m} \sum_{i=1}^{m} [log(1 - prob_{maj}) + log(prob_{min})]$
Adjust X_{gen} and Update the *Generator* with $line_{new}$ and X_{gen} by ascending the *Loss*.
if Loss converges or iteration stops **then**
    $X_{syn} = X_{gen}$
end if
return X_{syn}

closes to the majority samples. In this way, we expand the local data distribution reasonably and simplify the training model effectively. The details are shown in Algorithm 3.

$$Loss = \bigtriangledown \frac{1}{m} \sum_{i=1}^{m} [log(1 - prob_{maj}) + log(prob_{min})] \tag{5}$$

4.4. More details on GAME

After specifying the local linear function, the Generator and the Discriminator above, we summarize the whole GAME model in Algorithm 4. First, we compute G_n (Line 1), the amount of the generated samples and set the maximum iteration number Num_{Max_iter}. Then, the initial classification line of S_{maj} and S_{min} is calculated (Line 2). After that, we put these parameters into the local linear function (Line 3). Next, the Generator and the Discriminator perform data generation and update the loss according to their algorithms (Line 4–5). At last, the synthetic samples, the given minority and majority samples form balanced samples (Line 6). We update the local linear function and the Generator by ascending the stochastic gradient as Eq. (5) in the Discriminator. The iterative adjustment process of classification lines and synthetic samples gradually enlarges the data generating space.

Algorithm 4 the whole GAME algorithm

Input: S_{maj}, S_{min}, Num_{Max_iter}
Output: S_{syn}

1: $G_n = Num(S_{maj}) - Num(S_{min})$
2: $line_0 = Logistic\_Regression(S_{maj}, S_{min})$
3: $Local_k, D = Local\_Linear\_F(S_{maj}, S_{min}, line_0)$
4: $X_{gen} = Generator(G_n, Local_k, D, Num_{Max\_iter})$
5: $X_{syn} = Discriminator(X_{gen}, S_{maj}, S_{min})$
6: $S_{syn} = X_{syn} + S_{min} + S_{maj}$
7: **return** S_{syn}

In the imbalanced data problem, the minority samples can rarely stand for the whole minority sample distribution. So the synthetic samples generated by SMOTEs and GANs are not representative of the minority samples. In our view, the SMOTEs are uncontrollable and not adjustable, while the GANs are a fine-grained learning process with a coarse-grained generation. The main function of SMOTEs and GANs is to generate synthetic samples that satisfy the given data distribution. GAME is not only to maintain the original minority samples characteristics but also to break through the given data distribution to enlarge the data generating space. The local data generation complements the overall data shortage without calculating the overall classification hyperplane. A large number of hyperparameter calculations in the iteration are avoided, and computational resources are saved.

Table 1
The majority-minority ratio description.

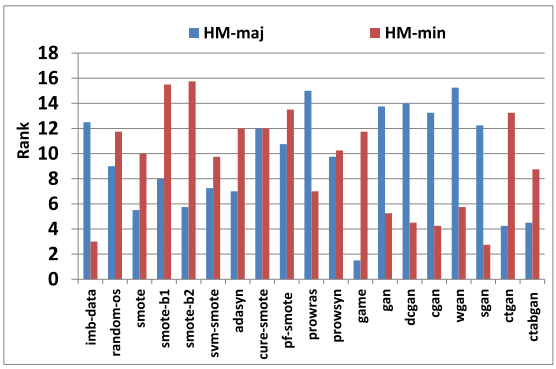
Dataset	original-maj:min	test-maj:min	train-maj:min
HB	255:81	67:67	65:13(5:1)
BCW1	444:239	200:200	200:38(5:1)
BCW2			200:20(10:1)
BCW3			210:14(15:1)
BCW4			244:12(20:1)
PID1	500:268	220:220	220:44(5:1)
PID2			240:24(10:1)
PID3		240:240	240:16(15:1)
PID4			240:12(20:1)
GM1	700:300	250:250	250:50(5:1)
GM2			250:25(10:1)
GM3			225:15(15:1)
GM4			260:13(20:1)
DCCC1	23 364:6636	5500:5500	5680:1136(5:1)
DCCC2		6036:6036	6000:600(10:1)
DCCC3		6222:6222	6210:414(15:1)
DCCC4		6320:6320	6320:316(20:1)
DCCC5		6506:6506	6500:130(50:1)
DCCC6		6571:6571	6500:65(100:1)
DCCC7		6623:6623	6500:13(500:1)
Shuttle1	45 586:12 414	10 345:10 345	10 345:2069(5:1)
Shuttle2		11 268:11 268	11 280:1128(10:1)
Shuttle3		11 639:11 639	11 625:775(15:1)
Shuttle4		11 823:11 823	11 820:591(20:1)
Shuttle5		12 171:12 171	12 150:243(50:1)
Shuttle6		12 292:12 292	12 200:122(100:1)
Shuttle7		12 390:12 390	12 000:24(500:1)
Shuttle8		12 402:12 402	12 000:12(1000:1)

5. Experiment results and discussions

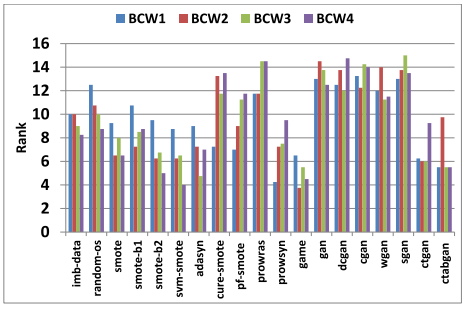
5.1. Datasets and experimental settings

We use 28 tabular datasets with various imbalanced ratios designed from six UCI real-world datasets of different domains to examine our ideas. Haberman's Survival (HM) dataset contains cases on the survival of breast cancer patients with 306 instances and 3 attributes. Breast Cancer Wisconsin Original (BCW) dataset records the clinical cases of Dr. Wolberg with 699 instances and 10 attributes, from which we remove the missing values in our experiment. The Pima Indians Diabetes (PID) dataset draws from the National Institute of Diabetes and Digestive and Kidney Diseases with 768 instances and 8 attributes, where the patients are all female Pima Indian heritage. German Credit Data (GM) classifies people described by a set of attributes as good or bad credit risks with 1000 instances and 20 attributes. Default of Credit Card Clients (DCCC) data set describes the cases of customers' default payments in Taiwan with 30,000 instances and 24 attributes. Statlog Dataset (Shuttle) collects shuttle logs with 58,000 instances and 9 attributes.

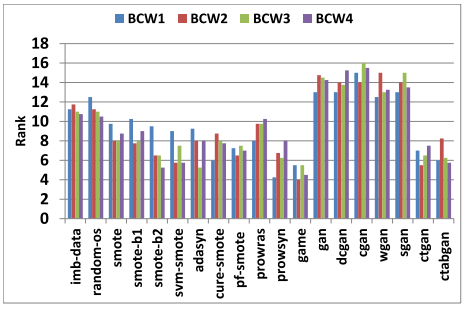
To fully compare the data generated by the algorithms, we design the datasets based on three principles. The first is to maximize the utilization of minority samples, which means to keep minority samples as many as possible when constructing different datasets. The second is to use a balanced dataset as the testing set, which is used to maximize the effect of distinguishing between majority and minority class tests. And the last is to keep the same number of majority samples in both training and testing sets, which is to minimize the problems caused by the differences in the amount of majority samples. From the perspective of the imbalanced ratio, our manuscript does not target a specific ratio but rather aims to accommodate as many ratios as possible. In this paper, we construct datasets with imbalanced ratios starting from 5:1



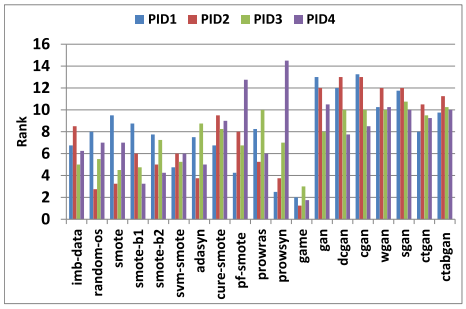
(a) Average Ranks in HM Majority & Minority



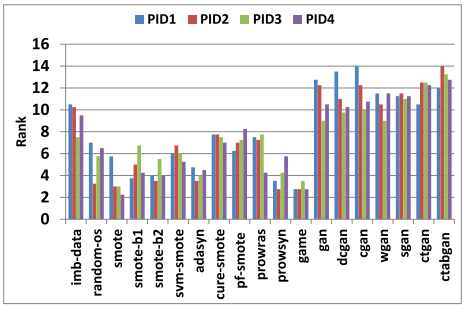
(b) Average Ranks of BCW Majority



(c) Average Ranks of BCW Minority



(d) Average Ranks of PID Majority



(e) Average Ranks of PID Minority

Fig. 4. F1 scores average ranks of different algorithms by datasets.

to 1000:1 as long as the total number of minority class samples is more than 10, so as to verify the actual effect of each algorithm. The specific description is shown in Table 1.

In order to measure the adaptability of the synthetic data generated by the algorithms to classifiers, we choose four classifiers of different design principles, Decision Tree(DT), Logistic Regression (LR), Support Vector Machine (SVM), and Naive Bayes (NB).

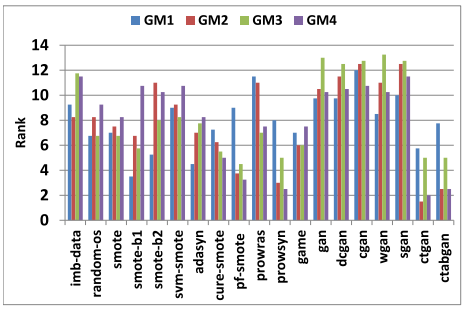
We select 18 methods as the baselines which are directly using the original imbalanced data (imb-data), random over-sampling (random-os), SMOTE, SMOTE-borderline 1 (Somte-b1), SMOTE-borderline 2 (Smote-b2), SMOTE-SVM (Smote-SVM), ADASYN, polynom-fit-SMOTE ($pf-Smote$), CURE-SMOTE, ProWSyn, ProWRAS, GAN, DCGAN, CGAN, WGAN, SGAN, CTGAN, and CTABGAN. In evaluation, we calculate the F1 scores of the majority and the minority separately, to compare the balanced effects of the algorithms for imbalanced data. For the convenience of statistics, the experimental results are the average values of 50 cross-validations for each dataset. The standard deviations of all 19 algorithms are measured in 10^{-2} and retained to 2 decimal places.

The parameters in our algorithm are set as follows. The maximum number of iterations $Num_{Max_{iter}}$ is set to 50. The k is set to 5 in the nearest neighbors of each minority sample, which contains at least 1 known sample from the majority samples. The structure of the neural network is not used in this method and there is no need to calculate the overall hyperplane, so no global hyper parameters are needed. The local data generation complements the overall data shortage without calculating the overall classification hyperplane. A large number of hyper parameters calculations in iteration are avoided, and computational resources are saved.

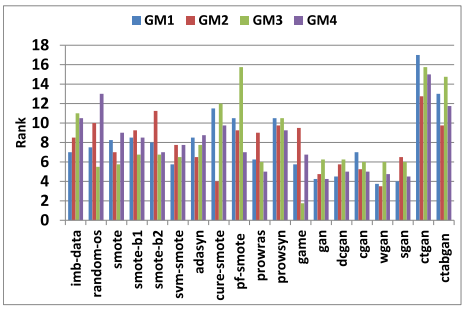
5.2. Analysis and discussion

The detailed experimental results are shown in Table 3 to Table 8. For the convenience of aggregating results in various datasets with different classifiers, we sorted out the ordinal ranking of each f1 score in Table 3 to Table 8 as the average rank, which is the average ranking position of the F1 scores from the highest to the lowest in all the result tables. The ranking details for all datasets are shown in Figs. 4a to 4k, and the final summary results are summarized in Table 2. We analyze the results and summarize the advantages of our algorithm below.

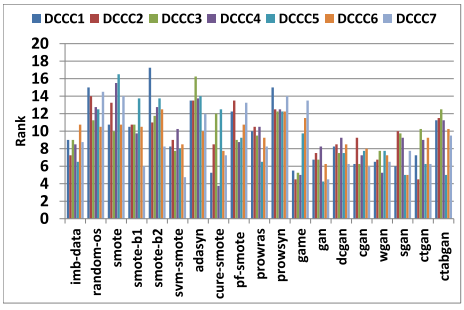
In this experiment, the minority samples have been generated in the enlarged space. The majority samples keep the original. If the enlargement space of the generated minority samples is too large and takes up the actual distribution space of the majority samples, the F1 Score of the majority samples will be affected. If the generating space



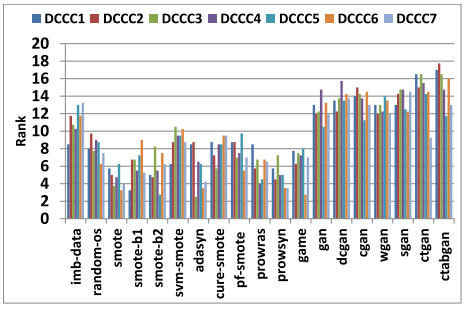
(f) Average Ranks of GM Majority



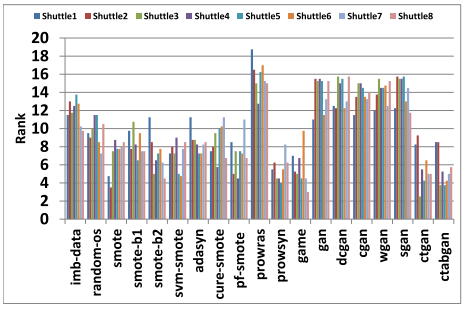
(g) Average Ranks of GM Minority



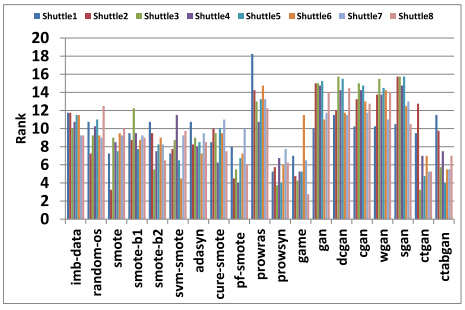
(h) Average Ranks of DCCC Majority



(i) Average Ranks of DCCC Minority



(j) Average Ranks of Shuttle Majority



(k) Average Ranks of Shuttle Minority

Fig. 4. (*continued*).

of the minority samples is not big enough, the F1 Score of the minority samples will be low. Therefore, while doing the whole data experiment, we separately count the F1 Scores of the majority and the minority samples, as Figs. 4a to 4k. From the Figures, almost every algorithm has a time when it performs optimally on a particular dataset, such as sgan in HM-min, svm-smote in BCW4-maj, smote in PID4-min, ctgan in GM2-maj, gan in DCCC5-maj and pf-smote in shuttle4-min. From Table 2a, it can be seen that our GAME algorithm has the best F1 scores in the majority samples. From Table 2b, the SMOTEs do well in DCCCs but get bad performance in HM, while the results are just the opposite for the GANs, where GAME keeps its stability in all the datasets. In Table 2c, the total average ranks of all the algorithms are listed, from which we can conclude that the GAME algorithm wins the most stable advantage ranking. From all 28 datasets and 4 classifiers, though not always perform best, the GAME method achieves the best-balanced effects for imbalanced data problems.

Then, the results should be discussed by imbalanced ratio and the amount of training data, as in Tables 3 to 8. The imbalanced ratios are considered to be high and low, where the boundary is 15:1. According to the amount, the datasets are divided into two groups here. The small group contains HM, BCWs, PIDs, and GMs, 13 datasets in all. The big group includes 15 datasets, DCCCs, and Shuttles. In the small group with low imbalanced ratios, the SMOTEs show good results, due to the limited data-generating space. In the big group with high imbalanced ratios, the imb-data method and random-os approach gain their advantages, which benefit from a sufficient sample number. As for GANs, they play a role in some low imbalanced ratios situations because of enough given data points. Our GAME method is stable in all these cases.

Also there is a limitation of the method. It is that for problems with low imbalanced ratios and a large number of known minority samples, the results should not have a significant advantage. Theoretically, the amount of known samples in such problems is large and representative, and the sample space enlargement has a certain effect but with limitations. For the problems with high imbalanced ratios and a few known minority samples, the effect will be better.

After that, it comes to the adaptation to classifiers. To the four classifiers, all the algorithms perform stably on DT and LR but fluctuate on SVM and NB. As shown in Tables 5 to 7, the SVM is relatively weak in classification effects dealing with the PIDs, GMs, and DCCCs, which is irrelevant to the algorithm and the imbalanced ratio. In other words, the four classifiers are more sensitive to the distribution of the dataset itself rather than the algorithm principles, which is suitable to judge the quality of the data generated by the algorithms.

At last, from the perspective of executive efficiency, because of the local linear perspective in GAME, the number of iterations is set as 50, within which the DNN model is difficult to converge. For the GANs, the epochs parameter depends on the amount of training data,

Table 2
Average rank in all datasets.

