SIMPOR: 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. Experimental results on forty-two datasets show that our technique significantly outperforms all compared state-of-the-art ones in terms of boosting deep-learning performance. It has been shown that a deep learning model can achieve up to 12% higher F1 score when trained with a data set augmented by the proposed technique than with the current state-of-the-art techniques. It also archives the highest winning times in F1-score and AUC over 41 datasets compared to others.**

***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. We also propose a minority oversampling-based technique 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 five state-of-the-art techniques.**

***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. We propose a class balancing solution 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 [7], Borderline SMOTE [6], Random Oversampling (ROS).

We categorize existing solutions into model-centric and data-centric approaches in which the first approach aims at modifying machine algorithms, and the latter looks for data balancing techniques. 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.* [6] 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. For example, if a sample Euclidean space-based group includes samples from other classes, they can be considered as samples near the border. Since this technique is entirely based on Euclidean distance from determining neighbors to generating synthetic data, it performs poorly in high dimensional space. 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 bounder. Leveraging the same way as SMOTE generates synthetic samples, another widely-used technique, ADASYN [7], 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 data statistical 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, we propose a minority oversampling technique 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, we leverage an entropy-based deep active learning technique that is able to select samples yielding high entropy to deep learning models. We denote the location of informative samples as the informative region. We then balance this region 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, we safely generate its synthetic neighbors 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, we find a direction to generate synthetic samples by maximizing their posterior probability based on Bayes's Theorem. However, maximizing the posterior probability is facing infeasible computation in the denominator. To overcome this, we maximize the posterior ratio 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 improves the training performance and alleviates the class imbalance problem. Our experiments indicate that we can achieve better classification results over widely-used techniques in all experimental cases by applying the proposed strategy.

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) We applied our technique to 41 real datasets with a diversity of imbalance ratio and the number of features.
4) We compare our technique with five different widely-used and recent techniques. The results show that the proposed technique outperforms others.

The rest of this paper is organized as follows. Section II introduces related concepts that will be used in this work, i.e., Imbalance Ratio, Macro F1-score, AUC, and Entropy-based active learning. Section III 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 IV. Section V discusses the technique implementation and complexity. We will show experiments on different datasets, including artificial and real datasets in Section VI. We also discuss experimental results in the same section. In Section VII, we briefly review other existing works. Section VIII concludes the study and discusses future work.

II. Preliminaries

In this section, we introduce related concepts that will be used in our work.

A. *Imbalance Ratio (IR)*

For binary classification problems, we use imbalance ratio (IR) 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, we evaluate balancing data techniques by classification performance. Specifically, we use F1-Score and Area Under the Curve (AUC) as evaluation metrics. We measure the Macro-averaging for measuring F1-scores in which we compute scores per class and take the average of all classes with the same weight regardless of how often they appear in the datasets. These are fair measurements for imbalanced test 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.

We also measure AUC [8] scores as it is an important metric to evaluate imbalanced data. AUC is derived from Receiver Operating Characteristic curve (ROC). In this work, we utilize a skit-learn library to compute AUC; the library can be found in sklearn.metrics.auc.

C. *Entropy-based Active Learning*

To find informative samples, we leverage entropy-based active learning. The technique gradually selects batch-by-batch samples that provide high information to the model based on information entropy theory [9]. 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.

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

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

In this work, we consider a dataset containing N pairs of samples X and corresponding labels y, and a deep neural network with parameter θ. At the first step $t^{(0)}$, we train the classifier with parameter $\theta^{(0)}$ on a random batch of k labeled samples and use the $\theta^{(0)}$ to predict the labels for the rest of the data (we assume their labels are unknown). We then compute the prediction entropy of each sample based on Equation 4. We are now able to collect the first batch of informative samples by selecting k samples based on the top k highest entropy. We query labels for this batch and concatenate them to existing labeled data to train the classifier parameter $\theta^{(1)}$ in the next step $t^{(1)}$. Steps are repeated until the number of informative samples reaches a preset informative portion (IP). For example, $IP = 0.3$ will select the top 30% high entropy samples as informative samples.

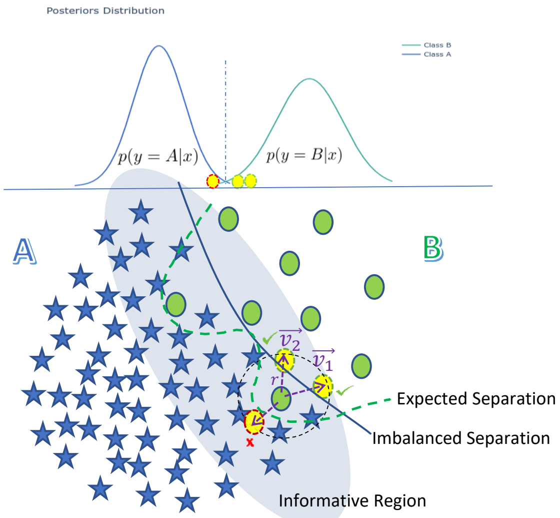


Fig. 1: Learning from imbalanced datasets

III. The Problem of Learning From Imbalanced Datasets

In this section, we review the problem of learning from imbalanced datasets. Although the problem may apply to different machine learning methods, we focus on deep learning in this work.

