Synthetic Data Generation for Fraud Detection using GANs

*

Charitos Charitou
Department of Computer Science
City, University of London
London, UK
charitos.charitou@city.ac.uk

Simo Dragicevic
BetBuddy
Playtech Plc
London, UK
simo.dragicevic@playtech.com

Artur d'Avila Garcez
Department of Computer Science
City, University of London
London, UK
a.garcez@city.ac.uk

***Abstract*—Detecting money laundering in gambling is becoming increasingly challenging for the gambling industry as consumers migrate to online channels. Whilst increasingly stringent regulations have been applied over the years to prevent money laundering in gambling, despite this, online gambling is still a channel for criminals to spend proceeds from crime. Complementing online gambling's growth more concerns are raised to its effects compared with gambling in traditional, physical formats, as it might introduce higher levels of problem gambling or fraudulent behaviour due to its nature of immediate interaction with online gambling experience. However, in most cases the main issue when organisations try to tackle those areas is the absence of high quality data. Since fraud detection related issues face the significant problem of the class imbalance, in this paper we propose a novel system based on Generative Adversarial Networks (GANs) for generating synthetic data in order to train a supervised classifier. Our framework Synthetic Data Generation GAN (SDG-GAN), manages to outperformed density based oversampling methods and improve the classification performance of benchmarks datasets and the real world gambling fraud dataset.**

***Index Terms*—fraud, GANs, gambling, synthetic data, class imbalanced**

I. Introduction

The major motivations for this research are to understand the problems faced by the gambling industry with regard to raising standards in money laundering detection. The AML process is conceptually similar to fraud detection, an area that has been the focus of a great deal of research in recent years. It has been shown that applying machine learning techniques to detect fraud can solve the problem to a certain degree, with the best results achieved with supervised learning. However, the problem with supervised learning is that it requires labelled data for both non-fraudulent and fraudulent behaviours in order to train a model [1]. Collaborating with online gambling operator which supported this research by making anonymous data available with labels describing high-risk (fraudulent customers) and no-risk of money laundering (non-fraudulent customers). Notwithstanding, the non-fraudulent customers are much greater in number compared to customers with high money laundering risk.

Most supervised learning algorithms are not designed to cope with a large difference in the number of cases belonging to different classes [2]. This problem is known in the literature as class imbalance and is an issue regularly encountered by researchers. The problem corresponds to the issue faced by inductive learning systems when dealing with domains where one class is represented by a large number of samples while the other class is represented by fewer samples. In such cases, the reliability and validity of the results are questionable since prediction algorithms tend to have a bias towards the majority class. Such an imbalanced dataset could lead to unintended model performance – for example, classifying all the cases as normal and managing to achieve almost perfect accuracy. This, however, is not helpful in real-world situations. Therefore, the problem arises of how to improve the identification of the minority class as opposed to achieving higher overall accuracy. The class imbalance problem has been the subject of extensive research [3], [4], [5] in different areas i.e. fraud detection, healthcare.

Many techniques for handling imbalanced data have emerged in the literature [6], [7], [8], [9]. Solutions have been implemented at the algorithmic, data and hybrid level. At the algorithmic level, algorithms are adjusted in order to reduce bias towards the majority class and improve classification. At the data level, sampling techniques are applied for synthetic data generation to balance the dataset. Finally, hybrid-level approaches combine data-level and algorithmic-level techniques. In Section II, we present the relevant literature. The class imbalance issue is observed in binary and multi-class classification problems. In this paper we focus on the binary classification problem as we try tackle the issue of fraud in the online gambling space.

A direct approach to the data generation process would be the use of a generative model that captures the actual data distribution [10] for generating synthetic data. Generative adversarial networks (GAN) are a recent method that uses neural networks to create generative models [11]. As previous studies have shown, GANs can be used effectively as an oversampling method to produce high-quality synthetic data [12]. In contrast

with other generative techniques, GANs are able to parallelise sample generation with sample classification. Further, they make no assumption about distribution and variational bounds. Finally, GANs make no use of Markov chain or maximum likelihood estimation [11], [13].

In this paper, we propose a GAN-based approach called synthetic data generation GAN (SDG-GAN), which, as the empirical results show, can be a powerful tool for tackling the imbalanced class problem on structured data by generating new high-quality instances. In Section III an overview of GANs is provided, where in Section IV we introduce our approach. Our method is validated in Section V and VI via experiments on benchmark datasets (Credit Card Fraud, Breast Cancer Wisconsin, Pima Diabetes) from different disciplines before applying it for generating new synthetic data for online gambling players in Section VII. Our method is evaluated in terms of its classification performance when combined with the classification models, namely logistic regression (LR), random forest (RF), multi-layer Perceptron (MLP) and XGBoost (XGB).

II. Related Work

Data-level methods are described as the sampling techniques used to balance a dataset [14]. This means that the number of instances of each class is adjusted either by increasing the instances of the minority class or by decreasing the instances of the majority class. In general, applying sampling algorithms will result in the alteration of the distribution of an imbalanced dataset until it is balanced. Various studies have shown that a balanced dataset can improve the performance of a classifier [15], [16]. Oversampling, undersampling and hybrid methods have been applied to achieve a balanced dataset.

