Synthetic Information towards Maximum Posterior Ratiofor deep learning on Imbalanced Data: SIMPOR

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*—Most state-of-the-art machine learning (ML) and deep learning classification techniques assume the input data are class-balanced. In fact, it is common in real-world applications that some classes naturally contain significantly less data than others. This reduced ML algorithms' performance as the classifiers, especially deep learning models, are biased toward the majority class. While there have been a number of data balancing techniques proposed for conventional machine learning algorithms (e.g., SVM, regression family), imbalanced data classification for deep learning models has not been studied thoroughly.**

This work explores how imbalanced data affects deep learning and proposes a data balancing technique by generating synthetic data in the minority class. Instead of randomly generating data, our approach prioritizes balancing the most informative samples based on entropy theory, maximizes the probability that generated synthetic data fall into the minority class, and preserves the best data topology. In classification using deep neural network models, not all samples contribute equally to the models; samples located in the informative region carry the most important information. Thus, we start with finding and balancing such informative instances by leveraging an entropy-based active learning technique which results in high entropy samples to deep learning models. Since these samples are practically near class boundaries and might be disputed by different classes, generating synthetic data is critical. Thus, we safely generate surrounding neighbors of each minority sample to preserve data topology. More importantly, we ensure that the synthetic samples fall into the targeted class (minority class) and fall apart from the majority class by maximizing the posterior ratio between classes derived from Bayes' Theorem. Experimental results show that our technique constantly outperforms widely-used techniques over different settings of imbalance ratio and data dimension.

***Index Terms*—data imbalance, deep learning, maximum fractional posterior, informative samples**

I. Introduction

Data imbalance is a common phenomenon; it could be caused by sampling procedures or simply the nature of data. For example, it is difficult to sample some of the 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 processes is much smaller than that of common samples. Thus, machine learning models will be dominated by the majority class; this results in a higher error rate in prediction. Other work also observed that 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 data imbalance on conventional machine learning algorithms by analytically studying algorithms and developing corresponding strategies. However, the problem on heuristic algorithms such as deep neural networks is often difficult to tackle. In this work, we address data imbalance for deep neural network models by providing a solution that utilizes both deep learning techniques and statistical derivations of Bayes' Theorem.

We categorize existing proposed solutions into model-centric and data-centric approaches. As the first approach aims at modifying machine algorithms, and the latter looks for data balancing techniques. Perhaps data-centric methods are more commonly used because they do not tie to a specific model. In this category, a simple data balancing technique is to re-sample minority samples to balance the sample quantity between classes. This can preserve the best data structure and reduce the negative effect of data imbalance to some degree. However, this puts too much weight on a few minority samples; as a result, it causes overfitting problems in deep learning when the imbalance ratio becomes higher.

Another widely-used method in this category is Synthetic Minority Oversampling Technique (SMOTE) [3], 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. In addition, if there are minority samples located in the majority class, the method will generate sample lines that cross the decision boundary, which leads to the shifting of decision boundaries and misclassification. To improve SMOTE, Hui Han, *et al.* [4] proposed a SMOTE-based method (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's neighbors in Euclidean space contain samples from other classes, then it can be consid-

sample near the border. Since this method is entirely based on Euclidean distance from determining neighbors to generating synthetic data, it performs poorly in high dimensional space. Leveraging the same way as SMOTE generate synthetic samples, another widely-used technique, ADASYN [5], controls the number of generated samples by the number of different samples in classes within small groups of samples. Again, this technique still suffers SMOTE's drawbacks.

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 at the informative region where provides the most important information to the deep learning models. Besides, the technique enhances the chance that synthetic data fall into the minority class and still preserves the data topology.

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 then balance these samples first. 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 critical because it can affect the decision boundary. Therefore, we designate a direction to generate synthetic samples that maximize their posterior probability of belonging to the minority class 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 will disappear. 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 randomly generating neighbors for each sample. Proposed technique results in a balanced dataset that 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) Explore the impact of class imbalance on deep learning.
2) A proposed synthetic minority oversampling technique, namely Synthetic Information towards Maximum Posterior Ratio, to balance data classes and alleviate data imbalance effects. Our technique is enhanced by 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.
    c) Approximating the likelihood in Bayes rule using kernel density estimation, which can approximate a complicated statistical model. Thus, proposed technique is able to work with large, distributively complex data.

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, 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. 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 work and discuss future work.

II. Preliminaries

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

A. Imbalance Ratio (IR)

For binary classification, 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 samples and 100 class-B samples, the Imbalance Ratio is 10:1.