(a) majority average rank

Maj-Rank	HM	BCW	PID	GM	DCCC	Shuttle	Average Rank
imb-data	12.5	9.3125	6.625	10.1875	8.535714	11.90625	9.844494
random-os	9	10.5	5.8125	7.75	12.92857	9.71875	9.28497
smote	5.5	7.5625	6.0625	7.375	12.96429	7.0625	7.754464
smote-b1	8	8.8125	5.6875	6.6875	10.28571	8.4375	7.985119
smote-b2	5.75	6.875	6.0625	8.625	12.46429	7.125	7.816964
svm-smote	7.25	6.375	5.5	9.3125	8.071429	7.1875	7.282738
adasyn	7	7	6.25	6.875	13.28571	8.53125	8.156994
cure-smote	12	11.4375	8.375	6	8.142857	8.625	9.096726
pf-smote	10.75	9.75	7.9375	5.125	10.96429	7.25	8.629464
prowras	15	13.125	7.375	9.25	9.214286	15.8125	11.62946
prowsyn	9.75	7.125	6.9375	4.625	12.96429	**5.59375**	7.832589
game	**1.5**	**5.0625**	**2**	6.625	7.857143	5.71875	**4.7939**
gan	13.75	13.4375	10.875	10.875	**6.32143**	14.0625	11.55357
dcgan	14	13.25	10.6875	11.0625	7.964286	14	11.82738
cgan	13.25	13.4375	11.1875	12	7.25	13.78125	11.81771
wgan	15.25	12.1875	10.625	10.75	6.821429	14.09375	11.62128
sgan	12.25	13.8125	11.125	11.6875	7.535714	14.25	11.77679
ctgan	4.25	6.875	9.3125	**3.5625**	7.535714	5.78125	6.219494
ctabgan	4.5	6.5625	10.3125	4.4375	10.17857	5.59375	6.930804

(b) minority average rank

Min-Rank	HM	BCW	PID	GM	DCCC	Shuttle	Average Rank
imb-data	3	11.1875	9.4375	9.25	11.32143	10.71875	9.15253
random-os	11.75	11.3125	5.625	9	8.142857	9.90625	9.289435
smote	10	8.625	3.5	7.5	**4.67857**	8.03125	7.055804
smote-b1	15.5	8.75	4.9375	8.25	6.25	9.34375	8.838542
smote-b2	15.75	6.9375	4.25	8.25	5.714286	8.15625	8.176339
svm-smote	9.75	7	6	6.9375	9.071429	8.15625	7.819196
adasyn	12	7.625	4.1875	7.875	5.75	8.71875	7.692708
cure-smote	12	7.625	7.5	9.3125	8.25	9.03125	8.953125
pf-smote	13.5	7.0625	7.1875	10.625	7.75	6.5	8.770833
prowras	7	9.4375	6.6875	6.5625	6.107143	13.71875	8.252232
prowsyn	10.25	6.3125	4.0625	10	4.928571	**5.6875**	6.873512
game	11.75	**4.875**	**2.9375**	5.9375	6.642857	5.90625	**6.34152**
gan	5.25	14.125	11.125	4.875	12.53571	13.34375	10.20908
dcgan	4.5	14	11.125	5.375	13.82143	13.34375	10.36086
cgan	4.25	15.125	11.75	5.8125	13.67857	13.125	10.62351
wgan	5.75	13.4375	10.625	**4.5**	12.82143	13.375	10.08482
sgan	**2.75**	13.875	11.25	5.25	13.71429	13.5625	10.06696
ctgan	13.25	6.625	11.9375	15.125	14.5	6.84375	11.38021
ctabgan	8.75	6.5625	13	12.3125	15.25	7.0625	10.48958

(c) all average rank

Rank	HM	BCW	PID	GM	DCCC	Shuttle	Average Rank
imb-data	7.75	10.25	8.03125	9.71875	9.928571	11.3125	9.498512
random-os	10.375	10.90625	5.71875	8.375	10.51786	9.8125	9.284226
smote	7.75	8.09375	4.78125	7.4375	8.803571	7.546875	7.402158
smote-b1	11.75	8.78125	5.3125	7.46875	8.25	8.890625	8.408854
smote-b2	10.75	6.90625	5.15625	8.4375	9.071429	7.640625	7.993676
svm-smote	8.5	6.6875	5.75	8.125	8.553571	7.671875	7.547991
adasyn	9.5	7.3125	5.21875	7.375	9.5	8.625	7.921875
cure-smote	12	9.53125	7.9375	7.65625	8.357143	8.828125	9.051711
pf-smote	12.125	8.40625	7.5625	7.875	9.339286	6.875	8.697173
prowras	11	11.28125	7.03125	7.90625	7.642857	14.76563	9.937872
prowsyn	10	6.71875	5.5	7.3125	8.946429	**5.64063**	7.353051
game	**6.625**	**4.96875**	**2.46875**	**6.28125**	**7.23214**	5.8125	**5.56473**
gan	9.5	13.78125	11	7.875	9.428571	13.70313	10.88132
dcgan	9.25	13.625	10.90625	8.21875	10.89286	13.67188	11.09412
cgan	8.75	14.28125	11.46875	8.90625	10.46429	13.45313	11.22061
wgan	10.5	12.8125	10.625	7.625	9.821429	13.73438	10.85305
sgan	7.5	13.84375	11.1875	8.46875	10.625	13.90625	10.92188
ctgan	8.75	6.75	10.625	9.34375	11.01786	6.3125	8.799851
ctabgan	**6.625**	6.5625	11.65625	8.375	12.71429	6.328125	8.710193

which is 5000 in the small group and 30,000 in the big group. The time consumptions are shown in the last column of each subtable of Tables 3 to 8. With the same hardware configuration, the SMOTE series completed data generation in one second, while the GANs take minutes even hours. Our GAME needs seconds to minutes, which is slower than SMOTEs but much faster than GANs.

In summary, we achieve more effective and more balanced results than the 18 mainstream algorithms.

6. Conclusion

In this paper, we propose a Generative Adversarial Minority Enlargement model to extend the data generating space. Based on the local linear property of samples, our GAME utilizes the adjustment of the generated samples and the parameters of the classification line, gradually approaching the majority category, enlarging the generating space, and getting a wide range of synthetic data. The target of our

Table 3
Classification performance on dataset HM.

HM ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	51.85 ± 1.57	67.5 ± 1.6	23.38 ± 0	69.11 ± 0	5.8 ± 0	**67.34 ± 0**	27.85 ± 0	69.84 ± 0	0:00:00
random-os	46.46 ± 2.53	68.64 ± 0.82	59.26 ± 0	58.65 ± 0	8.33 ± 0	66.33 ± 0	61.64 ± 0	54.1 ± 0	0:00:00.002000
smote	53.7 ± 4.58	68.75 ± 0.79	58.91 ± 2.16	61.87 ± 1.07	13.33 ± 2.21	66.32 ± 0.69	60.47 ± 1.86	63.31 ± 0.35	0:00:00.001999
smote-b1	50.91 ± 3.1	65.82 ± 1.53	58.46 ± 2.58	60.87 ± 1.69	15.38 ± 6.64	65.26 ± 1.33	46.03 ± 2.56	52.11 ± 1.03	0:00:00.027995
smote-b2	62.4 ± 3.56	67.13 ± 2.39	58.21 ± 2.9	58.21 ± 1.29	21.18 ± 6.4	63.39 ± 1.21	44.8 ± 1.96	51.75 ± 0.67	0:00:00.003999
svm-smote	56.41 ± 4.44	66.23 ± 2.03	56.9 ± 1.91	67.11 ± 0.66	10.67 ± 1.56	65.28 ± 0.54	52 ± 2.47	**71.43 ± 0.59**	0:00:00.005997
adasyn	52.83 ± 4.92	69.14 ± 3.04	57.81 ± 1.43	61.43 ± 1.42	15.19 ± 2.83	64.55 ± 0.64	54.26 ± 0.87	57.55 ± 0.23	0:00:00.003999
cure-smote	47.71 ± 4.8	64.15 ± 2.01	57.36 ± 2.87	60.43 ± 3.65	0 ± 0	65.33 ± 0.45	56.36 ± 2.23	69.62 ± 3.63	0:00:00.005997
pf-smote	51.33 ± 2.55	64.52 ± 1.21	56.14 ± 0	67.53 ± 0	10.39 ± 0	63.87 ± 0	45.28 ± 0	64.2 ± 0	0:00:00.003995
prowras	50 ± 4.2	68.29 ± 2.35	32.18 ± 3.15	67.4 ± 0.6	0 ± 0	66.67 ± 0	38.3 ± 0	66.67 ± 0	0:00:00.665222
prowsyn	52.99 ± 2.83	63.58 ± 2.65	52.34 ± 2.54	68.32 ± 0.84	13.33 ± 3.14	66.32 ± 1.07	41.24 ± 0.77	66.67 ± 0.62	0:00:00.012994
game	**65.6 ± 3.76**	**69.93 ± 1.9**	**60.43 ± 1.66**	57.36 ± 1.67	27.91 ± 3.92	65.93 ± 0.87	**62.32 ± 0.57**	60 ± 0.14	0:00:06.557920
gan	51.38 ± 1.53	66.67 ± 0.92	21.05 ± 0	68.75 ± 0	5.8 ± 0	67.34 ± 0	0 ± 0	66.67 ± 0	0:01:49.148332
dcgan	50.47 ± 1.84	67.08 ± 1.6	23.38 ± 0	69.11 ± 0	5.8 ± 0	67.34 ± 0	0 ± 0	66.67 ± 0.15	0:03:50.288416
cgan	51.85 ± 1.83	67.5 ± 1.26	21.05 ± 0	68.75 ± 0	5.8 ± 0	67.34 ± 0	0 ± 0	66.67 ± 0	0:02:33.768894
wgan	47.62 ± 2.05	66.26 ± 0.64	21.05 ± 0	68.75 ± 0	5.8 ± 0	67.34 ± 0	0 ± 0	66.67 ± 0.18	0:20:11.754763
sgan	55.56 ± 1.77	70 ± 1.29	21.05 ± 0	68.75 ± 0	5.8 ± 0	67.34 ± 0	0 ± 0	66.67 ± 0	0:04:59.387274
ctgan	56.2 ± 2.79	63.95 ± 2.06	59.68 ± 2.63	65.28 ± 3.24	**63.41 ± 1.15**	42.31 ± 3.86	46 ± 1.98	67.86 ± 0.38	0:02:47.407843
ctabgan	59.5 ± 2.43	66.67 ± 2.33	60.34 ± 1.56	**69.74 ± 2.14**	60 ± 16.64	49.15 ± 6.03	44 ± 1.79	66.67 ± 0.54	0:13:21.084600

Table 4
Classification performance on dataset BCW.

(a) BCW1

BCW1 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	93.62 ± 1.61	92.84 ± 1.2	95.15 ± 0	94.85 ± 0	66.78 ± 0	1 ± 0	96.22 ± 0	96.28 ± 0	0:00:00
random-os	88.44 ± 0.39	85.79 ± 0.27	95.59 ± 0	95.41 ± 0	66.78 ± 0	1 ± 0	96.2 ± 0	96.3 ± 0	0:00:00.002001
smote	90.32 ± 1.24	88.52 ± 0.85	95.59 ± 0.26	95.41 ± 0.24	89.5 ± 0.45	91.32 ± 0.67	96.22 ± 0	96.28 ± 0	0:00:00.002006
smote-b1	89.61 ± 0.74	87.74 ± 0.55	95.35 ± 0.14	95.14 ± 0.14	89.5 ± 0.19	91.32 ± 0.27	96.41 ± 0	96.59 ± 0	0:00:00.004986
smote-b2	88.44 ± 1.52	85.79 ± 0.96	96.48 ± 0.18	96.52 ± 0.17	90.41 ± 0.32	91.95 ± 0.45	95.88 ± 0	96.12 ± 0	0:00:00.003999
svm-smote	90.11 ± 1.66	88.22 ± 1.25	96.06 ± 0.13	95.94 ± 0.14	89.5 ± 0.09	91.32 ± 0.14	96.45 ± 0.13	96.55 ± 0.14	0:00:00.007997
adasyn	88.14 ± 1.89	84.99 ± 1.32	95.59 ± 0	95.41 ± 0	90.41 ± 0	91.95 ± 0	96.94 ± 0	97.06 ± 0	0:00:00.004001
cure-smote	89.37 ± 1.73	90.99 ± 1.23	95.38 ± 1.02	95.61 ± 0.93	**93.9 ± 0.53**	**93.05 ± 0.67**	97.54 ± 0.14	97.46 ± 0.14	0:00:00.008976
pf-smote	89.32 ± 1.32	91.03 ± 0.96	96.98 ± 0	97.01 ± 0	90.29 ± 0	87.96 ± 0	97.28 ± 0	97.22 ± 0	0:00:00.002992
prowras	92.59 ± 2.06	93.36 ± 1.73	94.85 ± 0.11	95.15 ± 0.11	1 ± 0	66.78 ± 0	97.26 ± 0	97.24 ± 0	0:00:01.143941
prowsyn	92.39 ± 3.37	93.08 ± 2.49	97.5 ± 0.22	97.5 ± 0.22	90.09 ± 0.11	87.64 ± 0.17	**97.79 ± 0.11**	**97.71 ± 0.11**	0:00:00.017953
game	93.69 ± 1.18	93.3 ± 0.96	96.98 ± 0.29	97.01 ± 0.3	88.58 ± 0.11	90.7 ± 0.19	96.41 ± 0	96.59 ± 0	0:00:09.012948
gan	94.48 ± 2.51	93.99 ± 1.94	83.93 ± 0.66	82.51 ± 0.46	66.56 ± 0	0.99 ± 0	95.58 ± 0	95.9 ± 0	0:01:28.576170
dcgan	93.14 ± 1.11	92.31 ± 0.8	84.49 ± 0.83	82.94 ± 0.45	66.78 ± 0.01	1 ± 0.25	95.58 ± 0.11	95.9 ± 0.12	0:02:33.678026
cgan	92.92 ± 1.97	92.02 ± 1.49	82.99 ± 1.01	79.11 ± 0.88	66.78 ± 0	1 ± 0	95.63 ± 0.11	95.86 ± 0.12	0:02:01.332138
wgan	93.4 ± 1.92	92.55 ± 1.44	84.82 ± 0.88	83.64 ± 0.96	66.78 ± 0	1 ± 0.12	95.58 ± 0.11	95.9 ± 0.12	0:16:10.522816
sgan	**95.17 ± 1.64**	**94.82 ± 1.26**	83.37 ± 1.23	82.08 ± 0.48	66.56 ± 0	0.99 ± 0.12	95.58 ± 0.11	95.9 ± 0.12	0:03:33.334966
ctgan	90.29 ± 1.29	91.17 ± 1.14	96.22 ± 0.52	96.28 ± 0.48	89.69 ± 0	87.01 ± 0	97.54 ± 0	97.46 ± 0	0:01:27.705598
ctabgan	89.89 ± 2.07	91.04 ± 1.7	**98 ± 0.23**	**98 ± 0.22**	89.69 ± 0	87.01 ± 0	97.54 ± 0	97.46 ± 0	0:06:09.132454

(b) BCW2

BCW2 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	86.84 ± 0	82.56 ± 0	88.05 ± 0	84.48 ± 0	66.67 ± 0	0 ± 0	96.73 ± 0	96.77 ± 0	0:00:00
random-os	81.39 ± 1.43	70.74 ± 0.87	91.71 ± 0	90.16 ± 0	66.67 ± 0	0 ± 0	96.73 ± 0	96.77 ± 0	0:00:00.000984
smote	89.19 ± 1.81	86.52 ± 1.13	91.28 ± 0.25	89.56 ± 0.17	92.47 ± 0.24	93.46 ± 0.32	96.48 ± 0.14	96.52 ± 0.13	0:00:00.002998
smote-b1	82.64 ± 2.01	73.42 ± 1.37	91.49 ± 0.17	89.86 ± 0.11	92.47 ± 0.39	93.46 ± 0.51	96.97 ± 0	97.03 ± 0	0:00:00.003977
smote-b2	84.32 ± 1.5	77.44 ± 0.86	92.99 ± 1.83	91.94 ± 1.32	92.18 ± 0.89	93.24 ± 1.15	96.97 ± 0.1	97.03 ± 0.12	0:00:00.004000
svm-smote	83.37 ± 1.19	75.69 ± 0.78	92.96 ± 0.25	91.98 ± 0.18	92.47 ± 0.1	93.46 ± 0.13	96.97 ± 0.13	97.03 ± 0.15	0:00:00.006991
adasyn	83.19 ± 0.81	75.31 ± 0.57	90.87 ± 0.22	88.95 ± 0.15	93.05 ± 0.34	93.9 ± 0.43	96.97 ± 0.11	97.03 ± 0.12	0:00:00.003997
cure-smote	84.9 ± 0.15	88.2 ± 0.11	88.46 ± 0.28	90.37 ± 0.23	1 ± 0	66.78 ± 0	96.26 ± 0.14	96.24 ± 0.13	0:00:00.004987
pf-smote	86.83 ± 0	89.39 ± 0	87.91 ± 0	89.91 ± 0	**95.01 ± 0**	**94.46 ± 0**	96.26 ± 0	96.24 ± 0	0:00:00.002992
prowras	89.43 ± 2.18	90.95 ± 1.64	83.52 ± 0.78	87.05 ± 0.51	0 ± 0	66.67 ± 0	96.5 ± 0.01	96.5 ± 0.01	0:00:00.013963
prowsyn	83.38 ± 2	86.74 ± 1.36	88.22 ± 0.25	90.11 ± 0.17	92.59 ± 0.36	91.3 ± 0.5	**97.06 ± 0.11**	96.94 ± 0.11	0:00:01.576786
game	90 ± 1.47	87.78 ± 0.97	**94.06 ± 0.74**	**93.4 ± 0.63**	88.27 ± 0	90.5 ± 0	96.97 ± 0.12	**97.03 ± 0.13**	0:00:11.756443
gan	87.22 ± 0	83.24 ± 0	75.72 ± 1.56	62.42 ± 0.74	66 ± 0	0 ± 0	95.12 ± 0.13	95.38 ± 0.14	0:01:52.235464
dcgan	88.79 ± 0	85.88 ± 0	74.74 ± 2.73	60.7 ± 2.28	66 ± 0	0 ± 0	95.36 ± 0.12	95.63 ± 0.1	0:03:58.365520
cgan	87.42 ± 0.32	83.57 ± 0.34	76.53 ± 3.55	59.39 ± 2.41	66.67 ± 0	0 ± 0	96.22 ± 0.14	96.28 ± 0.13	0:02:49.082396
wgan	85.9 ± 0	80.83 ± 0	81.78 ± 2.56	73.78 ± 1.8	66.67 ± 0	0 ± 0	95.12 ± 0	95.38 ± 0	0:23:15.030298
sgan	87.61 ± 0	83.91 ± 0	75.46 ± 3.17	62.22 ± 2.54	66 ± 0	0 ± 0	95.36 ± 0.02	95.63 ± 0.02	0:05:22.116722
ctgan	**91.01 ± 0.4**	**91.94 ± 0.34**	90.91 ± 1.52	92.02 ± 1.13	90.5 ± 0	88.27 ± 0	96.81 ± 0.23	96.69 ± 0.22	0:01:45.660612
ctabgan	86.03 ± 1.6	88.69 ± 1.09	86.98 ± 0.86	89.29 ± 0.57	90.91 ± 0	88.89 ± 0	96.81 ± 0.25	96.69 ± 0.24	0:06:52.770825

(c) BCW3

BCW3 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	84.62 ± 0	78.31 ± 0	88.2 ± 0	84.9 ± 0	66.67 ± 0	0 ± 0	96.95 ± 0	97.04 ± 0	0:00:00
random-os	79.68 ± 0.9	65.77 ± 0.46	91.59 ± 0	90.32 ± 0	66.67 ± 0	0 ± 0	96.95 ± 0	97.04 ± 0	0:00:00.001999
smote	81.93 ± 1.75	73.46 ± 1.16	91.42 ± 0	89.97 ± 0	90.71 ± 0.09	92.17 ± 0.13	96.95 ± 0.11	97.04 ± 0.11	0:00:00.002991

(*continued on next page*)

Table 4 (*continued*).

