Synthetic Information towards Maximum Posterior Ratio for deep learning on Imbalanced Data

Journal:	*IEEE Transactions on Artificial Intelligence*
Manuscript ID	TAI-2023-Jan-A-00016.R1
Manuscript Type:	Research Article
Date Submitted by the Author:	19-Mar-2023
Complete List of Authors:	Nguyen, Hung; University of South Florida, Chang, Morris; University of South Florida
Keywords:	Deep learning, Data mining, Machine learning

Synthetic Information towards Maximum Posterior Ratio for deep learning on Imbalanced Data

Hung Nguyen* and J. Morris Chang† Department of Electrical Engineering
University of South Florida
Tampa, Florida 33620
Email: *nsh@usf.edu, †chang5@usf.edu

***Abstract*—This work explores how class-imbalanced data affects deep learning and proposes a data balancing technique for mitigation by generating more synthetic data for the minority class. In contrast to random-based oversampling techniques, our approach prioritizes balancing the most informative region by finding high entropy samples. This approach is opportunistic and challenging because well-placed synthetic data points can boost machine learning algorithms' accuracy and efficiency, whereas poorly-placed ones can cause a higher misclassification rate. In this study, we present an algorithm for maximizing the probability of generating a synthetic sample in the correct region of its class by placing it toward maximizing the class posterior ratio. In addition, to preserve data topology, synthetic data are closely generated within each minority sample neighborhood. Overall, experimental results on forty-one datasets show that our technique significantly outperforms experimental methods in terms of boosting deep-learning performance.**

***Impact Statement*—Data class imbalance is a well-known problem in machine learning (ML) applications. This significantly reduces ML algorithms' performance because models are biased toward the majority class. While several strategies have been proposed to mitigate the problem for traditional ML, there is a lack of research for deep learning. In contrast to rule-based ML algorithms, deep learning is highly data-dependent, so understanding how a deep model is affected by data is crucial for finding the mitigations. We provide intuitive studies of different mitigation strategies on deep learning models to fill this gap. Besides a minority oversampling-based technique is proposed to address the problem, a combination of a heuristic technique to find high entropy samples and a conventional statistical theorem to determine where synthetic samples should be spawned. Because our technique is directly designed to tackle the issue of class imbalance for deep learning models, it has been shown to achieve the highest number of winning times (in two metrics, F1-score and AUC) over 41 real datasets compared to the other techniques. The Wilcoxon signed-rank test also shows the significance of the improvement.**

***Index Terms*—data imbalance, deep learning, maximum posterior ratio, high entropy samples**

I. Introduction

Class imbalance is a common phenomenon; it could be caused by the data collecting procedure or simply the nature of the data. For example, it is difficult to sample some rare diseases in the medical field, so collected data for these are usually significantly less than that for other diseases. This leads to the problem of class imbalance in machine learning. The chance of rare samples appearing in model training process is much smaller than that of common samples. Thus, machine learning models tend to be dominated by the majority class; this results in a higher prediction error rate. Existing work also observed that class imbalanced data cause a slow convergence in the training process because of the domination of gradient vectors coming from the majority class [1], [2].

In the last decades, a number of techniques have been proposed to soften the negative effects of class imbalance for conventional machine learning algorithms by analytically studying particular algorithms and developing corresponding strategies. However, the problem for heuristic algorithms such as deep learning is often more difficult to tackle. As suggested in the most recent deep learning with class imbalance survey [3], most of the works are emphasizing image data, and studies for other data types are missing. Thus, in this work, we focus on addressing the issue of tabular data with class imbalance for deep learning models. A class balancing solution is proposed that utilizes entropy-based sampling and data statistical information. As suggested in the survey ([3]) that techniques for traditional ML can be extended to deep learning and inspiring by the comparison in a recent relevant work, Gaussian Distribution Based Oversampling (GDO) [4], we compare the proposed technique with other widely-used and recent techniques such as GDO [4], SMOTE [5], ADASYN [6], Borderline SMOTE [7], DeepSMOTE [8].

Current solutions can be classified into two approaches: model-centric and data-centric. The former strives to alter machine algorithms, while the latter focuses on finding data balancing methods. Perhaps data-centric techniques are more commonly used because they do not tie to any specific model. In this category, a simple data balancing technique is to duplicate minority instances to balance the sample quantity between classes, namely random oversampling (ROS). This can preserve the best data structure and reduce the negative impact of data imbalance to some degree. However, this puts too much weight on a very few minority samples; as a result, it causes over-fitting problems in deep learning when the imbalance ratio becomes higher.

Another widely-used technique in this category is Synthetic Minority Oversampling Technique (SMOTE) [5], which randomly generates synthetic data on the connections (in Euclidean space) between minority samples. However, this easily breaks data topology, especially in high-dimensional space, because it can accidentally connect instances that are not supposed to be connected. In addition, if there are minority samples located in the majority class, the technique will generate sample lines across the decision boundary, which

leads to distorted decision boundaries and misclassification. To improve SMOTE, Hui Han, *et al.* [7] proposed a SMOTE-based technique (Borderline SMOTE), in which they only apply SMOTE on the near-boundary samples determined by the labels of their neighbors. Since this technique is entirely based on Euclidean distance from determining neighbors to generating synthetic data, it performs poorly in high dimensional space. To enhance oversampling with high dimensional data such as images, Dablain *et al.* introduced DeepSMOTE [8] in 2022 which is a combination of SMOTE and a GAN (generative adversarial network). Similar to SMOTE, if there is any poorly generated sample near the boundary, it will worsen the problem due to synthetic samples bridges across the border. Leveraging the same way as SMOTE generates synthetic samples, another widely-used technique, ADASYN [6], controls the number of generated samples by the number of samples in different classes within small groups. Again, this technique still suffers distortion of the decision boundary if the boundary region is class imbalanced. Additionally, such mentioned techniques have not utilized statistical data information. A recent work, Gaussian Distribution Based Oversampling (GDO) [4], balances data class based on the statistical information of data instead. However, its strong assumption of data distribution (data follow Gaussian) reduces the technique's effectiveness in real data.

To alleviate the negative effects of data imbalance and avoid the drawbacks of existing techniques, a minority oversampling technique is proposed that focuses on balancing the high-entropy region that provides the most critical information to the deep learning models. Besides, the technique enhances synthetic data's chance to fall into the minority class to reduce model errors. By carefully generating synthetic data near minority samples, our proposed technique also preserves the best data topology. Besides, our technique does not need any statistical assumption.

To find informative samples, an entropy-based deep active learning technique is used to select samples yielding high entropy to deep learning models. The region of informative samples is denoted as the informative region. This region is balanced first, and the remaining data are balanced later so that it would reduce the decision distortion mentioned earlier. For each minority sample in this region, its synthetic neighbors are safely generated so that the global data topology is still preserved. However, generating synthetic samples in this region is risky because it can easily fall across the decision boundary. Therefore, a direction for synthetic sample location can be chosen by maximizing its posterior probability based on Bayes's Theorem. However, maximizing the posterior probability is facing infeasible computation in the denominator. To overcome this, the posterior ratio is maximized instead so that the denominator computation can be avoided. This also ensures that the synthetic samples are not only close to the minority class but also far from the majority class. The remaining data are eventually balanced by a similar procedure.

The proposed technique alleviates the class imbalance problem. Overall, our experiments indicate that the proposed method can achieve better classification results over widely-used techniques.

Our work has the following main contributions:

1) Exploring the impact of class imbalance mitigations on deep learning via visualization and experiments.
2) Proposing a new minority oversampling-based technique, namely Synthetic Information towards Maximum Posterior Ratio, to balance data classes and alleviate data imbalance impacts. Our technique is enhanced by the following key points.
    a) Leveraging an entropy-based active learning technique to prioritize the region that needs to be balanced. It is the informative region where samples provide high information entropy to the model.
    b) Leveraging Maximum Posterior Ratio and Bayes's theorem to determine the direction to generate synthetic minority samples to ensure the synthetic data fall into the minority class and not fall across the decision boundary. To our best knowledge, this is the first work utilizing the posterior ratio for tackling class imbalanced data.
    c) Approximating the likelihood in the posterior ratio using kernel density estimation, which can approximate a complicated topology. Thus, the proposed technique is able to work with large, distributively complex data.
    d) Carefully generating synthetic samples surrounding minority samples so that the global data topology is still preserved.
3) The proposed technique is evaluated against commonly utilized and contemporary techniques across 41 actual datasets that vary in terms of imbalance ratio and feature count. The findings demonstrate that the proposed approach exhibits superior performance compared to others, on the whole.

The rest of this paper is organized as follows. Section II briefly review other existing works. Section III introduces related concepts that will be used in this work, i.e., Imbalance Ratio, Macro F1-score, Area Under the Curve (AUC), and Entropy-based active learning. Section IV will provide more detail on the problem of learning from an imbalanced dataset. Our proposed solution to balance dataset, Synthetic Information towards Maximum Posterior Ratio, will be explained comprehensively in Section V. Section VI discusses the technique implementation and complexity. Section VII shows experiments on different datasets, including artificial and real datasets. Experimental results are also discussed in the same section. Section VIII concludes the study and discusses future work.

II. Related Work

In the last few decades, many solutions have been proposed to alleviate the negative impacts of data imbalance in machine learning. However, most of them are not efficiently extended for deep learning. This section reviews algorithms to tackle class-imbalanced data that can be extended for deep learning. These techniques can be categorized into three main categories, i.e., sampling, cost-sensitive, and ensemble learning approaches.

A. Sampling-based approach.

Compared to other approaches, resampling techniques have attracted more research attention as they are independent of machine learning algorithms. This approach can be divided into two main categories, over-sampling, and under-sampling techniques. Such sampling-based methods e.g., [9]–[15] mainly generate a balanced dataset by either over-sampling the minority class or down-sampling the majority class. Some techniques are not designed for deep learning; however, they are still considered in this work since they are independent of the machine learning model architecture. In a widely used method, SMOTE [5], Chawla *et al.* attempt to oversample minority class samples by connecting a sample to its neighbors in feature space and arbitrarily drawing synthetic samples along with the connections. However, one of SMOTE drawbacks is that if there are samples in the minority class located in the majority class, it creates synthetic sample bridges toward the majority class [16]. This renders difficulties in differentiation between the two classes. Another SMOTE-based work, namely Borderline-SMOTE [7] was proposed in which its method aims to do SMOTE with only samples near the border between classes. The samples near the border are determined by the labels of its k distance-based neighbors. This "border" idea is similar to ours to some degree. However, finding a good k is critical for a heuristic machine learning algorithm such as deep learning, and it is usually highly data-dependent.

Among specific techniques for deep learning, generating synthetic samples in the minority class by sampling from data distribution is becoming more attractive as they outperform other methods in high dimensional data [17]. Regarding images, several deep learning generative-based methods have been proposed as deep learning is capable of capturing good image representations. [18] [19] [20] utilized Variational Autoencoder as a generative model to arbitrarily generate images from learned distributions. However, most assumed simple prior distributions, such as Gaussian for minor classes, tend to simplify data distribution and might fail in more sophisticated distributions. In addition, most of the works in this approach tackle image datasets, while our proposed method focuses on tabular datasets as this is a missing piece in the field [3].

Under the down-sampling category, existing techniques mainly down-sample the majority class to balance it with the minority class. There are several proposed techniques to simplify the majority. A straightforward way is to randomly remove the majority class samples. Other works, e.g., [21], [22] find near-border samples and authors believe the imbalance ratio in these areas is much smaller than that in the entire dataset. They then classify this small pool of samples to improve the performance and expedite the training process for the SVM-based method. However, this method was only designed for SVM-based methods, which mainly depend on the support vectors. Also, this potentially discards essential information of the entire dataset because only a small pool of data is used.

B. Cost-sensitive learning approach.

Cost-sensitive learning techniques usually require modifications of algorithms on the cost functions to balance each class's weight. Specifically, such cost-sensitive techniques put higher penalties on majority classes and less on minority classes to balance their contribution to the final cost. For example, [23] provided their designed formula $(1-\beta^n)/(1-\beta)$ to compute the weight of each class based on the effective number of samples n and a hyperparameter β which is then applied to re-balance the loss of a convolutional neural network model. [24], [25], [26] assign classes' weights inversely proportional to sample frequency appearing in each class. Hamed et al. [27] proposed an SVM-based cost-sensitive approach (SVMCS) that uses svm with a class-weighted loss function.

C. Ensemple learning approach.

Ensemble learning has achieved high performance in classification for its generalizability. Thus, it could reduce the bias due to class imbalance. Ensemble learning can be constructed by combining several base classifiers with different sampling-based approaches. In [28], [29], Chawla *et al.* and Seiffert *et al.* proposed variants of ensemble learning in which the data are balanced based on oversampling method SMOTE and then applying ensemble learning on balanced data. Similarly, authors in [30] generate cluster-based synthetic data and combine it with an evolutionary algorithm. Liu *et al.* in [31] balances the data by applying a fuzzy-based oversampling technique and building ensemble learning classifiers on this data. Zhou and Liu in [32] explore a method, namely Easy Ensemble classifier (EE), to perform ensemble learning on the random under-sampling balanced data.

III. Preliminaries

In this section, we introduce relevant concepts that will be utilized in our research.

A. Imbalance Ratio (IR)

For binary classification problems, imbalance ratio (IR) is used to depict the data imbalance as it has been widely used. IR is the ratio of the majority class samples to the minority class's samples. For example, if a dataset contains 1000 class-A and 100 class-B samples, the Imbalance Ratio is 10:1.

B. Evaluation Metrics

In this work, classification performance is used for evaluating techniques. Specifically, F1-Score and Area Under the Curve (AUC) are used for evaluation metrics. For F1-scores, Macro-averaging is measured as it is more relevant for evaluating imbalance datasets. F1 score is computed based on two factors Recall and Precision as follows:

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$F1 - score = \frac{2 * Recall * Precision}{Recall + Precision}, \tag{3}$$

where T and F stand for True and False; P and N stand for Positive and Negative.