B. F1 Score

In this work, we evaluate balancing data techniques by the classification results on balanced data. To measure the accuracy of classification, we use Macro-averaging F1-Score, in which we compute F1 score per class and average with the same weight regardless of how often they appear in the dataset. The F1 score is computed based on two factors Recall and Precision.

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

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

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

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

C. Entropy-based Active Learning

To find informative samples, we leverage entropy-based active learning. The method gradually selects batch-by-batch samples that provide high information to the model based on information entropy theory [6]. 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 as 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

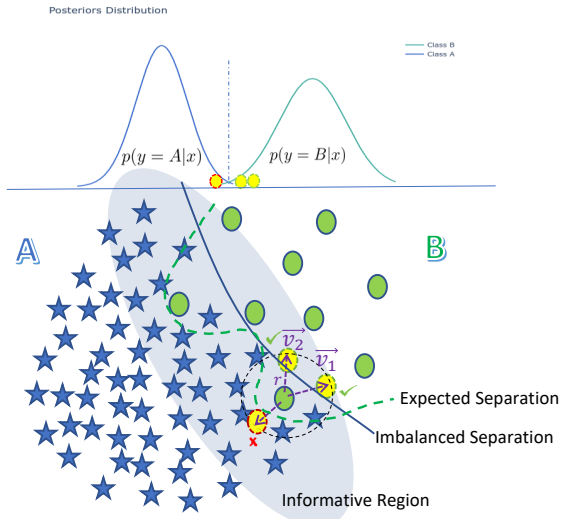


Fig. 1: Learning from imbalanced dataset

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.

In this work, we consider a ~~setting that includes 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 θ_0 on a random batch of k labeled samples and use the θ_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 to existing labeled data to train the classifier parameter θ_1 in the next step t_1. Steps are repeated until all sample's entropy are less than a threshold e.g., $Threshold = 0.7$.

III. The Problem of Learning From Imbalanced Dataset

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 separation errors. The dashed green line depicts the expected boundary, while the solid blue line is the model's boundary. Since the minority class is lacking data in this region, the majority class will dominate the model even with a few noisy samples, and this leads to a shift of the model's boundary. In contrast to the study by Ertekin *et al.* [7] which assumes the informative region is more balanced by nature and proposes a solution that only classifying over the informative samples, our assumption is different. We consider the cases that the informative region contains high imbalanced data, which we believe happens in most of the real cases. In a more complex setting such as high dimensional and topologically complex data, the problem could be more severe. 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 synthetic surrounding neighbors for minority samples. The remaining region is then fully balanced by arbitrarily generating minority samples' neighbors. We elaborate how our strategy is developed in the rest of this section.

A. Methodology Motivation

As Chazal and Michel mentioned in their work [8], the natural way to highlight the global topological structure of the data is to connect data points' neighbors; our proposed method preserve their observation but in the reverse procedure, generating surrounding synthetic neighbors for minority samples. Thus, our method not only generates more data for minority class but also preserve the underlying topological structure of the entire data.

Agreeing to the mutual idea in [7] and [9], 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 method 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 VI-A to find the samples that maximize entropy to the model. This technique is perhaps the most popular one and over-performs many other techniques on several datasets [10], [11] [12].

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 ~~in~~ the informative region because synthetic samples can unexpectedly jump across the decision boundary. This can be harmful to the model as this might reduce model's

performance. Therefore, we safely find vector $\vec{v}$ towards the minority class as this is also depicted in Figure 1. Our technique is described via a binary classification scenario as follows.

Let consider a binary classification between majority class A and minority class B. From the Bayes' theorem, the posterior probability $p(y' = A|x')$ and $p(y' = B|x')$ can be used to present the probabilities that a synthetic sample x' belongs to class A and 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 the synthetic data x' that maximize the probability of x' belonging to the minority class B and minimize the chance x' belonging to the majority class. Thus, we propose a method 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, and it has interesting property that is useful when we compute the derivatives for finding optima.

Approximation of priors in Equation 8: Also, we assume prior probabilities of observing samples in class A ($p(A)$) and class B ($p(B)$) (Equation 8) are constant. In fact, these probabilities do not affect our algorithm in terms of generating synthetic neighbors for minority samples as we only determine the relative direction between the minority and majority class. Thus, they can be canceled out at the end of the optimization problem.

Equation 8 reduction: The posterior ratio for each synthetic sample x' then can be estimated as follows:

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

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

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

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

where X_A and X_B are the subsets of dataset X containing class A and class B respectively, $X_A = \{x : y = A\}$ 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 dimension. h presents the width parameter of the Gaussian kernel.

