Wasserstein Generative Adversarial Network – Gradient Penalty (WGAN-GP) for class imbalance problems: A combined strategy using WGAN-GP and synthetic over-sampling techniques—SMOTE, Borderline-SMOTE, and ADASYN.

by

Nediff Amala Nixon

University of Nottingham

2023

A Dissertation presented in part consideration for the degree of MSc Business Analytics

Abstract

Class imbalance is one of the most common problems encountered by classification algorithms. To address this shortcoming, pre-processing methods that oversample minority classes are typically used. The Synthetic Minority Over-Sampling Technique (SMOTE), which focuses on local information but produces data that is not sufficiently realistic, is the foundation of traditional oversampling methodologies. In contrast, the Generative Adversarial Network (GAN) can also be used to generate synthetic samples for the minority class by capturing the real data distribution. However, GAN models are prone to mode collapse and unstable training. To overcome these problems, we propose a knowledge transfer-based two-phase over-sampling strategy that combines the benefits of traditional over-sampling techniques (SMOTE, Borderline-SMOTE and ADASYN) and Wasserstein Generative Adversarial Network – Gradient Penalty (WGAN-GP) for class imbalance problems. This combined strategy generates more realistic and diverse samples and overcomes the aforementioned problems.

Experiments on 6 different datasets with diverse class imbalance ratios (IR) empirically demonstrate that the proposed method outperforms the traditional over-sampling and GAN-based methods in 3 out of 6 datasets. The proposed strategy outperforms other over-sampling techniques when precision is chosen as the evaluation metric. The performance on other evaluation metrics (Recall, F1-score, G-mean and the area under the receiver operating characteristic curve) was inconclusive and might require further analysis.

Keywords: Class imbalance, Synthetic Minority Over-Sampling Technique (SMOTE), Borderline-SMOTE, ADASYN, Generative Adversarial Network (GAN), Wasserstein Generative Adversarial Network – Gradient Penalty (WGAN-GP), evaluation metrics.

Table of Contents

List of Tables

List of Figures

Introduction

Many real-world classification problems are imbalanced where the distribution of samples across the known classes is skewed or biased. The degree of imbalance can be mild, moderate, or extreme depending on the proportion of minority class typically ranging from 40% to less than 1% of the data set (Google Developers, 2023). The majority of machine learning methods used for classification were created assuming that each class had an equal number of examples, which makes class imbalance a difficulty for predictive modelling. As a result, models perform poorly in terms of prediction, particularly for the minority class. This is a concern since, in general, the minority class is more significant and, in many real-world scenarios, misclassifying members of the minority class might have more catastrophic consequences or greater related costs. For instance, in medical diagnostics, misdiagnosis of a rare disease (minority class) might be fatal, whereas misdiagnosis of a healthy person (majority class) may not have as severe repercussions.

Data imbalance can be commonly seen in fraud/fault/anomaly detection (Makki et al., 2019; Wei et al., 2013; Zhuo and Ze, 2020; Huang and Lei, 2020), medical diagnosis of lethal and rare diseases (Bria et al., 2020; Qin et al., 2020), software defection prediction (Rodriguez, 2019) and natural disaster. The commonly used pre-processing technique is over-sampling and the Synthetic Over-Sampling (SMOTE) technique is considered the golden standard. Although SMOTE can somewhat increase the accuracy of the minority class, it poses the risk of producing noisy samples and overfitting issues because the distribution of nearby samples is not considered (Meng and Li, 2022). Many variants (Borderline-SMOTE and ADASYN) were proposed in the literature to address the problems of SMOTE, but they still suffer from the inclusion of illegitimate samples which is still an active area of research (Zhu, Lin and Liu, 2020; Tao et al., 2020).

On the other hand, Generative Adversarial Network (GAN) and its variations are trending in the field of image processing to generate synthetic images that resemble real-life images. However, it can also be used to generate synthetic samples for class imbalance data sets and avoid over-fitting effectively (Zhang, Yang and Jiang, 2018). GAN and its variations have been successfully applied to many applications such as skin lesion classification for better diagnosis (Qin et al., 2020) or pipeline leakage in petrochemical systems (Xu, Du and Zhang, 2019). In 2018, Douzas and Bacao proposed that Conditional GAN (CGAN) can be used to generate new samples with an auxiliary condition as part of the input data. However, GAN models are prone to model collapse and unstable training (Kodali et al., 2017).

To overcome these problems, in 2020, Zheng et al. proposed the Conditional Wasserstein GAN - Gradient Penalty (CWGAN-GP) as an over-sampling approach for imbalanced datasets. The approach takes the output class labels of a dataset as an auxiliary condition and learns the global distribution of the data. The problem arises when the dataset is highly imbalanced, CWGAN-GP might fail to learn the fine details of minority class distribution. In another study by Sharma et al. (2021), they proposed a combined strategy using SMOTE and GAN, where the SMOTE-generated samples were further enhanced by the GAN model. Although this approach produces impressive results and solves the problem of CWGAN-GP, it still faces the problems of GAN mentioned before.

In this study, we propose a novel approach that extends the work of Sharma et al. (2021) by replacing the GAN model with the WGAN-GP model and replacing SMOTE with Borderline-SMOTE and ADASYN. We have obtained mixed results on 6 numerical datasets from the UCI library, Kaggle and GitHub. The nature of the performance depends upon the evaluation metric under consideration. The proposed method outperformed traditional over-sampling techniques and the GAN model in 3 out of 6 datasets.

Related Work

The benefits of sampling techniques for class imbalance problems include their greater adaptability and the fact that they can be used regardless of the classifiers that are chosen (Fernández et al., 2013). In random under-sampling, the desired number of samples from the majority class are randomly removed from the data set. Random under-sampling is computationally easier for learning compared to random over-sampling because it reduces the dataset size, avoids data duplication, and is more memory efficient. Random under-sampling discards samples from the majority class, it removes useful information that could be beneficial for better learning (Brandt and Lanzén, 2020; He and Garcia, 2009; Weiss and Provost, 2001). Popular under-sampling methods include the condensed nearest neighbour rule (CNN), Wilson's edited nearest neighbour rule (ENN), Tomek links, and one-sided selection (OSS).

Random over-sampling (ROS) adds samples of the minority class that are selected at random to the original data set. Although it is a simple method, it has been argued that exact replicas of samples from the minority class and highly detailed rules produced by this type of over-sampling can cause the algorithm to over-fit and over-generalize (Prati, Batista and Monard, 2004; Weiss and Provost, 2001). Because it doesn't capture the true underlying patterns in the data, lacks diversity, and amplifies noise within the minority class samples.

Therefore, to mitigate these problems, SMOTE, currently one of the most popular over-sampling techniques was proposed by Chawla et al. (2002). Although SMOTE performs remarkably well in many scenarios, it is still reported to have an over-generalization problem (Wang and Japkowicz, 2004). Over-generalization happens when the synthetic samples generated by SMOTE capture the noise or local variations present in the minority class rather than the true underlying distribution. This is especially problematic when the minority class is inherently noisy or contains outliers. Additionally, it creates synthetic samples regardless of the majority class, which causes the overlap between the classes to grow significantly (Prati, Batista and Monard, 2004). This problem causes samples from the majority class to be incorrectly classified as a member of the minority class and raises the false positive rate of learning algorithms. To overcome the SMOTE difficulties, much research is proposed in literature including safe-level SMOTE, ADASYN and Borderline-SMOTE that try to solve each other's shortcomings. The implementation of 85 variants of SMOTE in the Python library is shown by Kovács (2019). However, all these techniques face challenges when dealing with high-dimensional data, and data scarcity. This motivates us to find other solutions for synthetic over-sampling.

Generative Adversarial Network (GAN) is a class of machine learning framework that generates new samples with the same statistics as the original data by capturing the true data distribution (Goodfellow et al., 2014). Instead of over-sampling imbalanced classes, GAN was built to produce synthetic images from genuine images that should be difficult to tell apart. However due to its success in data augmentation for over-sampling, a variety of GAN variants have been developed to address issues with class imbalance.

In 2018, Douzas and Bacao proposed that Conditional GAN (CGAN) can be used to generate new samples of the minority class by learning the global distribution of the dataset, where the condition used in CGAN was the class labels.

However, this approach is problematic as GAN and CGAN are prone to mode collapse and unstable training. Mode collapse occurs when a GAN fails to produce a broad and varied sample set from the target data distribution. Instead, it creates a limited subset of samples that frequently converge to one mode or a few modes in the data distribution. This is not desired as the samples generated often lack diversity, variation, and quality. Unstable training refers to the difficulty of training GANs due to issues such as vanishing gradients, oscillating loss functions, or slow convergence. Finding a careful balance between the generator and discriminator networks is required while training a GAN. When this balance is disrupted, training can become highly unstable and challenging to control. To overcome these problems, in 2020, Zheng et al. proposed the Conditional Wasserstein GAN - Gradient Penalty (CWGAN-GP) as a novel and efficient synthetic oversampling approach for imbalanced datasets. Zheng et al. argued that CWGAN-GP resolves the aforementioned problems and produces more realistic data. Furthermore, they stated that based on their experiment on 15 different benchmarked datasets and two real imbalanced datasets, CWGAN-GP was able to increase the quality of synthetic data and outperform the other over-sampling techniques based on three evaluation metrics (F-measure, G-mean and the area under the receiver operating characteristic curve) for five classifiers.