Figure 1 illustrates our problem on a 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.* [10] 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. We consider the case that the informative region contains highly imbalanced data, which we believe happens in most real 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 of data imbalance by oversampling the minority class in an informative manner. The detail of our proposed technique will be described in Section IV.

IV. Synthetic Information towards Maximum Posterior Ratio

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. We first find the informative region where informative samples are located and balance this region by creating surrounding synthetic neighbors for minority samples. The remaining region is then fully balanced by arbitrarily generating minority samples' neighbors. We elaborate on how our strategy is developed in the rest of this section.

A. Methodology Motivation

As Chazal and Michel mentioned in their work [11], 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 [10] and [12], 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, we apply an entropy-based active learning strategy mentioned in II-C to find the samples that maximize entropy to the model. This strategy is perhaps the most popular active learning technique and over-performs many other techniques on several datasets [13], [14] [15].

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, we would like to generate each synthetic data x' that maximizes the probability of x' belonging to the minority class B and minimizes the chance x' falling into the majority class A. Thus, we propose a technique that maximizes the fractional posterior f,

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

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

Approximation of likelihoods in Equation 8: We use non-parametric kernel density estimates (KDE) 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 assume prior probabilities of observing samples in class A ($p(A)$) and class B ($p(B)$) (in Equation 8) are constant. Hence, these probabilities do not affect our algorithm in terms of generating synthetic neighbors for minority samples because we only determine the relative direction between the minority and the majority class. Thus, they can be canceled out at the end of the equation reduction.

Equation 8 reduction: 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:

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

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

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

$$\propto \frac{\sum_{i=1}^{N_B} e^{\frac{1}{2}(\frac{x' - X_{B_i}}{h})^2}}{\sum_{j=1}^{N_A} e^{\frac{1}{2}(\frac{x' - X_{A_j}}{h})^2}}. \tag{12}$$

Finding synthetic samples surrounding a minority sample: Because we want to generate neighbors for each minority sample that maximizes Function f in Equation 12, 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. The relationship between a synthetic sample x' and a minority sample can be described as follows,

$$\vec{x'} = \vec{x} + \vec{v}, \tag{13}$$

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

$$r \sim \mathcal{N}(0,\, (\alpha R)^2), \tag{14}$$

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:

$$R = \frac{1}{k} \sum_{1}^{k} ||x - x_j||, \tag{15}$$

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.

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 13).

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 12,

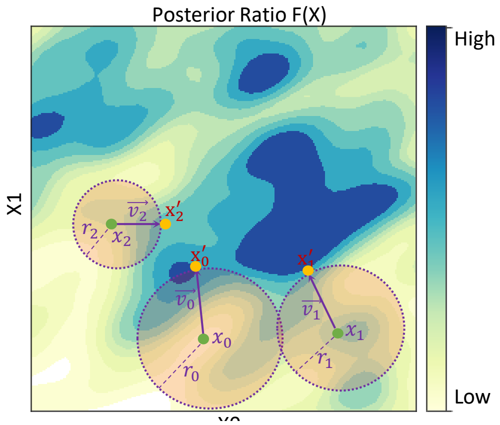


Fig. 2: Demonstration on how SIMPOR generates three synthetic samples x'_0, x'_1, x'_2, from three minority samples x_0, x_1, x_2, by maximizing the Posterior Ratio.

$$\max_{x'} f(x') \quad \text{s.t. } ||\vec{x'} - \vec{x}|| = r. \tag{16}$$

Solving optimization problem in Equation 16: Interestingly, the problem in Equation16 can be solved numerically. Function $f(x)$ in Equation 12 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 ([16]). Thus, a maximum exits as proved in [17]. To enhance the diversity of synthetic data, we accept either the global maximum or any local maximum so that the synthetic samples will not simply go to the same direction.

We solve the problem in Equation 16 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 16 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.

C. Algorithm

Our strategy can be described in Algorithm 2. The algorithm takes an imbalanced dataset as its input and results in a

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 $x_0$
 $\theta$ : Sample space $\theta \in [0, 2\pi]$
$lr$ : Gradient ascent learning rate
Output: An local maximum $x'$
Shift the Origin to $x_0$
Randomly initiate $x'_t$ on the sphere with radius $r$
while converge condition do
    Compute the gradient at $x'_t$
    $g_t(x'_t) = \nabla f(x'_t)$
    Project the gradient onto the sphere tangent plane
    $p_t = g_t - (g_t \cdot x'_t)x_t$
    Normalize projected vector
    $p_t = p_t/||p_t||$
    Update $x'$ on the constrained sphere
    $x'_{t+1} = x'_t cos(lr * \theta) + p_t sin(lr * \theta)$
end while
Shift back to the Origin
return  $x'_t$
```

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 12 (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'.

V. Algorithm Implementation and Complexity

Our proposed technique is straightforward in implementation. We first train a neural network model with initial samples and start querying the next batches of data based on the entropy scores from the previous model to find informative samples. The model is then updated with new batches of data until the entropy scores reach a certain threshold. All the informative samples are then balanced first, and the remaining data are balanced later. Each synthetic data point is generated by finding local maxima in Equation 12.

Perhaps 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 $N_A = n - 2$ respectively. We need to generate $n - 1$ synthetic samples to completely balance the dataset. Since each generated sample must loop through the entire dataset of size n to estimate the density, the algorithm complexity is $O(n^2)$.

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.
$\alpha$: 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 12.
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:         Compute maximum range $R$ based on k-nearest neighbors
7:         Randomly sample a radius $r \sim \mathcal{N}(0, \alpha R)$
8:         Generate a synthetic neighbor $x'$ from $x_i^c$:
           $x' = G(x_i^c, f, r)$
9:         Append $x'$ to $D'$
10:    end while
11: end for
   {Balance the remaining region}
12: for $c$ in $MI$ do
13:    while  $Count(c, D') \leq Count(MA, D')$  do
14:        Select a random $x_j^c \in \{X - S\}$
15:        Compute maximum range $R$ based on $k$
16:        Randomly sample a radius $r \sim \mathcal{N}(0, \alpha R)$
17:        Generate a synthetic neighbor $x'$ of $x_j^c$
18:        Append $x'$ to $D'$
19:    end while
20: end for
21: return
```