smote-b1	80 ± 1.7	66.67 ± 1.04	91.42 ± 0	89.97 ± 0	90.71 ± 0	92.17 ± 0	**97.2 ± 0**	**97.3 ± 0**	0:00:00.004997
smote-b2	82.5 ± 1.89	73.75 ± 1.32	94.87 ± 0.29	94.63 ± 0.26	90.71 ± 41.13	92.17 ± 11.77	96.68 ± 0.26	96.82 ± 0.32	0:00:00.006998
svm-smote	80.81 ± 1.24	68.85 ± 0.78	92.74 ± 0.37	91.69 ± 0.29	90.71 ± 0.37	92.17 ± 0.51	96.95 ± 0.13	97.04 ± 0.15	0:00:00.007998
adasyn	87.5 ± 1.42	84.09 ± 1	91.63 ± 0.26	90.27 ± 0.2	**92.18 ± 0**	**93.24 ± 0**	96.94 ± 0	97.06 ± 0	0:00:00.003995
cure-smote	84.09 ± 1.61	87.5 ± 0.91	91.78 ± 0	92.67 ± 0	0 ± 0	66.67 ± 0	96.04 ± 0.11	95.96 ± 0.12	0:00:00.006981
pf-smote	85.87 ± 2	88.38 ± 1.21	91.78 ± 0	92.67 ± 0	0 ± 0	66.67 ± 0	96.02 ± 0	95.98 ± 0	0:00:00.003989
prowras	81.66 ± 3.08	86.58 ± 1.5	80.36 ± 0.63	85.78 ± 0.33	0 ± 0	66.67 ± 0	96.3 ± 0.2	96.2 ± 0.22	0:00:01.879975
prowsyn	84.09 ± 0.92	87.5 ± 0.68	91.78 ± 0.16	92.67 ± 0.12	91.95 ± 0.09	90.41 ± 0.13	96.3 ± 0.18	96.2 ± 0.17	0:00:00.019946
game	88.69 ± 2.98	86.03 ± 2.06	**95.76 ± 0.39**	**95.74 ± 0.36**	88.89 ± 0	90.91 ± 0	96.41 ± 0.19	96.59 ± 0.22	0:00:09.276062
gan	86.65 ± 0	82.22 ± 0	70.28 ± 2.31	40 ± 1.16	66.22 ± 0	0 ± 0	94.33 ± 0.12	94.66 ± 0.15	0:01:57.744720
dcgan	86.46 ± 0	81.87 ± 0	71.56 ± 1.14	43.87 ± 0.67	66.67 ± 0	0 ± 0	94.87 ± 0	95.12 ± 0	0:03:55.276788
cgan	85.53 ± 0	80.12 ± 0	69.76 ± 2.63	37.55 ± 0.9	66.67 ± 0	0 ± 0	94.24 ± 0.09	94.26 ± 0.14	0:02:41.664425
wgan	87.22 ± 0	83.24 ± 0	76.86 ± 4.47	62.05 ± 3.35	66.67 ± 0	0 ± 0	94.87 ± 0.29	95.12 ± 0.39	0:21:49.504841
sgan	86.09 ± 0	81.18 ± 0	69.81 ± 1.6	40.74 ± 2.33	66.22 ± 0	0 ± 0	94.06 ± 0.31	94.43 ± 0.41	0:05:09.896768
ctgan	89.78 ± 1.75	91.12 ± 1.17	92.35 ± 0.5	93.11 ± 0.42	89.89 ± 0	87.32 ± 0	96.31 ± 0.11	96.18 ± 0.12	0:01:39.769356
ctabgan	**91.53 ± 0.76**	**92.42 ± 0.63**	91.49 ± 1.36	92.45 ± 1.1	90.91 ± 0.23	88.89 ± 0.35	96.57 ± 0.11	96.43 ± 0.12	0:06:44.533840

(d) BCW4

BCW4 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	87.17 ± 0.59	83.33 ± 0.3	86.59 ± 0	82.32 ± 0	66.67 ± 0	0 ± 0	97.23 ± 0	97.27 ± 0	0:00:00
random-os	82.33 ± 1.31	73.35 ± 0.49	88.99 ± 0	86.81 ± 0	66.67 ± 0	0 ± 0	97.46 ± 0	97.54 ± 0	0:00:00.002999
smote	81.84 ± 1.24	72.9 ± 0.49	89.45 ± 0.16	87.36 ± 0.12	91.6 ± 0.24	92.81 ± 0.31	97.23 ± 0	97.27 ± 0	0:00:00.033989
smote-b1	83.05 ± 0.82	75.61 ± 0.31	88.84 ± 0.39	86.43 ± 0.27	90.41 ± 0	91.95 ± 0	97.22 ± 0.21	97.28 ± 0.24	0:00:00.003998
smote-b2	83.76 ± 6.04	77.11 ± 2.91	**92.75 ± 1.56**	**92.23 ± 1.17**	91.6 ± 0.39	92.81 ± 0.52	97.22 ± 0.21	97.28 ± 0.21	0:00:00.004999
svm-smote	88.44 ± 3.12	85.14 ± 1.68	89.91 ± 0.16	87.91 ± 0.12	91.6 ± 0.24	92.81 ± 0.31	96.97 ± 0.11	97.03 ± 0.12	0:00:00.007006
adasyn	80.82 ± 8.2	70.48 ± 4.01	89.45 ± 0.41	87.36 ± 0.28	91.89 ± 0.24	**93.02 ± 0.32**	97.22 ± 0	97.28 ± 0	0:00:00.003992
cure-smote	79.17 ± 0	84.91 ± 0	89.32 ± 0.14	91.03 ± 0.09	0 ± 0	66.67 ± 0	95.96 ± 0	96.04 ± 0	0:00:00.008974
pf-smote	77.95 ± 1.03	84.43 ± 0.52	89.62 ± 0	91.24 ± 0	0 ± 0	66.67 ± 0	96.77 ± 0	96.73 ± 0	0:00:00.002992
prowras	82.9 ± 2.44	87.03 ± 1.46	78.31 ± 0.2	84.62 ± 0.1	0 ± 0	66.67 ± 0	95.41 ± 0.22	95.59 ± 0.2	0:00:01.898924
prowsyn	69.03 ± 1.24	80.41 ± 0.49	89.32 ± 0.27	91.03 ± 0.19	**93.02 ± 0.1**	91.89 ± 0.13	96.53 ± 0.11	96.46 ± 0.12	0:00:00.012965
game	**89.7 ± 3.26**	87.6 ± 2.24	89.36 ± 0.57	88.06 ± 0.5	87.64 ± 0	90.09 ± 0	**97.46 ± 0.07**	**97.54 ± 0.08**	0:00:07.866495
gan	85.22 ± 0.54	79.28 ± 0.28	77.75 ± 5.61	69.44 ± 0.74	66.67 ± 0	0 ± 0	95 ± 1.22	95 ± 0.86	0:02:00.592997
dcgan	85.22 ± 0.48	79.28 ± 0.25	76.46 ± 1.27	67.66 ± 0.47	66 ± 0	0 ± 0	94.71 ± 0.32	94.79 ± 0.2	0:03:35.725348
cgan	85.65 ± 0.48	80.59 ± 0.25	73.72 ± 2.21	48.71 ± 1.26	66.67 ± 0	0 ± 0	90.91 ± 2.98	89.49 ± 2.15	0:02:33.686637
wgan	87.36 ± 0.5	83.67 ± 0.26	77.68 ± 3.99	68.86 ± 1.02	66.67 ± 0	0 ± 0	95.26 ± 2.63	95.24 ± 1.77	0:23:13.340070
sgan	87.36 ± 0.59	83.67 ± 0.3	77.39 ± 10.51	69.41 ± 5.32	66 ± 0	0 ± 0	94.71 ± 1.19	94.79 ± 0.72	0:05:17.880592
ctgan	88.15 ± 1.49	90.16 ± 0.99	88.71 ± 2.38	90.62 ± 1.64	90.09 ± 0	87.64 ± 0	95.82 ± 0.11	95.67 ± 0.11	0:01:41.019011
ctabgan	89.37 ± 8	**90.99 ± 4.63**	89.62 ± 1.27	91.24 ± 0.93	90.7 ± 0.4	88.58 ± 0.6	96.08 ± 0.14	95.92 ± 0.13	0:06:18.765708

Table 5
Classification performance on dataset PID.

(a) PID1

PID1 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	70.09 ± 0.57	46.35 ± 1.57	71.6 ± 0	40.7 ± 0	66.67 ± 0	0 ± 0	71.82 ± 0	54.6 ± 0	0:00:00
random-os	66.91 ± 0.54	43.21 ± 1.81	72.73 ± 0	71.33 ± 0	66.67 ± 0	0 ± 0	73.25 ± 0	71.23 ± 0	0:00:00.002000
smote	69.16 ± 1.14	57.68 ± 2.03	69.53 ± 2.58	69.11 ± 2.75	66.67 ± 0.05	0 ± 0	70.59 ± 0.38	67.93 ± 1	0:00:00.002001
smote-b1	69.09 ± 1.17	60.26 ± 3.94	70.85 ± 1.84	70.05 ± 0.94	66.67 ± 0.06	0 ± 0	71.3 ± 1.27	68.57 ± 1.57	0:00:00.003999
smote-b2	67.56 ± 1.92	52.92 ± 4.81	73.73 ± 2.2	72.13 ± 2.24	66.67 ± 0.06	0 ± 0	73.01 ± 0.7	**71.5 ± 0.71**	0:00:00.003999
svm-smote	**71.4 ± 1.27**	58.5 ± 3.53	71.67 ± 2.12	66 ± 3.24	66.67 ± 0	0 ± 0	71.46 ± 1.31	64.63 ± 4.74	0:00:00.008999
adasyn	68.93 ± 0.43	52.72 ± 2.11	74.77 ± 1.53	**75.23 ± 1.18**	66.67 ± 0	0 ± 0	70.87 ± 0.4	68.1 ± 0.67	0:00:00.004000
cure-smote	69.33 ± 0.67	54.65 ± 4.77	**76.19 ± 0.69**	68.09 ± 1.5	**66.77 ± 0**	**0.9 ± 0**	68.01 ± 0.27	33.57 ± 1.47	0:00:00.006982
pf-smote	71.13 ± 0.89	57.7 ± 1.02	74.26 ± 0	69.95 ± 0	66.67 ± 0	0 ± 0	72.22 ± 0	55.88 ± 0	0:00:00.001995
prowras	70.1 ± 1.88	55.77 ± 3.61	68.5 ± 2.07	63.39 ± 2.55	66.67 ± 0	0 ± 0	71.21 ± 0.42	59.56 ± 0.48	0:00:01.884962
prowsyn	71.32 ± 0.49	59.34 ± 2.84	74.42 ± 1.82	70.27 ± 2.26	66.67 ± 0	0 ± 0	**75.73 ± 0.32**	71.14 ± 0.96	0:00:00.004987
game	71.15 ± 1.58	**60.96 ± 4.18**	74.77 ± 1.49	75.23 ± 1.39	**66.77 ± 0**	**0.9 ± 0**	73.6 ± 0.24	62.05 ± 0.5	0:00:16.344924
gan	68.56 ± 0.58	44.16 ± 0.81	66.67 ± 0	13.11 ± 0	66.67 ± 0	0 ± 0	65.74 ± 0.14	4.31 ± 0.37	0:01:34.067694
dcgan	69.75 ± 0.77	46.54 ± 1.78	65.53 ± 0.15	5.93 ± 1.01	66.67 ± 0	0 ± 0	66.26 ± 0.14	3.51 ± 0.44	0:03:06.831460
cgan	68.56 ± 1.06	44.16 ± 3.11	66.15 ± 1.37	9.21 ± 2.76	66.67 ± 0	0 ± 0	65.85 ± 0.58	3.48 ± 1.91	0:02:23.711914
wgan	70.34 ± 0.88	47.32 ± 2.92	66.77 ± 0.12	16.73 ± 1.12	66.67 ± 0	0 ± 0	65.74 ± 0.09	4.31 ± 0.02	0:22:27.501077
sgan	70.02 ± 1.09	48.3 ± 2.07	66.04 ± 0.16	9.96 ± 0.97	66.67 ± 0	0 ± 0	66.15 ± 0.05	7.63 ± 0.34	0:05:29.679758
ctgan	70.67 ± 0.61	47.13 ± 3.43	68.77 ± 1.33	19.51 ± 9.59	66.67 ± 0	0 ± 0	70.55 ± 1.15	41.89 ± 5.08	0:01:31.679975
ctabgan	70.27 ± 0.78	49.23 ± 4.23	66.67 ± 0.09	0 ± 0.81	66.67 ± 0	0 ± 0	67.18 ± 0.24	6.14 ± 3.04	0:06:31.667740

(b) PID2

PID2 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	67.89 ± 0.87	31.37 ± 2	69.66 ± 0	30.24 ± 0	66.67 ± 0	0 ± 0	72.18 ± 0	47.13 ± 0	0:00:00
random-os	69.58 ± 0.53	38.87 ± 0.9	**74.7 ± 0**	71.81 ± 0	66.67 ± 0	0 ± 0	74.9 ± 0	**70.59 ± 0**	0:00:00.001995
smote	70.49 ± 0.63	52.32 ± 3	74.07 ± 0.75	70.25 ± 1.66	66.67 ± 0	0 ± 0	72.21 ± 0.71	65.89 ± 0.84	0:00:00.002999
smote-b1	70.13 ± 1.47	51.1 ± 4.63	72.38 ± 1.5	66.67 ± 1.84	66.67 ± 0	0 ± 0	71.43 ± 0.59	64.49 ± 0.81	0:00:00.004999
smote-b2	68.93 ± 1.45	50.67 ± 5.64	72.55 ± 1.11	68.89 ± 0.86	66.67 ± 0	0 ± 0	73.97 ± 0.62	70.38 ± 1.02	0:00:00.005012
svm-smote	69.41 ± 1.89	47.16 ± 5.5	70.77 ± 0.28	54.4 ± 3.06	66.67 ± 0	0 ± 0	73.06 ± 0.52	63.39 ± 1.3	0:00:00.007998
adasyn	70.25 ± 1.55	51.51 ± 5.08	73.8 ± 0.53	68.65 ± 0.75	66.67 ± 0	0 ± 0	73.9 ± 0.55	68.51 ± 0.79	0:00:00.004020

(*continued on next page*)

Table 5 (*continued*).

cure-smote	67.93 ± 0.67	37.92 ± 2.54	69.19 ± 2.18	62.18 ± 3.29	66.67 ± 0	0 ± 0	68.71 ± 0.28	50.54 ± 3.47	0:00:00.005984
pf-smote	68.63 ± 0.16	36.08 ± 1.31	71.76 ± 0	68 ± 0	66.67 ± 0	0 ± 0	69.3 ± 0	52.52 ± 0	0:00:00.001995
prowras	70.41 ± 0.8	49.58 ± 3.35	72.92 ± 0.76	63.05 ± 1.01	66.67 ± 0	0 ± 0	70.53 ± 0.51	41.61 ± 1.28	0:00:02.333763
prowsyn	70.37 ± 1.18	51.91 ± 2.42	73.81 ± 0.54	71.05 ± 0.9	66.67 ± 0	0 ± 0	72.66 ± 0.83	67.28 ± 1.03	0:00:00.004987
game	**71.65 ± 1.36**	**54.99 ± 3.51**	74.15 ± 1.84	**72.02 ± 1.21**	66.67 ± 0	0 ± 0	**75.04 ± 0.19**	63.05 ± 0.61	0:00:10.693549
gan	68.1 ± 0.45	33.44 ± 2.07	65.43 ± 0.08	5.45 ± 0	66.67 ± 0	0 ± 0	65.25 ± 0.08	2.38 ± 0	0:01:24.922554
dcgan	67.29 ± 1.13	34.89 ± 2.85	65.34 ± 0.08	6.18 ± 0.35	66.67 ± 0	0 ± 0	65.35 ± 0.12	3.16 ± 0.56	0:02:31.980568
cgan	66.87 ± 1.13	31.85 ± 3.85	65.62 ± 0.87	6.95 ± 1.59	66.67 ± 0	0 ± 0	65.26 ± 0.45	0.8 ± 1.52	0:02:00.296316
wgan	67.9 ± 0.99	33.33 ± 2.6	65.62 ± 0.5	6.95 ± 1.6	66.67 ± 0	0 ± 0	65.25 ± 0.32	3.92 ± 1.31	0:16:25.825426
sgan	67.7 ± 1.11	33.23 ± 1.76	65.53 ± 0.04	6.2 ± 0.36	66.67 ± 0	0 ± 0	65.35 ± 0.16	3.16 ± 0	0:03:33.069685
ctgan	67.16 ± 1.73	24.14 ± 8.76	67.04 ± 0.21	3.28 ± 3.44	66.67 ± 0	0 ± 0	70.77 ± 0.82	38.71 ± 4.4	0:01:33.535017
ctabgan	67.36 ± 0.55	21.91 ± 2.61	66.67 ± 0.04	0 ± 0	66.67 ± 0	0 ± 0	66.95 ± 0.05	2.47 ± 0.45	0:06:28.220431

(c) PID3

PID3 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	69.77 ± 0.48	38.1 ± 1.03	68.29 ± 0	15.97 ± 0	66.67 ± 0	0 ± 0	69.8 ± 0	33.89 ± 0	0:00:00
random-os	68.98 ± 0.33	30.41 ± 0.68	67.15 ± 0	54.59 ± 0	66.67 ± 0	0 ± 0	71.59 ± 0	**63.16 ± 0**	0:00:00.001982
smote	69.62 ± 0.81	41.46 ± 3.28	70.19 ± 0.74	57 ± 0.88	66.67 ± 0	0 ± 0	71.1 ± 0.75	57.58 ± 2.03	0:00:00.004998
smote-b1	68.72 ± 0.85	34.73 ± 1.82	71.12 ± 1.06	52.08 ± 1.61	66.67 ± 0	0 ± 0	**71.67 ± 0.59**	52.78 ± 2.47	0:00:00.004981
smote-b2	66.56 ± 2.41	37.61 ± 7.6	71.31 ± 0.59	54.45 ± 0.87	66.67 ± 0	0 ± 0	70.87 ± 0.55	54.16 ± 1.37	0:00:00.003989
svm-smote	69.7 ± 0.56	40.25 ± 4.2	67.55 ± 0.21	21.99 ± 4.77	66.67 ± 0	0 ± 0	70.57 ± 1.4	43.29 ± 1.08	0:00:00.006991
adasyn	68.9 ± 0.84	41.44 ± 4.15	64.88 ± 1.05	50.63 ± 1.43	66.67 ± 0	0 ± 0	70.8 ± 0.51	58.23 ± 0.64	0:00:00.003999
cure-smote	66.57 ± 0.39	26.09 ± 4.34	**73.32 ± 0.35**	**72.07 ± 0.58**	66.67 ± 0	0 ± 0	66.06 ± 0.47	25.33 ± 4.94	0:00:00.005984
pf-smote	67.97 ± 0.62	27.8 ± 2.56	73.2 ± 0	70.87 ± 0	66.67 ± 0	0 ± 0	67.61 ± 0	35.51 ± 0	0:00:00.002993
prowras	65.77 ± 0.81	20.14 ± 5.29	70.36 ± 1.85	66.96 ± 3.68	66.67 ± 0	0 ± 0	66.34 ± 0.21	39.18 ± 0.92	0:00:02.623987
prowsyn	66.77 ± 0.54	35.98 ± 2	73.09 ± 0.74	71 ± 0.99	66.67 ± 0	0 ± 0	69.3 ± 1.22	57.57 ± 1.2	0:00:00.005985
game	69.62 ± 1.45	**47.29 ± 4.31**	71.64 ± 3.01	64.15 ± 4.22	66.67 ± 0	0 ± 0	71.12 ± 0.42	52.08 ± 1.07	0:00:08.292331
gan	**70.09 ± 0.47**	39.62 ± 1.08	66.39 ± 0	4.02 ± 0	66.67 ± 0	0 ± 0	65.55 ± 0	0 ± 0	0:01:42.887882
dcgan	69.24 ± 0.54	36.42 ± 1.09	66.29 ± 0.04	4.78 ± 0.36	66.67 ± 0	0 ± 0	65.55 ± 0	0 ± 0	0:03:49.640996
cgan	68.81 ± 0.63	33.33 ± 1.97	66.57 ± 0.15	5.58 ± 0.37	66.67 ± 0	0 ± 0	65.55 ± 0.21	0 ± 0	0:02:46.839240
wgan	69.57 ± 0.51	37.97 ± 1.45	66.2 ± 0.09	5.53 ± 0.36	66.67 ± 0	0 ± 0	65.55 ± 0.08	0 ± 0	0:24:27.195045
sgan	68.6 ± 0.57	32.24 ± 0.66	66.39 ± 0.1	4.02 ± 0.01	66.67 ± 0	0 ± 0	65.55 ± 0	0 ± 0	0:05:44.714214
ctgan	67.25 ± 0.74	20.07 ± 7.1	66.57 ± 0.1	0.83 ± 0.37	66.67 ± 0	0 ± 0	67.63 ± 0.46	17.71 ± 3.75	0:01:38.174133
ctabgan	66.76 ± 0.57	16.79 ± 5.56	66.67 ± 0	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.04	0 ± 0	0:06:26.202832