Although the result seems promising, both CGAN and CWGAN-GP in the case of moderate or extreme class imbalance are prone to loss of local details where they fail to capture fine-grained local details of the minority class. The generated samples may lack specific characteristics and nuances present in the original data, potentially leading to a loss of information. Furthermore, it is also prone to class collapse where the generated samples of the minority class become too similar or converge towards a limited set of representations. This can limit the diversity and variability of the synthetic samples and may not fully capture the details of the minority class distribution. Another approach by Sharma et al. (2021) suggests a two-step over-sampling technique called SMOTified-GAN that combines the strengths and overcomes the problems of two independent models that include SMOTE and GAN. In this approach, the authors first perform SMOTE and then use a GAN framework where the generator tries to learn the representation of the minority class in original data by sampling from the data generated by SMOTE. Although this approach produces impressive results and solves the problem of loss of local details, it still faces the problems of GAN mentioned before.

The work presented here extends the idea of SMOTified-GAN to address its problems by replacing GAN with Wasserstein Generative Adversarial Network – Gradient Penalty (WGAN-GP) and replacing SMOTE with other traditional over-sampling techniques such as Borderline-SMOTE and ADASYN. The benchmark datasets used for this study represent different class imbalance ratios (IR) to evaluate the effectiveness of this novel approach in real-world scenarios.

Proposed methods

Synthetic Minority Oversampling Technique (SMOTE) is a well-known oversampling technique for imbalanced data classification problems at the data pre-processing stage (Chawla, 2002). SMOTE effectively causes the region of the minority class to get wider by creating synthetic instances via linear interpolation within the minority class. In a nutshell, SMOTE discovers the k-nearest neighbours of a minority instance after choosing it at random. Then, on the line linking the instance and one random instance among its k neighbours in the feature space, a synthetic instance is produced at a randomly chosen position on the line.

SMOTE's operation is illustrated in Figure 1, where only a portion of the simulated cases are plotted. In this diagram, the instance X_i is one of the chosen instances from the minority class, and lines are used to connect it to its five closest neighbours. The synthetic instance X_{synth} is constructed along the line connecting X_i and $X_{neighbor}$ after a neighbour xneighbor has been randomly chosen (Zheng, 2020).

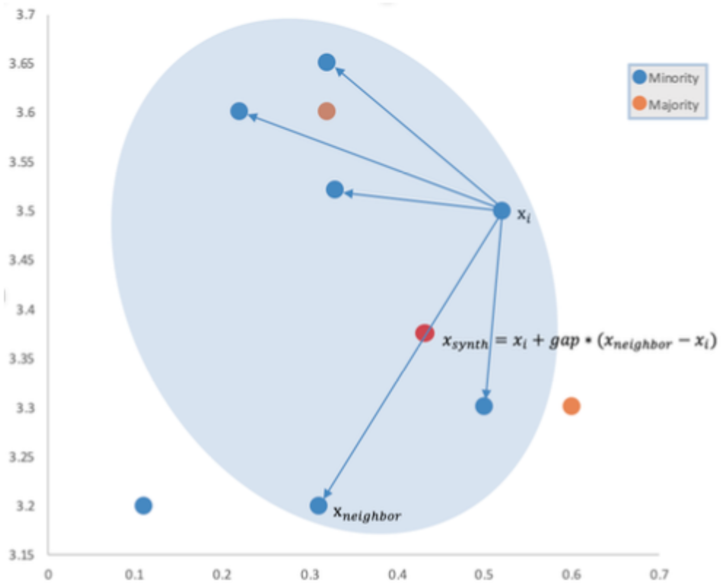


Figure 1 *Generating a synthetic instance for the minority class with SMOTE (Zheng, 2020).*

Although SMOTE performs remarkably well in many scenarios, it still faces the over-generalization problem (Wang and Japkowicz, 2004). Additionally, it generates synthetic samples regardless of the majority class, which causes the overlap between the classes to grow significantly (Prati, Batista and Monard, 2004). This problem causes samples from the majority class to be incorrectly classified as a member of the minority class and raises the false positive rate of learning algorithms.

In 2005, Han et al. proposed the Borderline-SMOTE1 and Borderline-SMOTE2 methods by oversampling the minority class borderline instances only. While borderline occurrences and those adjacent are more likely to be misclassified than those farther from the borderline, borderline-SMOTE algorithms try to learn the borderline of each class. All cases in the minority class of Borderline-SMOTE are split into three categories (see Figure 2): NOISE, DANGER, and SAFE (Cortesi and Chaki, 2012). The line between DANGER instances and their closest neighbours is then established when all instances of the minority class have been classified.

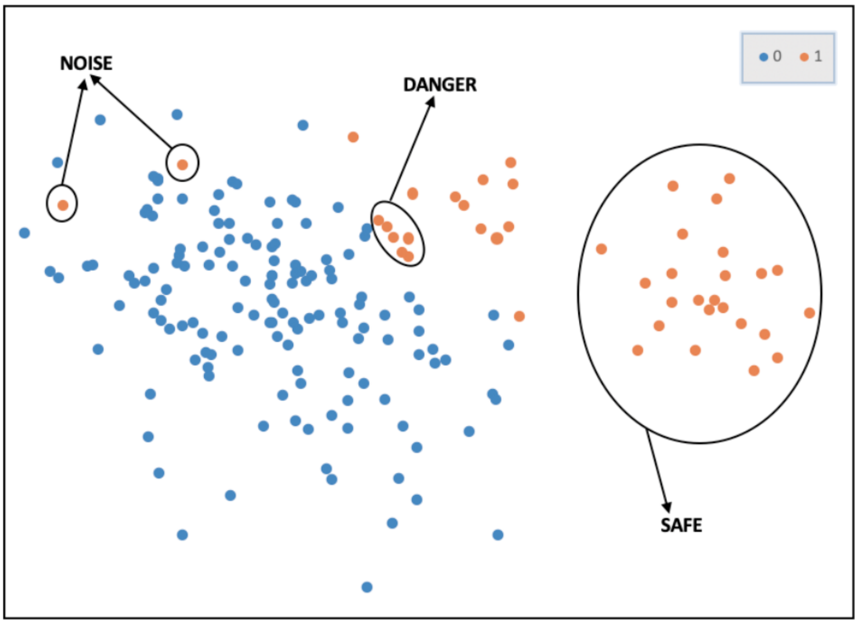


Figure 2 *Borderline-SMOTE categorising data samples (Zheng, 2020).*

Another over-sampling method was proposed by He et al. in 2008, addressing the problems of SMOTE by proposing the Adaptive Synthetic (ADASYN) sampling technique. The difficulty of learning certain minority class observations determines how many synthetic observations are generated by the ADASYN algorithm; more synthetic observations are generated for minority class observations that are significantly more difficult to learn (Brandt and Lanzén, 2020). However, all these techniques face challenges when dealing with high-dimensional data, and data complexity. They tend to generate samples that don't represent the over-sampled class distribution and lack diversity.

The Generative Adversarial Network (GAN) training process is a game between two adversarial networks: a generator network G maps a source of noise (in this case, SMOTE, Borderline-SMOTE or ADASYN) to the input space, and a discriminator network D determines whether a sample originated from the true data distribution or from the generator data distribution. The D and G play the two-player minimax game with:

$$min_G \, max_D V(D,G) = \mathbb{E}_{x\sim\mathbb{P}_r}[\log D(x)] + \mathbb{E}_{x\sim\mathbb{P}_g}[\log(1 - D(x))] \quad (1)$$

where P_r is the true data distribution and P_g is the generative data distribution implicitly defined by $x = G(z), z\sim p(z)$, for which z is sampled from a noise distribution p, such as uniform, normal, or a Gaussian distribution. The D is optimized to maximize the probability that the training samples and samples from

G are correctly classified, and G is optimized to minimize $\mathbb{E}_{x\sim\mathbb{P}_g}[\log(1 - D(x))]$ or $\mathbb{E}_{x\sim\mathbb{P}_g}[-\log D(x)]$ (Goodfellow et al., 2014).