Although generating synthetic data is only a one-time process, and this does not affect the classification performance in the testing phase, we still alleviate its weakness by providing parallelized implementations. We provide two suggestions, multiple CPU thread-based and GPU-based implementations. While the former simply computes each synthetic data sample in separated CPU threads, the later computes each exponential component in Equation 12 parallelly in GPUs' threads. More specifically, Equation 12 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 parallelly processed in GPUs. The latter is then implemented using Python Numba and Cupy libraries that utilize CUDA toolkit from NVIDIA

[18]. The consumption time for kernel density estimation for each synthetic data point is then reduced by $N_A + N_B = n$ times, which significantly simplifies the complexity to $O(n)$.

VI. Experiments and Discussion

In this section, we experiment on binary classification for an artificial dataset (i.e., Moon) and 41 real-world datasets (i.e., KEEL, UCI, Credit Card Fraud). Samples in Moon have two attributes, while other datasets contain various numbers of attributes 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 SIMPOR to SMOTE [5], Borderline-SMOTE [6], ADASYN [7], Random Oversampling (ROS), Gaussian Distribution Based Oversampling (GDO) [4]. To evaluate the classifications performance for skewed datasets, we measure widely-used metrics, i.e., F1-score and Area Under The Curve (AUC).

TABLE I: Dataset Description.

dataset	#samples	#attributes	IR
glass1	214	9	1.8
wisconsin	683	9	1.9
pima	768	8	1.9
glass0	214	9	2.1
yeast1	1484	8	2.5
haberman	306	3	2.8
vehicle1	846	18	2.9
vehicle2	846	18	2.9
vehicle3	846	18	3.0
creditcard	1968	30	3.0
glass-0-1-2-3_vs_4-5-6	214	9	3.2
vehicle0	846	18	3.3
ecoli1	336	7	3.4
new-thyroid1	215	5	5.1
new-thyroid2	215	5	5.1
ecoli2	336	7	5.5
glass6	214	9	6.4
yeast3	1484	8	8.1
ecoli3	336	7	8.6
page-blocks0	5472	10	8.8
yeast-2_vs_4	514	8	9.0
yeast-0-5-6-7-9_vs_4	528	8	9.4
vowel0	988	13	10.0
glass-0-1-6_vs_2	192	9	10.3
glass2	214	9	11.6
yeast-1_vs_7	459	7	14.3
glass4	214	9	15.5
ecoli4	336	7	15.8
page-blocks-1-3_vs_4	472	10	15.9
abalone9-18	731	8	16.4
yeast-1-4-5-8_vs_7	693	8	22.1
glass5	214	9	22.8
yeast-2_vs_8	482	8	23.1
car_eval_4	1728	21	25.6
wine_quality	4898	11	25.8
yeast_me2	1484	8	28.0
yeast4	1484	8	28.1
yeast-1-2-8-9_vs_7	947	8	30.6
yeast5	1484	8	32.7
yeast6	1484	8	41.4
abalone19	4174	8	129.4

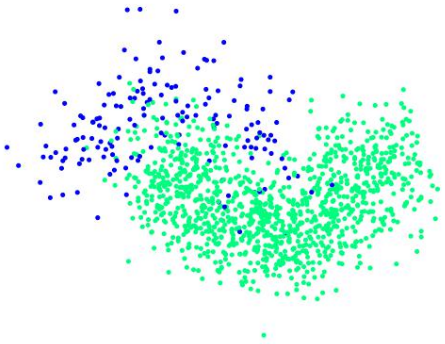


Fig. 3: Artificial class imbalanced Moon dataset with IR of 7:1.

A. Experimental Setup

This subsection describes the experimental setup for all datasets. In order to find the informative subset, we leverage entropy-based active learning as mentioned in Section II-C. The classifier for active learning is a fully connected neural network model containing 3 hidden layers with *relu* activation functions and 100 neurons in each layer. The output layer applies soft-max activation function. The model is trained in a maximum of 300 epochs with an early stop option until the loss does not change after updating weights.