Besides, AUC [33] score is computed as it is an important metric to evaluate imbalanced data. AUC is derived from the Receiver Operating Characteristic curve (ROC). In this work, a skit-learn library to compute AUC is utilized; the library can be found in sklearn.metrics.auc.

C. Entropy-based Active Learning

Entropy-based active learning (AL) is leveraged to find informative samples. The technique gradually selects batch-by-batch samples that provide high information to the model based on information entropy theory [34]. The information entropy is quantified based on the "surprise" to the model in terms of class prediction probability. Take a binary classification, for example; if a sample is predicted to be 50% belonging to class A and 50% belonging to class B, this sample has high entropy and is informative to the model. In contrast, if it is predicted to be 100% belonging to class A, it is certain and gives zero information to the model. The class entropy E for each sample can be computed as follows.

$$E(x,\theta) = -\sum_{i}^{n} P_\theta(y = c_i|x) \log_n P_\theta(y = c_i|x) \quad (4)$$

where $P_\theta(y = c_i|x)$ is the probability of data x belonging to the ith class of n classes with current model parameter θ.

We consider a dataset containing N pairs of samples X and corresponding labels y, and a deep neural network with parameter θ. AL implementation requires repeated phases, and a batch of informative data is selected for each phase. At the first phase $t^{(0)}$, a classifier is trained with parameter $\theta^{(0)}$ (Note that this classifier differs from the classifier for the final classification problem) on a random initial batch of labeled samples and use the model $\theta^{(0)}$ to estimate the entropy of the remaining data.

The entropy scores are then estimated for the remaining samples based on Equation 4. The first batch of informative samples is determined by selecting k highest entropy samples. This batch is then concatenated with the initial training data for the training classifier parameter ($\theta^{(1)}$) in the next phase ($t^{(1)}$) and also accumulated to the informative set. In the next phase, similarly, the updated model is then used to estimate the entropy of the remaining data. The next informative batch is selected and also added to the informative set. Phases are repeated until the number of accumulated informative samples reaches a pre-set informative portion (IP). For example, $IP = 0.3$ will select 30% training samples as informative samples.

IV. The Problem of Learning From Imbalanced Datasets

In this section, a concise overview of the challenge of acquiring knowledge from imbalanced datasets is presented. Although the problem may apply to different machine learning methods, this study only focus on deep learning.

Figure 1 illustrates our problem on binary classification. The imbalance in the informative region (light blue eclipse) could lead to classification errors. The dashed green line depicts the expected boundary, while the solid blue line is the model's boundary. Since the minority class lacks data in this region, the majority class will dominate the model even with a few noisy poorly-placed samples, which leads to a shift of the model's boundary. In contrast to the study by Ertekin *et al.* [21], which assumes the informative region is more balanced by nature and proposes a solution that only classifies over the informative samples, our assumption is different. This work contemplates the scenario where the informative region comprises extensively imbalanced data, which we believe is common in most real-world scenarios. The problem could be more severe in a more complex setting, such as high-dimensional and topologically complex data. Therefore, we proposed a technique to tackle the problem by oversampling the minority class in an informative manner. The detail of the technique will be described in Section V.

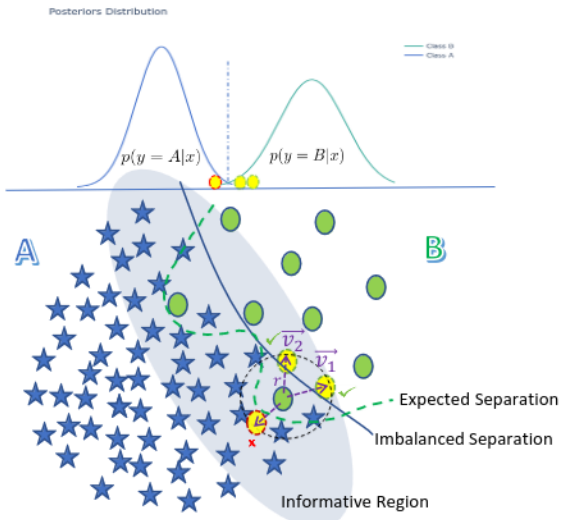


Fig. 1: Learning from imbalanced datasets

V. Methodology

To alleviate the negative effects of data imbalance, we propose a comprehensive approach, Synthetic Information towards Maximum Posterior Ratio (SIMPOR), which aims to generate synthetic samples for minority classes. First, the informative region that contains informative samples is determined and balanced by creating surrounding synthetic neighbors for minority samples. The remaining region is then fully balanced by arbitrarily generating minority samples' neighbors. The remainder of this section provides further information about how our approach was developed.

A. Methodology Motivation

As Chazal and Michel mentioned in their work [35], the natural way to highlight the global topological structure of the data is to connect data points' neighbors; our proposed method aligns with their observation by generating surrounding synthetic neighbors for minority samples to preserve data topology. Thus, our technique not only generates more data

for minority class but also preserve the underlying topological structure of the entire data.

Similar to [21] and [22], we believe that informative samples play the most important role in the prediction success of both traditional machine learning models (e.g., SVM, Naive Bayes) and modern deep learning approaches (e.g., neural network). Thus, our technique finds these informative samples and focuses on augmenting minority data in this region. In this work, an entropy-based active learning strategy mentioned in III-C is applied to find the samples that contain more information to the model. This strategy is perhaps the most popular active learning technique and over-performs many other techniques on several datasets [36], [37] [38].

B. Generating minority synthetic data

A synthetic neighbor x' and its label y' can be created surrounding a minority sample x by adding a small random vector v to the sample, $x' = x + v$. This lays on the d-sphere surface centered by x, and the d-sphere's radius is set by the length of vector $\vec{v}$, $|\vec{v}|$. It is, however, critical to generate synthetic data in the informative region because synthetic samples can unexpectedly jump across the decision boundary. This can be harmful to models as this might create outliers and reduce the model's performance. Therefore, we safely find vector $\vec{v}$ towards the minority class, such as $\vec{v}_0$ and $\vec{v}_1$ depicted in Figure 1. Our technique is described via a binary classification scenario as follows.

Let's consider a binary classification problem between majority class A and minority class B. From the Bayes' theorem, the posterior probabilities $p(y' = A|x')$ or $p(y' = B|x')$ can be used to present the probabilities that a synthetic sample x' belongs to class A or class B, respectively. Let the two posterior probabilities be f_0 and f_1; they can be expressed as follows.

$$p(y' = A|x') = \frac{p(x'|y' = A)\, p(A)}{p(x')} = f_0 \tag{5}$$

$$p(y' = B|x') = \frac{p(x'|y' = B)\, p(B)}{p(x')} = f_1 \tag{6}$$

As mentioned earlier, each synthetic data x' is generated so that it maximizes the probability of x' belonging to the minority class B and minimizes the chance x' falling into the majority class A. Thus, a technique that maximizes the fractional posterior f is proposed,

$$f = f_1/f_0 \tag{7}$$

$$= \frac{p(x'|y' = B)\, p(B)}{p(x'|y' = A)\, p(A)}. \tag{8}$$

Approximation of likelihoods in Equation 8: A non-parametric kernel density estimates (KDE) is selected to approximate the likelihoods $p(x'|y' = A)$ and $p(x'|y' = B)$ as KDE is flexible and does not require specific assumptions about the data distribution. One can use a parametric statistical model such as Gaussian to approximate the likelihood; however, it oversimplifies the data and does not work effectively with topological complex data, especially in high dimensions. In addition, parametric models require an assumption about the distribution of data which is difficult in real-world problems since we usually do not have such information. On the other hand, KDE only needs a kernel working as a window sliding through the data. Among different commonly used kernels for KDE, we choose Gaussian Kernel as it is a powerful continuous kernel that would also eases the derivative computations for finding optima.

Approximation of priors in Equation 8: Additionally, we estimate the prior probabilities of observing samples in class A ($p(A)$) and class B ($p(B)$) (in Equation 8) by the widely-used Empirical Bayes Method [39] to leverage the existing information from the original data. The estimates are denoted as $\widehat{p(A)}$ and $\widehat{p(B)}$ respectively.

Equation 8 Approximation: Let X_A and X_B be the subsets of dataset X which contain samples of class A and class B, $X_A = \{x : y = A\}$ and $X_B = \{x : y = B\}$. N_A and N_B are the numbers of samples in X_A and X_B. d is the number of data dimensions. h presents the width parameter of the Gaussian kernel. The posterior ratio for each synthetic sample x' then can be estimated as follows:

$$f = \frac{p(x'|y' = B)\, p(B)}{p(x'|y' = A)\, p(A)} \tag{9}$$

$$\propto \frac{\frac{1}{N_B h^d} \sum_{i=1}^{N_B} (2\pi)^{-\frac{d}{2}}\, e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}\, \widehat{p(B)}}{\frac{1}{N_A h^d} \sum_{j=1}^{N_A} (2\pi)^{-\frac{d}{2}}\, e^{\frac{1}{2}(\frac{x - X_{A_j}}{h})^2}\, \widehat{p(A)}} \tag{10}$$

$$\propto \frac{\frac{1}{N_B h^d} \sum_{i=1}^{N_B} e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}\, \widehat{p(B)}}{\frac{1}{N_A h^d} \sum_{j=1}^{N_A} e^{\frac{1}{2}(\frac{x - X_{A_j}}{h})^2}\, \widehat{p(A)}} \tag{11}$$

Selecting bandwidth parameter h for Gaussian kernel: The bandwidth is automatically selected for each dataset using the most common method, namely Scott's rule of thumb, proposed by Scott [40]. With an attempt to minimize the mean integrated squared error, the parameter is estimated as $h = N^{(-\frac{1}{d+4})}$ where N, d are the number of data points and the number of dimensions respectively. This study utilizes a scikitlearn python library for KDE, including bandwidth selection. The implementation detail can be found at [41].

Finding synthetic samples surrounding a minority sample: To generate neighbors for each minority sample that maximizes Function f in Equation 11, points on the r-radius sphere centered at a minority sample are considered synthetic instances. As a result, a vector $\vec{v}$ can be added to a minority sample for generating a new instance. The relationship between a synthetic sample x' and a minority sample can be described as follows,

$$\vec{x'} = \vec{x} + \vec{v}, \tag{12}$$

where $|\vec{v}| = r$, and r is sampled from a Gaussian distribution,

$$r \sim \mathcal{N}(0,\, (\alpha R)^2), \tag{13}$$

where αR is the standiviation of the Gaussian distribution and $0 < \alpha <= 1$. The range parameter R is relatively small and computed as the average distance of a minority sample x to its k-nearest neighbors. This will ensure that the generated sample will be surrounding the minority sample. The Gaussian

distribution with the mean of zero and the standiviation αR controls the distance between the synthetic samples and the minority sample. The standiviation is tuned from 0 to R by a coefficient $\alpha \in (0, 1]$. The larger the α is, the farther synthetic data created from its original sample. Consider a minority sample x and its k-nearest neighbors in the Euclidean space, R can be computed as follows:

$$R = \frac{1}{k}\sum_{1}^{k} ||x - x_j||, \tag{14}$$

where $||x - x_j||$ is the Euclidean distance between a minority sample x and its jth neighbor. k is a parameter indicating selected number of neighbors.

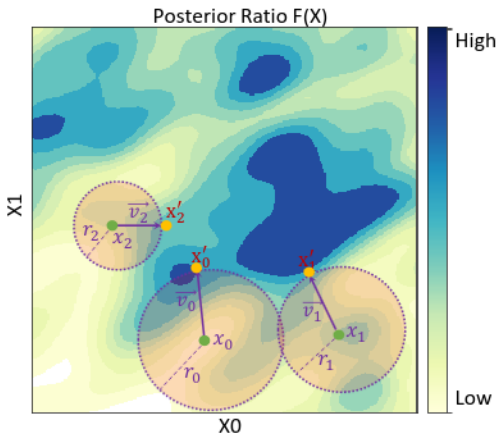


Fig. 2: Demonstration on how SIMPOR generates three synthetic samples x'_0, x'_1, x'_2, from three minority samples x_0, x_1, x_2, by maximizing the Posterior Ratio.

Figure 2 depicts a demonstration of finding 3 synthetic samples from 3 minority samples. In fact, one minority can be re-sampled to generate more than one synthetic sample. For a minority sample x_0, we find a synthetic sample x'_0 by maximizing the objective function $f(x'_0), x'_0 \in X$ with a constraint that the Euclidean length of $\vec{v_0}$ equals to a radius r_0, $||\vec{v_0}|| = r_0$ or $||\vec{x'_0} - \vec{x_0}|| = r_0$ (derived from Equation 12).

The problem can be described as a constrained optimization problem. For each minority sample x, we find a synthetic sample $x' \in \mathbb{R}^d$ lying on the d-sphere centered at x with radius r and maximizing function in Equation 11,

$$\max_{x'} f(x') \quad \text{s.t. } ||\vec{x'} - \vec{x}|| = r. \tag{15}$$

Solving optimization problem in Equation 15: Interestingly, the problem in Equation15 can be solved numerically. Function $f(x)$ in Equation 11 is defined and continuous for $x' \in (-\infty, +\infty)$ because all of the exponential components (Gaussian kernels) are continuous and greater than zero. In addition, the constraint, $||\vec{x'} - \vec{x}|| = r$, which contains all points on the sphere centered at x with radius r is a closed set ([42]). Thus, a maximum exits as proved in [43]. To enhance the diversity of synthetic data, either the global maximum or any local maximum can be accepted so that the synthetic samples will not simply go to the same direction.