While training, GAN exhibits some problematic behaviour; namely, model collapse where the generator G or the discriminator D starts to dominate, and unstable training (vanishing gradient), among other issues. Researchers have also noted that GAN would only be able to produce continuous data rather than discrete data when using the Jensen-Shannon (JS) divergence as a measure of generative samples (Arjovsky and Bottou, 2017):

$$JS\,(\mathbb{P}_r, \mathbb{P}_g) = KL\,(\mathbb{P}_r||\,\mathbb{P}_m) + KL\,(\mathbb{P}_g||\,\mathbb{P}_m) \quad (2)$$

where KL is the Kullback–Leibler divergence and P_m is the mixture $(P_r + P_g)/2$.

In contrast to GAN, the Wasserstein Generative Adversarial Network (WGAN) measure the distance between the true data distribution and the generative data distribution using the Earth-Mover (EM) distance, also known as Wasserstein-1, because of its superior smoothness to that of the JS divergence. Due to this, WGAN can theoretically solve the problem of vanishing gradients in GAN and previous experiments have indicated that WGAN can effectively address the problem of mode collapse by replacing the JS divergence with the EM distance (Arjovsky and Bottou, 2017; Arjovsky, Bottou and Chintala, 2017):

$$W(\mathbb{P}_r, \mathbb{P}_g) = inf_{\gamma\in\prod(\mathbb{P}_r,\mathbb{P}_g)}\mathbb{E}_{(x,y)\sim y}[||\,x - y\,||] \quad (3)$$

where $\prod(\mathbb{P}_r, \mathbb{P}_g)$ denotes the entire set of feasible joint distributions $\gamma(x, y)$ of the true data distribution P_r and generative data distribution P_g. Moreover, under modest assumptions, $W(\mathbb{P}_r, \mathbb{P}_g)$ is continuous and differentiable almost everywhere. However, Eq. (3) is highly complex to solve; thus, the EM distance could be reconstructed using the Kantorovich-Rubinstein duality (Villani, 2008):

$$W(\mathbb{P}_r, \mathbb{P}_g) = \frac{1}{k}\,sup_{||f||_L\leq K}\,\mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] + \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)] \quad (4)$$

where the supremum is over all the K-Lipschitz functions $f : x \rightarrow R$, and K denotes K-Lipschitz for constant K, where $K = 1$ for the original WGAN (Zheng et al., 2020). The objective function between the generator and the discriminator (critic) is the minimax Eq. (5):

$$min_G\,max_D V(D, G) = \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \mathbb{E}_{x\sim\mathbb{P}_g}[D(x)] \quad (5)$$

However, the WGAN still tends to generate poor samples or fails to converge. According to Gulrajani et al. (2017), the weight clipping in WGAN is a very poor approach to put a Lipschitz constraint on the discriminator and is the primary cause of these issues. Thus, they developed the Wasserstein Generative Adversarial Network – Gradient Penalty (WGAN-GP), an extension of WGAN, as an alternative to clipping weight by penalising the norm of the discriminator's gradient with respect to its input. According to their experimental findings, WGAN-GP performs better than normal WGAN and enables stable training of a wide range of GAN designs with essentially minimal hyperparameter adjustment. Formally, the new objective function of WGAN-GP is the minimax Eq. (6):

$$min_G\ max_D V(D,G) = \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(||\nabla_{\hat{x}}D(\hat{x})||_2 - 1)^2] \quad (6)$$

where λ is the gradient penalty coefficient and $\hat{x}$ is the interpolated samples between the true data distribution P_r and the generative data distribution P_g:

$$\hat{x} = \varepsilon x + (1-\varepsilon)\tilde{x}, \varepsilon \sim Uniform[0,1], x \sim \mathbb{P}_r, \tilde{x} \sim \mathbb{P}_g \quad (7)$$

Experimental methodology

The primary goal of this study is to evaluate the effectiveness of the two-step working of traditional over-sampling techniques (SMOTE, Borderline-SMOTE and ADASYN) and the Wasserstein Generative Adversarial Network – Gradient Penalty (WGAN-GP) generator's performance as an oversampling method for resolving the binary classification problem of imbalanced class data. Moreover, rather than sampling from random noise, the generator is designed to sample from the output generated by SMOTE, Borderline-SMOTE or ADASYN. First, the performance of the WGAN-GP generator is evaluated and compared with that of the traditional over-sampling techniques: SMOTE, Borderline-SMOTE, ADASYN, without sampling, and Generative Adversarial Network (GAN). Then, the performance of a combined strategy using traditional over-sampling techniques, GAN and WGAN-GP is evaluated and compared with that of the aforementioned step. The experiment[1] is implemented using a random repeated sampling strategy with 3 repetitions defined by 3 random states – 0, 22 and 42. Then, the results of these 3 random samples are averaged and reported (see Figure 3).

[1] Experiment virtual machine (VM) specifications: 20 vCPUs; 200 GiB RAM; 1x NVIDIA A10 GPU with 24GB VRAM; 1 TiB storage.

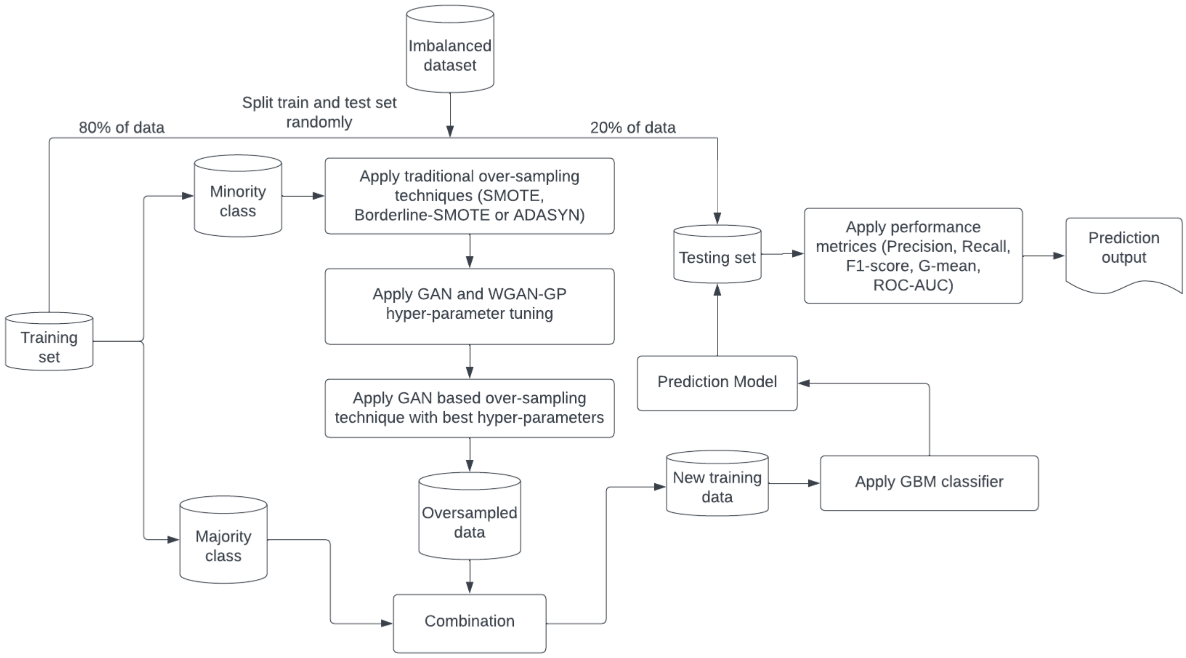


Figure 3 *Flowchart showing the experimental setup of 1 random sampling.*

In this study, we experimented with 5 different benchmark datasets from the UCI Machine Learning Repository (Dheeru and Karra, 2019) and Kaggle (www.kaggle.com, 2018; Feedzai, 2023). Moreover, the proposed approach was experimentally verified using a real imbalanced dataset from the National Aeronautics and Space Administration (NASA). The dataset consists of public software defects and is available on GitHub (Tantithamthavorn, 2023).

To create datasets with binary output class labels with various Imbalance Ratios (IRs), multiclass datasets from the UCI dataset were combined with one or more classes that were designated as the majority class. The IR is defined as the ratio between the number of instances in the majority class and the number of instances in the minority class. Other datasets obtained from Kaggle contained categorical input features, that were dropped from the dataset because the experiment emphasises evaluating the performance of different over-sampling algorithms and these algorithms support only numerical and continuous data. The real data obtained from NASA contains 5418 instances of software defects, of which 4583 instances are non-defective instances and the remaining 835 are defective instances. In this case, the IR is 5.49, which indicates a typical imbalanced dataset. Table 1 summarises the 6 datasets used in this experiment, including the total number of instances, number of features, number of instances in the majority class, number of instances in the minority class and the IR. The IR values of the imbalanced datasets range from 1.78 to 577.88, ensuring that the experiment is not biased towards a certain degree of imbalance.