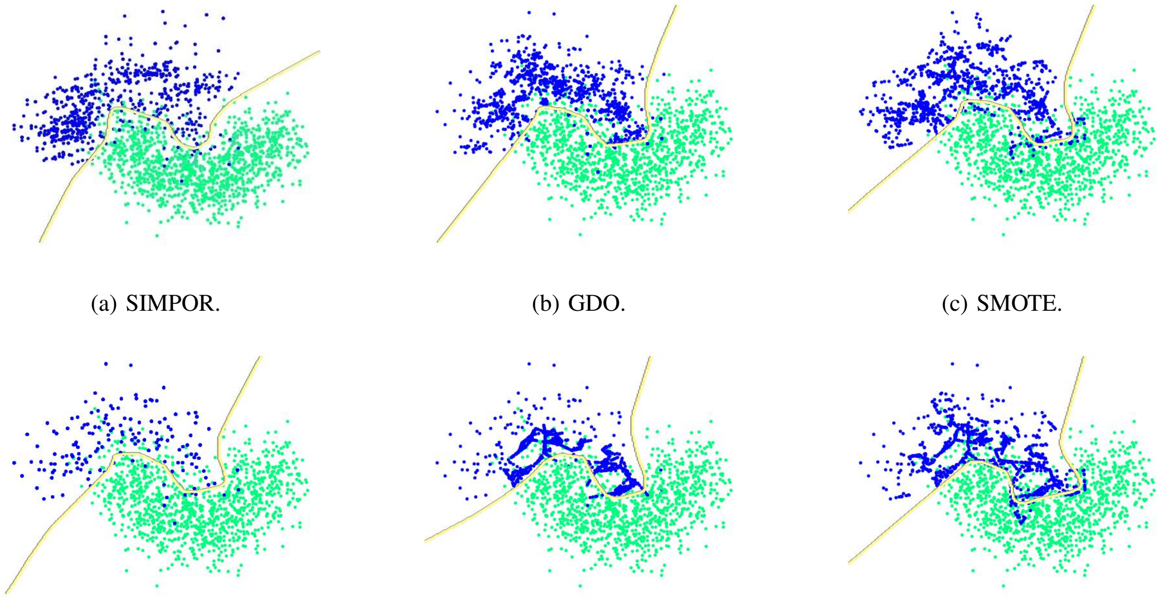
We randomly split the datasets into two parts, 80% for training and 20% for testing. Reported testing results for each dataset are the averages of 5 experimental trials. For SIMPOR to find optima of the function in Equation 16, we use a gradient ascent rate of 0.00001 and the iteration of 300. The architecture detail of the evaluation classifier and technique parameters are described in Table II.

TABLE II: Classification models' setting for each dataset.

Technique	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
ROS	sampling_strategy='auto', random_state=None, shrinkage=None
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 a demonstration purpose. We first generate the balanced MOON dataset using python library *sklearn.datasets.make_moons* including 3000 two-dimensional samples labeled in two classes. We then create an imbalanced dataset 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.



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

(d) Random Over Sampling. (e) BorderlineSMOTE. (f) ADASYN.

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

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. To classify the data, we use a fully connected neural network which is described in Table II.

TABLE III: Classification Result on Moon Dataset.

Metric	SIMPOR	SMOTE	Bl-SMOTE	ROS	ADASYN	GDO
F1-score	0.883	0.824	0.827	0.830	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 12% on F1-score and 0.6% on AUC. We can see that the Random Over Sampling technique (Figure 4d) which randomly duplicates minority samples might push the boundary towards the majority because the samples near the border carry significant weights. Due to the fact that SMOTE 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

This section, we compare the proposed technique on a total of 41 real datasets with a variety number of attributes and Imbalance Ratios, i.e., KEEL datasets [19], UCI datasets [20] and Credit Card Fraud [21] dataset. The datasets are described in Table I.

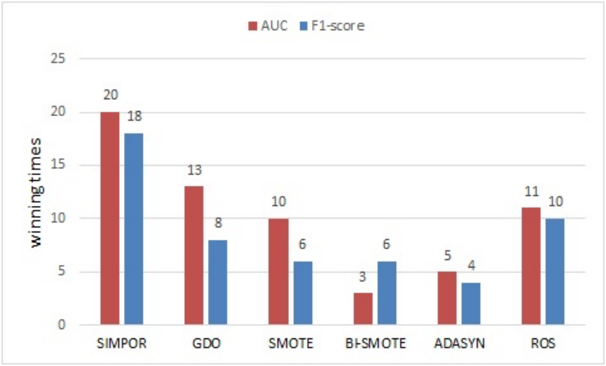


Fig. 5: Winning times over 41 datasets.

1) Classification results.: Table IV and V show the classification F1-score and AUC results respectively. The highest scores for each dataset are highlighted in bold style. 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 there more than two techniques sharing the same highest score, the winning time will be increased for each technique. A summary of winning times is shown in Figure 5. As we can see from the table, the proposed technique clearly outperforms others on both evaluation metrics, F1-score and AUC. More specifically, SIMPOR hits 20 F1-score wining times out of 62 and 18 AUC winning times out of 52. Its number of F1-score winning times at 20 nearly double the second winner (GDO) at 13, and its AUC winning times at 18 also nearly double the second AUC winner (ROS) at 10.