We solve the problem in Equation 15 by using the Projected Gradient Ascent approach in which we iteratively update the parameter to go up the gradient of the objective function. A local maximum is found if the objective value cannot be increased by any local update. For simplification, we rewrite the problem in Equation 15 by shifting the origin to the considered minority sample. The problem becomes finding the maximum of function $f(x')$, $x' \in \mathbb{R}^d$, constrained on a d-sphere, i.e., $||x'|| = r$. Our solution can be described in Algorithm 1. After shifting the coordinates system, we start by sampling a random point on the constraint sphere (line $1-2$). The gradient of the objective function at time t, $g_t(x'_t)$, is computed and projected onto the sphere tangent plane as p_t (line $4-5$). It is then normalized and used for update a new x'_{t+1} by rotating a small angle $lr * \theta$ (line $6-7$). The algorithm stops when the value of $f(x')$ is not increased by any update of x'. We finally shift to the original coordinates and return the latest x'_t.

Algorithm 1 Sphere-Constrained Gradient Ascent for Finding Maximum

Input: A minority sample x_0, objective function $f(x, X)$
Parameter:
r : The radius of the sphere centered at x_0
θ : Sample space $\theta \in [0, 2\pi]$
lr : Gradient ascent learning rate
Output: An local maximum x'

1: Shift the Origin to x_0
2: Randomly initiate x'_t on the sphere with radius r
3: **while** converge condition **do**
4: Compute the gradient at x'_t
$g_t(x'_t) = \nabla f(x'_t)$
5: Project the gradient onto the sphere tangent plane
$p_t = g_t - (g_t \cdot x'_t)x_t$
6: Normalize projected vector
$p_t = p_t / ||p_t||$
7: Update x' on the constrained sphere
$x'_{t+1} = x'_t cos(lr * \theta) + p_t sin(lr * \theta)$
8: **end while**
9: Shift back to the Origin
10: **return** x'_t

Avoiding synthesis of noise: To reduce the chance of misplacing synthetic samples on another class region because of noisy borderline and mislabeled minority samples, we set a policy for rejecting minority candidates which are selected for oversampling. The idea is to reject candidates surrounded mainly by other class samples. More specifically, we count the labels of the candidate's k-nearest neighbors and reject this candidate if there exists a class that its' number of samples is greater than the number of the minority samples). For example, the candidate is rejected when a class-A sample is selected for generating synthetic data, and its 5-nearest neighbors contain four class-B samples and one class-A sample. This is to avoid

selecting mislabeled samples and noisy borderline samples for oversampling.

C. Algorithm

Our strategy can be described in Algorithm 2. The algorithm takes an imbalanced dataset as its input and results in a balanced dataset which is a combination of the original dataset and synthetic samples. We first choose an active learning method $AL(\cdot)$ and find a subset of informative samples S by leveraging entropy-based active learning (lines $1-2$). We then generate synthetic data to balance S. For each random sample x_i^c in S and belonging to minority class c, we randomly sample a small radius r and find a synthetic sample that lies on the sphere centered at x_i^c and maximizes the posterior ratio in Equation 11 (lines $3-11$). The process is repeated until the informative set S is balanced. Similarly, the remaining region is balanced, which can be described in the pseudo-code from line 12 to line 20. The final output of the algorithm is a balanced dataset D'.

VI. Algorithm Time Complexity.

The costly part of SIMPOR is that each synthetic sample requires computing a kernel density estimation of the entire dataset. Elaborately, let n be the number of samples of the dataset. In the worst case, the numbers of samples of minority and majority class are $N_B = 1$ and $N_A = n-1$, respectively. We need to generate $n-2$ synthetic samples to balance the dataset completely. Since each generated sample must loop through the entire dataset of size n to estimate the density, the algorithm complexity is $O(n^2)$.

Although generating synthetic data is only a one-time process, and this does not affect the classification efficiency in the testing phase, we still try to alleviate its weakness by providing parallelized implementations to reduce the time complexity to $O(n)$. Specifically, each exponential component in Equation 11 is computed parallelly, utilizing GPU or CPU threads. Ellaborately, Equation 11 can be rewritten as N_B components of $e^{\frac{1}{2}(\frac{x-X_{B_i}}{h})^2}$ and N_A components of $e^{\frac{1}{2}(\frac{x-X_{A_i}}{h})^2}$. Fortunately, they are all independent and can be processed parallelly. Thus, with a sufficient hardware resource, the consumption time for the kernel density estimation of each synthetic data point is then reduced by $N_A + N_B = n$ times, which significantly simplifies the complexity to $O(n)$.

VII. Experiments

In this section, we explore the techniques via binary classification problems on an artificial dataset (i.e., Moon) and 41 real-world datasets (i.e., KEEL, UCI, Credit Card Fraud) with a diversity of imbalance ratios and different numbers of features. Samples in Moon have two features, while other datasets contain various numbers of features and imbalance ratios. Dataset details are described in Table I. The implementation steps to balance datasets follow Algorithm 2. To evaluate our proposed balancing technique, we compare the classification performance to different widely-used and state-of-the-art techniques. More specifically, We compare

Algorithm 2 SIMPOR

Input: Original Imbalance Dataset D including data X and labels y.
Parameter: MA is the majority class, MI is a set of other classes.
k: Number of neighbors of the considered sample which determines the maximum range of the sample to its synthetic samples.
α: preset radius coefficient $Count(c, P)$: A function to count class c sample number in population P.
$G(x_0, f, r)$: Algorithm 1, which returns a synthetic sample on sphere centered at x_0 with radius r and maximize Equation 11.
Output: Balanced Dataset D' including $\{X', y'\}$

1: Select an Active Learning Algorithm $AL()$
2: Query a subset of informative samples $S \in D$ using AL: $s \leftarrow AL(D)$
{Balance the informative region}
3: **for** $c \in MI$ **do**
4: **while** $Count(c, S) \leq Count(MA, S)$ **do**
5: Select a random $x_i^c \in S$
6: Reject and reselect x_i^c if its label is dominated among k-nearest labels
7: Compute maximum range R based on k-nearest neighbors
8: Randomly sample a radius $r \sim \mathcal{N}(0, \alpha R)$
9: Generate a synthetic neighbor x' from x_i^c: $x' = G(x_i^c, f, r)$
10: Append x' to D'
11: **end while**
12: **end for**
{Balance the remaining region}
13: **for** c in MI **do**
14: **while** $Count(c, D') \leq Count(MA, D')$ **do**
15: Select a random $x_j^c \in \{X - S\}$
16: Compute maximum range R based on k
17: Randomly sample a radius $r \sim \mathcal{N}(0, \alpha R)$
18: Generate a synthetic neighbor x' of x_j^c
19: Append x' to D'
20: **end while**
21: **end for**
22: **return**

SIMPOR to SMOTE [5], Borderline-SMOTE [7], ADASYN [6], DeepSMOTE [8], Gaussian Distribution Based Oversampling (GDO) [4], SVMCS [27], EE [32]. To evaluate the classifications performance for skewed datasets, we measure widely-used metrics, i.e., F1-score and Area Under The Curve (AUC).

A. Implementation Detail

This section describes the general settings and implementation details for the experimental techniques.

1) SIMPOR settings: In order to find the informative subset, we leverage entropy-based active learning. We first utilize

TABLE I: Dataset Description.

dataset	#samples	#features	IR
glass1	214	9	1.8 (138:76)
wisconsin	683	9	1.9 (444:239)
pima	768	8	1.9 (500:268)
glass0	214	9	2.1 (144:70)
yeast1	1484	8	2.5 (1055:429)
haberman	306	3	2.8 (225:81)
vehicle1	846	18	2.9 (629:217)
vehicle2	846	18	2.9 (628:218)
vehicle3	846	18	3.0 (634:212)
creditcard	1968	30	3.0 (1476:492)
glass-0-1-2-3_vs_4-5-6	214	9	3.2 (163:51)
vehicle0	846	18	3.3 (647:199)
ecoli1	336	7	3.4 (259:77)
new-thyroid1	215	5	5.1 (180:35)
new-thyroid2	215	5	5.1 (180:35)
ecoli2	336	7	5.5 (284:52)
glass6	214	9	6.4 (185:29)
yeast3	1484	8	8.1 (1321:63)
ecoli3	336	7	8.6 (301:35)
page-blocks0	5472	10	8.8 (4913:559)
yeast-2_vs_4	514	8	9.0 (463:51)
yeast-0-5-6-7-9_vs_4	528	8	9.4 (477:51)
vowel0	988	13	10.0 (898:90)
glass-0-1-6_vs_2	192	9	10.3 (175:17)
glass2	214	9	11.6 (197:17)
yeast-1_vs_7	459	7	14.3 (429:30)
glass4	214	9	15.5 (201:13)
ecoli4	336	7	15.8 (316:20)
page-blocks-1-3_vs_4	472	10	15.9 (444:28)
abalone9-18	731	8	16.4 (689:42)
yeast-1-4-5-8_vs_7	693	8	22.1 (663:30)
glass5	214	9	22.8 (205:9)
yeast-2_vs_8	482	8	23.1 (462:20)
car_eval_4	1728	21	25.6 (1663:65)
wine_quality	4898	11	25.8 (4715:183)
yeast_me2	1484	8	28.0 (1433:51)
yeast4	1484	8	28.1 (1433:51)
yeast-1-2-8-9_vs_7	947	8	30.6 (917:30)
yeast5	1484	8	32.7 (1440:44)
yeast6	1484	8	41.4 (1449:35)
abalone19	4174	8	129.4 (689:42)

a neural network model playing a role as a classifier to find high-entropy samples (Note that the classifier for finding the informative subset differs from the classifiers for the final classification evaluation after all balancing techniques are applied to the data). The detailed steps are introduced in Section III-C. The model contains two fully connected hidden layers with *relu* activation functions and 10 neurons in each layer. The output layer applies the soft-max activation function. The model is trained in a maximum of 300 epochs with an early stop option when the loss is not significantly improved after updating weights. The model is trained firstly on a random set of three samples each class (six samples two classes). This model is then used to estimate entropy scores for the remaining data. We then select next 20 highest entropy samples (k=20) for the next informative data batch. This batch is concatenated to the initial batch for updating the classifiers and accumulated to the informative set. The steps are repeated until the informative set reaches desire informative portion (IP). In these experiments, we set IP=0.3 corresponding to 30 percent of the training size selected for the informative set.

To solve the optimization problem in Equation 15 for finding optima (this differs from the classification optimization for the evaluation) introduced in Section V-B, we use a gradient ascent method with the gradient rate of $1e-5$ and the maximum iteration of 300.

2) Evaluation Classification settings: Considering each imbalanced dataset as a classification problem, we use the classification testing performance for the technique comparison. Each dataset is randomly split into two parts, 80% for training and 20% for testing. The classifiers are trained on training sets after applying the techniques. The results are reported on the raw testing sets (There isn't any technique applied on the testing sets; thus, they are also possibly class imbalanced). We use F1-score and AUC for the evaluation metrics as they are suitable and widely used to evaluate imbalanced data. Reported testing results for each dataset are the averages of 5 experimental trials.

The classifiers are constructed by neural networks with the input and output sizes corresponding to the number of datasets' features and unique labels. We use the same classifier structure (number of hidden layers, number of neurons in each layer, learning rate, optimizer) for all compared datasets. The detail of neural network implementation is described in Table II. For baseline technique settings, we follow the experimental parameter sets in [4] as we share very similar datasets and comparison techniques. For DeepSMOTE settings, the DCGAN input and output sizes are modified to adapt with each dataset, while other settings is taken from the initial parameter set in [8].

TABLE II: Classification models' setting for each dataset.

Method	Parameter
SIMPOR	k_neighbors=5, r_distribtuion=Gaussian(0,1), IP=0.3
GDO	k_neighbors=5, d=1
SMOTE	k_neighbors=5, sampling_strategy='auto',random_state=None
BL-SMOTE	k_neighbors=5, sampling_strategy='auto', random_state=None
ADASYN	k_neighbors=5, sampling_strategy='auto', random_state=None
EE	#estimators=10, Estimater=AdaBoostClassifier
DeepSMOTE	Sigma=1, Lambda=0.1
Classifier	**Parameter**
Architecture	neuron/layer=100, #layers=3
Optimization	optimizer='adam', epochs=200, batch_size=32, learning_rate=0.1, reduce_lr_loss(factor=0.9,epsilon=1e-4,patience=5)

B. SIMPOR on artificial Moon dataset

We implement techniques on an artificial 2-dimension dataset for demonstration purposes. We first generate the balanced synthetic MOON dataset using python library *sklearn.datasets.make_moons*. The generated MOON contains 3000 samples labeled in two classes, and each instance has two numerical features with values ranging from 0 to 1. We then make the dataset artificially imbalanced with an Imbalance Ratio of 7:1 by randomly removing 1285 samples from one class. As a result, the training dataset becomes imbalanced, as visualized in Figure 3.

Figure 4 captures the classification for different techniques. We also visualize the model decision boundaries to provide additional information on how the classification models are affected. We use a fully connected neural network described in Table II to classify the data.

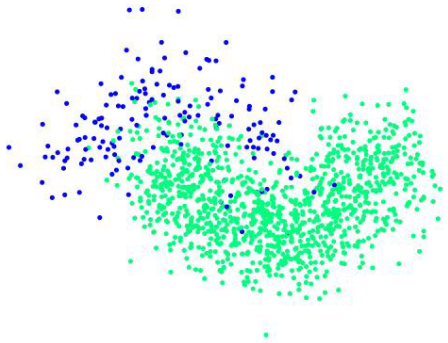


Fig. 3: Artificial class imbalanced Moon dataset with IR of 7:1.

TABLE III: Classification Result on Moon Dataset.