Table 1 *Description of the imbalanced datasets used in the experiment.*

ID	Dataset	#Instances	#Features	#Majority instances	#Minority instances	IR
1	Credit card	284807	30	284315	492	577.88
2	Page blocks	5473	11	4913	560	8.77
3	Yeast	514	9	463	51	9.07
4	Ionosphere	351	33	225	126	1.78
5	Bank account fraud	1000000	27	988971	11029	89.67
6	Software defect	5418	36	4583	835	5.49

The performance of a combined strategy using WGAN-GP and traditional over-sampling techniques was compared with those of traditional over-sampling techniques and generative approaches based on GAN; thus, the parameter settings focused on two aspects. The parameters settings of traditional over-sampling techniques were set to default values defined in imbalanced-learn.org (2023a), imbalanced-learn.org (2023b) and imbalanced-learn.org (2023c) and are listed in Table 2.

Table 2 Parameter settings of traditional over-sampling approaches.

Over-sampling approaches	k_neighbors	m_neighbors	n_neighbors
SMOTE	5	None	None
Borderline-SMOTE	5	10	None
ADASYN	None	None	5

In the case of generative approaches based on GAN, most hyperparameters were tuned to converge generator G loss and discriminator D loss and minimise the overall loss to generate reasonable samples. The generators of GAN and WGAN-GP were all 3–15 dense layers of neural networks (see Figure 4) where the noise is either the samples generated from the traditional over-sampling techniques or a normal distribution with the dimensions ranging from 9 to 36. Dropout and batch normalization were applied to only the generator. According to Foster (2019), the discriminator of WGAN-GP (also known as a critic) should not employ Batch Normalization, because batch normalization increases the correlation between samples in the same batch, which reduces the efficiency of the gradient penalty loss. Foster (2019) further states that many experiments have shown that WGAN-GP can still produce excellent samples without batch normalization in the discriminator. The hidden layers of the generator expand by a factor of 2 and then condense back to the input dimension space. Similarly, the hidden layers of the discriminator (see Figure 5) condense by a factor of 2 and produce a single output determining if the input sample is fake or real. Leaky Rectified Linear Unit (Leaky-ReLU) is used as an activation function for the hidden layers and the input layer for both the generator and the discriminator. In the case of GAN, the output layer of the discriminator used a sigmoid activation function, whereas a linear activation function was applied to the output layer in WGAN-GP. GAN and WGAN-GP were hyper-parameter tuned with batch size ranging from 32 to 512, epochs were set to 1000 with early stopping enabled, and the learning rate of Adam optimizer (Kingma and Ba, 2014) ranging from 0.00001 to 0.1. The loss functions of GAN and WGAN-GP were consistent with those mentioned by Foster (2019). In the case of GAN, one generator weights update was followed by one discriminator weights update. Whereas, in WGAN-GP, the discriminator weights were updated 3 - 7 times (critic steps) before one weight update of the generator. The gradient penalty weight of WGAN-GP determines the weight associated with the gradient penalty and helps in controlling the training of the discriminator. Gradient penalty weight was set to range from 2 to 20 with a step of 0.5 (see Table 3). The generative approaches based on GAN were implemented in Python using Google’s open-source framework TensorFlow (Abadi et al., 2016) with GPU enabled. The hyper-parameter tuning was implemented using the Python Optuna framework (Optuna, 2023) and the settings are mentioned in Table 4.

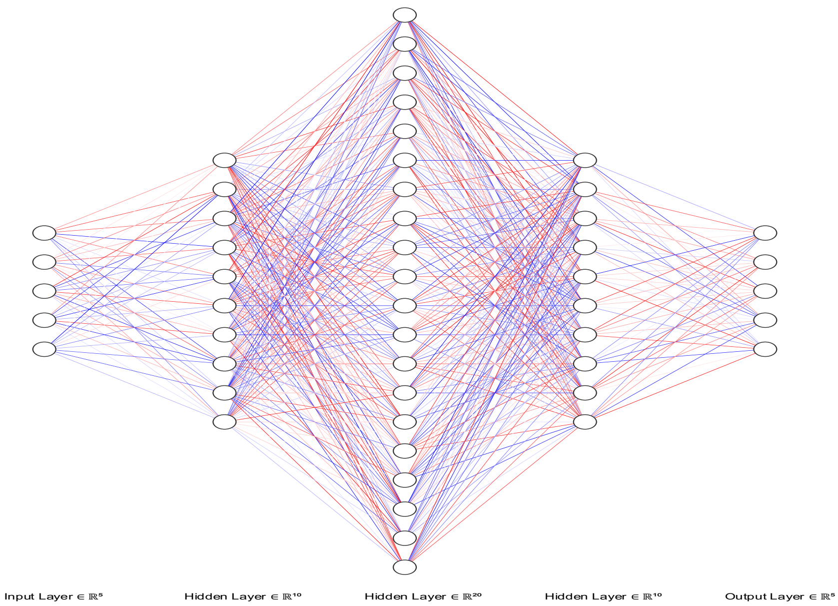


Figure 4 *Example architecture of GAN and WGAN-GP generator.*

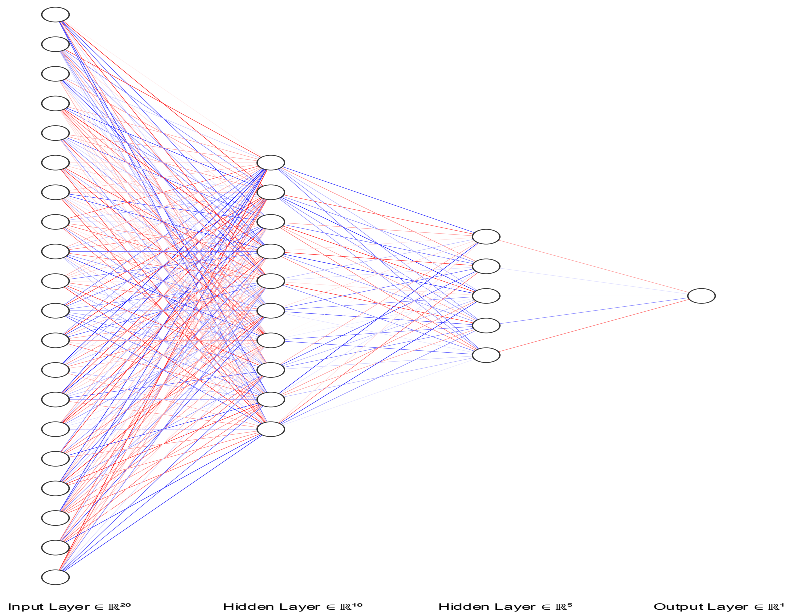


Figure 5 *Example architecture of GAN and WGAN-GP discriminator.*

Table 3 *GAN and WGAN-GP hyper-parameters and their search range.*

Parameter	GAN	WGAN-GP
Number of dense layers	3 - 15	3 - 15
Adam Optimizer learning rate	0.00001 - 0.1	0.00001 - 0.1
WGAN-GP critic steps	None	3 - 7

WGAN-GP gradient penalty weight	None	2 - 20 with a step = 0.5

Table 4 *Optuna hyper-parameter tuning framework settings.*

Parameter	Value
objective	minimize (minimize the total loss)
sampler	TPESampler (optuna.readthedocs.io, 2023a)
pruner	HyperbandPruner (optuna.readthedocs.io, 2023b)
n_trials	20

A Gradient Boosting Machine (GBM) was used as a Binary classification algorithm to evaluate the performance of the above over-sampling techniques. The Gradient Boosting Machine (GBM) classifier was implemented based on the Python library XGBoost, with default settings selected (xgboost.readthedocs.io, 2023). Additionally, early stopping rounds and the number of parallel threads for model training were set to 50 and 2, respectively.

Table 5 *Parameter settings of GBM classifier.*

Parameter	value
objective	binary:logistic
early_stopping_rounds	50
n_jobs	2
random_state	42

The performance of the eleven oversampling approaches and the approach without sampling on each dataset was empirically assessed using the GBM classification algorithms stated above. During pre-processing, each dataset was randomly split into an 80% training set and 20% testing set and only the training set was subjected to over-sampling to ensure that there was no information leak while evaluating the binary classifiers mentioned above. The training set was further split into an 80% sub-train set and a 20% validation set to ensure that the model results were generalised.

We evaluated the impact of each oversampling strategy on the datasets using the F-measure, G-mean, receiver operating characteristic (ROC) area under the curve (AUC), Precision (P) and Recall (R) performance metrics; these measures have all been applied in research on class imbalance with a similar focus (Sharma et al., 2021; Zheng et al., 2020; Barua et al., 2014). The performance evaluation metrics are defined as follows.

$$Precision = \frac{TP}{(TP+FP)} \tag{8}$$

$$Recall\ (Sensitivity) = \frac{TP}{(TP+FN)} \tag{9}$$

where TP (True Positive) represents a case where the real category is positive, and the predicted value is also positive; FP (False Positive) represents a case where the real category is negative, but the predicted value is positive; FN (False Negative) represents a case where the real category is positive, but the predicted value is negative (Zheng et al., 2020).

$$F - measure = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times (Precision + Recall)} \tag{10}$$

where β is usually equal to 1, as is the case in this study.

$$Specificity = \frac{TN}{(TN+FP)} \tag{11}$$

$$G - mean = \sqrt{Recall \times Specificity} \tag{12}$$

where TN (True Negative) represents a case where the real category is negative, and the predicted value is also negative.