TABLE IV: F1-score over different datasets.

dataset	SIMPOR	GDO	SMOTE	Bl-SMOTE	ADASYN	ROS
glass1	**0.729**	0.706	0.708	0.728	0.727	0.713
wisconsin	0.962	**0.966**	0.956	0.963	0.961	0.963
pima	**0.756**	0.744	0.754	0.718	0.728	0.740
glass0	**0.811**	0.760	0.775	0.757	0.771	0.790
yeast1	**0.689**	0.677	0.687	0.684	0.681	0.672
haberman	0.583	0.583	0.601	0.586	0.592	**0.615**
vehicle1	0.809	0.791	**0.824**	0.816	0.809	0.816
vehicle2	**0.982**	0.968	0.980	**0.982**	0.978	0.978
vehicle3	0.730	0.759	0.773	0.790	0.770	**0.801**
creditcard	**0.947**	0.931	**0.947**	0.945	0.943	0.944
glass-0-1-2-3_vs_4-5-6	0.905	0.908	0.904	0.910	**0.911**	**0.911**
vehicle0	**0.972**	0.962	0.969	0.964	0.966	0.965
ecoli1	**0.824**	0.799	0.814	0.807	0.798	0.797
new-thyroid1	0.957	**0.969**	0.946	0.953	0.953	0.946
new-thyroid2	0.934	**0.959**	0.935	0.935	0.948	0.948
ecoli2	0.903	0.856	0.900	0.865	0.873	**0.905**
glass6	0.866	**0.888**	0.876	0.861	0.873	0.873
yeast3	0.846	0.805	**0.850**	0.834	0.839	0.843
ecoli3	**0.811**	0.754	0.788	0.785	0.777	0.782
page-blocks0	**0.920**	0.902	0.911	0.901	0.891	0.916
yeast-2_vs_4	0.864	0.870	0.866	0.858	**0.874**	0.873
yeast-0-5-6-7-9_vs_4	0.778	0.746	0.766	0.783	0.762	**0.794**
vowel0	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
glass-0-1-6_vs_2	0.661	0.642	**0.723**	0.704	0.696	0.704
glass2	0.746	0.687	0.782	0.772	**0.797**	0.769
yeast-1_vs_7	**0.701**	0.641	0.613	0.638	0.588	0.664
glass4	**0.817**	0.803	0.805	0.797	0.814	0.807
ecoli4	**0.913**	0.843	0.890	0.892	0.892	0.892
page-blocks-1-3_vs_4	**0.981**	0.938	0.962	0.953	0.971	0.954
abalone9-18	**0.799**	0.717	0.791	0.757	0.772	0.782
yeast-1-4-5-8_vs_7	0.595	0.609	0.565	0.584	0.566	**0.629**
glass5	0.725	**0.923**	0.836	0.786	0.773	0.870
yeast-2_vs_8	**0.862**	0.705	0.712	0.722	0.718	0.787
car_eval_4	**1.000**	0.963	0.993	0.990	0.993	0.990
wine_quality	0.629	**0.673**	0.658	**0.673**	0.657	**0.673**
yeast_me2	0.653	**0.663**	0.642	**0.663**	0.639	0.644
yeast4	0.688	0.682	0.666	0.665	0.663	**0.689**
yeast-1-2-8-9_vs_7	0.658	0.625	0.611	0.660	0.596	**0.666**
yeast5	0.803	0.726	**0.868**	0.861	0.859	0.861
yeast6	0.688	0.685	0.704	**0.760**	0.702	0.736
abalone19	0.499	0.499	0.514	**0.520**	0.518	0.511

TABLE V: AUC result over different datasets.

dataset	SIMPOR	GDO	SMOTE	Bl-SMOTE	ADASYN	ROS
glass1	0.809	0.784	**0.810**	0.804	0.796	0.809
wisconsin	**0.994**	**0.994**	**0.994**	0.992	**0.994**	**0.994**
pima	0.840	**0.841**	0.825	0.818	0.824	0.833
glass0	0.886	0.871	0.879	0.867	0.877	**0.888**
yeast1	**0.788**	0.771	0.777	0.772	0.764	0.774
haberman	0.614	0.674	0.674	0.671	0.670	**0.681**
vehicle1	0.932	0.904	**0.933**	0.930	0.921	0.932
vehicle2	0.998	0.998	**0.999**	0.998	**0.999**	**0.999**
vehicle3	0.843	0.873	0.895	0.899	0.897	**0.904**
creditcard	0.968	0.967	0.965	0.965	0.963	**0.970**
glass-0-1-2-3_vs_4-5-6	**0.986**	**0.986**	0.981	0.977	0.980	0.984
vehicle0	0.993	0.994	**0.995**	0.993	**0.995**	0.993
ecoli1	0.947	0.948	**0.949**	0.943	0.945	0.945
new-thyroid2	**1.000**	0.998	0.996	0.998	0.999	0.998
new-thyroid1	0.998	**0.999**	0.997	0.997	0.997	0.998
ecoli2	**0.963**	0.959	0.962	0.948	0.951	0.960
glass6	0.867	**0.953**	0.856	0.851	0.848	0.905
yeast3	**0.959**	0.958	0.958	0.947	0.939	0.953
ecoli3	**0.885**	0.865	0.877	0.874	0.874	0.875
page-blocks0	**0.986**	0.985	0.985	0.983	0.984	**0.986**
yeast-2_vs_4	**0.979**	0.973	0.973	0.958	0.951	0.963
yeast-0-5-6-7-9_vs_4	0.899	**0.923**	0.890	0.910	0.883	0.899
vowel0	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
glass-0-1-6_vs_2	0.911	0.909	**0.927**	0.914	0.913	0.901
glass2	0.877	0.921	0.917	0.912	0.914	**0.929**
yeast-1_vs_7	**0.794**	0.793	0.743	0.768	0.733	0.746
glass4	**0.981**	0.978	0.955	0.970	0.964	0.954
ecoli4	**0.997**	0.979	0.984	0.988	0.989	0.981
page-blocks-1-3_vs_4	0.998	0.997	**0.999**	**0.999**	0.997	**0.999**
abalone9-18	**0.951**	0.916	0.930	0.940	0.930	0.935
yeast-1-4-5-8_vs_7	0.743	**0.779**	0.732	0.708	0.727	0.775
glass5	0.990	**0.995**	0.991	0.988	0.985	0.983
yeast-2_vs_8	**0.931**	0.801	0.823	0.844	0.839	0.822
car_eval_4	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
wine_quality	**0.810**	0.797	0.745	0.760	0.746	0.764
yeast_me2	0.864	**0.898**	0.814	0.831	0.812	0.817
yeast4	0.867	**0.868**	0.804	0.814	0.799	0.804
yeast-1-2-8-9_vs_7	**0.734**	0.724	0.732	0.711	0.714	0.699
yeast5	0.994	**0.999**	0.984	0.984	0.984	0.984
yeast6	**0.956**	0.937	0.908	0.948	0.891	0.907
abalone19	**0.684**	0.652	0.590	0.670	0.594	0.643