Finding synthetic samples surrounding a minority sample: Because we want to generate neighbors for each minority sample that maximize 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 uniform distribution $r \sim \mathcal{U}(0, R)$, $0 < r < R$. The range parameter R is relatively small and computed as the average distance of k-nearest neighbors of the minority sample x to itself. This will ensure that the generated sample will be surrounding the minority 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 number of the neighbors. In practice, we prefer to tune k from 5 to 20.

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

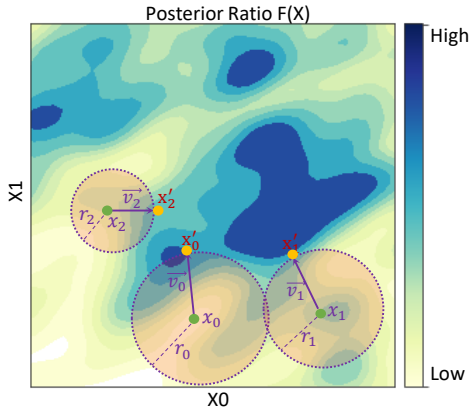


Fig. 2: Demonstration on how SIMPOR generates three synthetic samples x'_0, x'_1, x'_2, from minority three minority samples x_0, x_1, x_2, by maximize the Posterior Ratio.

sample $x' \in \mathbb{R}^d$ lying on the d-sphere centered at x with radius r and maximizing function in Equation 12,

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

Solving optimization problem 15: Interestingly, Problem 15 is solvable. 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 [13]. Thus, a maximum exits as proved in [14]. To enhance the diversity of synthetic data, we except either the global maximum and any local maximum, so that the synthetic samples will not simply go the same direction.

We solve Problem 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 Problem 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 with sampling a random point on the constraint sphere (line $1-2$). The gradient of objective function at time t, $g_t(x'_t)$, is computed and projected onto 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 $lr*\theta$. 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. Our 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 which is shown in lines $1-2$ in Algorithm 2. We choose entropy-based active learning for our experiments. 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 $3-10$). 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 11 to line 17. The final output of the algorithms is a balanced dataset D'.

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

V. Algorithm Implementation and Complexity

Our proposed method is straightforward in implementation. We first train a neural network model with initial samples and start querying next batches data based on the entropy scores from 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 a local maxima in Equation 12.

Perhaps the costly part of SIMPOR is that each synthetic sample requires to compute a kernel density estimation of the entire dataset. Let n be the number of samples of the dataset. In the worst case, the number of samples of minority and majority class are $N_B = 1$ and $N_A = n - 1$ respectively. We need to generate $n-1$ synthetic samples to completely balance the dataset. Since each generated sample must loop

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.
$Count(c, P)$: 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 $\{X', y'\}$

Select an Active Learning Algorithm $AL()$
Query a subset of informative samples $S \in D$ using AL: $s \leftarrow AL(D)$
{Balance the informative region}
for $c \in MI$ **do**
  while $Count(c, S) \leq Count(MA, S)$ **do**
    Select a random $x_i^c \in S$
    Compute maximum range R based on k
    Randomly sample a radius $r \sim \mathcal{U}(0, R)$
    Generate a synthetic neighbor x' from x_i^c: $x' = G(x_i^c, f, r)$
    Append x' to D'
  end while
end for
{Balance the remaining region}
for c in MI **do**
  while $Count(c, D') \leq Count(MA, D')$ **do**
    Select a random $x_j^c \in \{X - S\}$
    Compute maximum range R based on k
    Randomly sample a radius $r \sim \mathcal{U}(0, R)$
    Generate a synthetic neighbor x' of x_j^c
    Append x' to D'
  end while
end for
return

through the entire dataset of size n to estimate the density, the complexity is $O(n^2)$. Although generating synthetic data is only a one-time process, and this does not hurt the classification performance in the testing phase, we still alleviate its weakness by providing parallelized implementations. We provide two suggestion, multiple CPU thread-based and GPU-based implementations. While the former simply computes each synthetic data sample in a separated CPU thread, the later computes each exponential component in 12 parallelly in GPU's 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 which utilize CUDA toolkit from NVIDIA [15]. The consumption time for kernel density estimation for each synthetic data point is then $N_A + N_B = n$ times reduced which reduces the complexity to $O(n)$. Our source code is written in python and published on Github.