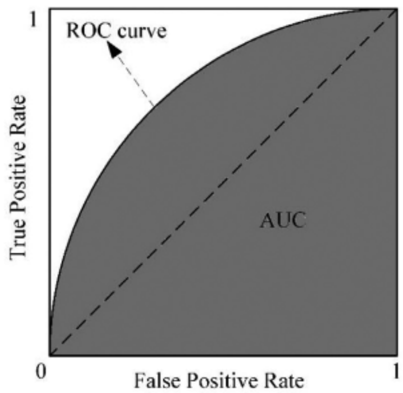
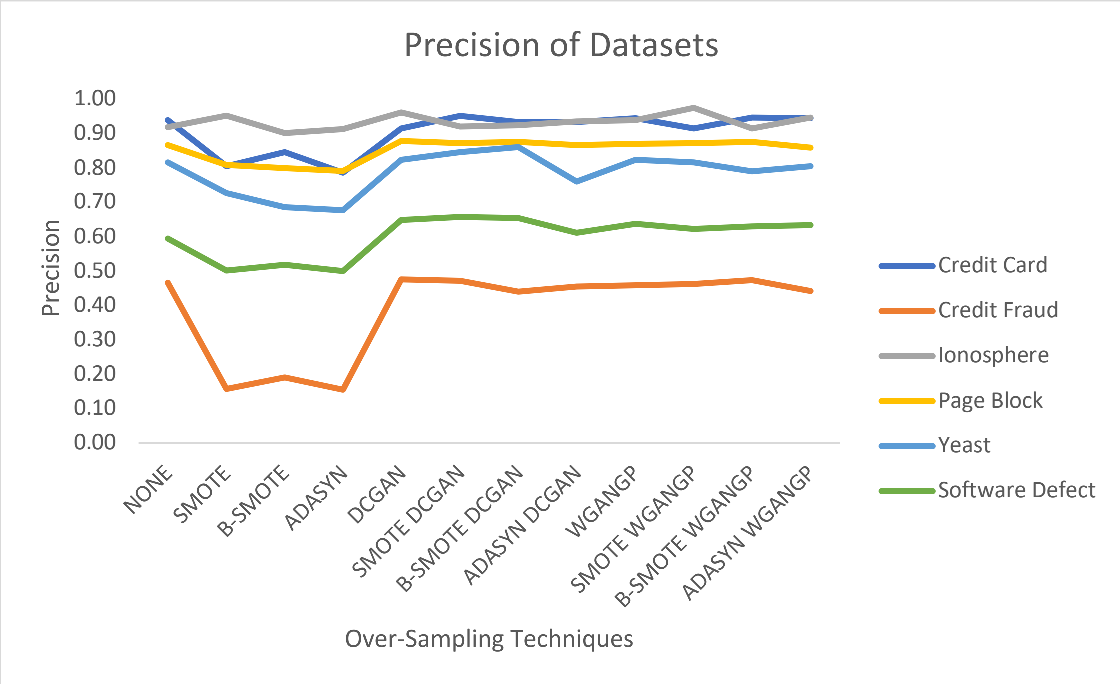


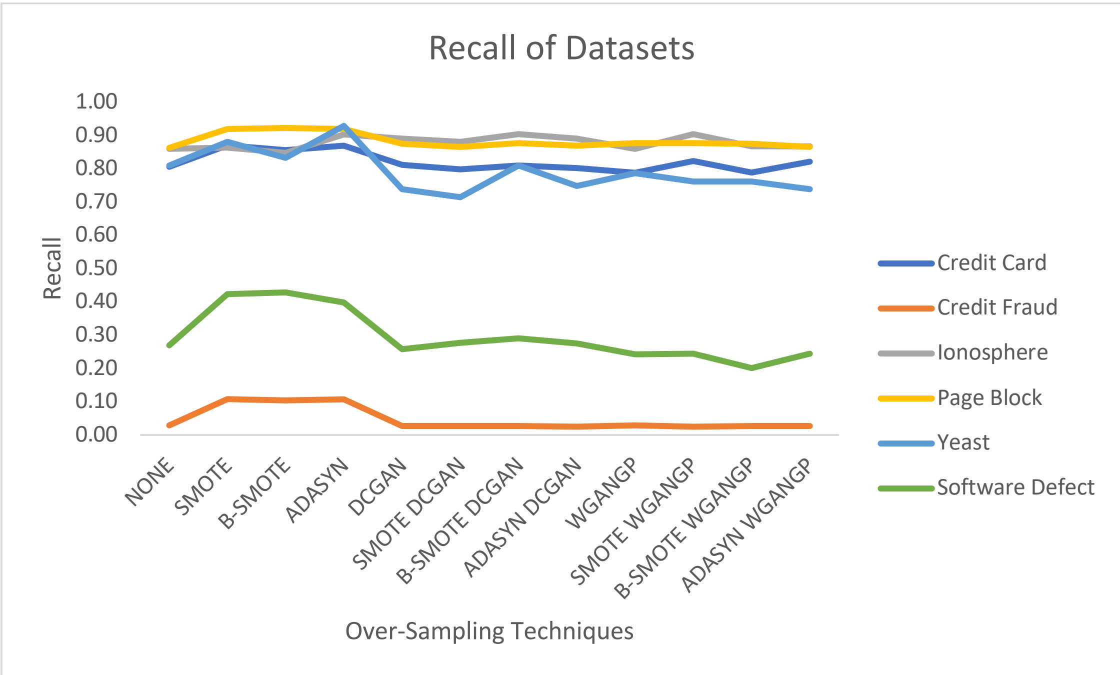
Figure 6 *ROC – AUC curve diagram (Zheng et al., 2020).*

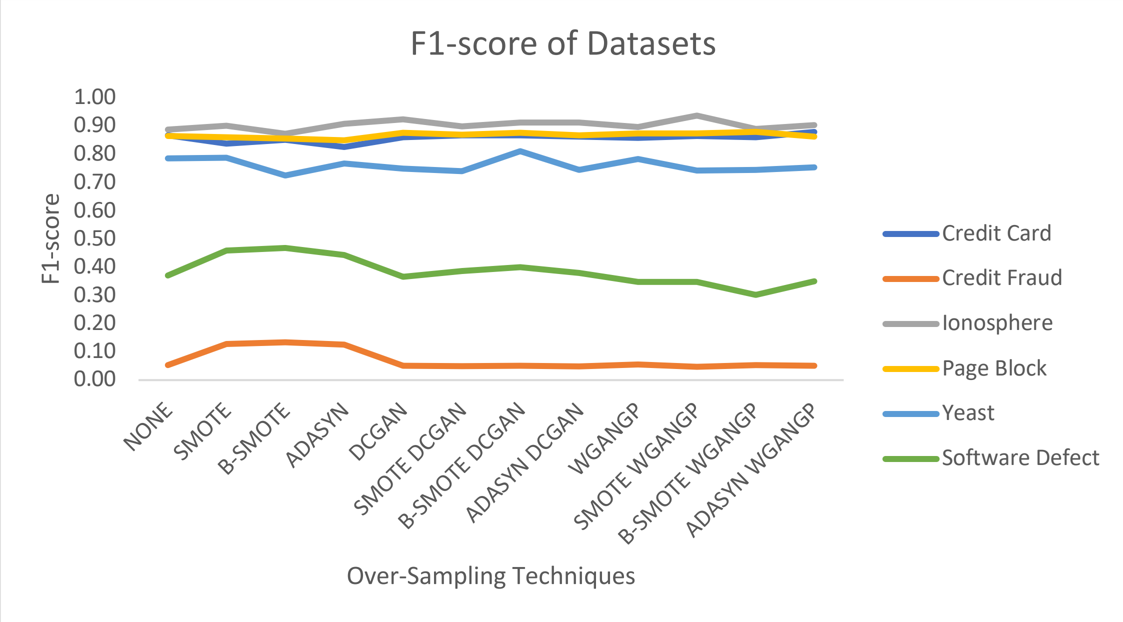
A larger AUC represents a more effective classifier. Figure 6 illustrates the AUC score on a two-dimensional diagram, where the grey area is the AUC value. The False Positive Rate (FPR), or the percentage of incorrectly categorised negative cases in relation to all negative cases, is shown on the x-axis of Figure 6. The True Positive Rate (TPR), or the proportion of accurately predicted positive instances to all positive cases, is shown on the y-axis. The AUC is a helpful performance evaluation parameter of the classifier since it is independent of the decision criterion, in contrast to the ROC curve, which is reliant on the classification threshold (Zheng et al., 2020; Douzas and Bacao, 2017).