(d) PID4

PID4 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	69.47 ± 0.5	31.19 ± 4.26	68.24 ± 0	24.03 ± 0	66.67 ± 0	0 ± 0	68.73 ± 0	24.82 ± 0	0:00:00
random-os	67.84 ± 0.19	20.29 ± 1.96	74.28 ± 0	65.2 ± 0	66.67 ± 0	0 ± 0	71.91 ± 0	60.85 ± 0	0:00:00.002007
smote	69.84 ± 0.65	42.42 ± 5.64	74.33 ± 0.64	64.52 ± 1.99	66.48 ± 0	0 ± 0	**73.89 ± 0.71**	**62.97 ± 0.56**	0:00:00.002997
smote-b1	69.54 ± 1.49	36.13 ± 5.81	74.61 ± 1.26	62.72 ± 3.6	66.67 ± 0	0 ± 0	72.66 ± 0.54	60.56 ± 1.87	0:00:00.005013
smote-b2	69.79 ± 1.86	45.16 ± 5.21	73.87 ± 1.57	61.14 ± 4.88	66.67 ± 0	0 ± 0	72.21 ± 0.57	60.25 ± 2.6	0:00:00.003993
svm-smote	68.22 ± 1.35	35.85 ± 5.98	74.91 ± 1.34	61.38 ± 5.14	66.67 ± 0	0 ± 0	71.86 ± 0.55	55.14 ± 1.27	0:00:00.005998
adasyn	68.29 ± 0.85	31.58 ± 2.9	74.51 ± 0.5	64.16 ± 2.12	66.67 ± 0	0 ± 0	73.24 ± 0.53	61.22 ± 0.92	0:00:00.004017
cure-smote	67.49 ± 0.56	32.15 ± 3.85	68.52 ± 0.72	59.52 ± 4.47	66.67 ± 0	0 ± 0	68.03 ± 0.47	49.46 ± 11.47	0:00:00.004987
pf-smote	66.25 ± 0.3	31.55 ± 3.63	66.16 ± 0	58.99 ± 0	66.67 ± 0	0 ± 0	66.23 ± 0	41.14 ± 0	0:00:00.002993
prowras	68.49 ± 0.93	42.01 ± 5.55	**75.33 ± 0.73**	**69.46 ± 1.08**	66.67 ± 0	0 ± 0	66.77 ± 0.15	35.08 ± 3.41	0:00:02.557165
prowsyn	66.23 ± 0.94	40.35 ± 3	65.9 ± 0.47	59.86 ± 1.64	66.67 ± 0	0 ± 0	65.94 ± 0.28	54.5 ± 0.36	0:00:00.005984
game	**70.88 ± 1.43**	**51.25 ± 4.75**	75.32 ± 0.72	66.17 ± 2.6	66.67 ± 0.07	0 ± 0	73.09 ± 0.34	54.75 ± 0.95	0:00:08.717187
gan	69.29 ± 0.35	32.11 ± 1.5	66.11 ± 0.13	1.63 ± 0.01	66.67 ± 0	0 ± 0	66.11 ± 0.08	0 ± 0	0:01:36.082069
dcgan	69.47 ± 0.67	31.19 ± 4.06	66.67 ± 0.08	3.25 ± 0.64	66.67 ± 0	0 ± 0	66.3 ± 0	0 ± 0	0:03:43.015282
cgan	68.78 ± 0.57	30.3 ± 3.56	66.67 ± 0.18	3.25 ± 1.43	66.67 ± 0	0 ± 0	66.3 ± 0.1	0 ± 0	0:02:23.301274
wgan	68.78 ± 0.39	30.3 ± 0.8	66.2 ± 0.29	2.43 ± 2.76	66.67 ± 0	0 ± 0	66.11 ± 0.17	0 ± 0	0:21:47.898755
sgan	68.78 ± 0.75	30.3 ± 5.61	66.39 ± 0.16	2.44 ± 0.34	66.67 ± 0	0 ± 0	66.11 ± 0.08	0 ± 0	0:04:24.908893
ctgan	67.07 ± 0.33	25.76 ± 4.76	66.76 ± 0.29	0.83 ± 2.53	66.67 ± 0	0 ± 0	69.37 ± 0.53	30.61 ± 3	0:01:33.808287
ctabgan	67.78 ± 0.87	29.8 ± 6.28	66.67 ± 0	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.11	1.65 ± 0.92	0:06:28.938520

GAME algorithm is not to follow a given data distribution, but to break through the given limited distribution and to enlarge the synthetic range of minority samples. The utilization of the local linear property of samples in GAME makes it simpler and faster than the GAN series models. On 28 datasets of different imbalanced ratios, by 4 classifiers, our approach obtains more effective and more balanced results than the 18 mainstream methods.

CRediT authorship contribution statement

Ke Wang: Conceptualization, Methodology, Software , Writing – original draft, Writing – review & editing. **Tongqing Zhou:** Data curation, Writing – review & editing. **Menghua Luo:** Visualization, Investigation. **Xionglve Li:** Validation, Writing – review & editing. **Zhiping Cai:** Supervision, Resources.

Declaration of competing interest

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

Data availability

The datasets in our work are open sourced UCI datasets.

Acknowledgments

This work was supported in part by Project of Guizhou Provincial Department of Science and Technology, China under Grant ZK[2021]338, and in part by Science Project of GuiZhou University of Finance and Economics, China under Grant 2019XQN04.

Table 6
Classification performance on dataset GM.

(a) GM1

GM1 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	45.33 ± 2.61	67.2 ± 0.75	38.51 ± 0	70.8 ± 0	0 ± 0	66.67 ± 0	70.57 ± 0	52.99 ± 0	0:00:00
random-os	42.82 ± 1.13	66.56 ± 0.48	67.24 ± 0	71.64 ± 0	0 ± 0	66.67 ± 0	70.53 ± 0	44.06 ± 0	0:00:00.001999
smote	39.89 ± 2.31	66.04 ± 0.98	66.81 ± 1.77	71.27 ± 1.53	0 ± 0	66.67 ± 0	71.41 ± 1.06	47.59 ± 1.25	0:00:00.002998
smote-b1	52.32 ± 4.17	67.01 ± 1.45	69.01 ± 2.65	70.93 ± 1.72	0 ± 0	66.67 ± 0.08	68.87 ± 1.52	28.38 ± 2.5	0:00:00.008996
smote-b2	41.78 ± 4.97	63.86 ± 1	**70.59 ± 0.9**	71.4 ± 1.2	0 ± 0	66.67 ± 0.1	**72.54 ± 1.6**	62.23 ± 0.77	0:00:00.003999
svm-smote	41.1 ± 4.56	66.14 ± 1.59	62.62 ± 2.04	72.03 ± 0.27	0 ± 0	66.67 ± 0	68.21 ± 0.9	65.7 ± 1.27	0:00:00.009998
adasyn	45.99 ± 4.37	65.91 ± 0.39	67.96 ± 1.19	**72.15 ± 0.8**	0 ± 0	66.67 ± 0	70.87 ± 0.54	41.92 ± 1.09	0:00:00.003999
cure-smote	47.69 ± 0.91	63.5 ± 0.56	63.2 ± 1.14	68.4 ± 2.15	0 ± 0	66.67 ± 0	54.81 ± 5.89	56.75 ± 3.11	0:00:00.008975
pf-smote	44.95 ± 1.84	63.91 ± 0.35	64.78 ± 0	70 ± 0	0 ± 0	66.67 ± 0	54.69 ± 0	56.47 ± 0	0:00:00.003989
prowras	45.79 ± 2.78	66.77 ± 1.42	27.81 ± 2.02	68.77 ± 0.27	0 ± 0	66.67 ± 0	54.31 ± 0.43	**70.3 ± 0.43**	0:00:00.056847
prowsyn	45.83 ± 3.81	66.23 ± 0.94	62.2 ± 1.48	67.41 ± 0.97	0 ± 0	66.67 ± 0.08	67.22 ± 0.87	50.63 ± 1.11	0:00:00.006982
game	50.4 ± 2.72	**70.49 ± 1.47**	64.81 ± 1.53	69.29 ± 1.6	0 ± 0	66.67 ± 0	15.92 ± 0.41	65.82 ± 0.06	0:00:18.425139
gan	49.75 ± 2.87	66.78 ± 0.27	42.51 ± 1.55	71.17 ± 0.35	0 ± 0	66.67 ± 0	1.59 ± 0	66.84 ± 0	0:01:43.808147
dcgan	52.71 ± 2.66	67.68 ± 0.44	40.48 ± 0.87	70.55 ± 0.19	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:03:46.404169
cgan	45.09 ± 1.69	66.77 ± 0.45	38.53 ± 3.64	70.13 ± 0.71	0 ± 0	66.67 ± 0	9.59 ± 0.35	66.39 ± 0.23	0:02:52.620182
wgan	53.04 ± 1.63	67.23 ± 0.45	41.09 ± 0.82	70.85 ± 0.1	0 ± 0	66.67 ± 0	5.45 ± 0	67.29 ± 0	0:23:28.683928
sgan	51.24 ± 1.56	67.22 ± 0.7	40.73 ± 1.52	70.94 ± 0.39	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.18	0:05:24.089319
ctgan	**54.42 ± 2.59**	64.04 ± 1.29	60.96 ± 5.7	67.28 ± 3.03	**24.65 ± 33.83**	56.85 ± 29.06	66.56 ± 0.67	47.03 ± 4.86	0:02:40.529967
ctabgan	46.35 ± 3.92	64.68 ± 0.94	57.67 ± 6.19	67.14 ± 1.84	0 ± 0	66.67 ± 0.08	67.81 ± 8.2	42.02 ± 6.76	0:24:17.027939

(b) GM2

GM2 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	28.38 ± 0.96	68.87 ± 0.25	3.92 ± 0	67.11 ± 0	0 ± 0	66.67 ± 0	66.28 ± 0	23.53 ± 0	0:00:00
random-os	15.44 ± 1.51	66.29 ± 0.47	61.26 ± 0	69.06 ± 0	0 ± 0	66.67 ± 0	65.43 ± 0	18.18 ± 0	0:00:00.002003
smote	27.54 ± 2.45	68.2 ± 1.42	60.77 ± 0.72	69.05 ± 0.15	0 ± 0	66.67 ± 0	64.66 ± 0.4	39.57 ± 1.13	0:00:00.002998
smote-b1	30.57 ± 1.47	68.22 ± 1.15	54.72 ± 1.06	66.67 ± 0.55	0 ± 0	66.67 ± 0	64.75 ± 0.63	49.27 ± 2.31	0:00:00.006995
smote-b2	23.1 ± 1.98	66.57 ± 1.22	53.08 ± 1.09	65.74 ± 0.98	0 ± 0	66.67 ± 0	48.12 ± 1.68	57.04 ± 3.37	0:00:00.003999
svm-smote	26.28 ± 5.43	66.57 ± 1.69	56.17 ± 2.58	69.17 ± 0.68	0 ± 0	66.67 ± 0	62.3 ± 4.71	51.03 ± 2.64	0:00:00.013994
adasyn	27.72 ± 1.42	68.58 ± 0.79	60.77 ± 0.78	69.05 ± 0.45	0 ± 0	66.67 ± 0	64.71 ± 0.6	34.76 ± 3.09	0:00:00.003996
cure-smote	51.16 ± 2.38	**69.17 ± 1.02**	58.06 ± 3.8	67.84 ± 0.78	0 ± 0	66.67 ± 0	54.29 ± 6.46	65.38 ± 2.46	0:00:00.009973
pf-smote	43.48 ± 1.64	67.09 ± 0.52	62.39 ± 0	66.92 ± 0	0 ± 0	66.67 ± 0	66.15 ± 0	64.2 ± 0	0:00:00.003989
prowras	28.21 ± 2.78	67.44 ± 0.71	2.35 ± 0.55	66.58 ± 0.06	0 ± 0	66.67 ± 0	27.67 ± 0.93	66.28 ± 0.34	0:00:00.030917
prowsyn	44.96 ± 2.42	65.25 ± 1.59	61.74 ± 0.92	67.41 ± 0.27	0 ± 0	66.67 ± 0	70.34 ± 1.09	61.78 ± 1.42	0:00:00.011969
game	32.4 ± 4.65	68.04 ± 1.06	64.26 ± 1.73	64.54 ± 1.07	0 ± 0	66.67 ± 0	17.59 ± 0.15	63.49 ± 0.02	0:00:06.205023
gan	29.51 ± 1.25	69.06 ± 0.66	5.41 ± 1.37	66.94 ± 0.38	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:01:38.280720
dcgan	26.4 ± 2.49	68.01 ± 0.72	6.18 ± 0.61	67.21 ± 0.23	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:03:03.361679
cgan	26 ± 1.27	68.29 ± 0.58	4.67 ± 2.38	67.03 ± 0.48	0 ± 0	66.67 ± 0	0.8 ± 1.16	**66.76 ± 0.28**	0:02:22.536756
wgan	27.42 ± 0.67	69.04 ± 0.5	11.19 ± 1.44	67.49 ± 0.35	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:18:57.550854
sgan	23.49 ± 1.19	67.52 ± 0.57	6.15 ± 1.01	67.03 ± 0.34	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.18	0:04:13.001600
ctgan	**56.81 ± 4.56**	67.02 ± 2.32	**66.09 ± 2.93**	**70.41 ± 1.55**	**41.46 ± 29.19**	40.75 ± 30.13	69.86 ± 1.18	32.9 ± 8.68	0:02:44.845434
ctabgan	51.82 ± 4.58	66.1 ± 1.17	61.33 ± 1.58	68.36 ± 0.85	0 ± 0	66.67 ± 0.1	**70.81 ± 1.24**	47.19 ± 4.98	0:24:20.620846

(c) GM3

GM3 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	14.79 ± 4.13	66.2 ± 3.51	12.73 ± 0	68.21 ± 0	0 ± 0	66.67 ± 0	63.38 ± 0	10.34 ± 0	0:00:00
random-os	21.77 ± 1.55	67.42 ± 4.32	59.22 ± 0	**71.43 ± 0**	0 ± 0	66.67 ± 0	63.65 ± 0	12.29 ± 0	0:00:00.002006
smote	25.17 ± 5.95	67.62 ± 3.29	58.74 ± 1.72	71.09 ± 0.76	0 ± 0	66.67 ± 0	63.56 ± 1	11.64 ± 0.51	0:00:00.002998
smote-b1	26.92 ± 0.78	66.86 ± 2.54	51.16 ± 1.29	69.17 ± 1.81	0 ± 0	66.67 ± 0.83	64.14 ± 1.22	21.66 ± 0.58	0:00:00.009997
smote-b2	27.85 ± 5.89	66.67 ± 2.61	43.38 ± 2.1	68.84 ± 1.55	0 ± 0	66.67 ± 0	63.14 ± 6.67	38.83 ± 1.29	0:00:00.005998
svm-smote	23.92 ± 2.54	67.24 ± 3.15	33.12 ± 3.79	69.39 ± 1.92	0 ± 0	66.67 ± 0	63.64 ± 0.97	13.51 ± 0.62	0:00:00.006998
adasyn	23.03 ± 2.36	66.38 ± 2.52	58.45 ± 1.67	70.65 ± 0.83	0 ± 0	66.67 ± 0	63.47 ± 1.68	11 ± 0.77	0:00:00.004998
cure-smote	**62.77 ± 2.56**	60.78 ± 3.23	61.91 ± 0.72	63.26 ± 1.23	0 ± 0	66.67 ± 0	44.27 ± 1.51	63.92 ± 0.37	0:00:00.003989
pf-smote	56.77 ± 1.16	62.43 ± 3.85	65.36 ± 0	63.8 ± 0	**66.67 ± 0**	0 ± 0	46 ± 0	64 ± 0	0:00:00.002992
prowras	25.74 ± 2.76	67.72 ± 0.62	22.3 ± 1.87	68.72 ± 0.25	0 ± 0	66.67 ± 0	**68.1 ± 0.12**	59.73 ± 0.72	0:00:00.006982
prowsyn	58.67 ± 1.51	63.79 ± 4.53	65.74 ± 1.03	65.05 ± 0.49	0 ± 0	66.67 ± 0.42	45.06 ± 1.02	64.13 ± 0.29	0:00:00.004987
game	29.68 ± 3.37	**68.41 ± 3.62**	**66.39 ± 2.03**	69.47 ± 1.38	0 ± 0	66.67 ± 12.86	23.33 ± 0.11	**67.14 ± 0.01**	0:00:03.234974
gan	13.48 ± 1.81	66.02 ± 2.05	19.35 ± 0.83	68.79 ± 0.24	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.04	0:01:25.971284
dcgan	16.67 ± 2.83	66.29 ± 2.48	18.71 ± 0.18	68.7 ± 0.42	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.08	0:02:34.798059
cgan	13.43 ± 2.69	65.83 ± 0.56	22.46 ± 1.27	69.09 ± 0.85	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.01	0:02:00.112176
wgan	14.29 ± 3.98	66.67 ± 1.93	15.94 ± 0.63	67.96 ± 0.52	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.07	0:16:25.620331
sgan	14.13 ± 3.6	66.11 ± 1.94	19.35 ± 1.15	68.79 ± 0.4	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0.3	0:03:33.824970
ctgan	28.57 ± 3.82	64.98 ± 0.96	53.86 ± 5.46	65.62 ± 1.69	0.79 ± 1.04	66.4 ± 0.16	63.68 ± 0.71	26.59 ± 4.91	0:02:46.129005
ctabgan	31.74 ± 4.43	65.77 ± 1.64	44.32 ± 4.47	67.3 ± 0.83	0.16 ± 0.36	66.49 ± 0.12	64.07 ± 2.6	27.71 ± 11.62	0:15:32.501790