Synthetic oversampling (i.e. generating new synthetic instances) and random oversampling (ROS) are the two methods of oversampling. In ROS, minority samples are added to the training set by randomly replicating minority class samples. Although the performance of a prediction algorithm can be improved with ROS [17], Chawla [18] has suggested that it could also cause overfitting – since the same data may be used more than once – and could be more computationally expensive.

Notwithstanding the problems originating from ROS, advancements in the field of imbalanced classification show that most issues can be overcome with synthetic oversampling. Synthetic oversampling methods generate new synthetic instances in order to balance a dataset. Examples of synthetic oversampling techniques include but are not limited to ADAptive SYNthetic sampling (ADASYN) [8] and Synthetic Minority Oversampling TEchnique (SMOTE) [7]. A popular extension to SMOTE includes selecting instances of the minority class that are misclassified, such as with a k-nearest neighbour classification model. This modified SMOTE method is called Bordeline-SMOTE (B-SMOTE) [9].

The key difference between ADASYN and SMOTE is that ADASYN uses a density distribution criterion to automatically decide how many samples need to be generated for each minority data point. First, it improves learning by reducing the bias caused by the imbalance in class priors. Second, it improves performance because the classification decision boundary is adaptively shifted toward 'difficult examples' [8].

Finally, hybrid methods incorporate both oversampling and undersampling techniques. Ganguly and Sadaoui [19] utilised a hybrid method of data oversampling and undersampling to improve effectiveness in addressing the issue of highly imbalanced auction fraud datasets. Their results showed a significant classification improvement for various well-known classifiers. Other popular hybrid methods in the literature include SMOTE+TOMEK [20] and SMOTE+ENN [21]. SMOTE+TOMEK aims to clean overlapping data points for each of the classes distributed in sample space, while SMOTE+ ENN deletes any instance of the majority class which its nearest neighbours are misclassified.

GANs are one of the most popular and successful generative technique for synthetic data generation [11], especially image generation [22] [23]. Literature on using GANs for oversampling structured data has also emerged. Douzas and Bacao [10] used a conditional GAN (cGAN) to approximate the true data distribution and generate data for the minority classes of various imbalanced datasets. They compared their results against standard oversampling approaches and showed improvements in the quality of data generation.

Lei et al. [24] designed CTGAN, a cGAN-based method to balance tabular datasets with both continuous and discrete columns. They designed a benchmark with seven simulated and eight real datasets and several Bayesian network baselines. CTGAN outperformed Bayesian methods on most of the real datasets while other deep learning methods did not. The authors in [25] proposed oversampling by training a GAN with vanilla GAN loss on only minority class observations. They compared their method against SMOTE and no oversampling and reported mixed results. Experiments showed that a classifier trained on the augmented dataset outperformed the same classifier trained on the original data. In this work, GANs are examined for tackling the imbalanced class issue, through the generation of synthetic data.

III. GANs

GANs are generative models based on a game-theoretic scenario in which a generator (G) network is competing against a discriminator (D) [11]. The generator, with noise variable Z as input, generates fake samples with distribution p_g that match the true data distribution, p_{data}. However, the discriminator network is trained to distinguish the real samples (drawn from the training data) and fake samples generated from G [1].

A common analogy in the literature for GANs [26] is to think of one network as an art forger and another as an art expert. The forger, known in the literature as the generator, G, creates forgeries with the aim of making realistic images. The expert, known as the discriminator, D, receives both forgeries and real images and aims to tell them apart. Both are trained simultaneously and in competition with each other.

Typically, the discriminator model is trained to maximise its ability to distinguish real input data from fake data. The generator tries to fool the discriminator by producing better fake samples. Mathematically, the generator and discriminator play a min-max two-player game with value function V(G,D) [11]:

$$\min_G \max_D V(G,D) = E_{x\sim p_{data}}[log(D(x))] + E_{z\sim p_z}[1 - log(D(G(z)))] \quad (1)$$

where E is the expectation, p_{data} is the real data distribution and p_z is a noise distribution. The training of a GAN could be characterised as an optimisation process for both the generator and discriminator. The output of the generator is defined as p_g. As equation (1) suggests, GANs aim to minimise the Jensen–Shannon divergence between the data distribution p_{data} and the generative distribution p_g with perfect minimisation reached when $p_g = p_{data}$. The optimisation equations for the generator and the discriminator are defined respectively as follows:

$$\min_G E_{z\sim p_z}[1 - log(D(G(z)))] \quad (2)$$

$$\max_D E_{x\sim p_{data}}[log(D(x))] + E_{z\sim p_z}[1 - log(D(G(z)))] \quad (3)$$

Although GANs are very promising for synthetic data generation, training a GAN is challenging [11] and often unstable which could lead to the following 'symptoms' [27]–[29]:

- Difficulties in making both the discriminator and generator converge [27].
- Collapse of the generator model by producing similar samples from different inputs [28].
- The discriminator converging quickly to zero [29], providing no reliable path for gradient updates to the generator.