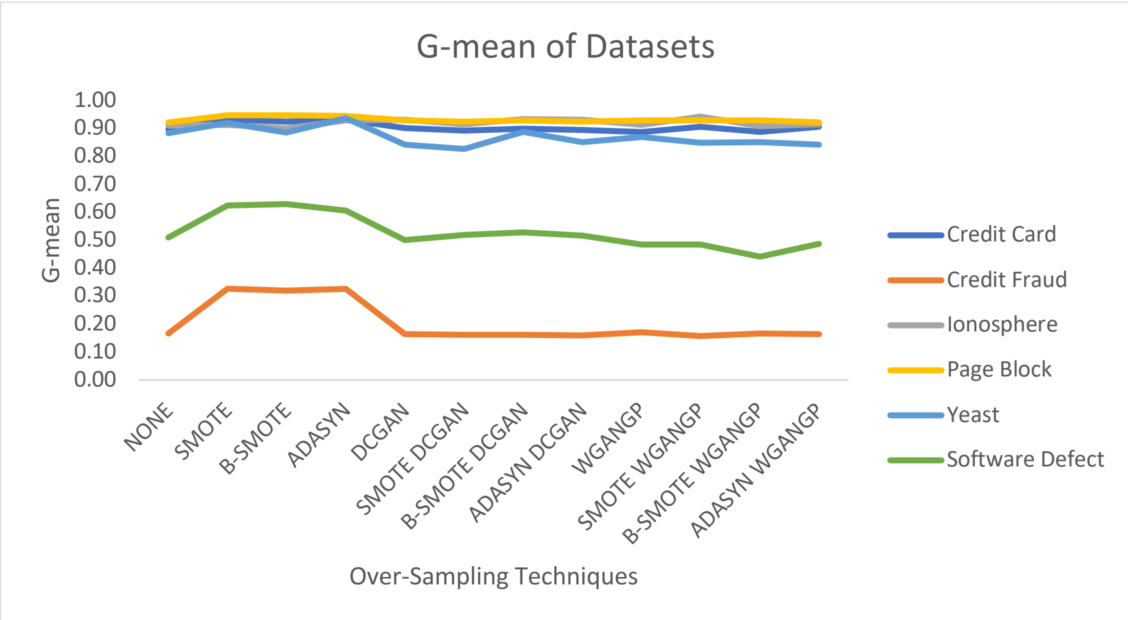
Results

In Figure 7, each plot corresponds to an evaluation metric for the Gradient Boosting Machine (GBM) classifier. The x- and y-axes of each plot represent the different over-sampling techniques and the score of the evaluation metric, respectively. Therefore, each plot includes six lines, corresponding to six datasets mentioned in Table 1.






F1-score of Datasets
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
F1-score
NONE
SMOTE
B-SMOTE
ADASYN
DCGAN
SMOTE DCGAN
B-SMOTE DCGAN
ADASYN DCGAN
WGANGP
SMOTE WGANGP
B-SMOTE WGANGP
ADASYN WGANGP
Over-Sampling Techniques
Credit Card
Credit Fraud
Ionosphere
Page Block
Yeast
Software Defect


G-mean of Datasets
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
G-mean
NONE
SMOTE
B-SMOTE
ADASYN
DCGAN
SMOTE DCGAN
B-SMOTE DCGAN
ADASYN DCGAN
WGANGP
SMOTE WGANGP
B-SMOTE WGANGP
ADASYN WGANGP
Over-Sampling Techniques
Credit Card
Credit Fraud
Ionosphere
Page Block
Yeast
Software Defect

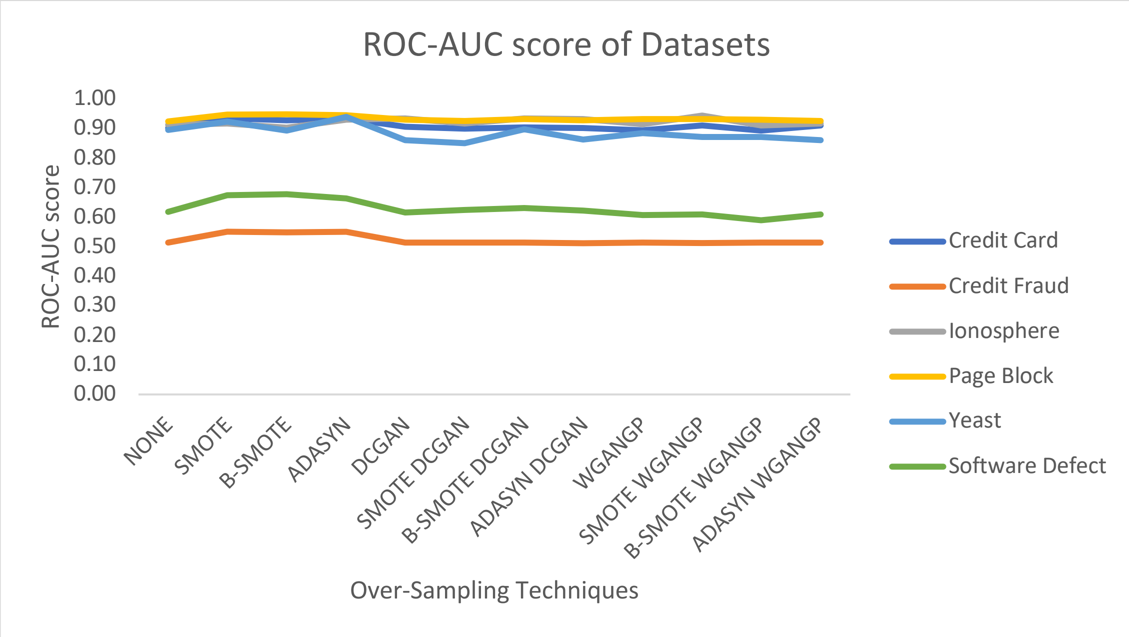


Figure 7 *Performance results of each over-sampling technique using 6 distinct imbalanced datasets and the approach without over-sampling.*

Figure 7 shows the following: (1) the GAN and CWGAN-GP-based over-sampling approaches, under precision as the evaluation metric, outperformed the traditional over-sampling approaches; (2) the improved performance in precision comes at a cost of the recall, the GAN and CWGAN-GP based over-sampling approaches fall behind in improving the recall of the GBM classifier; (3) Under f1-score as the evaluation metric, the GAN and CWGAN-GP based over-sampling approaches outperforms the traditional over-sampling approaches only in 3 out of 6 datasets, making the result inconclusive; (4) the GAN and CWGAN-GP based over-sampling approaches performed similarly under G-mean and ROC-AUC score as an evaluation metric, either it didn't show any improvement or it performed poorly as compared to traditional over-sampling techniques.

The results in Figure 7 demonstrate that the imbalance ratio (IR) for five evaluation metrics for the performance of the oversampling techniques did not behave consistently. The best and worst performance scores of all oversampling procedures for the GBM classifier are 0.97 and 0.02, respectively, and the IR changes significantly for the imbalanced datasets. It is therefore impossible to draw the conclusion that a smaller or larger IR causes better or poorer performance. According to Figure 7, the data distribution may have an impact on how well oversampling approaches perform (i.e., the higher the complexity of data distribution, the less the oversampling approach can learn about the characteristics of the data distribution). As a result, the generated data will be less realistic, which will worsen performance. In contrast, the more characteristics the oversampling algorithms can learn from the data distribution, the simpler the data distribution. As a result, the performance will be better, and the generated data will be more realistic (Zheng et al., 2020).