Metric	SIMPOR	SMOTE	BL-SMOTE	DeepSMOTE	ADASYN	GDO
F1-score	0.883	0.824	0.827	0.842	0.785	0.817
AUC	0.961	0.957	0.955	0.959	0.955	0.959

1) Results and Discussion: From the visualization shown in Figure 4 and the classification performance results in Table III, it is clear that SIMPOR performs better than others by up to 10% on F1-score and 1.1% on AUC. We can see that DeepSMOTE (DeepSM) creates dense squared noise and pushes the decision boundary to the majority class. Due to the fact that SMOTE-based methods does not take the informative region into account, unbalanced data in this area lead to a severe error in decision boundary. In Figures 4f and 4e, BorderlineSMOTE (BL-SMOTE) and ADASYN focus on the area near the model's decision boundary, but they inherit a drawback from SMOTE; any noise or mislabeled samples can, unfortunately, create very dense bridges crossing the expected border and lead to decision errors. Figure 4b shows that GDO also generates local gaussian groups of samples near the boder and thus create errors. This phenomenon might cause by a few mis-labeled sample points. In contrast, by generating neighbors of minority samples in the direction towards the minority class and balancing the informative region, SIMPOR (Figure 4a) helps the classifier to make a better decision with a solid smooth decision boundary. Poorly-placed synthetic samples are significantly less than that of others.

C. SIMPOR on forty-one real datasets

In this section, we compare the proposed technique on 41 real two-class datasets with a variable number of features and Imbalance Ratios, i.e., KEEL datasets [44], [45], UCI datasets fetched from Sklearn tool [46], [47] and Credit Card Fraud [48] dataset. Since the original Credit Card Fraud contains a large number of banking normal and fraud transaction samples (284,807) which significantly reduces our experimental efficiency, we reduced the dataset size by randomly removing normal class transactions to reach an imbalance ratio of 3.0. Other datasets are kept as their original versions after removing bad samples (containing Null values). The datasets are described in Table I.

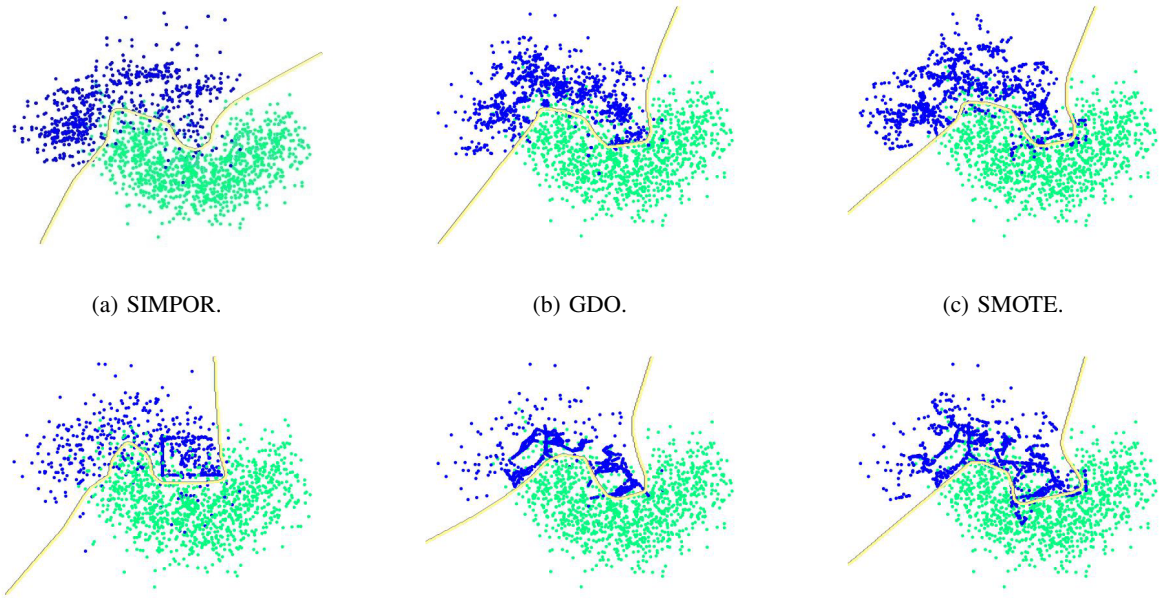
TABLE IV: F1-score over different datasets.

	SIMPOR	GDO	SMOTE	BL-SMOTE	ADASYN	DeepSM	SVMCS	EE
glass1	0.729	**0.741**	0.707	0.729	0.729	0.706	0.719	0.705
wisconsin	**0.962**	0.959	0.953	0.958	0.956	0.960	0.958	0.957
pima	**0.777**	0.699	0.714	0.720	0.700	0.721	0.742	0.731
glass0	**0.840**	0.799	0.804	0.795	0.806	0.813	0.835	0.811
yeast1	**0.715**	0.676	0.675	0.685	0.672	0.673	0.685	0.678
haberman	**0.601**	0.599	0.589	0.587	0.580	0.587	0.586	0.584
vehicle1	**0.824**	0.815	0.807	0.796	0.817	0.785	0.784	0.808
vehicle2	**0.987**	0.967	0.977	0.978	0.981	0.954	0.976	0.981
vehicle3	**0.821**	0.766	0.785	0.792	0.806	0.780	0.782	0.785
creditcard	**0.954**	0.935	0.946	0.944	0.943	0.947	0.907	0.939
glass-0-1-2-3_vs_4-5-6	0.850	0.923	0.918	0.912	0.915	**0.929**	0.907	0.905
vehicle0	0.933	0.956	**0.970**	0.965	0.965	0.952	**0.970**	0.969
ecoli1	0.831	0.822	0.838	0.818	0.815	**0.853**	0.824	0.827
new-thyroid1	0.970	**0.979**	0.946	0.953	0.953	0.902	0.946	0.946
new-thyroid2	0.962	**0.982**	0.938	0.938	0.938	0.872	0.930	0.930
ecoli2	**0.922**	0.880	0.905	0.864	0.884	0.887	0.909	0.914
glass6	**0.952**	0.899	0.875	0.880	0.869	0.864	0.880	0.880
yeast3	0.862	0.818	0.842	0.836	0.829	0.831	0.867	**0.879**
ecoli3	0.806	0.791	0.790	0.792	0.792	**0.829**	0.827	0.824
page-blocks0	**0.926**	0.904	0.909	0.900	0.900	0.913	0.919	0.912
yeast-2_vs_4	0.883	0.875	**0.893**	0.844	0.866	0.817	0.807	0.772
yeast-0-5-6-7-9_vs_4	**0.824**	0.752	0.754	0.781	0.758	0.747	0.813	0.805
vowel0	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.997	**1.000**	0.997
glass-0-1-6_vs_2	**0.771**	0.692	0.733	0.725	0.707	0.524	0.685	0.646
glass2	0.737	0.717	**0.839**	0.805	0.801	0.779	0.701	0.666
yeast-1_vs_7	**0.710**	0.663	0.595	0.654	0.608	0.614	0.681	0.691
glass4	0.795	0.871	0.846	0.850	0.859	**0.892**	0.811	0.819
ecoli4	**0.909**	0.841	0.893	0.883	0.883	0.863	0.893	0.893
page-blocks-1-3_vs_4	0.982	0.944	0.964	0.972	0.964	0.982	**0.990**	**0.990**
abalone9-18	0.777	0.763	0.760	0.767	0.773	0.752	**0.817**	0.792
yeast-1-4-5-8_vs_7	0.593	**0.637**	0.584	0.618	0.628	0.487	0.489	0.489
glass5	0.912	0.843	0.792	**0.919**	0.780	0.792	0.633	0.633
yeast-2_vs_8	**0.884**	0.758	0.746	0.772	0.750	0.823	0.876	0.876
car_eval_4	**1.000**	0.967	0.997	0.994	0.997	0.993	**1.000**	**1.000**
wine_quality	**0.753**	0.681	0.660	0.690	0.674	0.669	0.675	0.674
yeast_me2	0.701	0.638	0.668	0.655	0.656	0.690	**0.707**	0.702
yeast4	**0.793**	0.698	0.682	0.690	0.671	0.738	0.752	0.752
yeast-1-2-8-9_vs_7	**0.775**	0.642	0.612	0.633	0.607	0.698	0.750	0.750
yeast5	0.667	0.854	0.871	0.877	**0.881**	0.786	0.839	0.844
yeast6	**0.745**	0.734	0.730	0.724	0.705	0.696	0.708	0.738
abalone19	0.498	0.500	**0.526**	0.518	0.524	0.497	0.498	0.498

TABLE V: AUC result over different datasets.

	SIMPOR	GDO	SMOTE	BL-SMOTE	ADASYN	DeepSM	SVMCS	EE
glass1	0.798	**0.818**	0.788	0.782	0.807	0.804	0.800	0.795
wisconsin	**0.995**	0.992	0.992	0.992	0.992	0.991	**0.995**	0.994
pima	**0.858**	0.790	0.800	0.804	0.782	0.810	0.826	0.818
glass0	**0.901**	0.879	0.885	0.859	0.873	0.891	0.896	0.882
yeast1	**0.811**	0.758	0.753	0.753	0.746	0.776	0.782	0.774
haberman	0.675	0.662	0.660	0.673	0.667	0.686	0.686	**0.689**
vehicle1	**0.936**	0.917	0.922	0.920	0.929	0.924	0.920	0.928
vehicle2	**0.999**	0.998	**0.999**	**0.999**	**0.999**	0.991	**0.999**	**0.999**
vehicle3	**0.918**	0.871	0.895	0.897	0.900	0.901	0.903	0.904
creditcard	**0.974**	0.969	0.966	0.962	0.962	0.961	0.968	0.945
glass-0-1-2-3_vs_4-5-6	0.968	0.987	**0.989**	0.976	0.988	0.987	0.985	0.985
vehicle0	0.975	0.991	0.995	0.995	**0.996**	0.992	0.995	**0.996**
ecoli1	0.949	0.948	**0.952**	0.942	0.943	0.951	0.951	0.950
new-thyroid1	**0.999**	**0.999**	0.997	0.997	0.997	0.982	0.997	0.997
new-thyroid2	**0.999**	**0.999**	0.998	0.998	0.997	0.977	0.998	**0.999**
ecoli2	0.950	0.953	0.957	0.946	0.958	0.957	0.959	**0.960**
glass6	**0.963**	0.960	0.920	0.833	0.841	0.894	0.939	0.877
yeast3	**0.968**	0.943	0.935	0.927	0.937	0.946	0.966	0.967
ecoli3	0.883	0.879	0.878	0.880	0.883	0.891	**0.897**	0.885
page-blocks0	**0.990**	0.986	0.969	0.982	0.984	0.981	0.986	0.986
yeast-2_vs_4	0.974	**0.976**	0.961	0.959	0.960	0.907	0.972	0.949
yeast-0-5-6-7-9_vs_4	0.915	**0.923**	0.881	0.904	0.866	0.876	0.918	0.914
vowel0	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
glass-0-1-6_vs_2	**0.942**	0.897	0.905	0.892	0.910	0.886	0.907	0.941
glass2	0.929	0.917	0.923	0.923	**0.952**	0.919	0.940	0.932
yeast-1_vs_7	**0.848**	0.777	0.677	0.761	0.685	0.702	0.791	0.795
glass4	0.955	0.976	0.954	**0.987**	0.949	0.979	0.972	0.975
ecoli4	**0.997**	0.978	0.984	0.984	0.989	0.953	0.990	0.988
page-blocks-1-3_vs_4	**1.000**	0.997	0.999	0.999	0.999	0.999	0.999	0.999
abalone9-18	0.934	0.920	0.933	0.929	0.919	0.898	0.930	**0.940**
yeast-1-4-5-8_vs_7	**0.823**	0.746	0.721	0.734	0.721	0.721	0.769	0.754
glass5	0.987	**0.990**	0.985	0.983	0.987	0.987	**0.990**	0.988
yeast-2_vs_8	0.855	0.845	0.835	**0.865**	0.853	0.802	0.809	0.805
car_eval_4	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
wine_quality	**0.852**	0.783	0.727	0.756	0.740	0.793	0.781	0.805
yeast_me2	**0.896**	0.887	0.793	0.817	0.787	0.832	0.871	0.874
yeast4	**0.935**	0.889	0.796	0.810	0.792	0.817	0.832	0.848
yeast-1-2-8-9_vs_7	**0.761**	0.699	0.702	0.687	0.685	0.695	0.745	0.756
yeast5	0.835	0.991	0.985	0.984	0.985	0.983	0.992	**0.993**
yeast6	0.960	0.936	0.905	0.946	0.906	0.933	0.959	**0.963**
abalone19	**0.782**	0.557	0.616	0.676	0.575	0.721	0.763	0.771

1) Classification results: Table IV, V, VI and VII show the classification F1-score, AUC, Precision, and Recall results, respectively. The highest scores for each dataset are highlighted in bold style. We also provide the summary of the F1 and AUC scores by "winning times" scores. We count the number of datasets for which a technique achieves the highest scores among the compared techniques and name this number "winning times". For convention, if more than two techniques share the same highest score, the winning



(a) SIMPOR. (b) GDO. (c) SMOTE.

(d) DeepSMOTE. (e) BorderlineSMOTE. (f) ADASYN.

Fig. 4: Data plot and model's decision boundary visualization for Moon Dataset over different techniques.

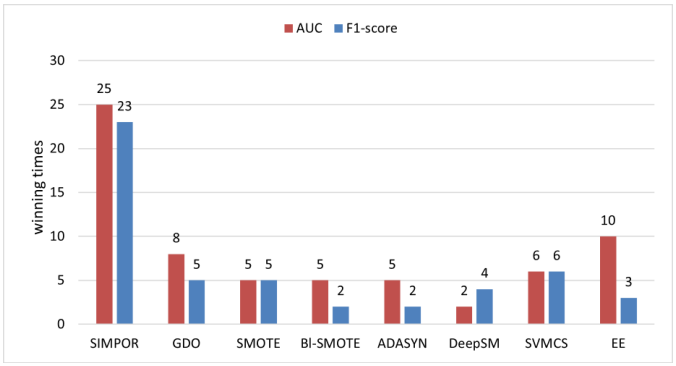


Fig. 5: Winning times over 41 datasets.

times will be increased for each technique. Figure 5 shows a summary of winning times.