(d) GM4

GM4 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	25.32 ± 2.51	66.76 ± 0.53	7.6 ± 0	67.03 ± 0	0 ± 0	66.67 ± 0	55.41 ± 0	30.26 ± 0	0:00:00
random-os	12.77 ± 2.2	65.74 ± 0.63	41.99 ± 0	64.3 ± 0	0 ± 0	66.67 ± 0	57.14 ± 0	29.18 ± 0	0:00:00.001001
smote	26.89 ± 4.31	67.91 ± 1.04	40.85 ± 0.7	64.21 ± 0.65	0 ± 0	66.67 ± 0	54.04 ± 0.37	36.28 ± 4.22	0:00:00.001999
smote-b1	20 ± 3.02	67.32 ± 0.93	39.55 ± 1.08	66.87 ± 0.9	0 ± 0	66.67 ± 0	45.17 ± 0.5	47.95 ± 2	0:00:00.003992

(*continued on next page*)

Table 6 (*continued*).

smote-b2	26.4 ± 4.12	68.01 ± 1.08	40.33 ± 0.55	66.14 ± 0.75	0 ± 0	66.67 ± 0.1	41.59 ± 0.37	56.29 ± 3.16	0:00:00.003996
svm-smote	26.2 ± 3.47	66.38 ± 0.52	38.21 ± 6.01	68.87 ± 0.55	0 ± 0	66.67 ± 0	42.86 ± 0.32	50.93 ± 2.72	0:00:00.007995
adasyn	26.8 ± 2.68	67.72 ± 1.21	42.28 ± 0.63	66.24 ± 0.46	0 ± 0	66.67 ± 0	53.03 ± 0.07	35.93 ± 1.22	0:00:00.002999
cure-smote	33.33 ± 3.85	66.27 ± 0.34	60.49 ± 2.3	67.28 ± 0.9	0 ± 0	66.67 ± 0	57.58 ± 3.4	52.54 ± 1.97	0:00:00.007979
pf-smote	36.42 ± 1.59	66.36 ± 0.29	69.62 ± 0	69.98 ± 0	0 ± 0	66.67 ± 0	64.54 ± 0	54.13 ± 0	0:00:00.003989
prowras	34.76 ± 2.1	68.15 ± 0.83	6.92 ± 2.1	67.3 ± 0.24	0 ± 0	66.67 ± 0	65.31 ± 0	59.44 ± 0	0:00:00.061836
prowsyn	34.71 ± 0.66	66.36 ± 1.05	**69.72 ± 0.3**	69.48 ± 0.21	0 ± 0	66.67 ± 0.08	**67.33 ± 0.28**	23.41 ± 2.89	0:00:00.009973
game	33.02 ± 3.8	**69.2 ± 0.86**	47.26 ± 1.26	61.96 ± 0.97	0 ± 0	66.67 ± 0	18.71 ± 0.3	63.48 ± 0.06	0:00:03.807787
gan	28.12 ± 1.71	67.25 ± 0.9	9.77 ± 1.92	67.3 ± 0.19	0 ± 0	66.67 ± 0	0 ± 0	**66.67 ± 0**	0:01:25.564697
dcgan	27.01 ± 1.47	67.05 ± 0.46	9.77 ± 0.89	67.3 ± 0.14	0 ± 0	66.67 ± 0	0 ± 0	**66.67 ± 0**	0:02:28.042186
cgan	30.63 ± 1.89	67.35 ± 0.67	8.33 ± 2.75	67.12 ± 0.47	0 ± 0	66.67 ± 0	0 ± 0	**66.67 ± 0.36**	0:01:55.647470
wgan	27.01 ± 2.45	67.05 ± 0.6	14.65 ± 0.27	67.95 ± 0.14	0 ± 0	66.67 ± 0	0 ± 0	**66.67 ± 0**	0:16:05.655506
sgan	26.54 ± 1.08	67.15 ± 0.4	9.77 ± 2.26	67.3 ± 0.3	0 ± 0	66.67 ± 0	0 ± 0	**66.67 ± 0.19**	0:03:32.182935
ctgan	**47.9 ± 4.19**	64.54 ± 1.73	69.35 ± 2.06	69.84 ± 2.21	**66.12 ± 0.69**	8.86 ± 5.54	66.39 ± 0.24	13.57 ± 2.01	0:02:46.635649
ctabgan	38.86 ± 5.46	67.08 ± 2.33	66.08 ± 2.21	**71.45 ± 1.31**	12.63 ± 0.9	66.49 ± 0.14	66.76 ± 1.29	16.2 ± 4.11	0:24:27.547824

Table 7
Classification performance on dataset DCCC.

(a) DCCC1

DCCC1 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	68.31 ± 0.12	47.71 ± 0.66	66.68 ± 0	0.11 ± 0	66.65 ± 0	0.04 ± 0	32.69 ± 0	67.7 ± 0	0:00:00
random-os	67.92 ± 0.08	45.33 ± 0.31	59.32 ± 0	63.86 ± 0	66.68 ± 0	0.44 ± 0	26.01 ± 0	67.72 ± 0	0:00:00.008000
smote	68.2 ± 0.3	47.53 ± 0.61	65.38 ± 0.07	43.5 ± 1.13	**66.76 ± 0.02**	2.43 ± 0.17	28.03 ± 0.58	67.47 ± 0.07	0:00:00.085660
smote-b1	68.31 ± 0.34	48.69 ± 0.57	65.22 ± 0.28	44.03 ± 1.58	66.73 ± 0.01	2.43 ± 0.19	28.5 ± 0.68	**67.77 ± 0.18**	0:00:00.377983
smote-b2	67.98 ± 0.23	47.26 ± 0.7	58.95 ± 1.05	**64.41 ± 7.41**	1.84 ± 0.02	**66.17 ± 0.21**	27.57 ± 0.54	67.31 ± 0.1	0:00:00.354857
svm-smote	68.12 ± 0.36	48.49 ± 0.94	66.98 ± 0.5	34.38 ± 1.73	66.7 ± 0.04	1.51 ± 0.09	32.32 ± 1.84	67.14 ± 0.27	0:00:03.056012
adasyn	66.98 ± 0.55	46.22 ± 1.65	65.33 ± 0.05	43.47 ± 1.31	66.73 ± 0.02	1.87 ± 0.13	27.8 ± 0.18	67.47 ± 0.04	0:00:00.362863
cure-smote	68.49 ± 0.11	48.25 ± 0.48	**67.13 ± 0.62**	29.81 ± 9.46	66.69 ± 29.08	0.9 ± 29.13	40.89 ± 1.46	64.8 ± 0.15	0:00:06.694001
pf-smote	67.8 ± 0.07	46.76 ± 0.18	64.62 ± 0	53.74 ± 0	66.7 ± 0	0.98 ± 0	36.53 ± 0	65.34 ± 0	0:00:00.017986
prowras	68.44 ± 0.24	47.58 ± 0.33	65.87 ± 0.07	31.2 ± 1.09	66.68 ± 0.01	0.83 ± 0.09	30.79 ± 0.45	66.39 ± 0.1	0:00:01.080655
prowsyn	67.8 ± 0.23	**49.22 ± 0.67**	65.85 ± 1.76	25.67 ± 15.17	1.66 ± 0	65.97 ± 0	33.53 ± 0.08	65.63 ± 0.03	0:00:02.899699
game	68.41 ± 0.31	47.95 ± 0.77	65.98 ± 1.29	46.19 ± 3.92	66.7 ± 0.01	2.53 ± 0.07	**66.67 ± 0**	0 ± 0	0:06:35.555324
gan	68.39 ± 0.19	47.47 ± 0.67	66.68 ± 0.01	0.11 ± 0.07	66.65 ± 0	0.04 ± 0	49.29 ± 0.07	62.66 ± 0.07	0:08:30.714959
dcgan	68.23 ± 0.15	46.63 ± 0.34	66.68 ± 0.01	0.11 ± 0.1	66.65 ± 0	0.04 ± 0	49.23 ± 0	62.72 ± 0.01	0:14:32.346290
cgan	68.59 ± 0.15	47.54 ± 0.41	66.67 ± 0.02	0.07 ± 0.24	66.65 ± 0	0.04 ± 0	49.6 ± 0.08	62.64 ± 0.13	0:11:42.667788
wgan	68.27 ± 0.23	46.79 ± 0.54	66.68 ± 0.01	0.11 ± 0.02	66.68 ± 0	0.4 ± 0	49.31 ± 0	62.72 ± 0	1:46:18.660336
sgan	68.56 ± 0.12	47.33 ± 0.47	66.68 ± 0.01	0.11 ± 0.05	66.65 ± 0	0.04 ± 0.03	49.25 ± 0.06	62.7 ± 0.03	0:20:12.390144
ctgan	**68.66 ± 0.28**	44.64 ± 1.06	66.67 ± 0	0.07 ± 0.03	66.67 ± 0	0.01 ± 0	45.18 ± 4.86	65.11 ± 0.97	1:17:13.108348
ctabgan	67.12 ± 0.42	41.06 ± 1.17	66.67 ± 0.02	0.07 ± 0.1	66.67 ± 0	0.01 ± 0	48.3 ± 3.11	63.67 ± 0.55	2:30:10.377896

(b) DCCC2

DCCC2 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	68.03 ± 0.05	36.85 ± 0.57	66.68 ± 0	0.2 ± 0	66.67 ± 0	0.03 ± 0	57.1 ± 0	64.72 ± 0	0:00:00
random-os	68.03 ± 0.19	31.93 ± 0.83	64.03 ± 0	49.24 ± 0	66.7 ± 0	0.33 ± 0	23.11 ± 0	67.38 ± 0	0:00:00.007997
smote	67.87 ± 0.26	38.24 ± 0.35	64.13 ± 0.29	47.83 ± 2.58	66.73 ± 0.02	1.38 ± 0.14	25.98 ± 0.86	67.35 ± 0.16	0:00:00.029984
smote-b1	68.05 ± 0.41	37.07 ± 1.36	64.98 ± 0.17	45.81 ± 0.5	66.73 ± 0.01	1.7 ± 0.08	26.78 ± 0.62	67.35 ± 0.13	0:00:00.234919
smote-b2	67.74 ± 0.08	36.92 ± 0.52	65.3 ± 0.21	52.62 ± 0.45	66.74 ± 28.96	1.61 ± 28.94	27.3 ± 0.55	**67.51 ± 0.23**	0:00:00.229913
svm-smote	67.8 ± 0.3	36.45 ± 1.08	**66.77 ± 0.19**	31.42 ± 1.06	66.71 ± 0.01	0.89 ± 0.15	27.62 ± 2.51	67.48 ± 0.32	0:00:01.866399
adasyn	67.55 ± 0.37	35.76 ± 1.33	64.35 ± 0.1	49.29 ± 0.54	66.74 ± 0.01	1.35 ± 0.09	25.9 ± 0.32	67.32 ± 0.05	0:00:00.237925
cure-smote	68.37 ± 0.11	37.53 ± 0.85	66.51 ± 1.12	33.02 ± 12.11	66.67 ± 35.49	40.75 ± 36	44.02 ± 2.3	64.47 ± 0.35	0:00:00.008072
pf-smote	68.29 ± 0.09	36.58 ± 0.32	64.02 ± 0	50.04 ± 0	66.67 ± 0	0.66 ± 0	37.17 ± 0	65.12 ± 0	0:00:00.002013
prowras	68.37 ± 0.33	37.79 ± 1.36	64.2 ± 0.47	55.85 ± 0.68	66.67 ± 0.01	0.73 ± 0.08	31.15 ± 0.69	66.76 ± 0.16	0:00:00.100254
prowsyn	68.55 ± 0.18	**39.04 ± 0.68**	64.19 ± 0.08	47.6 ± 0.38	1.87 ± 0	**66.12 ± 0**	32.86 ± 0.37	65.72 ± 0.04	0:00:00.015625
game	68.6 ± 0.22	38.26 ± 0.58	65.11 ± 0.19	**55.89 ± 1.11**	**66.74 ± 0.01**	1.93 ± 0.09	66.64 ± 0	0.13 ± 0	0:03:03.636084
gan	68.51 ± 0.11	38.22 ± 0.48	66.67 ± 0.01	0 ± 0.05	66.67 ± 0	0.03 ± 0	56.3 ± 0.03	59.7 ± 0.11	0:09:20.245777
dcgan	68.21 ± 0.13	37.27 ± 0.63	66.67 ± 0.01	0.1 ± 0.07	66.67 ± 0	0.03 ± 0	56.27 ± 0.02	59.67 ± 0.02	0:20:00.444797
cgan	67.98 ± 0.14	36.55 ± 0.26	66.67 ± 0.01	0.03 ± 0.05	66.67 ± 0	0.03 ± 0	56.3 ± 0.16	59.47 ± 0.13	0:14:12.894632
wgan	68.1 ± 0.07	37.12 ± 0.62	66.69 ± 0	0.2 ± 0.05	66.67 ± 0	0.03 ± 0	56.31 ± 0.02	59.63 ± 0.03	1:33:18.741894
sgan	67.72 ± 0.12	36.18 ± 0.66	66.67 ± 0.01	0.13 ± 0.09	66.67 ± 0.01	0.03 ± 0.08	56.27 ± 0.06	59.63 ± 0.08	0:21:41.850980
ctgan	**68.97 ± 0.48**	35.8 ± 1.22	66.67 ± 0	0.17 ± 0.06	66.67 ± 0	0.07 ± 0	66.93 ± 1.3	5.37 ± 18.11	1:15:24.004548
ctabgan	67.63 ± 0.37	32.41 ± 2.09	66.66 ± 0.01	0.07 ± 0.17	66.66 ± 0.01	0.03 ± 0.02	**67.54 ± 3.36**	23.88 ± 20.62	2:04:56.284062

(c) DCCC3

DCCC3 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	68.14 ± 0.05	33.95 ± 0.3	66.67 ± 0	0.06 ± 0	66.67 ± 0	0.03 ± 0	41.87 ± 0	66.41 ± 0	0:00:00
random-os	**68.73 ± 0.09**	32.55 ± 0.66	63.31 ± 0	50.62 ± 0	66.68 ± 0	0.1 ± 0	22.9 ± 0	**66.96 ± 0**	0:00:00.007013
smote	68.57 ± 0.18	37.18 ± 0.49	63.05 ± 0.14	46.79 ± 0.38	66.7 ± 0.01	0.61 ± 0.09	26.45 ± 0.41	66.55 ± 0.03	0:00:00.019987
smote-b1	67.77 ± 0.37	34.88 ± 1.58	65.11 ± 0.13	40.41 ± 0.46	66.7 ± 0.01	0.61 ± 0.1	33.42 ± 0.33	66.07 ± 0.09	0:00:00.121955
smote-b2	67.69 ± 0.2	33.11 ± 1.03	64.51 ± 1.23	43.52 ± 6.23	**66.71 ± 9.1**	0.7 ± 9.18	30.97 ± 0.95	66.07 ± 0.26	0:00:00.139957
svm-smote	68.21 ± 0.35	33.87 ± 1.52	**67.39 ± 0.25**	16.76 ± 8.78	66.68 ± 0.02	0.19 ± 0.16	33.25 ± 0.75	65.95 ± 0.21	0:00:01.151624
adasyn	68.15 ± 0.13	36.17 ± 1.18	62.69 ± 0.08	48.01 ± 0.71	1.51 ± 0.02	**66.26 ± 0.01**	25.48 ± 0.3	66.75 ± 0.02	0:00:00.132956

(*continued on next page*)

Table 7 (*continued*).

cure-smote	68.3 ± 0.32	35.18 ± 1.21	64.36 ± 0.66	46.77 ± 4.57	1.51 ± 35.7	66.13 ± 35.84	41.49 ± 6.1	65.13 ± 10.19	0:00:00.215425
pf-smote	68.55 ± 0.1	34.32 ± 0.57	62.8 ± 0	48.96 ± 0	66.7 ± 0	0.61 ± 0	38.68 ± 0	65.35 ± 0	0:00:00.011968
prowras	68.09 ± 0.1	35.16 ± 1.15	64.8 ± 0.17	31.62 ± 0.58	66.71 ± 0.01	0.45 ± 0.02	36.44 ± 0.55	66.24 ± 0.14	0:00:00.405916
prowsyn	68.04 ± 0.23	**38.11 ± 0.68**	64.16 ± 0.19	27.96 ± 1.54	66.69 ± 0	0.26 ± 0.04	35.62 ± 0.45	65.86 ± 0.06	0:00:00.149600
game	68.28 ± 0.25	34.94 ± 1.16	64.9 ± 0.36	**55.69 ± 2.67**	66.71 ± 0.01	0.86 ± 0.09	**66.67 ± 0**	0 ± 0.01	0:02:08.688508
gan	68.17 ± 0.04	35.11 ± 0.5	66.69 ± 0	0.19 ± 0.03	66.67 ± 0	0.03 ± 0	58.37 ± 0.07	57 ± 0.02	0:08:56.541324
dcgan	68.17 ± 0.08	33.88 ± 0.25	66.67 ± 0	0.06 ± 0	66.67 ± 0	0.03 ± 0	58.39 ± 0.01	57.1 ± 0.03	0:23:17.553625
cgan	68.34 ± 0.11	34.54 ± 0.36	66.67 ± 0	0 ± 0	66.67 ± 0	0.03 ± 0	58.54 ± 0.05	56.91 ± 0.06	0:17:04.188959
wgan	68.07 ± 0.05	33.42 ± 0.11	66.68 ± 0	0.26 ± 0.04	66.67 ± 0	0.06 ± 0	58.41 ± 0.01	57.11 ± 0.03	1:47:02.708529
sgan	67.69 ± 0.19	32.64 ± 0.57	66.68 ± 0	0.16 ± 0	66.67 ± 0	0.03 ± 0	58.34 ± 0.07	57.08 ± 0.1	0:20:54.156634
ctgan	67.7 ± 0.14	28.1 ± 1.36	66.67 ± 0	0 ± 0	66.67 ± 0	0 ± 0	60.16 ± 4.12	61.01 ± 1.45	0:38:35.409296
ctabgan	67.54 ± 0.52	27.8 ± 2.77	66.67 ± 0	0 ± 0	66.67 ± 0	0 ± 0	46.72 ± 5.02	64.79 ± 0.7	2:36:05.115130