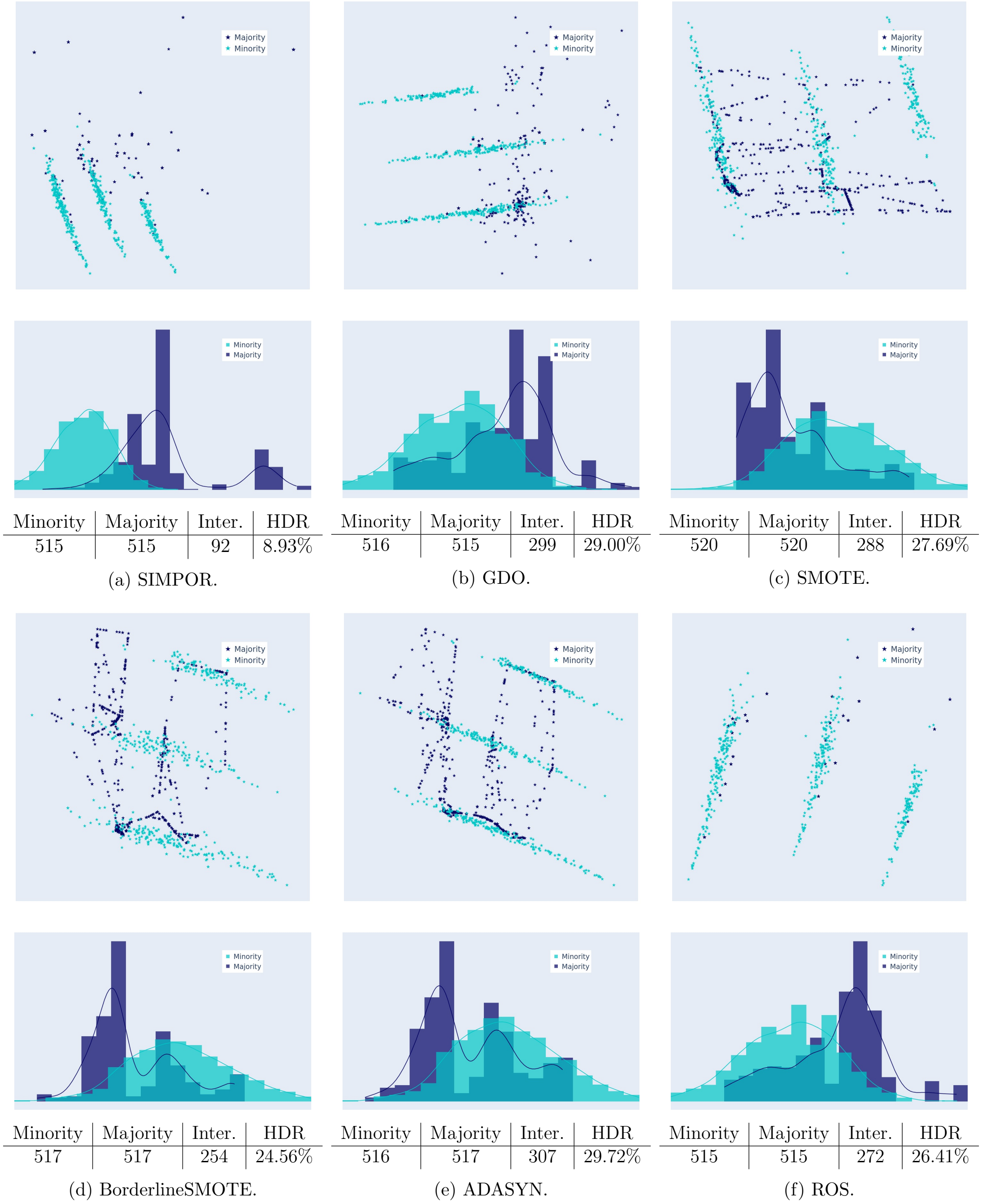
2) Data visualization.: To explore more on how the techniques perform, we visualize the generated data by projecting them onto lower dimension space (i.e., one and two dimensions) using the Principle Component Analysis technique (PCA) [22]. Data's 2-Dimension (2D) plots and 1-Dimension histograms are presented along with a hard-to-differentiate ratio (HDR) for each technique. A hard-to-differentiate ratio is defined as the ratio of intersection between 2 classes in the 1D histogram to the total of minority samples ($HDR = \frac{No.\ Intersection\ samples}{No.\ Minority\ samples}$). 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 are unable to be distinguished in the projected 1D space. Other than 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 ratio at 8.93%. 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-places samples.