VI. Experiments and Discussion

In this section, we experiment on binary classification for both artificial dataset (i.e., Moon) for demonstration and real-world datasets (i.e., Breast Cancer, Credit Card Fraud). Samples in artificial Moon have two dimensions while samples in Breast Cancer and Credit Card Fraud are both 30-dimension numerical data. Compare to original Breast Cancer dataset size (569 samples), the other original Credit Card dataset contains much larger amount of data, 284,907 samples. The implementation steps to balance datasets are following Algorithm 2. To evaluate our proposed balancing technique, we compare the classification performance to different widely-used techniques. We measure Recall, Precision, F1-score, Receiver Operating Characteristic Curve (ROC) and Area Under The Curve (AUC) for each classification result as they are powerful and widely-used metrics to evaluate classification performance especially for skewed datasets. More specifically, We compare our proposed technique SIMPOR to SMOTE [3], Borderline-SMOTE [4], ADASYN [5], Random Oversampling and Raw data which does not apply any balancing technique.

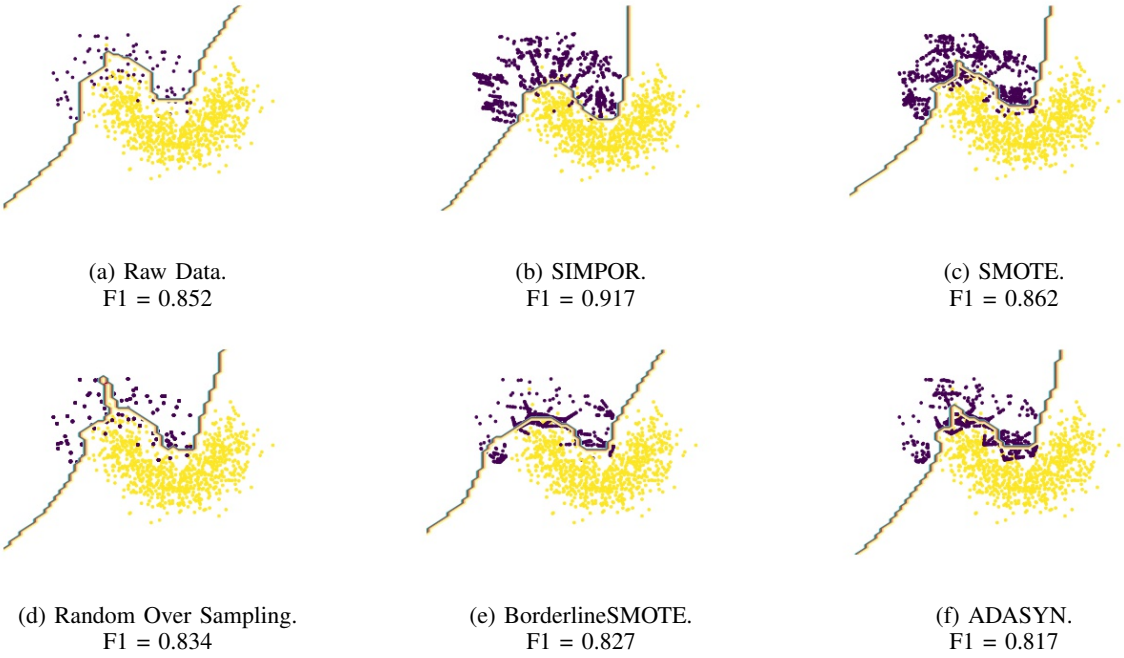
A. Sharing Settings

This subsection describes settings sharing for all datasets. In order to find the informative subset, we leverage entropy-based active learning as mentioned in Section . We use a fully connected neural network containing 3 hidden layers with *relu* activation functions and 100 neurons each layer. The output layer uses softmax activation function. The models are trained in maximum of 300 epochs with the early stop option when the loss does not change after updating weights.

In addition, we randomly split the data into two parts, 75% for training and 25% for testing. Reported results for each dataset are the averages of 5 experimental trials. The detail of model architecture for each dataset is described in Table I. Also, for SIMPOR to find optima of Problem 1, we use gradient ascent rate of 0.00001 and the maximum iteration of 300.

TABLE I: Classifier models' setting for each dataset.

Model Setting	Moon	Breast Cancer	Credit Card
Neurons/Layer	100	50	200
Hidden Layers	3	3	3
Hidden Activation	ReLu	ReLu	ReLu
Output Activation	Softmax	Softmax	Softmax
Epochs	200	150	200
Batch size	32	32	64
Optimizer	Adam	Adam	Adam
Learning Rate	0.01	0.01	0.01



(a) Raw Data. F1 = 0.852
(b) SIMPOR. F1 = 0.917
(c) SMOTE. F1 = 0.862
(d) Random Over Sampling. F1 = 0.834
(e) BorderlineSMOTE. F1 = 0.827
(f) ADASYN. F1 = 0.817

Fig. 3: Balanced data plot and model's decision boundary for Moon Dataset.