As we can see from the table, the proposed technique outperforms others on both evaluation metrics, F1-score and AUC. More specifically, SIMPOR hits 23 F1-score winning times and 25 AUC winning times. Its number of F1-score winning times at 23 four times better than the second winner (SVMCS) at 6, and its AUC winning times at 25 doubles the second AUC winners (EE) at 10.

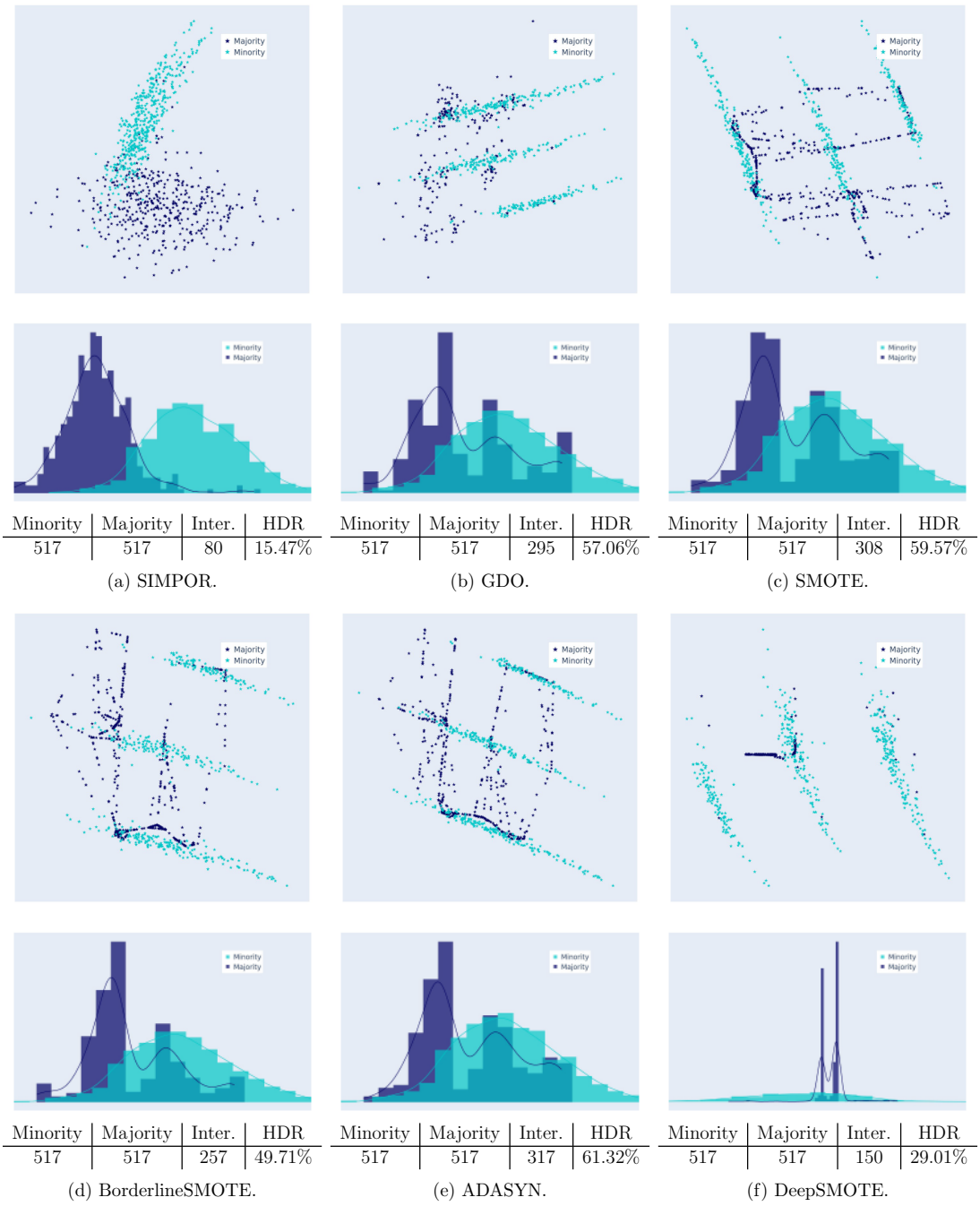
D. Statistical Test.

To further evaluate the effectiveness of the technique, we also performed a Wilcoxon Signed Rank Test [49] on the 41 dataset results (F1 score and AUC). Wilcoxon hypothesis test is relevant to our study as it is a non-parametric statistical test and does not require a specific distribution assumption for the results. On the other hand, 41 data points (corresponding to 41 datasets results) are sufficient to support this test. Our null hypothesis is that the difference between the proposed technique results and those of the other technique is insignificant. Wilcoxon signed-rank test outputs are computed over the 41 dataset results and return a p-value for each technique pair. We then compare the p-value with the significant value $\alpha = 0.05$. Suppose the p-value is smaller than α. In that case, the evidence is sufficient to reject the hypothesis, which means the proposed technique does make a significant difference from the others, and vice versa. Table VIII shows the Wilcoxon p-value results.

As we can see from Table VIII, the p-values are all smaller than the critical value of 0.05. Thus, the null hypothesis can be rejected as the supporting evidence is sufficient. In other words, the statistical result shows that the proposed technique makes a significant improvement compared to others.

E. Data visualization

To explore more on how the techniques perform, we visualize the generated data by projecting them onto lower dimension space (i.e., one and two dimensions) using the Principle Component Analysis technique (PCA) [50]. Data's 2-Dimension (2D) plots and 1-Dimension histograms are presented with a hard-to-differentiate ratio (HDR) for each technique. 1D histograms are computed by dividing one-dimensional-reduced data into 20 bins (intervals) and counting the number of samples within the interval of each bin. A hard-to-differentiate ratio is defined as the ratio of the number of samples in the intersection between 2 classes to the total of minority samples ($HDR = \frac{No.\,Intersection\,samples}{No.\,Minority\,samples}100\%$) where the number of intersection samples is estimated by counting samples in the overlapped bins between the two classes in the 1D histograms. This ratio is expected to be as



Minority	Majority	Inter.	HDR
517	517	80	15.47%

(a) SIMPOR.

Minority	Majority	Inter.	HDR
517	517	295	57.06%

(b) GDO.

Minority	Majority	Inter.	HDR
517	517	308	59.57%

(c) SMOTE.

Minority	Majority	Inter.	HDR
517	517	257	49.71%

(d) BorderlineSMOTE.

Minority	Majority	Inter.	HDR
517	517	317	61.32%

(e) ADASYN.

Minority	Majority	Inter.	HDR
517	517	150	29.01%

(f) DeepSMOTE.

Fig. 6: Abalone9-18: Generated training data projected onto 2-dimension space and their histograms in 1-Dimension space using Principle Component Analysis dimension reduction technique. The bottom tables illustrate the number of samples in two classes, 1-Dimension histogram intersection between 2 classes, and the hard-to-differentiate ratio between the number of intersection samples to the number of minority samples ($HDR = \frac{Inter.}{Minority}100\%$).

small as 0% if the two classes are well separated; in contrast, 100% indicates that the two classes cannot be distinguished in the projected 1D space. Besides HDR, we show the absolute numbers of Minority, Majority, and Intersection samples for each technique in the bottom tables. From the plots, we observe how the data are distributed in 2D space and quantify samples that are hard to be differentiated in the 1D space histograms.

To save space, we only show the plot of one dataset (i.e., Abalone9-18 dataset) in Figure 6. Many other datasets are observed to have similar patterns. We observe that the proposed technique does not poorly generate synthetic samples as many as other techniques do. HDR results show that SIMPOR achieves the least number of hard-to-differentiate

TABLE VI: Precision results over 41 datasets.

	SIMPOR	GDO	SMOTE	BL-SMOTE	ADASYN	DeepSM	SVMCS	EE
glass1	0.733	**0.744**	0.710	0.732	0.730	0.710	0.726	0.711
wisconsin	**0.959**	0.955	0.952	0.956	0.954	0.957	0.956	0.955
pima	**0.776**	0.694	0.714	0.715	0.700	0.727	0.752	0.737
glass0	**0.836**	0.782	0.796	0.782	0.795	0.809	0.825	0.808
yeast1	**0.727**	0.674	0.670	0.680	0.668	0.687	0.702	0.689
haberman	**0.610**	0.588	0.578	0.574	0.570	0.592	0.603	0.608
vehicle1	**0.835**	0.803	0.811	0.804	0.819	0.797	0.793	0.817
vehicle2	**0.986**	0.954	0.970	0.973	0.977	0.944	0.971	0.977
vehicle3	**0.826**	0.760	0.779	0.797	0.807	0.788	0.794	0.798
creditcard	**0.961**	0.933	0.955	0.953	0.951	0.958	0.904	0.954
glass-0-1-2-3_vs_4-5-6	0.845	0.915	0.922	0.915	0.911	**0.925**	0.907	0.911
vehicle0	0.921	0.935	0.960	0.960	0.956	0.959	**0.971**	0.969
ecoli1	0.810	0.795	0.811	0.792	0.784	**0.827**	0.813	0.816
new-thyroid1	**0.977**	0.964	0.950	0.953	0.953	0.903	0.950	0.950
new-thyroid2	**0.974**	0.971	0.966	0.966	0.966	0.886	0.963	0.963
ecoli2	**0.928**	0.849	0.901	0.852	0.867	0.867	0.915	0.924
glass6	0.977	0.922	0.941	**0.980**	0.926	0.920	**0.980**	**0.980**
yeast3	0.868	0.777	0.825	0.815	0.786	0.829	0.867	**0.876**
ecoli3	0.791	0.726	0.739	0.743	0.734	0.806	**0.842**	0.838
page-blocks0	0.929	0.858	0.897	0.870	0.854	**0.936**	0.922	0.913
yeast-2_vs_4	0.874	0.846	**0.907**	0.864	0.865	0.830	0.750	0.818
yeast-0-5-6-7-9_vs_4	0.835	0.688	0.741	0.755	0.721	0.748	0.861	**0.868**
vowel0	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.999	**1.000**	0.999
glass-0-1-6_vs_2	**0.822**	0.710	0.760	0.728	0.720	0.531	0.664	0.619
glass2	0.731	0.653	**0.765**	0.764	0.723	0.758	0.739	0.649
yeast-1_vs_7	**0.779**	0.613	0.580	0.632	0.584	0.693	0.740	0.759
glass4	0.756	0.845	0.910	0.927	0.853	**0.985**	0.887	0.899
ecoli4	**0.896**	0.792	0.879	0.862	0.862	0.808	0.879	0.879
page-blocks-1-3_vs_4	0.967	0.903	0.935	0.949	0.935	0.969	**0.982**	**0.982**
abalone9-18	0.806	0.706	0.717	0.731	0.731	0.762	**0.903**	0.866
yeast-1-4-5-8_vs_7	**0.668**	0.574	0.557	0.582	0.579	0.479	0.479	0.479
glass5	0.912	0.823	0.824	**0.954**	0.803	0.824	0.673	0.673
yeast-2_vs_8	0.913	0.728	0.670	0.752	0.682	0.825	**0.921**	**0.921**
car_eval_4	**1.000**	0.939	0.994	0.988	0.994	0.999	**1.000**	**1.000**
wine_quality	**0.758**	0.659	0.682	0.713	0.682	0.727	0.723	0.736
yeast_me2	0.737	0.593	0.666	0.652	0.641	0.738	**0.852**	0.843
yeast4	0.800	0.635	0.640	0.660	0.629	0.738	0.829	**0.831**
yeast-1-2-8-9_vs_7	0.909	0.598	0.587	0.620	0.589	0.842	0.957	**0.988**
yeast5	0.629	0.759	0.804	0.807	**0.821**	0.753	0.807	0.816
yeast6	0.785	0.662	0.671	0.729	0.657	0.714	0.740	**0.801**
abalone19	0.497	0.498	0.516	**0.517**	0.511	0.497	0.497	0.497

TABLE VII: Recall results over 41 datasets.