D. 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 14. The classification performance is measured under different 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 result figure shows that the classification performance is not very sensitive to the r factor with the radius distribution standard deviation between 0.6R and R. While there are only small changes within the α range from 0.6 to 1, the performance is increasing in the range from 0.2 to 0.6 (i.e.,



Minority	Majority	Inter.	HDR
515	515	92	8.93%

(a) SIMPOR.

Minority	Majority	Inter.	HDR
516	515	299	29.00%

(b) GDO.

Minority	Majority	Inter.	HDR
520	520	288	27.69%

(c) SMOTE.

Minority	Majority	Inter.	HDR
517	517	254	24.56%

(d) BorderlineSMOTE.

Minority	Majority	Inter.	HDR
516	517	307	29.72%

(e) ADASYN.

Minority	Majority	Inter.	HDR
515	515	272	26.41%

(f) ROS.

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 intersectional samples to the number of minority samples ($HDR = \frac{Inter.}{Minority}100\%$).

ecoli1, abalone9-18,yeast4). These observations suggest us to use α from 0.6 to 1 for the selected datasets.

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

This section studies the empirical impact of the informative portion (IP) in Section II-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 VI-D. Different values of IP ranging from 0.1 to 1 are applied, and the classification performance results are shown in Figure 8.

As we observe from the figure, while the performance is not significantly affected to datasets achieved high performance (new-thyroid1, ecoli1), there are obvious peaks within the

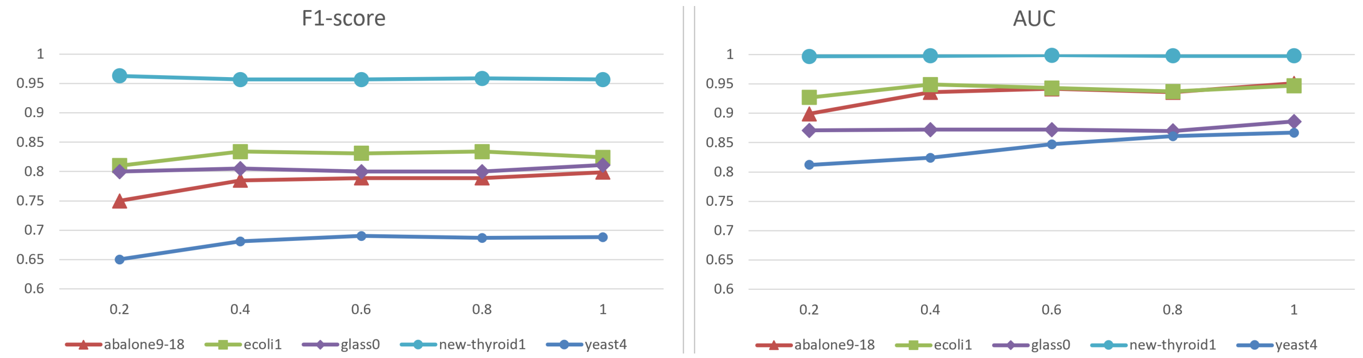


Fig. 7: F1-score and AUC results with varying Gaussian standard deviation (ranging form 0.2R to R) for r's distribution.

IP values of (0.2,0.6) in both F1-score and AUC score for other datasets (abalone9-18, glass0, yeast4). This suggests that tuning IP for each dataset between a range of (0.2,0.6) could achieve higher performance. For example, by adjusting IP from 0.1 to 0.3 in ablalone9-18 dataset experiment, we can increase the performance by 5% for both F1-score and AUC score.

VII. 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 for deep learning that can be extended to deep learning. These techniques can be categorized into two main categories, i.e., data-centric and model-centric approaches.

Model-centric approaches usually require modifications of algorithms on the cost functions to balance each class's weight. Specifically, such cost-sensitive approaches 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.

Compared to model-centric-based manners, data-centric approaches have attracted more research attention as they are independent of machine learning algorithms. This category divides into two main approaches, i.e., sampling-based and generative approaches. Sampling-based methods [27], [28], [29], [30], [31] 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, but we still consider them 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 the drawbacks of SMOTE is that if there are samples in the minority class located in the majority class, it creates synthetic sample bridges toward the majority class [32]. This renders difficulties in differentiation between the two classes. Another SMOTE-based work, namely Borderline-SMOTE [6] 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.

Under the down-sampling category, other works [10], [12] leverage active learning techniques to find informative samples which 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.