B. SIMPOR on artificial Moon dataset

We implement our technique on an artificial 2-dimension numeric dataset as a demonstration of our proposed method. Figure 3 captures the classification F1 result for different techniques. We also visualize model decision boundary to provide additional information on how a the classification model is affected by different techniques. To classify the data, we use a fully connected neural network which is described in Table I.

1) Dataset: We first generate a balanced dataset using python library *sklearn.datasets.make_moons* including 3000 two-dimensional samples labeled in two classes, A and B. We then create an imbalanced dataset with a Imbalance Ratio of 3:1 by randomly removing 1000 samples from class B. We then split the data into two parts, 80% for training and 20% for testing. As a result, the training dataset becomes imbalanced as visualized in Figure 3a, which contains 400 samples of class B and 1200 samples of class A. The testing data contains 100 samples of class B and 300 samples of class A.

2) Results and Discussion: From the results show in Figure 3, it is clear that SIMPOR performs better than others at the F1 score of 0.9170 (Figure 3b). Without any balancing techniques, it is no surprise that the classification result on the raw imbalanced data achieves the lower F1 Score at 0.852 (Figure 3a). SMOTE technique does help to improve the classification F1-score to 0.862; however, it has not reached SIMPOR's performance. Other techniques perform poorly in this case; they could not achieve F1-score as high as the raw data can achieve.

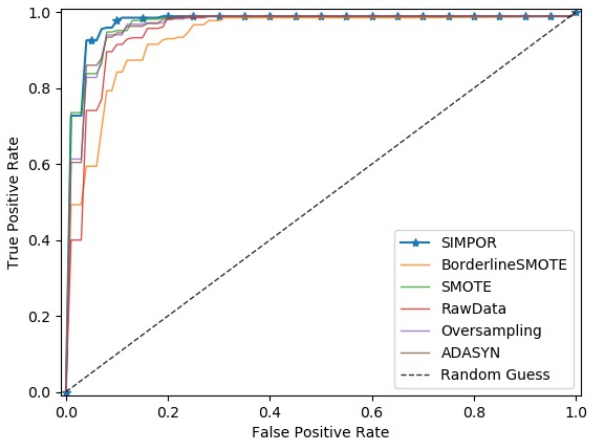
Additionally, SIMPOR, from our observation, results in a smooth and robust model decision boundary. We can see that the Random Over Sampling which randomly duplicates minority samples might cause overfitting where samples are duplicated many times and significantly increase their weights. SMOTE does better than Random Over Sampling. However, because of generating synthetic samples on the path connecting between samples in euclidean space, SMOTE cannot generate samples in several areas in the feature space whereas it is very dense at edges. Besides, due to the fact that SMOTE does not take the informative region into account, unbalanced data in this area led to a severe error in decision boundary. In Figures 3f and 3e, BorderlineSMOTE and ADASYN focus on the area near model's decision boundary, but they inherit a drawback from SMOTE; any noise or mislabeled samples can create very dense connections crossing the expected border, and it also leads to decision errors. In contrast, by generating neighbors of samples in the direction towards the minority class and balancing the informative region, SIMPOR (Figure 3b) helps the classifier to make a better decision with a solid smooth decision line. It is also worth to notice that the classifier decision boundary lines of all other techniques are rougher than that of SIMPOR. This is because they randomly generate synthetic samples, and it might cause data imbalance in the informative region.

C. SIMPOR on Breast Cancer dataset

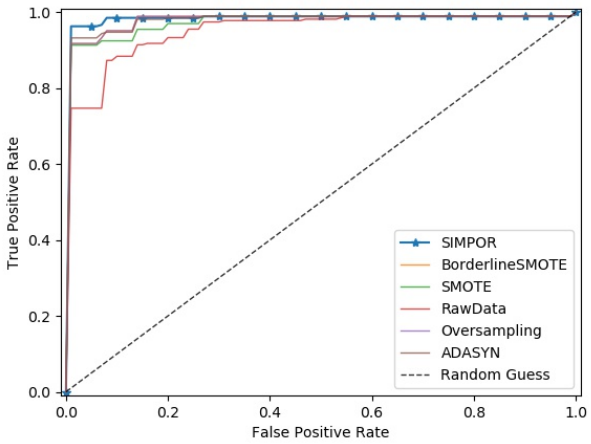
1) Dataset: Breast Cancer is a real-world dataset containing information of cancer patients collected by Dua and Graff [16]. This has two classes including 357 negatives and 212 positives; each record includes 30 features extracted from a digitized image of a fine needle aspirate of a breast mass e.g., radius, area, and perimeter. Some of the records in the training

TABLE II: Classification results of difference data balancing techniques on Breast Cancer Dataset.