(d) DCCC4

DCCC4 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	67.71 ± 0.11	25.59 ± 0.49	66.66 ± 0	0.13 ± 0	66.67 ± 0	0 ± 0	36.65 ± 0	66.46 ± 0	0:00:00
random-os	67.73 ± 0.09	22.61 ± 0.37	64.65 ± 0	42.64 ± 0	66.67 ± 0	0.06 ± 0	29.16 ± 0	**66.64 ± 0**	0:00:00.007998
smote	67.67 ± 0.22	27.79 ± 1.32	64.71 ± 0.34	43.92 ± 0.7	66.66 ± 0.01	0.28 ± 0.05	32.55 ± 0.13	66.58 ± 0.08	0:00:00.045986
smote-b1	67.75 ± 0.06	27.8 ± 0.54	65.2 ± 0.3	44.35 ± 2.92	66.67 ± 0.01	0.25 ± 0.03	34.87 ± 0.18	66.43 ± 0.09	0:00:00.132953
smote-b2	67.65 ± 0.22	28.46 ± 1.08	65.22 ± 0.12	47.17 ± 0.25	66.67 ± 0.02	0.25 ± 0.07	31.38 ± 0.23	66.35 ± 0.06	0:00:00.116956
svm-smote	67.52 ± 0.17	25.49 ± 0.51	67.15 ± 0.2	11.16 ± 6.92	66.67 ± 0.01	0.03 ± 0.18	34.84 ± 0.19	66.5 ± 0.06	0:00:00.952695
adasyn	67.76 ± 0.36	26.64 ± 1.42	64.79 ± 0.07	43.72 ± 0.53	66.66 ± 0.04	0.16 ± 0.01	32.2 ± 0.05	66.54 ± 0.04	0:00:00.119947
cure-smote	**68.05 ± 0.08**	26.46 ± 0.73	**67.16 ± 0.27**	17.48 ± 5.9	66.69 ± 0	0.38 ± 0.02	46.58 ± 0.6	63.4 ± 0.21	0:00:01.076652
pf-smote	67.97 ± 0.18	24.71 ± 0.67	59.23 ± 0	**63.74 ± 0**	66.69 ± 0	0.57 ± 0	41.11 ± 0	64.74 ± 0	0:00:00.021994
prowras	67.54 ± 0.26	28.74 ± 1.3	65.55 ± 0.48	48.11 ± 1.02	66.71 ± 0.01	0.88 ± 0.02	35.9 ± 0.36	65.89 ± 0.08	0:00:00.832745
prowsyn	67.82 ± 0.24	**30.56 ± 1.2**	64.85 ± 0.12	38.89 ± 0.29	1.91 ± 0.09	**66.16 ± 0.01**	36.25 ± 0.21	65.43 ± 0.12	0:00:00.572816
game	67.97 ± 0.19	27.37 ± 1.05	64.73 ± 0.25	48.59 ± 1.28	**66.75 ± 0.01**	1.47 ± 0.05	**66.67 ± 0**	0 ± 0	0:01:23.655626
gan	67.68 ± 0.17	25.24 ± 0.62	66.65 ± 0	0.06 ± 0	66.67 ± 0	0 ± 0	62.57 ± 0.02	51.43 ± 0.06	0:08:26.322965
dcgan	67.59 ± 0.08	24.43 ± 0.3	66.66 ± 0	0.09 ± 0	66.67 ± 0	0 ± 0	62.52 ± 0.01	51.4 ± 0.03	0:15:18.190145
cgan	67.65 ± 0.13	24.97 ± 0.52	66.66 ± 0	0.13 ± 0	66.67 ± 0	0 ± 0	62.68 ± 0.01	51.52 ± 0.04	0:12:27.157882
wgan	67.87 ± 0.16	26.32 ± 0.63	66.66 ± 0	0.13 ± 0	66.67 ± 0	0 ± 0	62.61 ± 0.01	51.63 ± 0.01	1:47:15.817282
sgan	67.58 ± 0.05	25.15 ± 0.46	66.65 ± 0	0.06 ± 0	66.67 ± 0	0 ± 0	62.58 ± 0.02	51.51 ± 0.03	0:21:23.616219
ctgan	67.15 ± 0.78	19.26 ± 4.12	66.67 ± 0	0 ± 0	66.67 ± 0.43	0.03 ± 0	62.23 ± 5.18	54.25 ± 25.3	0:38:54.330707
ctabgan	67.34 ± 0.27	21.2 ± 1.76	66.67 ± 0.02	0 ± 0	66.67 ± 0	0.06 ± 0	57.22 ± 5.91	61.37 ± 3.07	2:35:32.792755

(e) DCCC5

DCCC5 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	67.24 ± 0.05	14.04 ± 0.43	66.66 ± 0	0 ± 0	66.67 ± 0	0 ± 0	64.94 ± 0	12.59 ± 0	0:00:00
random-os	67.26 ± 0.07	11.52 ± 0.56	61.71 ± 0	47.7 ± 0	66.67 ± 0	0 ± 0	29.49 ± 0	**67.52 ± 0**	0:00:00.007996
smote	67.17 ± 0.12	13.64 ± 1.14	63.05 ± 0.09	47.58 ± 0.62	2.18 ± 0.02	66 ± 0	29.37 ± 0.43	67.32 ± 0.1	0:00:00.028063
smote-b1	67.12 ± 0.17	14.72 ± 1.17	65.34 ± 0.13	43.58 ± 0.25	66.66 ± 0	0.03 ± 0	34.9 ± 0.08	66.86 ± 0.06	0:00:00.045985
smote-b2	67.27 ± 0.1	15.62 ± 1	63.81 ± 0.18	48.13 ± 2.58	1.89 ± 0.11	**66.11 ± 0.01**	30.37 ± 0.47	66.94 ± 0.1	0:00:00.042985
svm-smote	67.19 ± 0.18	13.95 ± 1.61	**67.31 ± 0.21**	20.9 ± 5.78	66.67 ± 0	0 ± 0	35.49 ± 0.26	66.91 ± 0.07	0:00:00.488844
adasyn	67.23 ± 0.2	14.13 ± 1.42	63.55 ± 0.05	46.86 ± 0.37	66.66 ± 0	0.03 ± 0	29.72 ± 0.26	67.35 ± 0.04	0:00:00.045981
cure-smote	67.19 ± 0.11	13.67 ± 0.65	65.97 ± 0.44	43.24 ± 2.89	1.47 ± 0.01	66.02 ± 0.01	46.39 ± 0.74	63.17 ± 0.28	0:00:00.108963
pf-smote	67.19 ± 0.05	11.77 ± 0.54	64.19 ± 0	44.57 ± 0	**66.68 ± 0**	0.28 ± 0	39.8 ± 0	65.01 ± 0	0:00:00.043990
prowras	**67.62 ± 0.14**	17.27 ± 1.07	65.5 ± 0.11	**50.71 ± 2.42**	66.68 ± 0	0.21 ± 0	36.41 ± 0.5	65.67 ± 0.17	0:00:00.864731
prowsyn	67.48 ± 0.1	**18.86 ± 1.29**	64.81 ± 0.19	42.23 ± 0.28	1.41 ± 0	66.07 ± 0	34.36 ± 0.48	65.88 ± 0.15	0:00:00.072972
game	67.3 ± 0.15	14.7 ± 0.88	62.24 ± 0.04	49.86 ± 0.13	66.64 ± 0.01	0.31 ± 0.05	66.45 ± 0	3.69 ± 0.05	0:00:26.079013
gan	67.3 ± 0.04	14.75 ± 0.37	66.66 ± 0.01	0 ± 0	66.67 ± 0	0 ± 0	65.47 ± 0.01	34.66 ± 0.15	0:09:12.329669
dcgan	67.12 ± 0.05	13.58 ± 0.56	66.66 ± 0	0 ± 0	66.67 ± 0	0 ± 0	65.49 ± 0.01	34.52 ± 0.04	0:21:32.078098
cgan	67.16 ± 0.06	14.11 ± 0.51	66.66 ± 0	0 ± 0	66.67 ± 0	0 ± 0	65.42 ± 0.06	34.66 ± 0.51	0:14:17.579995
wgan	67.04 ± 0.05	12.65 ± 0.48	66.66 ± 0	0 ± 0	66.67 ± 0	0 ± 0	65.52 ± 0	34.3 ± 0.02	1:42:46.067843
sgan	67.3 ± 0.03	13.9 ± 0.41	66.66 ± 0	0 ± 0	66.67 ± 0	0 ± 0	65.46 ± 0.01	34.65 ± 0.07	0:25:43.671022
ctgan	67.07 ± 0.17	10.3 ± 1.38	66.67 ± 0	0 ± 0	66.67 ± 0	0.03 ± 0	**67.67 ± 1.94**	14.43 ± 20.35	0:38:31.495285
ctabgan	67.29 ± 0.15	12.54 ± 1.29	66.67 ± 0.01	0.03 ± 0.13	66.67 ± 0	0.03 ± 0.02	57.03 ± 5.56	61.16 ± 26.51	2:10:02.821317

(f) DCCC6

DCCC6 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	66.82 ± 0.11	6.15 ± 0.88	66.68 ± 0	0.49 ± 0	66.67 ± 0	0 ± 0	47.72 ± 0	64.6 ± 0	0:00:00
random-os	**67.2 ± 0.04**	7.28 ± 0.27	62.64 ± 0	46.25 ± 0	66.67 ± 0	0.06 ± 0	39.73 ± 0	**65.78 ± 0**	0:00:00.006998
smote	66.98 ± 0.09	10.37 ± 0.65	63.02 ± 0.08	46.34 ± 0.49	66.69 ± 0	0.27 ± 0.02	42.1 ± 0.24	65.36 ± 0.02	0:00:00.013990
smote-b1	66.96 ± 0.06	9.08 ± 0.78	62.41 ± 0.12	30.7 ± 2.91	66.68 ± 0	0.21 ± 0	59.15 ± 0.2	54.65 ± 0.07	0:00:00.027971
smote-b2	66.9 ± 0.1	7.97 ± 1.12	60.33 ± 0.14	35.28 ± 8.94	66.69 ± 28.88	0.27 ± 29.42	47.61 ± 0.79	61.56 ± 0.29	0:00:00.024987
svm-smote	66.99 ± 0.03	7.85 ± 0.63	65.99 ± 0.05	17.42 ± 2.78	66.67 ± 0	0.06 ± 0	57.9 ± 0.53	57.73 ± 0.23	0:00:00.263914
adasyn	67.08 ± 0.09	10.24 ± 0.7	62.95 ± 0.07	46.81 ± 0.48	66.69 ± 0.03	0.27 ± 0	41.95 ± 0.16	65.34 ± 0.04	0:00:00.027980
cure-smote	67.08 ± 0.08	8.7 ± 0.91	66.51 ± 0.46	30.94 ± 6.22	66.67 ± 0	0.03 ± 0	53.42 ± 0.67	61.55 ± 0.35	0:00:00.010005
pf-smote	67.15 ± 0.06	9.06 ± 0.45	65.94 ± 0	37.36 ± 0	1.75 ± 0	**65.99 ± 0**	48.21 ± 0	63.74 ± 0	0:00:00.015623
prowras	66.95 ± 0.06	11.14 ± 0.56	**67.11 ± 0.1**	32.46 ± 0.42	66.67 ± 0	0.03 ± 0	44.9 ± 0.36	64.4 ± 0.02	0:00:00.115874
prowsyn	67.12 ± 0.15	**11.87 ± 1.02**	66.2 ± 0.05	36.64 ± 0.68	1.74 ± 0	65.96 ± 0.01	42.82 ± 0.26	64.84 ± 0.03	0:00:00.015625
game	66.97 ± 0.1	9.43 ± 0.65	62.05 ± 0.04	**50.33 ± 0.19**	**66.71 ± 0**	0.73 ± 0	44.42 ± 0.3	65.39 ± 0.45	0:00:11.432351
gan	66.99 ± 0.05	6.89 ± 0.62	66.68 ± 0	0.49 ± 0.04	66.67 ± 0	0 ± 0	66.56 ± 0.01	12.22 ± 0.35	0:09:26.732850
dcgan	66.88 ± 0.05	6.1 ± 0.28	66.68 ± 0	0.49 ± 0	66.67 ± 0	0 ± 0	66.56 ± 0	12.22 ± 0.03	0:20:33.835280
cgan	66.89 ± 0.06	6.38 ± 0.45	66.68 ± 0	0.49 ± 0	66.67 ± 0	0 ± 0	66.56 ± 0.01	11.84 ± 0.25	0:15:10.367544
wgan	66.98 ± 0.03	6.97 ± 0.18	66.68 ± 0	0.49 ± 0	66.67 ± 0	0 ± 0	66.54 ± 0	11.96 ± 0.03	1:45:21.591930

(*continued on next page*)

Table 7 (*continued*).

sgan	67.04 ± 0.08	7.58 ± 0.56	66.68 ± 0	0.49 ± 0.03	66.67 ± 0	0 ± 0	66.57 ± 0.02	12.27 ± 0.34	0:28:06.694229
ctgan	67.01 ± 0.14	5.5 ± 0.99	66.67 ± 0	0 ± 0	66.67 ± 0	0 ± 0	48.08 ± 0.69	64.08 ± 0.15	0:38:41.933899
ctabgan	66.94 ± 0.1	6.77 ± 1.09	66.67 ± 0.01	0.06 ± 0.11	66.66 ± 0	0 ± 0	**66.59 ± 0**	0.36 ± 0	2:08:57.113136
(g) DCCC7									
DCCC7 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	66.69 ± 0.02	1.2 ± 0.3	66.66 ± 0	0 ± 0	66.67 ± 0	0 ± 0	64.86 ± 0	11.67 ± 0	0:00:00
random-os	66.65 ± 0.02	0.48 ± 0.14	60.87 ± 0	53.97 ± 0	66.67 ± 0	0 ± 0	31.56 ± 0	**67.28 ± 0**	0:00:00.006998
smote	66.65 ± 0.05	1.9 ± 0.28	60.74 ± 0.06	53.21 ± 0.17	66.67 ± 0	0.03 ± 0.02	36.86 ± 0.5	66.33 ± 0.14	0:00:00.012993
smote-b1	66.78 ± 0.01	2.58 ± 0.1	64.35 ± 0.07	47.22 ± 0.24	66.67 ± 0	0.03 ± 0	53.69 ± 0.04	61.55 ± 0.03	0:00:00.015994
smote-b2	66.71 ± 0.01	2.05 ± 0.08	65.42 ± 0.18	41.03 ± 0.2	66.67 ± 0	0.03 ± 0	49.81 ± 0.45	62.73 ± 0.11	0:00:00.013997
svm-smote	66.72 ± 0.04	2.05 ± 0.39	**67.52 ± 0.31**	19.15 ± 5.45	66.67 ± 0	0 ± 0	62.46 ± 0.35	51.59 ± 1.11	0:00:00.129958
adasyn	66.7 ± 0.04	1.91 ± 0.32	60.76 ± 0.03	52.59 ± 0.16	66.67 ± 0	0.03 ± 0.01	37.61 ± 0.33	65.97 ± 0.04	0:00:00.016994
cure-smote	66.75 ± 0.04	1.85 ± 0.41	64.24 ± 0.41	37.11 ± 5.27	66.67 ± 0	0 ± 0	58.83 ± 9.96	56.78 ± 4.59	0:00:00.008072
pf-smote	66.72 ± 0.04	1.47 ± 0.27	63.6 ± 0	45.23 ± 0	1.85 ± 0	**66.09 ± 0**	48.36 ± 0	63.66 ± 0	0:00:00.002013
prowras	66.75 ± 0.02	3.1 ± 0.21	63.83 ± 0.09	43.04 ± 0.32	66.67 ± 0	0 ± 0	52.29 ± 0.43	62.41 ± 0.11	0:00:00.100254
prowsyn	66.71 ± 0.04	**3.96 ± 0.59**	63.53 ± 0.07	45.25 ± 0.04	2.19 ± 0.15	65.96 ± 0.04	44.58 ± 0.26	64.47 ± 0.04	0:00:00.015625
game	66.71 ± 0.02	1.76 ± 0.25	60.01 ± 0.04	**58.48 ± 0.12**	1.33 ± 0	66.02 ± 0.02	65.42 ± 0.04	10.44 ± 0.01	0:00:18.625013
gan	66.71 ± 0.03	1.67 ± 0.34	66.67 ± 0.05	0.09 ± 0.72	66.67 ± 0	0 ± 0	66.84 ± 0.02	6.16 ± 0.17	0:08:07.026274
dcgan	66.69 ± 0.02	1.46 ± 0.1	66.67 ± 0.02	0.03 ± 0.24	66.67 ± 0	0 ± 0	66.85 ± 0	6.08 ± 0	0:14:27.851337
cgan	66.71 ± 0.03	1.61 ± 0.31	66.7 ± 0.05	0.69 ± 0.59	66.67 ± 0	0 ± 0	66.82 ± 0.01	5.57 ± 0.13	0:11:16.547728
wgan	66.71 ± 0.03	1.85 ± 0.31	66.66 ± 0.01	0.03 ± 0.16	66.67 ± 0	0 ± 0	66.84 ± 0	6.16 ± 0.02	1:42:10.098739
sgan	66.66 ± 0.04	0.93 ± 0.41	66.66 ± 0.07	0 ± 0	66.67 ± 0	0 ± 0	**66.85 ± 0.02**	6.11 ± 0.19	0:21:04.588117
ctgan	**66.8 ± 0.08**	1.88 ± 0.65	66.67 ± 0.03	0 ± 0	66.67 ± 0	0 ± 0	49.47 ± 1.16	63.82 ± 0.48	0:38:38.510127
ctabgan	66.67 ± 0.09	0.54 ± 1.13	66.67 ± 0.01	0 ± 0	66.67 ± 0	0 ± 0	52.25 ± 1.22	61.22 ± 1.27	2:06:06.135376

Table 8
Classification performance on dataset Shuttle.