	SIMPOR	GDO	SMOTE	BL-SMOTE	ADASYN	DeepSM	SVMCS	EE
glass1	0.725	**0.739**	0.705	0.726	0.727	0.702	0.713	0.699
wisconsin	**0.965**	0.963	0.955	0.961	0.959	0.962	0.960	0.958
pima	**0.778**	0.705	0.714	0.727	0.701	0.715	0.733	0.726
glass0	0.844	0.817	0.814	0.810	0.818	0.817	**0.847**	0.815
yeast1	**0.704**	0.679	0.680	0.691	0.677	0.660	0.669	0.668
haberman	0.593	**0.611**	0.601	0.599	0.590	0.583	0.570	0.563
vehicle1	0.814	**0.827**	0.803	0.789	0.816	0.774	0.777	0.800
vehicle2	**0.988**	0.980	0.983	0.983	0.985	0.965	0.982	0.986
vehicle3	**0.817**	0.772	0.791	0.787	0.806	0.772	0.770	0.773
creditcard	**0.948**	0.937	0.937	0.936	0.935	0.936	0.911	0.925
glass-0-1-2-3_vs_4-5-6	0.857	0.932	0.915	0.910	0.920	**0.935**	0.909	0.901
vehicle0	0.947	**0.979**	**0.979**	0.971	0.973	0.946	0.969	0.969
ecoli1	0.854	0.852	0.866	0.847	0.849	**0.880**	0.837	0.839
new-thyroid1	0.965	**0.995**	0.942	0.953	0.953	0.901	0.942	0.942
new-thyroid2	0.951	**0.995**	0.913	0.913	0.913	0.859	0.900	0.900
ecoli2	**0.916**	0.913	0.911	0.878	0.902	0.910	0.903	0.905
glass6	**0.931**	0.880	0.818	0.800	0.820	0.816	0.800	0.800
yeast3	0.856	0.863	0.862	0.860	0.878	0.834	0.869	**0.883**
ecoli3	0.823	**0.870**	0.850	0.850	0.861	0.855	0.816	0.814
page-blocks0	0.923	**0.955**	0.924	0.932	0.952	0.892	0.917	0.913
yeast-2_vs_4	0.895	**0.908**	0.881	0.827	0.868	0.810	0.875	0.741
yeast-0-5-6-7-9_vs_4	0.814	**0.832**	0.770	0.815	0.802	0.771	0.772	0.752
vowel0	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.995	**1.000**	0.995
glass-0-1-6_vs_2	**0.741**	0.680	0.735	0.735	0.730	0.520	0.716	0.682
glass2	0.747	0.800	**0.939**	0.869	0.907	0.812	0.677	0.689
yeast-1_vs_7	0.655	**0.722**	0.611	0.678	0.634	0.562	0.633	0.635
glass4	0.841	**0.906**	0.795	0.795	0.872	0.821	0.759	0.761
ecoli4	**0.934**	0.901	0.911	0.909	0.909	0.928	0.911	0.911
page-blocks-1-3_vs_4	0.998	0.992	0.996	0.997	0.996	0.998	**0.999**	**0.999**
abalone9-18	0.751	**0.830**	0.812	0.812	0.823	0.748	0.751	0.733
yeast-1-4-5-8_vs_7	0.549	**0.723**	0.618	0.665	0.697	0.495	0.499	0.499
glass5	**0.912**	0.868	0.768	0.893	0.766	0.768	0.602	0.602
yeast-2_vs_8	**0.872**	0.835	0.858	0.838	0.858	0.845	0.848	0.848
car_eval_4	**1.000**	0.997	**1.000**	0.999	**1.000**	0.986	**1.000**	**1.000**
wine_quality	**0.750**	0.706	0.639	0.670	0.667	0.620	0.634	0.623
yeast_me2	0.671	**0.693**	0.670	0.658	0.673	0.647	0.610	0.609
yeast4	**0.790**	0.776	0.732	0.725	0.722	0.741	0.690	0.690
yeast-1-2-8-9_vs_7	0.685	**0.694**	0.641	0.649	0.627	0.602	0.622	0.604
yeast5	0.716	**0.979**	0.955	0.964	0.956	0.830	0.883	0.883
yeast6	0.714	**0.827**	0.813	0.723	0.776	0.691	0.681	0.691
abalone19	0.499	0.503	0.537	0.518	**0.539**	0.497	0.500	0.500

TABLE VIII: Wilcoxon Signed Rank Hypothesis Test results.

	p-value	
SIMPOR vs.	F1-score	AUC
GDO	1.82E-03	1.93E-03
SMOTE	2.66E-03	6.64E-05
BL_SMOTE	4.22E-03	1.63E-04
ADASYN	2.89E-03	6.13E-04
DeepSM	6.40E-04	2.57E-04
SVMCS	2.74E-03	3.40E-02
EE	1.99E-03	2.17E-02

ratio at 15.47%. As shown in the 2D visualization sub-figures, other techniques poorly-place synthetic data crossed the other class. This causes by outliers or noises near the border between the two classes that other techniques do not pay attention to and mistakenly create more noise. In contrast, SIMPOR safely produces synthetic data towards the minority class by maximizing the posterior ratio ;thus it can reduce the number of poorly-placed samples.

TABLE IX: Processing time (in seconds) over 41 datasets.

	SIMPOR	GDO	SMOTE	BL-SMOTE	ADASYN	DeepSM
glass1	0.1147	0.0576	0.0020	0.0033	0.0032	0.8587
wisconsin	2.0805	0.1769	0.0024	0.0044	0.0046	1.2004
pima	0.2032	0.2066	0.0025	0.0049	0.0050	1.2297
glass0	0.2157	0.0553	0.0023	0.0035	0.0036	0.8601
yeast1	0.2457	0.4749	0.0035	0.0108	0.0104	1.4846
haberman	0.0517	0.1560	0.0022	0.0033	0.0036	0.9246
vehicle1	0.4365	0.1237	0.0025	0.0059	0.0059	1.4147
vehicle2	6.2913	0.1512	0.0029	0.0053	0.0061	1.3976
vehicle3	0.2821	0.1237	0.0024	0.0060	0.0061	1.3487
creditcard	2.1200	0.3783	0.0087	0.0184	0.0182	1.7980
glass-0-1-2-3_vs_4-5-6	0.3376	0.0459	0.0023	0.0035	0.0035	0.8407
vehicle0	7.3645	0.1198	0.0024	0.0054	0.0058	1.2953
ecoli1	0.0418	0.0337	0.0010	0.0018	0.0017	0.9310
new-thyroid1	0.5352	0.0304	0.0015	0.0024	0.0024	0.8590
new-thyroid2	0.3881	0.0359	0.0025	0.0033	0.0031	0.8747
ecoli2	0.2516	0.0266	0.0011	0.0017	0.0016	0.9733
glass6	0.3196	0.0268	0.0014	0.0025	0.0023	1.0744
yeast3	0.1374	0.2422	0.0023	0.0060	0.0059	1.6699
ecoli3	0.0658	0.0378	0.0015	0.0025	0.0024	0.9647
page-blocks0	7.9654	2.0918	0.0045	0.0143	0.0138	3.6029
yeast-2_vs_4	2.4310	0.0624	0.0017	0.0028	0.0028	1.0286
yeast-0-5-6-7-9_vs_4	0.0868	0.0632	0.0016	0.0029	0.0027	0.9809
vowel0	4.7675	0.1312	0.0018	0.0039	0.0037	1.2410
glass-0-1-6_vs_2	0.0482	0.0207	0.0013	0.0023	0.0022	0.9133
glass2	0.0501	0.0227	0.0013	0.0024	0.0024	0.8855
yeast-1_vs_7	0.4697	0.0420	0.0017	0.0026	0.0026	1.0355
glass4	0.1141	0.0197	0.0012	0.0024	0.0023	0.9469
ecoli4	0.1087	0.0310	0.0015	0.0024	0.0024	0.9393
page-blocks-1-3_vs_4	1.8742	0.0445	0.0015	0.0027	0.0026	0.9992
abalone9-18	2.9722	0.0716	0.0015	0.0028	0.0026	1.2095
yeast-1-4-5-8_vs_7	0.0881	0.0673	0.0017	0.0031	0.0028	1.0803
glass5	0.2815	0.0241	0.0017	0.0033	0.0036	0.8550
yeast-2_vs_8	0.1239	0.0441	0.0016	0.0027	0.0028	0.9849
car_eval_4	0.4381	0.1746	0.0026	0.0066	0.0049	1.6616
wine_quality	0.1622	0.8587	0.0030	0.0144	0.0137	3.3128
yeast_me2	0.1060	0.1379	0.0018	0.0042	0.0039	1.7350
yeast4	0.1083	0.1386	0.0018	0.0041	0.0039	1.6314
yeast-1-2-8-9_vs_7	0.0924	0.0757	0.0017	0.0031	0.0030	1.3069
yeast5	0.1188	0.1312	0.0019	0.0037	0.0040	1.6181
yeast6	0.0613	0.1419	0.0018	0.0037	0.0036	1.6382
abalone19	0.0890	0.3161	0.0022	0.0053	0.0054	3.0168

F. Processing Time.

Data processing times for oversampling-based approaches on 41 datasets are compared to provide a more comprehensive comparison. We don't compare them to the other approaches, i.e., cost-sensitive learning and ensemble learning, because they only need negligible data processing time as they focus on classifiers other than improving the data. The processing time was recorded from our machine, which uses an Intel i7 32-thread processor and two NVIDIA 3090 Ti GPUs. Table IX shows the recorded processing time over 41 datasets. Overall, our technique takes longer than others as we have to compute the kernel estimation for each data point, as mentioned in Section VI. Similarly, DeepSMOTE generally suffers high time consuming cost because it heavily relies on underlying heuristic methods. In other words, the proposed technique is slower, but it provides better F1 and AUC scores than others.

G. Empirical study on the impact of radius factor r.

In this section, we study how the classification performance is impacted by different generation radius factor r in Equation 13. The classification performance is measured under different

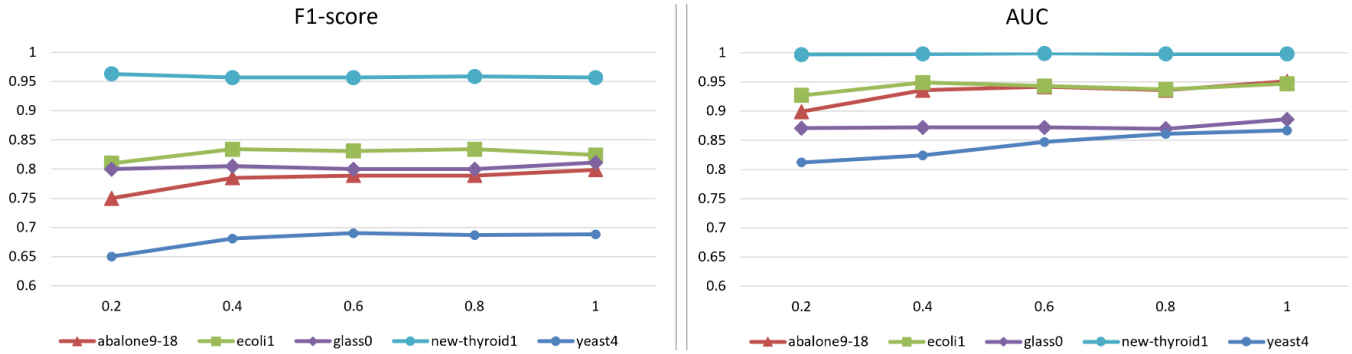


Fig. 7: F1-score and AUC results with varying Gaussian standard deviation (ranging from 0.2R to R).

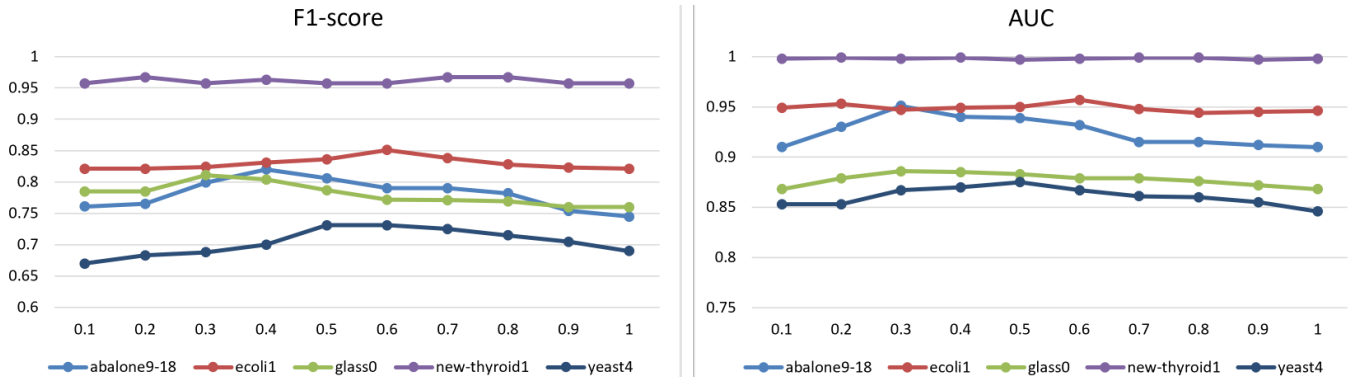


Fig. 8: F1-score and AUC results with varying informative portion IP.

distribution settings of the radius r as it controls how far synthetic data are generated from its original minority sample. We use different parameters for the Gaussian distribution $\mathcal{N}(\mu, (\alpha R)^2)$. Particularly, we fix the mean value to zero and change α from 0.2 to 1 with steps of 0.2 so that the Gaussian standard deviation αR will range from 0.2R to R. To save space, we arbitrarily select 5 datasets to conduct this experiment. The classification results are shown in Figure 7.

The figure obtained from the experiment indicates that the r factor, with a radius distribution standard deviation ranging from 0.6R to R, has minimal impact on the classification performance. While there are slight variations within the α range of 0.6 to 1, the performance improves between 0.2 to 0.6 (such as for ecoli1, abalone9-18, and yeast4). This is because the performance mainly depends on the classifier's decision boundary, and the synthetic data are placed far away from the decision boundary towards the minority class area; thus, the radius does not have much effect on the accuracy results. However, in the case of multi-classed data, the performance might be affected by a significant value of R.

H. Empirical study on the impact of informative portion (IP).

This section studies the empirical impact of the informative portion (IP) in Section III-C. This portion works as a threshold to adjust how many samples are taken into consideration of informative samples. To save space, we study five datasets used in Section VII-G. Different values of IP ranging from 0.1 to 1 are applied, and the classification performance results are shown in Figure 8.

As we can see from the figure, while datasets with outstanding performance (new-thyroid1, ecoli1) have little impact, there are fluctuations in other datasets' F1-score and AUC score (abalonce9-18, glass0, yeast4). This is because, for the easy-separated dataset such as new-thyroid1 and ecoli1, the IP change does not affect the classification performance as the data classes are easily separated. While in more challenging datasets, IP changes might affect the balance at the informative region; thus, this leads to performance variations. The resulting figure also suggests tuning IP for each dataset between a range of (0.2, 0.6) could achieve higher performance.

VIII. Conclusion

A data balancing technique by oversampling the minority class is proposed. The technique aims at balancing datasets and preventing the creation of noise in data by directing the synthetic samples toward the minority class. Our experiment results show that the proposed technique outperforms other experimental techniques over 41 real-world datasets. For future work, we would like to investigate the class imbalance for image data type and enhance our approach to adapt to image datasets.