Table 6 *Mean score of eleven over-sampling techniques and without sampling (NONE) under five evaluation metrics (Precision, Recall, F1-score, G-mean and ROC-AUC score). Bold values indicate the best score across all over-sampling techniques for each row.*

Datasets	NONE	SMOTE	B-SMOTE	ADASYN	DCGAN	SMOTE DCGAN	B-SMOTE DCGAN	ADASYN DCGAN	WGANGP	SMOTE WGANGP	B-SMOTE WGANGP	ADASYN WGANGP
Credit Card	0.9403	0.8063	0.8472	0.7866	0.9158	**0.9519**	0.9333	0.9344	0.9447	0.9150	0.9470	0.9457
	0.8052	**0.8701**	0.8557	0.8698	0.8118	0.7986	0.8093	0.8022	0.7878	0.8222	0.7881	0.8220
	0.8675	0.8369	0.8514	0.8261	0.8607	0.8684	0.8668	0.8630	0.8589	0.8651	0.8602	**0.8795**
	0.8973	**0.9326**	0.9249	0.9324	0.9010	0.8936	0.8994	0.8955	0.8875	0.9065	0.8877	0.9066
	0.9026	**0.9349**	0.9277	0.9347	0.9059	0.8993	0.9046	0.9010	0.8938	0.9110	0.8940	0.9110
Bank Account Fraud	0.4669	0.1577	0.1909	0.1548	**0.4760**	0.4713	0.4411	0.4551	0.4597	0.4627	0.4738	0.4419
	0.0280	0.1071	0.1033	**0.1072**	0.0267	0.0261	0.0267	0.0256	0.0293	0.0249	0.0277	0.0273
	0.0529	0.1275	**0.1340**	0.1266	0.0505	0.0495	0.0503	0.0485	0.0550	0.0472	0.0524	0.0514
	0.1673	0.3261	0.3205	**0.3264**	0.1633	0.1614	0.1629	0.1599	0.1709	0.1577	0.1663	0.1651
	0.5138	0.5503	0.5492	**0.5504**	0.5132	0.5129	0.5132	0.5126	0.5144	0.5123	0.5137	0.5135
Ionosphere	0.9199	0.9519	0.9019	0.9139	0.9615	0.9217	0.9246	0.9352	0.9396	**0.9753**	0.9147	0.9467
	0.8594	0.8628	0.8475	**0.9029**	0.8902	0.8798	0.9029	0.8909	0.8594	0.9029	0.8664	0.8664
	0.8886	0.9020	0.8730	0.9083	0.9242	0.8995	0.9136	0.9123	0.8974	**0.9374**	0.8894	0.9044
	0.9102	0.9128	0.9005	0.9289	0.9322	0.9165	0.9329	0.9306	0.9127	**0.9427**	0.9088	0.9158
	0.9119	0.9157	0.9027	0.9294	0.9334	0.9174	0.9335	0.9316	0.9148	0.9437	0.9105	0.9177
Page-Block	0.8675	0.8099	0.8000	0.7916	0.8789	0.8729	**0.8769**	0.8678	0.8714	0.8719	0.8761	0.8599
	0.8627	0.9192	**0.9224**	0.9192	0.8750	0.8661	0.8772	0.8690	0.8769	0.8776	0.8741	0.8651
	0.8645	0.8602	0.8551	0.8498	0.8765	0.8689	0.8766	0.8680	0.8738	0.8746	**0.8798**	0.8623
	0.9216	0.9469	**0.9475**	0.9454	0.9288	0.9238	0.9299	0.9251	0.9294	0.9299	0.9283	0.9226
	0.9239	0.9474	**0.9480**	0.9459	0.9308	0.9259	0.9317	0.9270	0.9311	0.9315	0.9301	0.9246
Yeast	0.8175	0.7265	0.6855	0.6773	0.8237	0.8472	**0.8611**	0.7616	0.8237	0.8175	0.7900	0.8056
	0.8095	0.8809	0.8333	**0.9286**	0.7381	0.7143	0.8095	0.7489	0.7857	0.7619	0.7619	0.7381

	0.7858	**0.7890**	0.7251	0.7668	0.7500	0.7411	0.8113	0.7450	0.7844	0.7433	0.7460	0.7532
	0.8836	0.9202	0.8866	**0.9371**	0.8427	0.8270	0.8872	0.8511	0.8694	0.8498	0.8504	0.8432
	0.8941	0.9225	0.8934	**0.9393**	0.8602	0.8501	0.8977	0.8619	0.8839	0.8703	0.8702	0.8602
Software Defect	0.5944	0.5019	0.5182	0.5002	**0.6482**	0.6578	0.6548	0.6120	0.6380	0.6235	0.6303	0.6334
	0.2690	0.4240	**0.4283**	0.3980	0.2581	0.2777	0.2897	0.2755	0.2423	0.2445	0.2010	0.2434
	0.3702	0.4592	**0.4683**	0.4431	0.3671	0.3876	0.3997	0.3787	0.3488	0.3487	0.3020	0.3514
	0.5097	0.6253	**0.6297**	0.6074	0.5007	0.5185	0.5294	0.5160	0.4845	0.4859	0.4417	0.4866
	0.6176	0.6732	**0.6776**	0.6624	0.6158	0.6252	0.6305	0.6214	0.6080	0.6086	0.5894	0.6088

Discussion

An improvement in the quality of classification has been observed in 3 out of 6 datasets (Credit Card, Ionosphere and Page Block datasets) with the introduction of the proposed traditional over-sampling (SMOTE, Borderline-SMOTE and ADASYN) assisted WGAN-GP. WGAN-GP-based over-sampling has shown no major improvement in recall. They show better precision than the traditional over-sampling techniques. Overall, in most datasets, traditional over-sampling-assisted WGAN-GP represents a balanced performance as compared to traditional over-sampling techniques.

Considering the impact of the imbalance ratio (IR), the performance of the proposed over-sampling technique has shown mixed results. For example, datasets such as Credit Card, Page Block and Ionosphere have an IR of 577.88, 8.77, and 1.78, respectively, and have shown good results. Whereas datasets such as Bank Account Fraud, Software defects and Yeast have an IR of 89.67, 5.49, and 9.07, respectively, have shown poor results. Although only a small improvement or no improvement is observed between GAN-based over-sampling and WGAN-GP-based oversampling, WGAN-GP is more stable during training and rarely experiences mode collapse. Because WGAN-GP uses Earth Mover (EM) distance as opposed to Jensen-Shannon (JS) divergence and Kullback–Leibler (KL) divergence to measure the distance between real and generated distributions.

The limitation of our approach is that WGAN-GP has a training process that requires the hyper-parameters to be tuned before generating new samples. This poses a significant challenge in finding the optimal hyper-parameters for a specific problem and requires more time and effort as compared to traditional over-sampling techniques. However, based on the objectives of a specific problem, it may be worthwhile to exchange limited time and effort for higher-quality synthetic data as it might substantially improve the performance of classification algorithms.

The proposed over-sampling technique in this paper faced many challenges. WGAN-GP can effectively handle training instability than GAN-based oversampling techniques to some extent. However, this issue is still worth researching, as the stability of data creation must be examined using various designs under various network architectures. Theoretical study into WGAN-GP is necessary to speed up its training and convergence procedures, as the WGAN-GP training process uses more time and resources than traditional over-sampling techniques. Although all oversampling techniques can produce the necessary data, the created data's quality and diversity are not the same as those of actual data. Researchers must create new strategies and put forth more practical techniques to significantly improve the quality and diversity of data generated in many application domains (Zheng et al., 2020).

Conclusion

Imbalanced data are a common occurrence in both theoretical research and practical implementations. The application of conventional classification techniques relies fairly on even distributions of output class labels of a dataset. These algorithms could, however, contain varying degrees of flaws, making them ineffective. Resampling strategies, the majority of which include oversampling methods, have been presented to address the issue brought on by imbalanced data classification. The foundation of oversampling is the idea of producing minority data to balance an unbalanced dataset. Even though it has been demonstrated that these oversampling methods improve performance, the generated data is not sufficiently realistic because traditional oversampling methods concentrate on interpolating information from the imbalanced dataset. Furthermore, mode collapse and unstable training are constant issues with over-sampling methods based on Generative Adversarial Network (GAN). We presented traditional over-sampling methods (SMOTE, Borderline-SMOTE and ADASYN) assisted Wasserstein Generative Adversarial Network – Gradient Penalty (WGAN-GP), a novel over-sampling method built on WGAN-GP and inspired by SMOTified-GAN proposed by Sharma et al. (2021), to solve these issues. By studying the distributional properties of the existing samples of the minority class, our method generates new samples.

The proposed method was empirically evaluated by comparing it with eight other over-sampling approaches (SMOTE, Borderline-SMOTE, ADASYN, DCGAN, SMOTE DCGAN, Borderline-SMOTE DCGAN, ADASYN DCGAN, and WGAN-GP) as well as an approach without sampling. We used the Gradient Boosting Machine (GBM) algorithm on 6 datasets with diverse imbalance ratios (IR) to avoid bias toward a specific IR. According to the experimental findings, WGAN-GP performed better than traditional over-sampling techniques across a few evaluation metrics, only in 3 out of 6 datasets.

A possible explanation for such behaviour could be that the proposed method is highly dependent on its hyper-parameters and there is a possibility that the optimal hyper-parameters are stuck in local minima. Moreover, WGAN-GP generates better samples with more stable training than GAN, since WGAN-GP uses the Earth Mover (EM) distance, whereas GAN uses the Jensen-Shannon (JS) divergence, which causes mode collapse and unstable training. In future studies, we intend to examine the performance of Conditional WGAN-GP by replacing it with WGAN-GP proposed in this study. We also aim to compare the performance of GAN-based over-sampling techniques to that of complex over-sampling approaches.