(a) Shuttle1									
Shuttle1 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	99.94 ± 0	99.94 ± 0	81.02 ± 0	86.07 ± 0	11.05 ± 0	67.99 ± 0	76.77 ± 0	82.88 ± 0	0:00:00
random-os	99.87 ± 0	99.87 ± 0	**94.88 ± 0**	**94.85 ± 0**	31.93 ± 0	71.17 ± 0	76.35 ± 0	82.06 ± 0	0:00:00.007007
smote	99.97 ± 0.01	99.97 ± 0.01	94.81 ± 1.59	94.77 ± 1.46	48.16 ± 0.27	74.55 ± 0.06	76.06 ± 0.06	81.37 ± 0.12	0:00:00.053970
smote-b1	99.94 ± 0	99.94 ± 0	92.02 ± 1.07	91.63 ± 0.69	40.06 ± 0.2	72.74 ± 0.05	73.74 ± 0.1	81.49 ± 0.04	0:00:00.188939
smote-b2	99.91 ± 0	99.91 ± 0	89.9 ± 0.58	89.03 ± 0.47	49.81 ± 0.68	74.95 ± 0.19	73.85 ± 0.04	81.77 ± 0.02	0:00:00.190938
svm-smote	**99.98 ± 0**	**99.98 ± 0**	91.36 ± 0.28	90.42 ± 0.35	40.06 ± 0.46	72.74 ± 0.1	74 ± 0.77	81.73 ± 0.28	0:00:01.799422
adasyn	99.94 ± 0	99.94 ± 0	90.25 ± 0.38	89.36 ± 0.33	36.37 ± 0.04	72 ± 0.01	73.82 ± 0.18	81.73 ± 0.06	0:00:00.196954
cure-smote	99.97 ± 0.04	99.97 ± 0.04	90.68 ± 1.55	91.41 ± 1.3	44.4 ± 0.24	73.67 ± 0.05	74.92 ± 1.13	80.38 ± 1.4	0:00:17.932652
pf-smote	99.91 ± 0	99.91 ± 0	87.04 ± 0	89.05 ± 0	49 ± 0	74.75 ± 0	**78.15 ± 0**	**84.06 ± 0**	0:00:00.015623
prowras	99.89 ± 0.04	99.89 ± 0.03	79.26 ± 1.45	85.22 ± 0.75	8.28 ± 0.02	67.64 ± 0	70.06 ± 0.02	81.07 ± 0.01	0:01:10.259488
prowsyn	99.92 ± 0.02	99.92 ± 0.02	91.28 ± 2.4	91.59 ± 2.35	**88.02 ± 0.1**	**84.24 ± 0.18**	77.83 ± 0.06	83.6 ± 0.02	0:00:00.986358
game	99.96 ± 0.02	99.96 ± 0.02	91.46 ± 2.64	90.13 ± 2	85.79 ± 0.02	80.16 ± 0.04	72.98 ± 0.01	81.63 ± 0.01	0:04:08.866247
gan	99.96 ± 0.01	99.96 ± 0.01	80.18 ± 0.29	85.54 ± 0.25	11.12 ± 0.03	68 ± 0	75.08 ± 0.21	82.77 ± 0.09	0:08:54.917476
dcgan	99.96 ± 0.01	99.96 ± 0.01	80.03 ± 0.1	85.32 ± 0.13	11.07 ± 0	67.99 ± 0	74.95 ± 0.12	82.75 ± 0.05	0:15:05.572446
cgan	99.95 ± 0.01	99.95 ± 0.01	80.26 ± 0.24	85.61 ± 0.15	11.1 ± 0	68 ± 0	75.17 ± 0.27	82.81 ± 0.11	0:12:00.799369
wgan	99.95 ± 0.01	99.95 ± 0.01	79.96 ± 0.37	85.34 ± 0.2	11.12 ± 0	68 ± 0	75.4 ± 0.25	82.94 ± 0.1	1:45:08.549852
sgan	99.95 ± 0.01	99.95 ± 0.01	79.97 ± 0.05	85.39 ± 0.05	11.07 ± 0.04	67.99 ± 0.01	75.43 ± 0.01	82.95 ± 0	0:20:19.172990
ctgan	99.92 ± 0.03	99.92 ± 0.03	87.68 ± 3.23	87.47 ± 2.04	83.92 ± 0.55	76.31 ± 1.25	76.75 ± 0.27	82.35 ± 0.29	0:22:18.622084
ctabgan	99.92 ± 0.03	99.92 ± 0.03	81.02 ± 5.99	83.04 ± 5.14	83.93 ± 0.58	76.32 ± 1.3	76.8 ± 0.16	82.41 ± 0.18	5:12:44.322134
(b) Shuttle2									
Shuttle2 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	99.89 ± 0.05	99.89 ± 0.05	79.44 ± 0	85.31 ± 0	1.76 ± 0	66.86 ± 0	74.08 ± 0	82.25 ± 0	0:00:00
random-os	99.87 ± 0	99.87 ± 0	91.39 ± 0	91.74 ± 0	18.39 ± 0	69 ± 0	76.25 ± 0	82.75 ± 0	0:00:00.007998
smote	**99.96 ± 0.05**	**99.96 ± 0.05**	**94.61 ± 0.98**	**94.64 ± 0.84**	35.71 ± 0.2	71.87 ± 0.04	76.51 ± 0.82	82.55 ± 0.28	0:00:00.024990
smote-b1	99.92 ± 0.01	99.92 ± 0.01	91.76 ± 1.38	90.25 ± 1.14	9.78 ± 0.06	67.83 ± 0.01	75.91 ± 1.69	81.54 ± 0.88	0:00:00.135956
smote-b2	99.91 ± 0	99.91 ± 0	90.6 ± 1.04	88.56 ± 1.16	22.46 ± 0.45	69.6 ± 0.07	75.81 ± 0.49	81.46 ± 0.24	0:00:00.136956
svm-smote	99.9 ± 0.02	99.9 ± 0.02	92.47 ± 0.68	91.3 ± 0.88	16.55 ± 0.37	68.73 ± 0.05	75.22 ± 0.26	82.39 ± 0.23	0:00:00.955693
adasyn	99.92 ± 0.01	99.92 ± 0.01	91.85 ± 0.44	90.41 ± 0.45	9.58 ± 0.01	67.8 ± 0	74.06 ± 0.02	82.2 ± 0.02	0:00:00.128958
cure-smote	99.84 ± 0	99.84 ± 0	91.62 ± 0.72	92.15 ± 0.56	30.49 ± 2.02	70.92 ± 0.34	83.97 ± 0.3	80.33 ± 0.57	0:00:02.898723
pf-smote	99.95 ± 0.01	99.95 ± 0.01	88.26 ± 0	89.92 ± 0	38.71 ± 0	72.46 ± 0	81.97 ± 0	85.11 ± 0	0:00:00.015623
prowras	99.73 ± 0.07	99.73 ± 0.06	79.23 ± 0.95	85.23 ± 0.44	1.13 ± 0	66.79 ± 0	74.96 ± 0.15	83.08 ± 0.05	0:01:06.494816
prowsyn	99.8 ± 0.04	99.8 ± 0.04	92.1 ± 1.3	92.45 ± 1.11	87.77 ± 0.08	83.81 ± 0.14	**88.2 ± 0.44**	**86.24 ± 0.79**	0:00:00.378855
game	99.92 ± 0.01	99.92 ± 0.01	92.3 ± 1.92	91.04 ± 2.51	**89.18 ± 0.12**	**86.18 ± 0.21**	73.01 ± 0	81.96 ± 0	0:04:19.781748
gan	99.85 ± 0.06	99.85 ± 0.06	79.4 ± 0.14	85.31 ± 0.08	1.76 ± 0.03	66.86 ± 0	65.39 ± 0.24	79.45 ± 0.08	0:09:28.409815
dcgan	99.92 ± 0.03	99.92 ± 0.03	79.55 ± 0.12	85.39 ± 0.04	1.76 ± 0	66.86 ± 0	65.45 ± 0.33	79.47 ± 0.12	0:21:01.256789
cgan	99.91 ± 0.06	99.91 ± 0.06	79.28 ± 0.16	85.24 ± 0.08	1.77 ± 0	66.87 ± 0	65.53 ± 0.56	79.5 ± 0.2	0:14:49.309013
wgan	99.9 ± 0.06	99.9 ± 0.06	79.13 ± 0.21	85.15 ± 0.14	1.79 ± 0	66.87 ± 0	65.67 ± 0.44	79.55 ± 0.16	1:45:18.196753
sgan	99.85 ± 0.04	99.85 ± 0.04	78.99 ± 0.11	85.11 ± 0.06	1.76 ± 0.04	66.86 ± 0	65.62 ± 0.59	79.53 ± 0.21	0:25:40.101041

(*continued on next page*)

Table 8 (*continued*).

ctgan	99.82 ± 0.07	99.82 ± 0.07	83.79 ± 7.28	85.17 ± 6.32	82.95 ± 0.06	74.14 ± 0.13	79.37 ± 2.69	81.08 ± 1.26	0:44:57.998122
ctabgan	99.84 ± 0.03	99.85 ± 0.03	83.3 ± 6.34	85.67 ± 5.17	83.02 ± 0.69	74.3 ± 1.51	80.69 ± 1.89	81.7 ± 0.86	3:27:02.144177
(c) Shuttle3									
Shuttle3 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	99.58 ± 0.01	99.58 ± 0.01	83.93 ± 0	87.68 ± 0	0.77 ± 0	66.75 ± 0	76.49 ± 0	82.72 ± 0	0:00:00
random-os	99.56 ± 0	99.56 ± 0	91.58 ± 0	91.75 ± 0	12.46 ± 0	68.18 ± 0	76.17 ± 0	81.21 ± 0	0:00:00.007001
smote	99.56 ± 0.05	99.56 ± 0.05	93.1 ± 1.74	93.14 ± 1.66	28.19 ± 0.23	70.52 ± 0.04	77.97 ± 0.21	80.84 ± 0.08	0:00:00.017994
smote-b1	99.5 ± 0	99.51 ± 0	88.5 ± 0.48	87.61 ± 0.61	8.96 ± 0.06	67.73 ± 0.01	84.98 ± 0.37	81.74 ± 0.62	0:00:00.116963
smote-b2	99.88 ± 0	**99.88 ± 0**	88.21 ± 0.39	86.82 ± 0.53	18.03 ± 0.05	68.94 ± 0.01	**86.98 ± 0.22**	**84.24 ± 0.37**	0:00:00.117962
svm-smote	99.59 ± 0.05	99.59 ± 0.05	89.51 ± 1.1	88.49 ± 1.27	13.45 ± 0.41	68.31 ± 0.05	85.32 ± 0.34	81.62 ± 0.58	0:00:00.994681
adasyn	99.88 ± 0	99.88 ± 0	88.85 ± 1.05	88.07 ± 1.28	8.1 ± 0.03	67.62 ± 0	75.1 ± 0.03	80.95 ± 0.04	0:00:00.108964
cure-smote	99.68 ± 0.11	99.68 ± 0.11	85.19 ± 2.7	88.02 ± 1.71	17.77 ± 4.84	68.91 ± 0.68	77.71 ± 2.95	81.14 ± 0.77	0:00:01.200696
pf-smote	99.77 ± 0.02	99.77 ± 0.02	87.51 ± 0	89.65 ± 0	30.83 ± 0	70.98 ± 0	79.1 ± 0	83.71 ± 0	0:00:00.015623
prowras	99.69 ± 0.01	99.7 ± 0.01	76.81 ± 0.35	84.15 ± 0.16	0.63 ± 0	66.74 ± 0	73.75 ± 0.24	82.63 ± 0.11	0:01:11.315836
prowsyn	99.81 ± 0.02	99.81 ± 0.02	92.78 ± 2.65	92.97 ± 2.48	88.21 ± 0.4	84.57 ± 0.7	81.96 ± 0.55	83.91 ± 0.32	0:00:00.253869
game	99.84 ± 0.03	99.84 ± 0.03	96.04 ± 2.8	95.78 ± 3.95	**88.68 ± 0.09**	**85.38 ± 0.14**	73.9 ± 0.01	81.15 ± 0.01	0:04:33.080555
gan	99.54 ± 0.04	99.54 ± 0.04	78.93 ± 0.26	85.09 ± 0.12	0.77 ± 0	66.75 ± 0	51.24 ± 0.39	75.31 ± 0.11	0:09:12.485471
dcgan	99.55 ± 0.02	99.56 ± 0.02	78.17 ± 0.26	84.72 ± 0.13	0.77 ± 0	66.75 ± 0	51.05 ± 0.16	75.27 ± 0.05	0:20:12.853963
cgan	99.55 ± 0.02	99.56 ± 0.02	78.63 ± 0.34	84.91 ± 0.18	0.77 ± 0	66.75 ± 0	51.62 ± 0.19	75.41 ± 0.06	0:13:59.287525
wgan	99.52 ± 0.02	99.52 ± 0.02	78.64 ± 0.27	84.92 ± 0.13	0.77 ± 0	66.75 ± 0	51.86 ± 0.16	75.47 ± 0.05	1:34:11.666271
sgan	99.54 ± 0.04	99.54 ± 0.04	78.93 ± 0.18	85.06 ± 0.09	0.77 ± 0	66.75 ± 0	50.63 ± 0.06	75.16 ± 0.02	0:19:26.567802
ctgan	**99.88 ± 0.03**	99.88 ± 0.03	93.5 ± 4.53	93.54 ± 4.06	83.22 ± 0.04	74.75 ± 0.09	85.59 ± 0.46	82.9 ± 0.51	0:22:45.456215
ctabgan	99.82 ± 0.05	99.82 ± 0.05	**97.51 ± 7.11**	**97.42 ± 5.94**	83.51 ± 0.16	75.4 ± 0.36	79.32 ± 0.95	80.8 ± 0.2	1:45:09
(d) Shuttle4									
Shuttle4 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	99.39 ± 0.02	99.39 ± 0.02	78.59 ± 0	84.98 ± 0	0.2 ± 0	66.69 ± 0	76.32 ± 0	82.33 ± 0	0:00:00
random-os	99.35 ± 0.01	99.36 ± 0.01	90.46 ± 0	90.71 ± 0	9.15 ± 0	67.75 ± 0	77.41 ± 0	81.17 ± 0	0:00:00.008009
smote	99.35 ± 0.06	99.36 ± 0.06	91.41 ± 0.97	91.68 ± 0.91	24.53 ± 0.1	69.93 ± 0.02	83.48 ± 0.49	82.82 ± 1.35	0:00:00.016992
smote-b1	99.58 ± 0.02	99.59 ± 0.02	89.96 ± 0.34	88.56 ± 0.42	7.03 ± 0.03	67.49 ± 0	84.93 ± 0.37	80.88 ± 2.06	0:00:00.109964
smote-b2	**99.93 ± 0.02**	**99.93 ± 0.02**	88.16 ± 0.7	85.78 ± 1.03	15.7 ± 2.05	68.61 ± 0.27	83.61 ± 0.18	78.7 ± 1.38	0:00:00.109959
svm-smote	99.36 ± 0.04	99.37 ± 0.04	91.53 ± 0.83	90.33 ± 1.03	8.98 ± 0.32	67.73 ± 0.04	84.03 ± 0.45	78.53 ± 1.62	0:00:00.775746
adasyn	99.85 ± 0	99.85 ± 0	88.66 ± 1.15	87.24 ± 1.28	5.4 ± 0.06	67.29 ± 0.01	84.89 ± 0.25	84.09 ± 1.25	0:00:00.101973
cure-smote	99.92 ± 0.11	99.92 ± 0.1	91.2 ± 0.82	91.83 ± 0.65	16.34 ± 0.84	68.7 ± 0.12	81.02 ± 2.31	75.71 ± 6.2	0:00:00.022128
pf-smote	99.91 ± 0	99.91 ± 0	90.52 ± 0	91.79 ± 0	23.17 ± 0	69.71 ± 0	**87.7 ± 0**	**87.13 ± 0**	0:00:00.022128
prowras	99.77 ± 0.04	99.77 ± 0.04	76.54 ± 0.78	84.03 ± 0.37	0.34 ± 0	66.7 ± 0	76.67 ± 0.03	83.29 ± 0.01	0:01:14.242587
prowsyn	99.9 ± 0.03	99.9 ± 0.03	91.58 ± 1.21	91.89 ± 1.08	88.67 ± 0.06	85.35 ± 0.1	79.57 ± 0.33	67.95 ± 0.94	0:00:00.200494
game	99.76 ± 0.02	99.76 ± 0.02	94.08 ± 0.23	93.39 ± 0.32	**90.64 ± 0.25**	**88.48 ± 0.06**	75.18 ± 0.01	79.72 ± 0.06	0:05:04.069633
gan	99.36 ± 0.07	99.37 ± 0.07	78.33 ± 0.22	84.85 ± 0.11	0.2 ± 0	66.69 ± 0	28 ± 0.15	70.49 ± 0.02	0:09:10.108637
dcgan	99.37 ± 0.06	99.37 ± 0.06	78.34 ± 0.17	84.86 ± 0.1	0.2 ± 0	66.69 ± 0	27.35 ± 0.09	70.38 ± 0.01	0:21:31.198489
cgan	99.37 ± 0.04	99.37 ± 0.04	77.58 ± 0.27	84.46 ± 0.14	0.2 ± 0	66.69 ± 0	29.78 ± 0.15	70.79 ± 0.03	0:15:14.322630
wgan	99.37 ± 0.1	99.38 ± 0.1	77.75 ± 0.32	84.53 ± 0.16	0.2 ± 0	66.69 ± 0	29.71 ± 0.1	70.78 ± 0.02	1:31:41.557579
sgan	99.37 ± 0.08	99.37 ± 0.08	77.72 ± 0.25	84.53 ± 0.13	0.2 ± 0	66.69 ± 0	27.51 ± 0.08	70.41 ± 0.01	0:25:05.405948
ctgan	99.88 ± 0.04	99.88 ± 0.04	**97.73 ± 2.86**	**97.67 ± 2.76**	83.76 ± 0.07	75.95 ± 0.16	75.38 ± 1.19	54.68 ± 3.69	0:22:41.417333
ctabgan	99.87 ± 0.06	99.87 ± 0.06	97.21 ± 3.73	97.23 ± 3.7	83.52 ± 0.21	75.42 ± 0.46	79.04 ± 2.75	65.74 ± 10	1:45:21.635026
(e) Shuttle5									
Shuttle5 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	99.36 ± 0.05	99.37 ± 0.05	77.56 ± 0	84.48 ± 0	0.08 ± 0	66.68 ± 0	76.24 ± 0	82.62 ± 0	0:00:00
random-os	99.35 ± 0	99.36 ± 0	94.99 ± 0	94.88 ± 0	4.12 ± 0	67.14 ± 0	79.09 ± 0	81.07 ± 0	0:00:00.007997
smote	99.55 ± 0.03	99.55 ± 0.03	94.64 ± 1.01	94.51 ± 0.99	13.79 ± 0.2	68.35 ± 0.03	81.88 ± 0.14	81.8 ± 0.39	0:00:00.012995
smote-b1	99.64 ± 0.11	99.64 ± 0.11	95.86 ± 0.54	95.63 ± 0.58	4.28 ± 0.07	67.16 ± 0.01	83.72 ± 0.41	77.8 ± 1.62	0:00:00.085972
smote-b2	**99.95 ± 0.09**	**99.95 ± 0.09**	93.41 ± 1.54	92.54 ± 1.85	5.62 ± 0.12	67.32 ± 0.02	80.46 ± 0.67	71.22 ± 3.61	0:00:00.086972
svm-smote	99.6 ± 0.06	99.61 ± 0.05	96.34 ± 0.28	96.12 ± 0.33	5.85 ± 0.1	67.34 ± 0.01	85.89 ± 0.18	81.23 ± 0.94	0:00:00.443857
adasyn	99.82 ± 0	99.82 ± 0	95.23 ± 0.96	94.98 ± 1.1	3.27 ± 0.02	67.04 ± 0	83.72 ± 0.18	77.72 ± 0.69	0:00:00.078974
cure-smote	99.55 ± 0.07	99.55 ± 0.07	90.37 ± 2.53	91.02 ± 1.6	7.44 ± 2.17	67.54 ± 0.26	79.03 ± 2.39	73.44 ± 4.76	0:00:00.184874
pf-smote	99.57 ± 0	99.57 ± 0	89.68 ± 0	91.2 ± 0	6.7 ± 0	67.45 ± 0	86.36 ± 0	**87.32 ± 0**	0:00:00.015624
prowras	99.48 ± 0.11	99.49 ± 0.1	72.53 ± 3.85	82.26 ± 1.63	0.05 ± 0	66.67 ± 0	74.62 ± 0.15	82.99 ± 0.05	0:01:19.136955
prowsyn	99.58 ± 0.05	99.58 ± 0.05	96.28 ± 0.92	96.31 ± 0.87	**91.45 ± 0.19**	**89.69 ± 0.28**	84.63 ± 0.13	82.11 ± 0.29	0:00:00.122378
game	99.86 ± 0.03	99.86 ± 0.03	91.5 ± 0.11	89.93 ± 0.14	8.21 ± 0.17	67.55 ± 0.03	**88.93 ± 0.08**	86.19 ± 0.4	0:03:28.141134
gan	99.32 ± 0.07	99.33 ± 0.07	76.44 ± 0.79	83.98 ± 0.36	0.08 ± 0	66.68 ± 0	0.74 ± 0	66.71 ± 0	0:08:25.702111
dcgan	99.37 ± 0.08	99.38 ± 0.08	75.86 ± 0.44	83.69 ± 0.2	0.08 ± 0	66.68 ± 0	0.72 ± 0	66.71 ± 0	0:14:51.770716
cgan	99.48 ± 0.08	99.49 ± 0.08	74.72 ± 0.75	83.2 ± 0.34	0.08 ± 0	66.68 ± 0	0.74 ± 0	66.71 ± 0	0:11:31.271121
wgan	99.4 ± 0.05	99.4 ± 0.05	75.78 ± 0.93	83.67 ± 0.42	0.08 ± 0	66.68 ± 0	0.74 ± 0	66.71 ± 0	1:43:59.276282
sgan	99.35 ± 0.09	99.35 ± 0.09	75.6 ± 0.51	83.59 ± 0.24	0.08 ± 0	66.68 ± 0	0.74 ± 0	66.71 ± 0	0:20:30.953745
ctgan	99.72 ± 0.08	99.73 ± 0.08	**97.43 ± 3.62**	**97.36 ± 3.62**	83.69 ± 0.04	75.79 ± 0.09	83.51 ± 0.79	78.83 ± 1.96	0:22:40.722027
ctabgan	99.83 ± 0.08	99.83 ± 0.08	95.38 ± 4.91	95.01 ± 4.44	83.81 ± 0.23	76.05 ± 0.49	83.95 ± 0.31	81.99 ± 0.82	1:45:35.772426
(f) Shuttle6									
Shuttle6 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	99.27 ± 0.07	99.28 ± 0.06	76.87 ± 0	84.18 ± 0	0.05 ± 0	66.67 ± 0	75.25 ± 0	81.3 ± 0	0:00:00
random-os	99.3 ± 0	99.31 ± 0	94.01 ± 0	94 ± 0	2.28 ± 0	66.92 ± 0	82.42 ± 0	76.64 ± 0	0:00:00.008958
smote	99.25 ± 0.04	99.26 ± 0.04	96.28 ± 3.75	96.27 ± 3.34	7.67 ± 0.08	67.56 ± 0.01	78.99 ± 0.41	69.3 ± 1.08	0:00:00.010012
smote-b1	99.34 ± 0	99.35 ± 0	94.46 ± 2.07	94.26 ± 2.28	2.46 ± 0.07	66.94 ± 0.01	45.42 ± 0.12	72.92 ± 0.07	0:00:00.070976