References

[1] Q. Ya-Guan, M. Jun, Z. Xi-Min, P. Jun, Z. Wu-Jie, W. Shu-Hui, Y. Ben-Sheng, and L. Jing-Sheng, "EMSGD: An Improved Learning Algorithm of Neural Networks With Imbalanced Data," *IEEE Access*, vol. 8, pp. 64 086–64 098, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9055020/

[2] Y. Liu, X. Li, X. Chen, X. Wang, and H. Li, "High-Performance Machine Learning for Large-Scale Data Classification considering Class Imbalance," *Scientific Programming*, vol. 2020, pp. 1–16, May 2020. [Online]. Available: https://www.hindawi.com/journals/sp/2020/1953461/

[3] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, Dec. 2019. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0192-5

[4] Y. Xie, M. Qiu, H. Zhang, L. Peng, and Z. Chen, "Gaussian distribution based oversampling for imbalanced data classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 667–679, 2022.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002. [Online]. Available: https://www.jair.org/index.php/jair/article/view/10302

[6] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

[7] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 878–887.

[8] D. Dablain, B. Krawczyk, and N. V. Chawla, "Deepsmote: Fusing deep learning and smote for imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[9] L. Shen, Z. Lin, and Q. Huang, "Learning deep convolutional neural networks for places2 scene recognition," *CoRR*, vol. abs/1512.05830, 2015. [Online]. Available: http://arxiv.org/abs/1512.05830

[10] Y. Geifman and R. El-Yaniv, "Deep active learning over the long tail," *CoRR*, vol. abs/1711.00941, 2017. [Online]. Available: http://arxiv.org/abs/1711.00941

[11] Haibo He and E. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5128907/

[12] L. Li, H. He, and J. Li, "Entropy-based Sampling Approaches for Multi-Class Imbalanced Problems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2159–2170, Nov. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8703114/

[13] S. Ertekin, J. Huang, and C. L. Giles, "Active learning for class imbalance problem," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07*. Amsterdam, The Netherlands: ACM Press, 2007, p. 823. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1277741.1277927

[14] W. Feng, G. Dauphin, W. Huang, Y. Quan, W. Bao, M. Wu, and Q. Li, "Dynamic synthetic minority over-sampling technique-based rotation forest for the classification of imbalanced hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2159–2169, 2019.

[15] W. Feng, W. Huang, and W. Bao, "Imbalanced hyperspectral image classification with an adaptive ensemble method based on smote and rotation forest with differentiated sampling rates," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 12, pp. 1879–1883, 2019.

[16] D. S. Goswami, "Class Imbalance, SMOTE, Borderline SMOTE, ADASYN," Nov. 2020. [Online]. Available: https://towardsdatascience.com/class-imbalance-smote-borderline-smote-adasyn-6e36c78d804

[17] A. Liu, J. Ghosh, and C. E. Martin, "Generative oversampling for mining imbalanced datasets," in *Proceedings of the 2007 International Conference on Data Mining, DMIN 2007, June 25-28, 2007, Las Vegas, Nevada, USA*, R. Stahlbock, S. F. Crone, and S. Lessmann, Eds. CSREA Press, 2007, pp. 66–72.

[18] M. Rashid, J. Wang, and C. Li, "Convergence analysis of a method for variational inclusions," *Applicable Analysis*, vol. 91, no. 10, pp. 1943–1956, Oct. 2012. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/00036811.2011.618127

[19] W. Dai, K. Ng, K. Severson, W. Huang, F. Anderson, and C. Stultz, "Generative Oversampling with a Contrastive Variational Autoencoder," in *2019 IEEE International Conference on Data Mining (ICDM)*. Beijing, China: IEEE, Nov. 2019, pp. 101–109. [Online]. Available: https://ieeexplore.ieee.org/document/8970705/

[20] S. S. Mullick, S. Datta, and S. Das, "Generative Adversarial Minority Oversampling," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 1695–1704. [Online]. Available: https://ieeexplore.ieee.org/document/9008836/

[21] S. Ertekin, J. Huang, L. Bottou, and L. Giles, "Learning on the border: active learning in imbalanced data classification," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07*. Lisbon, Portugal: ACM Press, 2007, p. 127. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1321440.1321461

[22] U. Aggarwal, A. Popescu, and C. Hudelot, "Active Learning for Imbalanced Datasets," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Snowmass Village, CO, USA: IEEE, Mar. 2020, pp. 1417–1426. [Online]. Available: https://ieeexplore.ieee.org/document/9093475/

[23] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-Balanced Loss Based on Effective Number of Samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 9260–9269. [Online]. Available: https://ieeexplore.ieee.org/document/8953804/

[24] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning Deep Representation for Imbalanced Classification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 5375–5384. [Online]. Available: http://ieeexplore.ieee.org/document/7780949/

[25] V. K. Rangarajan Sridhar, "Unsupervised Text Normalization Using Distributed Representations of Words and Phrases," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Denver, Colorado: Association for Computational Linguistics, 2015, pp. 8–16. [Online]. Available: http://aclweb.org/anthology/W15-1502

[26] D. Mahajan, R. B. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, "Exploring the limits of weakly supervised pretraining," *CoRR*, vol. abs/1805.00932, 2018. [Online]. Available: http://arxiv.org/abs/1805.00932

[27] H. Masnadi-Shirazi, N. Vasconcelos, and A. Iranmehr, "Cost-sensitive support vector machines," *CoRR*, vol. abs/1212.0975, 2012. [Online]. Available: http://arxiv.org/abs/1212.0975

[28] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003*, N. Lavrač, D. Gamberger, L. Todorovski, and H. Blockeel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 107–119.

[29] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[30] P. Lim, C. K. Goh, and K. C. Tan, "Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2850–2861, 2017.

[31] S. Liu, Y. Wang, J. Zhang, C. Chen, and Y. Xiang, "Addressing the class imbalance problem in twitter spam detection using ensemble learning," *Computers and Security*, vol. 69, pp. 35–49, 2017, security Data Science and Cyber Threat Management. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404816301754

[32] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, 2009.

[33] W. contributors, "Receiver operating characteristic Wikipedia. the free encyclopedia," 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Receiver_operating_characteristic&oldid=1081635328

[34] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948. [Online]. Available: https://ieeexplore.ieee.org/document/6773024

[35] S. Leroueil, "Compressibility of Clays: Fundamental and Practical Aspects," *Journal of Geotechnical Engineering*, vol. 122, no. 7, pp. 534–543, Jul. 1996. [Online]. Available: http://ascelibrary.org/doi/10.1061/%28ASCE%290733-9410%281996%29122%3A7%28534%29

[36] D. Gissin and S. Shalev-Shwartz, "Discriminative active learning," 2019. [Online]. Available: https://openreview.net/forum?id=rJl-HsR9KX

[37] Z. Qiu, D. J. Miller, and G. Kesidis, "A maximum entropy framework for semisupervised and active learning with unknown and label-scarce classes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 917–933, 2017.

[38] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*. Honolulu, Hawaii: Association for Computational Linguistics, 2008,

p. 1070. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1613715.1613855
[39] "Empirical bayes method," Feb 2023. [Online]. Available: https://en.wikipedia.org/wiki/Empirical_Bayes_method
[40] D. W. Scott, *Multivariate density estimation: Theory, practice, and visualization*. Wiley, 2015.
[41] "sklearn.neighbors.KernelDensity." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.neighbors.KernelDensity.html
[42] "Sphere," Aug 2021. [Online]. Available: https://en.wikipedia.org/wiki/Sphere
[43] "Maximal and minimal points of functions theory." [Online]. Available: https://aipc.tamu.edu/~schlump/24section4.1_math171.pdf
[44] "Keel: Software tool. evolutionary algorithms for data mining." [Online]. Available: https://sci2s.ugr.es/keel/category.php?cat=imb#sub1
[45] A. Fernández, J. Luengo, J. Derrac, J. Alcalá-Fdez, and F. Herrera, "Implementation and integration of algorithms into the keel data-mining software tool," *Intelligent Data Engineering and Automated Learning - IDEAL 2009*, p. 562–569, 2009.
[46] "fetch datasets 2014; Version 0.10.0 — imbalanced-learn.org," https://imbalanced-learn.org/stable/references/generated/imblearn.datasets.fetch_datasets.html, [Accessed 16-Dec-2022].
[47] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html
[48] "Credit Card Fraud Detection — kaggle.com," https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud, [Accessed 16-Dec-2022].
[49] Wikipedia contributors, "Wilcoxon signed-rank test — Wikipedia, the free encyclopedia," 2022, [Online; accessed 22-December-2022]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Wilcoxon_signed-rank_test&oldid=1109073866
[50] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

SIMPOR: SUMMARY OF CHANGES

We are immensely pleased to be offered the helpful suggestions, and have thoroughly revised the manuscript according to the reviewers' comments. In this revision, we have taken the opportunity to address reviewers' concerns that were kindly drawn to our attention. We thoroughly considered each comment and made changes to clarify in the manuscript accordingly. The following items summarize the main changes in the latest revision conducted to the paper.

1) Section V (Methodology) and VII (Experiments) were thoroughly revised, especially sections V.B, VII.A, VII.C, and VII.E. Specifically, we added more information to improve the readability of the manuscript, including formulation descriptions, result explanations, discussion, and implementation details.
2) Section VII (Related Work) was reorganized to improve reading comprehension and moved to the second section. Two suggested papers from the reviewer were also considered and referred to in this section.
3) The prior estimation (was estimated as constants) in Equation 9 of the proposed method has been changed to use the Empirical Bayes method for a better approximation.
4) To have a better comparison, we replaced the Random Oversampling technique with a recently published work (DeepSMOTE) which is a deep learning-based oversampling method.
5) The experiments on 41 datasets were re-conducted according to the changes in Items 3 and 4. Figures 4, 5, 6, and Tables III, IV, V, VI, VII, and VIII were updated to reflect new experimental results.
6) All mistakes and typos from reviewers' comments have been reviewed and addressed. Several sentences thoughout the paper have been rewritten to improve the readability.

I. RESPONSE TO REVIEWER 1

Reviewer's Comment. Comments to the Author This work explores how class imbalanced data affects deep learning and proposes a data balancing technique for mitigation by generating more synthetic data for the minority class. The paper is well-written and easy to read. Several comments are listed below.

Comment 1.1 The detail of entropy-based active learning should be improved. For example, how many samples are used for model training initially? On p.3, the classifier with θ^0 is trained using a random batch of k selected samples. So only k samples are randomly drawn from the training set to train the model? Besides, the authors further state that the informative samples are obtained by selecting k samples based on the top k highest entropy. These two k are identical?

Response 1.1: Thank you for your comments and sorry for any confusions. We would like to response to your question and revise Section III.C as follows.

Active learning requires repeated phases. In the initial phase, a classifier is only trained on a small initial batch of data. The classifier is then used to estimate the remaining data entropy scores. Then a batch of informative samples are collected from the top entropy samples. This batch is concatenated to the training data for the next phase and also added to informative set. This phase is repeated so that the classifier is trained on larger dataset after every phase. For example, if the initial set contains 6 samples and each next informative batch contains 20 samples, the training size at the 4th phase is $6 + 20 * 4 = 86$.

In this revision, the detail of entropy-based active learning implementation under the experimental setting section is enhanced to improve the readability (e.g., including number of initial training samples). Additionally, the "Entropy-based Active Learning paragraph" is rewritten for better readability. The two changed paragraphs are listed below.

(This modification can be viewed at page 4, line 23.)

"' ... We consider a dataset containing N pairs of samples X and corresponding labels y, and a deep neural network with parameter θ. AL implementation requires repeated phases, and a batch of informative data is selected for each phase. At the first phase $t^{(0)}$, a classifier is trained with parameter $\theta^{(0)}$ (Note that this classifier differs from the classifier for the final classification problem) on a random initial batch of labeled samples and use the model $\theta^{(0)}$ to estimate the entropy of the remaining data.

The entropy scores are then estimated for the remaining samples based on Equation 4. The first batch of informative samples is determined by selecting k highest entropy samples. This batch is then concatenated with the initial training data for the training classifier parameter ($\theta^{(1)}$) in the next phase ($t^{(1)}$) and also accumulated to the informative set. In the next phase, similarly, the updated model is

then used to estimate the entropy of the remaining data. The next informative batch is selected and also added to the informative set. Phases are repeated until the number of accumulated informative samples reaches a pre-set informative portion (IP). For example, $IP = 0.3$ will select 30% training samples as informative samples. ”’

“‘ ... In order to find the informative subset, we leverage entropy-based active learning. We first utilize a neural network model playing a role as a classifier to find high-entropy samples (Note that the classifier for finding the informative subset differs from the classifiers for the final classification evaluation after all balancing techniques are applied to the data). The detailed steps are introduced in Section III-C. The model contains two fully connected hidden layers with *relu* activation functions and 10 neurons in each layer. The output layer applies the soft-max activation function. The model is trained in a maximum of 300 epochs with an early stop option when the loss is not significantly improved after updating weights. The model is trained firstly on a random set of three samples each class (six samples two classes). This model is then used to estimate entropy scores for the remaining data. We then select next 20 highest entropy samples (k=20) for the next informative data batch. This batch is concatenated to the initial batch for updating the classifiers and accumulated to the informative set. The steps are repeated until the informative set reaches desire informative portion (IP). In these experiments, we set IP=0.3 corresponding to 30 percent of the training size selected for the informative set. ”’

Comment 1.2 The authors simplify Eq. (8) by assume that the prior probabilities in the two classes are identical to cancel out this term. I can understand this assumption can make it easy to calculate, but is it reasonable? The priors for the majority and minority classes are apparently different. I suggest the authors consider the empirical priors to conduct experiments.