Breast Cancer							
Majority:Minority (IR)	Metric	**SIMPOR**	**SMOTE**	**Borderline SMOTE**	**Over-Sampling**	**ADASYN**	**RawData**
357:120 (3:1)	F1	**0.953**	0.931	0.891	0.935	0.835	0.944
	AUC	**0.974**	**0.974**	0.964	**0.974**	**0.974**	0.971
	Precision	0.97	0.924	0.868	0.935	0.79	**0.975**
	Recall	**0.938**	**0.938**	0.922	0.936	0.928	0.919
357:50 (7:1)	F1	**0.943**	0.903	0.865	0.939	0.821	0.879
	AUC	**0.968**	**0.968**	0.967	0.967	**0.968**	0.966
	Precision	**0.952**	0.899	0.846	0.948	0.805	0.873
	Recall	**0.936**	0.918	0.903	0.932	0.879	0.907



(a) IR 3:1



(b) IR 7:1

Fig. 4: AUC-ROC Curve for Breast Cancer on two different Imbalance Ratios.

dataset are randomly removed to create different imbalance ratios.

2) Results and Discussions: Table II shows the classification results of different data balancing techniques on Breast Cancer dataset over the imbalance ratios of 3:1 and 7:1. Overall, SIMPOR outperforms other techniques on both imbalance ratios at F1-score of 0.947 and 0.935 for IR of 3:1 and 7:1, respectively. SIMPOR also achieves highest AUC results at 0.977 and 0.986 for both imbalance ratios. Both tests over the raw data (without any balancing technique) received lowest F1-score as expected.

Figure 4 shows the receiver operating characteristic curve (ROC) which illustrate the ability of the classifiers. As shown in the figure, SIMPOR's ROC curves are closest to the upper left corner. In other words, SIMPOR reaches higher true positive rate and lower fall positive rates.

D. SIMPOR on Credit Card Fraud

1) Dataset: In this section, we experiment our technique on Credit Card Fraud dataset [17]. The original dataset contains 492 fraud records out of 284,807 transactions; each record includes 30 features. During the experiments, we fix the size of fraud records as the minority class and randomly select a part of the remaining normal transactions to generate new datasets with different imbalance ratios. Classification model for experiments under this dataset can be found in Table I.

2) Results and Discussions: Table III shows the classification results on Credit Card Fraud dataset over different balancing techniques. SIMPOR constantly overperforms other techniques over all the settings and achieve the highest F1 and AUC score. While SIMPOR, SMOTE improve classification performance in most of the settings, ADASYN, BorderlineSMOTE fail to create good synthetic samples and reduce the classification performance.

To better understand how techniques perform, we visualize the generated data by projecting them onto lower dimensions (i.e., one and two dimensions) space using Principle Component Analysis technique (PCA) [18]. Data's 2-Dimension (2D) plots and 1-Dimention histograms are presented in Figure 5. A hard-to-differentiate ratio (HDR) is defined as the ratio of intersection between 2 classes in the 1D histogram to the total Fraudulent samples ($HDR = \frac{No.\ Intersection\ samples}{No.\ Fraud\ samples}$). This ratio should be as small as 0% if after reducing the dimension to one, the two classes are well separated; in contrast, 100% indicates that the two classes are unable to be distinguished. Other than HDR, the bottom tables in Figure 5 also show the absolute numbers of Fraudulent, Normal and Intersection

TABLE III: Classification results of difference data balancing techniques on Credit Card Fraud Dataset.