(*continued on next page*)

Table 8 (*continued*).

smote-b2	99.71 ± 0	99.72 ± 0	89.53 ± 2.61	88.19 ± 3.03	3.64 ± 0.13	67.08 ± 0.02	41.24 ± 1.04	65.53 ± 2.13	0:00:00.070977
svm-smote	99.59 ± 0	99.59 ± 0	96.63 ± 1.8	96.59 ± 1.85	3.03 ± 0.07	67.01 ± 0.01	78.67 ± 1.49	78.64 ± 5.88	0:00:00.295906
adasyn	**99.73 ± 0**	**99.74 ± 0**	90 ± 2.21	89.05 ± 2.36	2.32 ± 0.03	66.93 ± 0	74.4 ± 0.11	73.48 ± 0.03	0:00:00.067989
cure-smote	99.32 ± 0.04	99.33 ± 0.04	83.42 ± 2.56	87.11 ± 1.42	1.69 ± 1.05	66.86 ± 0.12	78.76 ± 0.22	81.54 ± 2.81	0:00:00.084624
pf-smote	99.35 ± 0	99.36 ± 0	84.39 ± 0	87.67 ± 0	2.9 ± 0	67 ± 0	**84.47 ± 0**	**85.38 ± 0**	0:00:00.031247
prowras	98.66 ± 0.46	98.7 ± 0.44	69.96 ± 2.27	81.23 ± 0.88	0.02 ± 0	66.67 ± 0	71.58 ± 0.14	81.72 ± 0.05	0:01:20.870264
prowsyn	99.42 ± 0.06	99.42 ± 0.06	88.75 ± 3.52	89.97 ± 2.96	47.12 ± 0.69	74.3 ± 0.17	80.7 ± 0.14	76.61 ± 0.25	0:00:00.084623
game	99.22 ± 0.08	99.23 ± 0.08	94.97 ± 0.07	94.48 ± 0.08	8.76 ± 0.21	67.63 ± 0.03	66.19 ± 0.03	63.11 ± 0.08	0:04:06.026587
gan	99.46 ± 0.21	99.47 ± 0.2	75.46 ± 0.29	83.54 ± 0.13	0.05 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:08:43.172881
dcgan	99.4 ± 0.04	99.41 ± 0.04	75.44 ± 0.07	83.53 ± 0.03	0.05 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:19:44.241255
cgan	99.38 ± 0.21	99.39 ± 0.2	74.13 ± 0.08	82.92 ± 0.04	0.05 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:14:21.180743
wgan	99.28 ± 0.02	99.29 ± 0.02	75.05 ± 0.12	83.34 ± 0.05	0.05 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	1:30:27.150720
sgan	99.39 ± 0.27	99.39 ± 0.26	74.4 ± 0.14	83.06 ± 0.06	0.05 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:26:10.908405
ctgan	99.3 ± 0.1	99.31 ± 0.1	96.62 ± 0.43	96.62 ± 0.42	83.68 ± 0.04	75.78 ± 0.08	77.94 ± 0.71	70.62 ± 1.29	0:22:37.412358
ctabgan	99.3 ± 0.17	99.31 ± 0.17	**97.22 ± 0.48**	**97.12 ± 0.44**	**84.94 ± 0.31**	**78.45 ± 0.65**	82.25 ± 0.63	77.03 ± 1.4	1:41:13.529192

(g) Shuttle7

Shuttle7 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	99.39 ± 2.77	99.4 ± 2.26	69.77 ± 0	81.15 ± 0	0 ± 0	66.67 ± 0	75.44 ± 0	81.84 ± 0	0:00:00
random-os	99.24 ± 2.16	99.25 ± 1.94	93.52 ± 0	93.52 ± 0	0.05 ± 0	66.67 ± 0	**79.39 ± 0**	69.79 ± 0	0:00:00.008002
smote	99.38 ± 1.76	99.39 ± 1.58	87.01 ± 0.7	88.11 ± 0.51	0.93 ± 0.01	66.77 ± 0	72.79 ± 0.12	52.29 ± 0.04	0:00:00.007985
smote-b1	99.38 ± 1.79	99.39 ± 1.61	85.02 ± 0.35	87.21 ± 0.24	1.36 ± 0.01	66.82 ± 0	73.39 ± 0.06	54.29 ± 0.06	0:00:00.056003
smote-b2	99.86 ± 0	99.86 ± 0	85.09 ± 0.63	86.97 ± 0.4	2.5 ± 0.07	66.95 ± 0.01	70.33 ± 0.32	39.82 ± 0.54	0:00:00.057981
svm-smote	**99.46 ± 1.76**	**99.47 ± 1.58**	82.96 ± 0.65	86.7 ± 0.33	0.85 ± 0.01	66.76 ± 0	73.74 ± 1.35	56.63 ± 0.28	0:00:00.128944
adasyn	99.37 ± 1.95	99.38 ± 1.77	88.17 ± 1.69	88.46 ± 1.74	0.95 ± 0.01	66.77 ± 0	70.6 ± 0.12	42.22 ± 0.21	0:00:00.055978
cure-smote	94.78 ± 1.74	95.27 ± 1.54	84.3 ± 3.62	87.14 ± 2.6	0.63 ± 0.09	66.74 ± 0.01	74.53 ± 1.96	78.97 ± 0.79	0:00:00.031247
pf-smote	95.2 ± 2.14	95.62 ± 1.91	81.49 ± 0	86.13 ± 0	0.45 ± 0	66.72 ± 0	75.51 ± 0	**82.48 ± 0**	0:00:00.015625
prowras	98.19 ± 2.27	98.25 ± 1.95	68.75 ± 1.32	80.74 ± 0.48	0 ± 0	66.67 ± 0	64.94 ± 0.08	79.36 ± 0.03	0:00:47.896837
prowsyn	95.19 ± 2.18	95.61 ± 1.96	84.67 ± 0.67	87.53 ± 0.4	3.19 ± 0.08	67.03 ± 0.01	76.37 ± 0.06	81.84 ± 0.03	0:00:00.015624
game	99.55 ± 0.03	99.55 ± 0.03	92.1 ± 1.46	91.53 ± 1.55	**88.91 ± 0.16**	**85.74 ± 0.02**	71.09 ± 0.04	37.48 ± 0.05	0:05:15.021223
gan	99.34 ± 1.29	99.35 ± 1.14	71.19 ± 4.11	81.71 ± 1.5	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:08:34.740977
dcgan	99.43 ± 1.73	99.43 ± 1.52	68.19 ± 2.4	80.55 ± 1.01	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:15:19.072951
cgan	99.33 ± 1.8	99.33 ± 1.58	72.15 ± 1.11	82.11 ± 0.42	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:12:11.429552
wgan	99.42 ± 1.64	99.43 ± 1.43	69.72 ± 2.94	81.13 ± 1.22	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	1:35:07.246435
sgan	99.26 ± 1.43	99.27 ± 1.26	68.99 ± 1.4	80.85 ± 0.56	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:20:45.562541
ctgan	99.57 ± 0.08	99.58 ± 0.08	80.5 ± 1.14	85.17 ± 0.54	83.19 ± 0.02	74.68 ± 0.04	75.77 ± 0.89	80.38 ± 0.42	0:22:54.247819
ctabgan	99.24 ± 1.9	99.25 ± 1.71	**95.72 ± 5.99**	**95.89 ± 4.05**	18.95 ± 28.84	69.08 ± 4.53	76.92 ± 0.35	81.1 ± 0.14	1:39:27.543390

(h) Shuttle8

Shuttle8 ± std(10^{-2})	DT-maj	DT-min	LR-maj	LR-min	SVM-maj	SVM-min	NB-maj	NB-min	Time
imb-data	99.19 ± 1.73	99.2 ± 1.54	55.07 ± 0	76.33 ± 0	0 ± 0	66.67 ± 0	75.09 ± 0	78.18 ± 0	0:00:00
random-os	95.92 ± 0	96.23 ± 0	89.2 ± 0	89.04 ± 0	0 ± 0	66.67 ± 0	68.4 ± 0	55.08 ± 0	0:00:00.007011
smote	95.92 ± 1.39	96.23 ± 1.14	89.19 ± 0.95	89.36 ± 0.72	0.8 ± 0	66.76 ± 0	68.85 ± 0.27	64.05 ± 0.6	0:00:00.009013
smote-b1	95.92 ± 0	96.23 ± 0	91.92 ± 1.24	92.34 ± 0.72	1.28 ± 0.01	66.81 ± 0	68.52 ± 0.22	63.51 ± 0.44	0:00:00.055969
smote-b2	**99.64 ± 0.02**	**99.64 ± 0.02**	93.36 ± 0.27	92.84 ± 0.29	2.03 ± 0.04	66.9 ± 0	68.52 ± 0.99	57.65 ± 2.29	0:00:00.056002
svm-smote	95.12 ± 0.93	95.56 ± 0.77	92.27 ± 0.95	92.89 ± 0.88	0.77 ± 0.01	66.75 ± 0	69.26 ± 0.45	66 ± 1.02	0:00:00.100960
adasyn	95.12 ± 0.92	95.56 ± 0.77	90.32 ± 0.85	90.53 ± 0.66	0.8 ± 0	66.76 ± 0	71.1 ± 0.03	72.25 ± 0.11	0:00:00.053995
cure-smote	96.03 ± 1.38	96.33 ± 1.13	92.6 ± 0.62	93.11 ± 0.51	0.06 ± 0	66.67 ± 0	77.9 ± 3.68	69.77 ± 7.98	0:00:00.010026
pf-smote	98.11 ± 1.67	98.18 ± 1.45	89.16 ± 0	90.73 ± 0	0.11 ± 0	66.68 ± 0	**86.91 ± 0**	**88.06 ± 0**	0:00:00.010028
prowras	83.03 ± 6.99	87.33 ± 4.11	61.08 ± 3.25	78.1 ± 1.03	0 ± 0	66.67 ± 0	66.35 ± 0.24	79.89 ± 0.09	0:00:47.467494
prowsyn	95.92 ± 0	96.23 ± 0	92.18 ± 0.46	92.79 ± 0.36	0.4 ± 0.03	66.71 ± 0	85.25 ± 0.21	82.3 ± 0.39	0:00:00.010026
game	99.48 ± 0.01	99.48 ± 0.01	**94.09 ± 2.69**	**93.4 ± 2.43**	6.05 ± 0.04	67.35 ± 0	74.01 ± 0.03	74.78 ± 0.1	0:03:39.428752
gan	90.9 ± 1.79	92.3 ± 1.61	55.02 ± 4.86	76.32 ± 1.47	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:08:45.521951
dcgan	90.9 ± 2.56	92.3 ± 2.2	50.18 ± 3.53	75.04 ± 1.15	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:20:49.773649
cgan	93.22 ± 2.3	94.03 ± 1.89	64.65 ± 4.11	79.29 ± 1.21	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:13:27.394133
wgan	90.91 ± 3.45	92.31 ± 2.86	54.65 ± 7.23	76.22 ± 2.42	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	1:33:27.744728
sgan	96.12 ± 2.08	96.4 ± 1.73	72.7 ± 4.14	82.33 ± 1.18	0 ± 0	66.67 ± 0	0 ± 0	66.67 ± 0	0:24:55.031054
ctgan	99.4 ± 0.08	99.41 ± 0.08	80.52 ± 1.63	84.63 ± 1.1	**83.92 ± 0.08**	**76.3 ± 0.18**	83.27 ± 0.71	77.2 ± 1.89	0:21:47.413971
ctabgan	99.47 ± 0.12	99.48 ± 0.12	82.73 ± 3.02	84.26 ± 2.34	1.07 ± 0.27	66.79 ± 0.03	79.79 ± 0.33	71.02 ± 1.02	1:35:30.156954

Appendix

See Tables 3–8.

References

Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019). *IEEE Access*, *9*, 26766–26791.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv e-prints arXiv:1701.07875.

Barua, S., Islam, M. M., & Murase, K. (2013). ProWSyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning. In *Advances in knowledge discovery and data mining: 17th Pacific-Asia conference, PAKDD 2013, gold coast, Australia, april 14-17, 2013, proceedings, part II 17* (pp. 317–328). Springer.

Bej, S., Schulz, K., Srivastava, P., Wolfien, M., & Wolkenhauer, O. (2021). A multi-schematic classifier-independent oversampling approach for imbalanced datasets. *IEEE Access*, *9*, 123358–123374.

Bellinger, C., Sharma, S., Japkowicz, N., & Zaïane, O. R. (2020). Framework for extreme imbalance classification: SWIM—sampling with the majority class. *Knowledge and Information Systems*, *62*, 841–866.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357.

Chen, Z., Duan, J., Kang, L., & Qiu, G. (2021). Class-imbalanced deep learning via a class-balanced ensemble. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15. http://dx.doi.org/10.1109/TNNLS.2021.3071122.

Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). Deepsmote: Fusing deep learning and SMOTE for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 1–15. http://dx.doi.org/10.1109/TNNLS.2021.3136503.

Dai, W., Ng, K., Severson, K., Huang, W., Anderson, F., & Stultz, C. (2019). Generative oversampling with a contrastive variational autoencoder. In *2019 IEEE international conference on data mining (ICDM)* (pp. 101–109).

Du, J., Zhou, Y., Liu, P., Vong, C.-M., & Wang, T. (2021). Parameter-free loss for class-imbalanced deep learning in image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1–7. http://dx.doi.org/10.1109/TNNLS.2021.3110885.

Gazzah, S., & Amara, N. E. B. (2008). New oversampling approaches based on polynomial fitting for imbalanced data sets. In *2008 the eighth iapr international workshop on document analysis systems* (pp. 677–684). IEEE.

Goodfellow, I. (2017). NIPS 2016 tutorial: Generative adversarial networks. arxiv e-prints, arXiv:1701.00160.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *International conference on neural information processing systems* (pp. 2672–2680).

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on advances in intelligent computing* (pp. 878–887).

Hayashi, T., Fujita, H., & Hernandez-Matamoros, A. (2021). Less complexity one-class classification approach using construction error of convolutional image transformation network. *Information Sciences*, [ISSN: 0020-0255] *560*, 217–234.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), IJCNN 2008* (pp. 1322–1328).

Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., & Belongie, S. (2017). Stacked generative adversarial networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1866–1875).

Kovács, G. (2019). An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Applied Soft Computing*, [ISSN: 1568-4946] *83*, Article 105662.

Ma, L., & Fan, S. (2017). CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. *BMC Bioinformatics*, *18*(1), 1–18.

Mehrannia, P., Bagi, S. S. G., Moshiri, B., & Al-Basir, O. A. (2021). Deep representation of imbalanced spatio-temporal traffic flow data for traffic accident detection. CoRR, arXiv:2108.09506.

Mienye, I. D., & Sun, Y. (2021). Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Informatics in Medicine Unlocked*, [ISSN: 2352-9148] *25*, Article 100690.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv e-prints arXiv:1411.1784.

Orooji, A., & Kermani, F. (2021). Machine learning based methods for handling imbalanced data in hepatitis diagnosis. *Frontiers in Health Informatics*, *10*(1), 57.

Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., & Swami, A. (2016). Practical black-box attacks against deep learning systems using adversarial examples. CoRR, arXiv:1602.02697.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv e-prints, arXiv:1511.06434.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., et al. (2016). Improved techniques for training GANs. In *Advances in neural information processing systems 29* (pp. 2234–2242). Curran Associates, Inc..

Wang, Q., Luo, Z., Huang, J., Feng, Y., & Liu, Z. (2017). A novel ensemble method for imbalanced data learning: Bagging of extrapolation-SMOTE SVM. *Computational Intelligence and Neuroscience*, *2017*, 1827016:1–1827016:11.

Xia, S., Zheng, S., Wang, G., Gao, X., & Wang, B. (2021). Granular ball sampling for noisy label classification or imbalanced classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1–12. http://dx.doi.org/10.1109/TNNLS.2021.3105984.

Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling tabular data using conditional GAN. *vol. 32*, In *Advances in neural information processing systems* (pp. 7335–7345).

Xu, Y., Yu, Z., Chen, C. L. P., & Liu, Z. (2021). Adaptive subspace optimization ensemble method for high-dimensional imbalanced data classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14. http://dx.doi.org/10.1109/TNNLS.2021.3106306.

Zhang, T., Chen, J., Li, F., Zhang, K., Lv, H., He, S., et al. (2022). Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions. *ISA Transactions*, *119*, 152–171.

Zhao, Y., Hao, K., Tang, X.-S., Chen, L., & Wei, B. (2021). A conditional variational autoencoder based self-transferred algorithm for imbalanced classification. *Knowledge-Based Systems*, *218*, Article 106756.

Zhao, Z., Kunar, A., Birke, R., & Chen, L. Y. (2021). Ctab-gan: Effective table data synthesizing. In *Asian conference on machine learning* (pp. 97–112). PMLR.