Response 1.2: Thank you for the insightful comment and suggestion. In this revision, we adopted the Empirical Bayes method to estimate the priors for equations (8) as suggested. The changes are made in the following paragraph (Section V.B) of the manuscript:

(This change can be viewed at page 5, line 8)

“‘

Approximation of priors in Equation 8: Additionally, we estimate the prior probabilities of observing samples in class A ($p(A)$) and class B ($p(B)$) (in Equation 8) by the widely-used Empirical Bayes Method [39] to leverage the existing information from the original data. The estimates are denoted as $\widehat{p(A)}$ and

$\widehat{p(B)}$ respectively.

Equation 8 Approximation: Let X_A and X_B be the subsets of dataset X which contain samples of class A and class B, $X_A = \{x : y = A\}$ and $X_B = \{x : y = B\}$. N_A and N_B are the numbers of samples in X_A and X_B. d is the number of data dimensions. h presents the width parameter of the Gaussian kernel. The posterior ratio for each synthetic sample x' then can be estimated as follows:

$$f = \frac{p(x'|y' = B)\, p(B)}{p(x'|y' = A)\, p(A)} \tag{9}$$

$$\propto \frac{\frac{1}{N_B h^d} \sum_{i=1}^{N_B} (2\pi)^{-\frac{d}{2}}\, e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}\, \widehat{p(B)}}{\frac{1}{N_A h^d} \sum_{j=1}^{N_A} (2\pi)^{-\frac{d}{2}}\, e^{\frac{1}{2}(\frac{x - X_{A_j}}{h})^2}\, \widehat{p(A)}} \tag{10}$$

$$\propto \frac{\frac{1}{N_B h^d} \sum_{i=1}^{N_B} e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}\, \widehat{p(B)}}{\frac{1}{N_A h^d} \sum_{j=1}^{N_A} e^{\frac{1}{2}(\frac{x - X_{A_j}}{h})^2}\, \widehat{p(A)}} \tag{11}$$

,,,,,,

Comment 1.3 The proposed method involves many hyper-parameters. For example, h in Eq. (12). The values of these hyper-parameters and how to obtain them should be presented in the manuscript.

Response 1.3: Thank you for your suggestions. In this revision, we updated missing hyper-parameters and explain how we selected them. For example, the update for the bandwidth parameter h in Eq. 12 is as follows.

(This change be viewed at page 5, line 21.)

‘‘‘ ... ***Selecting bandwidth parameter*** h ***for Gaussian kernel:*** The bandwidth is automatically selected for each dataset using the most common method, namely Scott's rule of thumb, proposed by Scott [40]. With an attempt to minimize the mean integrated squared error, the parameter is estimated as $h = N^{(-\frac{1}{d+4})}$ where N, d are the number of data points and the number of dimensions respectively. This study utilizes a scikitlearn python library for KDE, including bandwidth selection. The implementation detail can be found at [41].

’’’

Comment 1.4 The comparison methods should include recently published methods and deep learning methods. Among the comparison methods, only GOD was a method published in 2022. SMOTE,

Boderline-SMOTE, EE, and ADASYN where published before 2010. SVMCS was in 2012.

Response 1.4: Thank you for your recommendation. In the revised version, we have included a comparison with DeepSMOTE, a data balancing technique based on deep learning that was published in 2022. Further details can be found in Section VII (Experiment).

Comment 1.5 In Section VI. E (Data Visualization), how to calculate No. of intersection samples?

Response 1.5: In order to provide further clarification on the calculation of the number of intersection samples, we have added additional details to the paragraph (Section VII.E), which now reads as follows.

"‘ ... A hard-to-differentiate ratio is defined as the ratio of the number of samples in the intersection between 2 classes to the total of minority samples ($HDR = \frac{No.\ Intersection\ samples}{No.\ Minority\ samples} 100\%$) where the number of intersection samples is estimated by counting samples in the overlapped bins between the two classes in the 1D histograms ”’

Comment 1.6 . In Fig. 6, how to obtain the densities of the results? If they are obtained by using KDE, are they unfair to the other methods that don't use KDE?

Response 1.6: Thank you for your feedback. The densities presented in Figure 6 were estimated using KDE on the dimensionally reduced data, but this was done only for visualization purposes. All the quantified comparisons, such as HDR, were actually computed based on the 1D histogram, as explained in Comment 1.5. To make this clearer and to avoid any confusion, we have revised the relevant paragraph (Section VII.E) accordingly.

(The modifications can be viewed at page 10, line 17.)

"‘ To explore more on how the techniques perform, we visualize the generated data by projecting them onto lower dimension space (i.e., one and two dimensions) using the Principle Component Analysis technique (PCA) [50]. Data's 2-Dimension (2D) plots and 1-Dimension histograms are presented with a hard-to-differentiate ratio (HDR) for each technique. 1D histograms are computed by dividing one-dimensional-reduced data into 20 bins (intervals) and counting the number of samples within the interval of each bin. A hard-to-differentiate ratio is defined as the ratio of the number of samples in the intersection between 2 classes to the total of minority samples ($HDR = \frac{No.\ Intersection\ samples}{No.\ Minority\ samples} 100\%$) where the number of intersection samples is estimated by counting samples in the overlapped bins between the two classes

in the 1D histograms. This ratio is expected to be as small as 0% if the two classes are well separated; in contrast, 100% indicates that the two classes cannot be distinguished in the projected 1D space. Besides HDR, we show the absolute numbers of Minority, Majority, and Intersection samples for each technique in the bottom tables. From the plots, we observe how the data are distributed in 2D space and quantify samples that are hard to be differentiated in the 1D space histograms. ”’

Comment 1.7 In Fig. 7 & 8, the results show that α and IP do not significantly affect the performance, but they are counter-intuitive. The authors should provide more discussion to explain the results rather than only presenting the results.

Response 1.7: We appreciate your valuable feedback. In this revised version, we have included additional discussion and explanation of the results accordingly in Section VII.G as follows.

(This modification can be viewed at page 13, line 9, and page 13, line 1.)

“‘ ... The figure obtained from the experiment indicates that the r factor, with a radius distribution standard deviation ranging from 0.6R to R, has minimal impact on the classification performance. While there are slight variations within the α range of 0.6 to 1, the performance improves between 0.2 to 0.6 (such as for ecoli1, abalone9-18, and yeast4). This is because the performance mainly depends on the classifier’s decision boundary, and the synthetic data are placed far away from the decision boundary towards the minority class area; thus, the radius does not have much effect on the accuracy results. However, in the case of multi-classed data, the performance might be affected by a significant value of R. ”””

“‘ ... As we can see from the figure, while datasets with outstanding performance (new-thyroid1, ecoli1) have little impact, there are fluctuations in other datasets’ F1-score and AUC score (abalonce9-18, glass0, yeast4). This is because, for the easy-separated dataset such as new-thyroid1 and ecoli1, the IP change does not affect the classification performance as the data classes are easily separated. While in more challenging datasets, IP changes might affect the balance at the informative region; thus, this leads to performance variations. The resulting figure also suggests tuning IP for each dataset between a range of (0.2, 0.6) could achieve higher performance. ”””

Comment 1.8 . In Section VI. F (Processing Time), the sentence ”To explore more ... the To better evaluate the technique” should be re-phrased.

Response 1.8: Thank you for your suggestion. The sentence in your comment has been reworded in the revised version, and we have also rephrased several other sentences to enhance the readability of the manuscript. The suggested sentence was modified as follows.

“‘ Data processing times for oversampling-based approaches on 41 datasets are compared to provide a more comprehensive comparison.... ”””

II. RESPONSE TO REVIEWER 2

Reviewer's Comment. This work explores how class-imbalanced data affects deep learning and proposes a data-balancing technique to mitigate by generating more synthetic data for minority classes. The author has done a lot of work, and the experimental results are very good. The content of the article is substantial. I hope the following comments could be useful for the further improvement of the paper:

Comment 2.1 The language of the manuscript is not academic enough. There are too many "we" in the text. I suggest the writer use the passive voice

Response 2.1:

Thank you for your feedback. In this version, we have thoroughly revised the manuscript according to the comments. Multiple sentences and paragraphs throughout the manuscript have been rephrased to refine the language and enhance its comprehensibility. A few examples are provided below.

Example 1:

"‘ ...We elaborate on how our strategy is developed in the rest of this section. "’’’

is rewritten as

"‘ ...The remainder of this section provides further information about how our approach was developed. "’’’

Example 2:

"‘ ... Because we want to generate neighbors for each minority sample that maximizes Function f in Equation 11, we examine points lying on the sphere centered at the minority sample with a small radius r. As a result, we can find a vector $\vec{v}$ so that it can be added to the sample to generate a new sample. "’’’

is rewritten as

"‘ ...To generate neighbors for each minority sample that maximizes Function f in Equation 11, points on the r-radius sphere centered at a minority sample are considered synthetic instances. As a result, a vector $\vec{v}$ can be added to a minority sample for generating a new instance. "’’’

Comment 2.2 The contribution: 3) We applied our technique to 41 real datasets with a diversity of imbalance ratio and the number of features. I don't think this is a contribution. This is a statement of work.

Response 2.2: Thank you for your feedback. We have revised accordingly by moving the contribution 3 to the statement of work in Section VII as follows.

(The change can be viewed at page 7, line 43.)

“‘ In this section, we explore the techniques via binary classification problems on an artificial dataset (i.e., Moon) and 41 real-world datasets (i.e., KEEL, UCI, Credit Card Fraud) with a diversity of imbalance ratios and different numbers of features. Samples in Moon have two features, while other datasets contain various numbers of features and imbalance ratios. ... ”””

Comment 2.3 AUC is an acronym. It is better to use an acronym after mentioning the full name in the “Introduction”.

Response 2.3:

Thanks for your suggestion. We have included the complete name of the AUC abbreviation in this version (Section I).

(The change can be viewed at page 2, line 36.)

“‘ ... Section III introduces related concepts that will be used in this work, i.e., Imbalance Ratio, Macro F1-score, Area Under the Curve (AUC), and Entropy-based active learning... ”””

Comment 2.4 The letters used in the formulas are not explained clearly, especially formula (9).

Response 2.4: Thanks for your feedback. To enhance the comprehensiveness of the revision, we have included descriptions for every letter in formula (9) (Section V.B). The detail is listed as follows.

(This modification can be viewed in column 2 on page 5.)

“‘ ***Approximation of priors in Equation 8:*** Additionally, we estimate the prior probabilities of observing samples in class A ($p(A)$) and class B ($p(B)$) (in Equation 8) by the widely-used Empirical Bayes Method [39] to leverage the existing information from the original data. The estimates are denoted as $\widehat{p(A)}$ and $\widehat{p(B)}$ respectively.

Equation 8 Approximation: Let X_A and X_B be the subsets of dataset X which contain samples of class A and class B, $X_A = \{x : y = A\}$ and $X_B = \{x : y = B\}$. N_A and N_B are the numbers of samples in X_A and X_B. d is the number of data dimensions. h presents the width parameter of the Gaussian kernel. The posterior ratio for each synthetic sample x' then can be estimated as follows:

$$f = \frac{p(x'|y' = B)\, p(B)}{p(x'|y' = A)\, p(A)} \tag{9}$$

$$\propto \frac{\frac{1}{N_B h^d} \sum_{i=1}^{N_B} (2\pi)^{-\frac{d}{2}}\, e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}\, \widehat{p(B)}}{\frac{1}{N_A h^d} \sum_{j=1}^{N_A} (2\pi)^{-\frac{d}{2}}\, e^{\frac{1}{2}(\frac{x - X_{A_j}}{h})^2}\, \widehat{p(A)}} \tag{10}$$

$$\propto \frac{\frac{1}{N_B h^d} \sum_{i=1}^{N_B} e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}\, \widehat{p(B)}}{\frac{1}{N_A h^d} \sum_{j=1}^{N_A} e^{\frac{1}{2}(\frac{x - X_{A_j}}{h})^2}\, \widehat{p(A)}} \tag{11}$$

Selecting bandwidth parameter h ***for Gaussian kernel:*** The bandwidth is automatically selected for each dataset using the most common method, namely Scott's rule of thumb, proposed by Scott [40]. With an attempt to minimize the mean integrated squared error, the parameter is estimated as $h = N^{(-\frac{1}{d+4})}$ where N, d are the number of data points and the number of dimensions respectively. This study utilizes a scikitlearn python library for KDE, including bandwidth selection. The implementation detail can be found at [41]. "'

Comment 2.5 More new published methods are suggested to be used as the comparison algorithms

Response 2.5: Thank you for your recommendation. In the revised version, we have included a comparison with DeepSMOTE, a data balancing technique based on deep learning that was published in 2022. (More detail can be viewed in Section VII at Page 7.)

Comment 2.6 "Related work" should be put to the second section of the paper. In addition, the following references are suggested to be present and cited:

1

Wei Feng*, Gabriel Dauphin, Wenjiang Huang, Yinghui Quan, Wenxin g Bao, Mingquan Wu, Qiang Li, "Dynamic synthetic minority over-sampling technique based rotation forest for the classification of imbal a nced hyperspectral data," IEEE Journal of Selected Topics in AppliedEarth Observations and Remote Sensing, vol. 12, no. 7, pp. 2159–2169, 2019.

2

Wei Feng*, Wenjiang Huang and Wenxing Bao, "Imbalanced Hyperspectral Image Classification With

an Adaptive Ensemble Method Based on SMOTE and Rotatio n F orest With Differentiated Sampling Rates," IEEE Geoscience and Remote Sensing Le tters, vol. 16, no. 12, pare suggestedggestedc. 2019.

Response 2.6: Thank you for your suggestions. In this revision, the two suggested papers have been taken into account and are listed as [14] and [15]. Besides, Section "Related Work" (was Section VII in previous version) has been moved to Section II to improve readability of the manuscript.