Credit Card Fraud							
Majority:Minority (IR)	Metric	**SIMPOR**	**SMOTE**	**Borderline SMOTE**	**Over-Sampling**	**ADASYN**	**RawData**
1470:490 (3:1)	F1	**0.943**	0.903	0.865	0.939	0.821	0.879
	AUC	**0.968**	**0.968**	0.967	0.967	**0.968**	0.966
	Precision	**0.952**	0.899	0.846	0.948	0.805	0.873
	Recall	**0.936**	0.918	0.903	0.932	0.879	0.907
3430:490 (7:1)	F1	**0.953**	0.931	0.891	0.935	0.835	0.944
	AUC	**0.974**	**0.974**	0.964	**0.974**	**0.974**	0.971
	Precision	0.97	0.924	0.868	0.935	0.79	**0.975**
	Recall	**0.938**	**0.938**	0.922	0.936	0.928	0.919
4900:490 (10:1)	F1	**0.952**	0.901	0.863	0.658	0.906	0.93
	AUC	**0.972**	0.957	0.967	0.964	0.969	0.975
	Precision	**0.979**	0.891	0.855	0.662	0.885	0.931
	Recall	0.929	0.925	0.918	0.845	0.933	**0.936**
7350:490 (15:1)	F1	**0.952**	0.909	0.885	0.917	0.883	0.944
	AUC	**0.968**	0.963	0.955	0.963	0.965	0.960
	Precision	0.966	0.892	0.850	0.909	0.849	**0.984**
	Recall	**0.940**	0.931	0.931	0.935	0.931	0.911

samples for each technique. From the plots, we can observe how the data distribute in 2D space and quantify hard-to-differentiate samples in the histograms.

While some techniques help to reduce hard-to-differentiate ratio, others increase this ratio and worsen the data distribution. For example, in the 1D histogram of the raw imbalanced data in Figure 5d, there are 2115 samples in the histogram intersection between 2 classes; they accounts for 81.82% of the total 2585 Fraudulent samples, which is worse than HDR of Rawdata (69.75%). In addition, the 2-Dimension plot illustrates that many synthetic samples (Fraudulent class) cross the majority (Normal class). Similarly, Figure 5e shows that BorderlineSMOTE also poorly generates synthetic minority data in this setting (CreditCard Fradulent dataset with IR of 7:1) with the HDR of 75.98%. Among all techniques, SIMPOR achieves the best HDR of 11.14% and the synthetic data are far away from the majority class which helps to improve the classification results as shown in Table III.

VII. Related Work

In the last few decades, there have been a number of solutions proposed to alleviate the negative effect of data imbalance in machine learning. Some of them might work with modern models, but some are not efficient when it comes to high-dimensional data and deep learning. In this section, we review algorithms that aim at deep learning and strategies inherited from conventional machine learning methods. These techniques can mainly be categorized into two different categories, i.e., data-centric and model-centric approaches.

Model-centric approaches usually require modifications of algorithms on the cost functions in order to balance the weight of each class. 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, [19] 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. [20], [21], [22] assign classes' weights inversely proportional to sample frequency appearing in each class.

Comparing to model-centric-based manners, data-centric approaches have been attracting more researching attention as it is independent to machine learning algorithms. In this category, we divide into two main approaches, i.e. sampling-based and generative approaches. Sampling-based methods [23], [24], [25], [26], [27] mainly generate a balanced dataset by either over-sampling minority classes or down-sampling majority classes. Some methods 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 [3], Chawla *et al.* attempt to oversampling minority class samples by connecting an sample to its neighbors in feature space and arbitrarily drawing synthetic samples along 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 towards the majority class [28]. This makes classifiers difficult to classify the two classes. Another SMOTE-based work namely Borderline-SMOTE [4] 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 (if more than a half of neighbors belong to the other class, the sample is considered to be on the border). This "border" idea is similar to ours to some degree. However, finding a good *k* is critical, and it is usually highly data-dependent. In addition, Borderline-SMOTE again faces the problems of SMOTE.

Under the down-sampling category, other works [7], [9] leverage active learning techniques to find informative samples which authors believe the imbalance ratio in these areas are much smaller than that in the entire dataset. They then classify

on 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 important 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 [29]. 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. [30] [31] [32] 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 not succeed in sophisticated distributions. Our solution also falls into this category; however, we leverage the idea of a mixture model to tackle this issue for image data.

VIII. Conclusion

We propose a data balancing technique by generating synthetic data for minority samples, which maximize the posterior ratio to embrace the chance they fall into the minority class. While maximizing the posterior ratio, we use kernel density estimation to estimate the likelihood so that it is able to work with topologically complex data. 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 we have shown in our experiments. We also find that applying our technique to the image dataset might create synthetic images such as a noisy version of minority images. This might help to improve the model to some degree; however, we believe that generating more meaningful synthetic images will improve the model better. In future work, we would like to investigate imbalanced image datasets and enhance our technique to adapt to image data.

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] 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

[4] 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.

[5] 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.

[6] 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

[7] 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

[8] 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

[9] 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/

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

[11] 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.

[12] 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

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

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

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

[16] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[17] 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/

[18] 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.