Generative approaches which generate synthetic samples in minor classes by sampling from data distribution are becoming more attractive as they are outperforming other methods in high dimensional data [33]. When it comes to images, a number of deep learning generative-based methods have been proposed as deep learning is capable of capturing good image representations. [34] [35] [36] utilized Variational Autoencoder as a generative model to arbitrarily generate images from learned distributions. However, most of them assumed simple prior distributions such as Gaussian for minor classes; they tend to simplify data distribution and might fail in sophisticated distributions. In addition, most of the works in this approach are tackling image datasets, while our proposed method focuses on tabular datasets as this is a missing piece in the field [3].

VIII. Conclusion

We propose a data balancing technique by generating synthetic data for minority samples maximizing the posterior ratio to embrace the chance they fall into the minority class and do not fall across the expected decision boundary. While maximizing the posterior ratio, we use kernel density estimation to estimate the likelihood so that it is able to work with

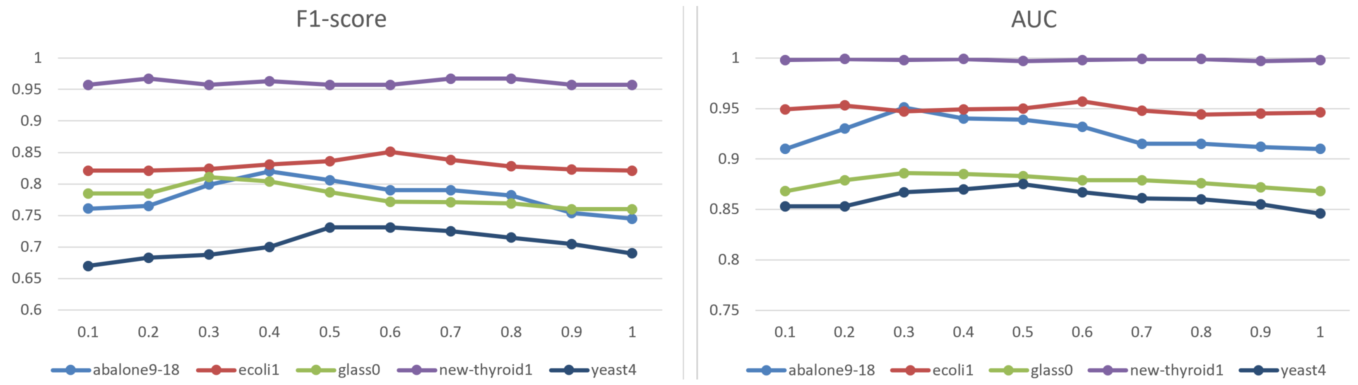


Fig. 8: F1-score and AUC results with varying informative portion IP.

complex distribution data without requiring data distribution assumptions. In addition, our technique leverage entropy-based active learning to find and balance the most informative samples. This is important to improve model performance as shown in our experiments on 41 real-world datasets. For future work, we would like to investigate the class imbalance for image data type and enhance our technique to adapt to image datasets.

Acknowledgment

Efforts sponsored in whole or in part by United States Special Operations Command (USSOCOM), under Partnership Intermediary Agreement No. H92222-15-3-0001-01. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. [1]

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. 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.

[7] 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.

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

[9] 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

[10] 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

[11] 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

[12] 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/

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

[14] 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.

[15] 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

[16] "Sphere," Aug 2021. [Online]. Available: https://en.wikipedia.org/wiki/Sphere

[17] "Maximal and minimal points of functions theory." [Online]. Available: https://aipc.tamu.edu/~schlump/24section4.1_math171.pdf

[18] NVIDIA, P. Vingelmann, and F. H. Fitzek, "Cuda, release: 10.2.89," 2020. [Online]. Available: https://developer.nvidia.com/cuda-toolkit

[19] 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.

[20] 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

[21] G. Goy, C. Gezer, and V. C. Gungor, "Credit Card Fraud Detection with Machine Learning Methods," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*. Samsun, Turkey: IEEE, Sep. 2019, pp. 350–354. [Online]. Available: https://ieeexplore.ieee.org/document/8906995/

[1] The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Special Operations Command.

[22] 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.

[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] 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

[28] 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

[29] 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/

[30] 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/

[31] 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

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

[33] 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.

[34] 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

[35] 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/

[36] 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/

Hung Nguyen received the M.Sc. degree and currently pursuing his Ph.D. degree in Department of Electrical Engineering, University of South Florida, FL, USA. His current research interests include machine learning, artificial intelligence, federated learning, cyber security, privacy enhancing technologies. Hung is a member of IEEE.

J. Morris Chang (SM'08) is a professor in the Department of Electrical Engineering at the University of South Florida. He received his Ph.D. degree from the North Carolina State University. His past industrial experiences include positions at Texas Instruments, Microelectronic Center of North Carolina and AT&T Bell Labs. He received the University Excellence in Teaching Award at Illinois Institute of Technology in 1999. His research interests include: cyber security, wireless networks, and energy efficient computer systems. In the last six years, his research projects on cyber security have been funded by DARPA. Currently, he is leading a DARPA project under Brandeis program focusing on privacy-preserving computation over Internet. He is a handling editor of Journal of Microprocessors and Microsystems and an editor of IEEE IT Professional. He is a senior member of IEEE.