References

1. A. Bria, C. Marrocco, and F. Tortorella (2020), *"Addressing class imbalance in deep learning for small lesion detection on medical images,"* Computers in Biology and Medicine, vol. 120, p. 103735.
2. A. Fernández, V. López, M. Galar, M. D. Jesus and F. Herrera (2013), *"Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches"*, Knowl. -Based Syst., vol. 42, pp. 97-110.
3. Agostino Cortesi and Nabendu Chaki. Computer information systems and industrial management. 2012.
4. Anuraganand Sharma, Prabhat Kumar Singh and Rohitash Chandra (2021). *"SMOTified-GAN for class imbalanced pattern classification problems."* arXiv. Available: https://doi.org/10.1109/ACCESS.2022.3158977
5. B. X. Wang and N. Japkowicz (2004), "Imbalanced data set learning with synthetic samples", Proc. IRIS Mach. Learn. Workshop.
6. Brandt, J. and Lanzén, E. (2020). A Comparative Review of SMOTE and ADASYN in Imbalanced Data Classification. [online] Available at: https://www.diva-portal.org/smash/get/diva2:1519153/FULLTEXT01.pdf.
7. C. Villani (2008), Optimal Transport: Old and New, Springer Science & Business Media.
8. D. Dheeru and E. Karra Taniskidou (2019), *UCI Machine Learning Repository.* University of California, Irvine, School of Information and Computer Sciences. [Online]. Available: http://archive.ics.uci.edu/ml

9. D. Rodriguez, I. Herraiz, R. Harrison, J. Dolado, and J. C. Riquelme (2014), *"Preliminary comparison of techniques for dealing with imbalance in software defect prediction,"* in Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, ser. EASE '14. New York, NY, USA: Association for Computing Machinery.
10. D.P. Kingma, J. Ba (2014), *Adam: a method for stochastic optimization*, arXiv:1412.6980.
11. Feedzai (2023). Bank Account Fraud (BAF) Tabular Dataset Suite. [online] Available at: https://github.com/feedzai/bank-account-fraud/tree/main [Accessed 29 Jul. 2023].
12. Foster, D. (2019). *Generative deep learning: teaching machines to paint, write, compose, and play.* Beijing; Boston; Farnham; Sebastopol; Tokyo O'reilly.
13. G. Douzas and F. Bacao (2018), "Effective data generation for imbalanced learning using conditional generative adversarial networks," Expert Systems with Applications, vol. 91, pp. 464–471. [Online]. Available at: https://www.sciencedirect.com/science/article/pii/S0957417417306346.
14. G. Douzas, F. Bacao (2017), *Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning*, Expert Syst. Appl. 82, pp. 40–52.
15. G. Kovács (2019), *"Smote-variants: A python implementation of 85 minority oversampling techniques"*, Neurocomputing, vol. 366, pp. 352–354.
16. G. M. Weiss and F. Provost (2001), *"The Effect of Class Distribution on Classifier Learning: An Empirical Study"*.
17. Google Developers. (2023). Imbalanced Data. [online] Available at: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data.
18. H. He and E. A. Garcia (2009), *"Learning from imbalanced data"*, IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263-1284.
19. He, Haibo; Bai, Yang, Garcia, Edwardo A.; Li, Shutao. 2008. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. IEEE World Congress on Computational Intelligence: 2008 IEEE International Joint Conference on Neural Networks. Hong Kong, China, 1322-1328.

20. Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. pages 878–887, 2005.
21. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014), *"Generative Adversarial Nets,"* in Advances in Neural Information Processing Systems, pp. 2672-2680.
22. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio (2014), *Generative adversarial nets, in: Advances in Neural Information Processing Systems*, pp. 2672–2680.
23. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville (2017), *Improved training of wasserstein gans*, in: Advances in Neural Information Processing Systems, pp. 5767–5777.
24. imbalanced-learn.org. (2023a), *SMOTE — Version 0.9.0.* [online] Available at: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html [Accessed 29 Jul. 2023].
25. imbalanced-learn.org. (2023b), *BorderlineSMOTE — Version 0.10.1.* [online] Available at: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.BorderlineSMOTE.html [Accessed 29 Jul. 2023].
26. imbalanced-learn.org. (2023c). *ADASYN — Version 0.9.1.* [online] Available at: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.ADASYN.html [Accessed 29 Jul. 2023].
27. Kodali, N., Abernethy, J., Hays, J. and Kira, Z. (2017). ON CONVERGENCE AND STABILITY OF GANS. [online] Available at: https://arxiv.org/pdf/1705.07215.pdf.
28. L. Zhang, H. Yang, and Z. Jiang (2018), "Imbalanced biomedical data classifica- tion using self-adaptive multilayer ELM combined with dynamic GAN," BioMedical Engineering OnLine, vol. 17, no. 1, p. 181.
29. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard (2016), *TensorFlow: a system for large-scale machine learning*.
30. M. Arjovsky, L. Bottou (2017), *Towards principled methods for training generative adversarial networks*, in: arXiv e-prints.

31. M. Arjovsky, S. Chintala, L. Bottou (2017), *Wasserstein generative adversarial networks*, in: International Conference on Machine Learning, pp. 214–223.
32. M. Zheng, T. Li, R. Zhu, Y. Tang, M. Tang, L. Lin, and Z. Ma (2020), *"Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification"*, Information Sciences, vol. 512, pp. 1009–1023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519309715
33. M. Zheng, T. Li, R. Zhu, Y. Tang, M. Tang, L. Lin, and Z. Ma (2020), *"Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification"*, Information Sciences, vol. 512, pp. 1009–1023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025519309715
34. Meng, D. and Li, Y. (2022). An imbalanced learning method by combining SMOTE with Center Offset Factor. Applied Soft Computing, 120, p.108618.
35. N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer (2002), *"SMOTE: Synthetic minority over-sampling technique"*, J. Artif. Intell. Res., no. 16, pp. 341-378.
36. Optuna (2023). Optuna - A hyperparameter optimization framework. [online] Available at: https://optuna.org.
37. optuna.readthedocs.io. (2023a). optuna.samplers.TPESampler — Optuna 3.3.0 documentation. [online] Available at: https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html#optuna.samplers.TPESampler [Accessed 5 Sep. 2023].
38. optuna.readthedocs.io. (2023b). optuna.pruners.HyperbandPruner — Optuna 3.3.0 documentation. [online] Available at: https://optuna.readthedocs.io/en/stable/reference/generated/optuna.pruners.HyperbandPruner.html [Accessed 5 Sep. 2023].
39. P. Xu, R. Du, and Z. Zhang (2019), *"Predicting pipeline leakage in petrochemical system through GAN and LSTM,"* Knowledge-Based Systems, vol. 175, pp. 50–61.
40. R. C. Prati, G. E. A. P. A. Batista and M. C. Monard (2004), *"Class imbalances versus class overlapping: An analysis of a learning system behavior",* Proc. Adv. Artif. Intell., pp. 312-321.

41. S. Barua, M.M. Islam, X. Yao, K. Murase (2014), *MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning*, IEEE Trans. Knowl. Data Eng. 26, pp. 405–425.
42. S. Huang and K. Lei (2020), *"IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks,"* Ad Hoc Networks, vol. 105, p. 102177.
43. S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. Hacid, and H. Zeineddine (2019), *"An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection"*, IEEE Access, vol. 7, pp. 93010–93022.
44. T. Zhu, Y. Lin, and Y. Liu (2020), *"Improving interpolation-based oversampling for imbalanced data learning,"* Knowledge-Based Systems, vol. 187, p. 104826.
45. Tantithamthavorn, C. (Kla) (2023). NASADefectDataset. [online] GitHub. Available at: https://github.com/klainfo/NASADefectDataset [Accessed 28 Jul. 2023].
46. W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen (2013), *"Effective detection of sophisticated online banking fraud on extremely imbalanced data"*, World Wide Web, vol. 16, no. 4, pp. 449–475.
47. www.kaggle.com. (2018). Credit Card Fraud Detection. [online] Available at: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.
48. X. Tao, Q. Li, W. Guo, C. Ren, Q. He, R. Liu, and J. Zou (2020), *"Adaptive weighted over-sampling for imbalanced datasets based on density peaks clustering with heuristic filtering,"* Information Sciences, vol. 519, pp. 43–73.
49. xgboost.readthedocs.io. (2023). Python API Reference — xgboost 1.7.5 documentation. [online] Available at: https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBClassifier [Accessed 25 Aug. 2023].
50. Y. Zhuo and Z. Ge (2020), *"Gaussian Discriminative Analysis aided GAN for imbalanced big data augmentation and fault classification"*, Journal of Process Control, vol. 92, pp. 271–287.
51. Z. Qin, Z. Liu, P. Zhu, and Y. Xue (2020), *"A GAN-based image synthesis method for skin lesion classification,"* Computer Methods and Programs in Biomedicine, vol. 195, p. 105568.

52. Zheng, X. (2020). SMOTE Variants for Imbalanced Binary Classification: Heart Disease Prediction. [online] escholarship.org. Available at: https://escholarship.org/uc/item/99x0w9w0 [Accessed 28 Jul. 2023].