[19] 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/

[20] 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/

[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.

[21] 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

[22] 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

[23] 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

[24] 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

[25] 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/

[26] 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/

[27] 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

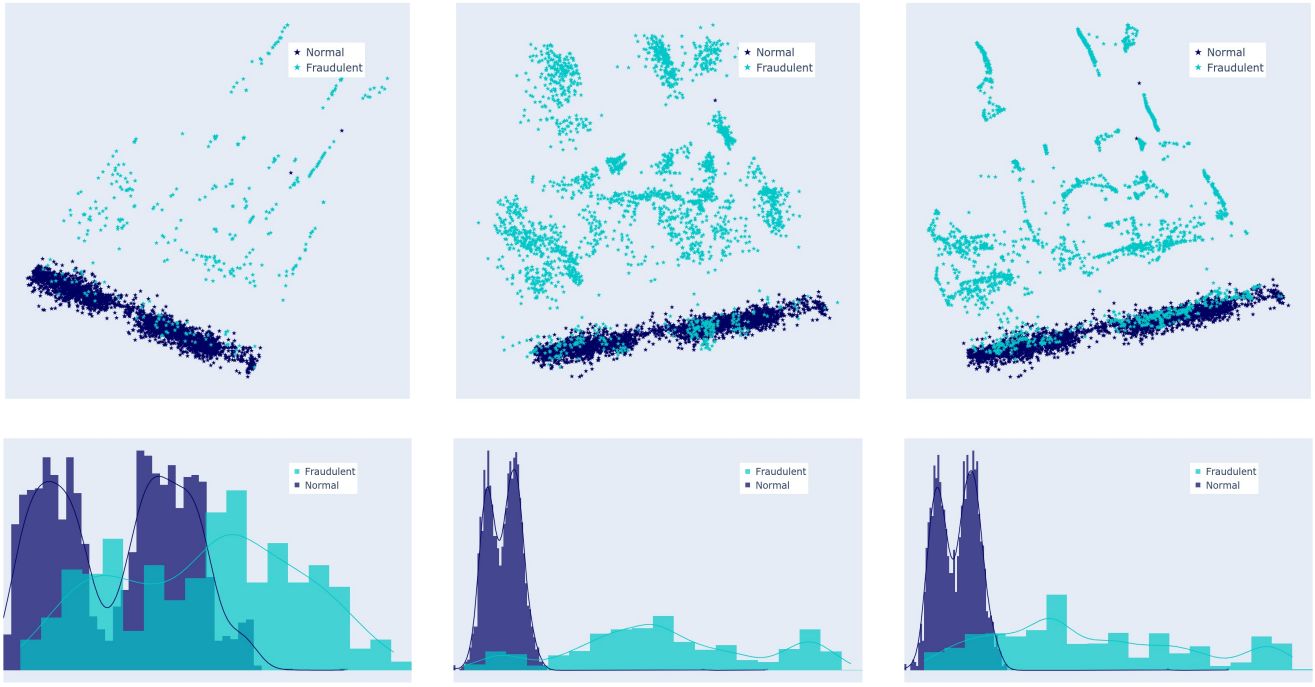
[28] D. S. Goswami, "Class Imbalance, SMOTE, Borderline SMOTE, ADASYN," Nov. 2020. [Online]. Available: https://towardsdatascience.com/class-imbalance-smote-borderline-smote-adasyn-6e36c78d804

[29] 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.

[30] 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

[31] 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/

[32] 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/



Fraud	Normal	Inter.	HDR
367	2585	256	69.75%

(a) Raw Data.

Fraud	Normal	Inter.	HDR
2585	2585	288	**11.14**%

(b) Generated data by SIMPOR.

Fraud	Normal	Inter.	HDR
2585	2585	534	20.66%

(c) Generated data by SMOTE.

Fraud	Normal	Inter.	HDR
2585	2585	2115	81.82%

(d) Generated data by ADASYN.

Fraud	Normal	Inter.	HDR
2585	2585	1964	75.98%

(e) Generated data by BorderlineSMOTE.

Fraud	Normal	Inter.	HDR
2585	2585	544	21.4%

(f) Generated data by Oversampling.

Fig. 5: Generated training data projected onto 2-dimension space and their histograms in 1-Dimension space using Principle Component Analysis dimension reduction technique. The explanation tables illustrates the number of samples in each class (Fraudulent and Normal), 1-Dimension histogram intersection between 2 classes and the hard-to-differentiate ratio ($HDR = \frac{Inter.}{Fraud}100\%$).