Researchers have considered several approaches to overcome such issues. They have been experimenting with architectural changes [28], different loss functions [30] and both. Our SDG-GAN tries to eliminate the above problems by combining a different loss function and architecture compared to the original vanilla GAN [11]. In Section III-A, we give an overview of conditional GANs (cGANs) [31] and in Section IV, we explain how they were used to build the SDG-GAN architecture.

A. Conditional GANs

Conditional GANs [32] are a simple extension of the original GAN framework, which conditions the generator on class labels to generate output for a specific class [31]. The conditioning is achieved by feeding the class label y into both the discriminator and generator as additional input. Thus, the generator estimates the distribution of $p_{X|y}$, and the discriminator learns to estimate $D(X,y) = P(fake|X,y)$. The modification of the generator and discriminator with the conditional rule allows for the generation of samples belonging to a specific class. Furthermore, the conditional discriminator ensures that the generator does not ignore class labels [31]. Formally, the objective value function between generator G and discriminator D is the min-max in equation (4).

$$\min_G \max_D V(G,D) = E_{x\sim p_{data}}[log(D(x|y))] + E_{z\sim p_z}[1 - log(D(G(z|y)))] \quad (4)$$

During cGAN training, the discriminator is trained first with batches of only real features, $y_{real} = 1$ and then with batches of only fake ones, $y_{fake} = 0$ before the generator training continues through the GAN model. The GAN model assumes that the generated features will always be real, $y_{gan} = 1$. As cGAN is an extension of the original generative adversarial framework, it exhibits the same problematic behavior, i.e. mode collapse and unstable training, due to the vanishing gradient problem [33].

IV. Synthetic Data Generation GAN

In the SDG-GAN framework the generator and discriminator of SDG-GAN are both feedforward networks with a MLP architecture. The generator of a regular GAN aims to generate fake data that are close to the real distribution. The discriminator of a regular GAN is used to identify whether an input is real or fake from the generator.

The process of generating new instances of the minority class requires training the GAN to estimate the distribution of the data. When the training phase is completed, new synthetic data can be generated utilising the generator's abilities. The cGAN architecture of estimating the conditional distribution, $p_{x|y}$, is adapted in our method to generate the minority class samples. Instead of regular loss, feature matching loss is adapted by the SDG-GAN. Feature matching loss was introduced by [28] as a method for improving GAN training.

Here, we propose a GAN architecture based on cGANs. The generator is a feedforward neural network that tries to learn the actual data distribution. In contrast with a cGAN generator, we use a feature matching technique to train the generator. Feature matching changes the cost function for the generator to minimise the statistical differences between the features of the real data and generated data. This changes the scope of the generative network from fooling the opponent to matching features in the real data. The objective function of feature matching loss is defined as follows:

$$||E_{x\sim p_{data}} f(x) - E_{z\sim p_z(z)} f(G(z))||_2^2 \quad (5)$$

where $f(x)$ is the feature vector extracted by an intermediate layer in the discriminator. Feature matching addresses the instability of GANs by specifying a new objective for the generator that prevents it from over-training. Instead of directly maximising the output of the discriminator, the new objective requires the generator to generate data that match the statistics of the real data, while we use the discriminator only to specify the statistics we think are worth matching. Specifically, we train the generator to match the expected value of the features on an intermediate layer of the discriminator. This is a natural choice of statistics for the generator to match

because by training the discriminator, we ask it to find the features that are most discriminative of real data versus data generated by the current model [28].

To oversample an imbalanced dataset, we first trained the SDG-GAN's generator with imbalanced samples to estimate the data distribution. Once the training was completed, we could oversample the data by specifying to the generator how many new synthetic instances of the minority class we wanted to produce. We used a cGAN structure to estimate the conditional distribution, $p_{X|y}$, which allowed us to sample the minority class explicitly by conditioning the generator on the minority class label, $X_{new} = G(z, y = y^{minority})$.

The discriminator was trained similarly to a regular GAN discriminator. As with regular cGAN training, the objective had a fixed point where G exactly matched the distribution of the training data. We had no guarantee of reaching this fixed point in practice, but our empirical results indicated that feature matching is indeed effective in situations wherein a regular GAN becomes unstable [28]. Thus, we achieve the following objective function:

$$\min_G \max_D \underbrace{||E_{x\sim p_{data}} f(x|y) - E_{z\sim p_z(z|y)} f(G(z))||_2^2}_{FM_{Loss}} + E_{x\sim p_{data}}[log(D(x|y))] \quad (6)$$

where FM is the feature matching loss and the rest of the objective function is the binary cross entropy between true class label $y \in (0, 1)$ and the predicted class probability.

A. Hyperparameter Settings

Our proposed method has many hyperparameters that need to be tuned in order to achieve optimal performance. After experimenting with different set of hyperparameters, the hyperparameters below have been chosen after showing on producing the best results. We selected the settings presented in Table I. Future work could include optimising those hyperparameters for the oversampling task. The noise parameter distribution was set to be a Gaussian distribution with size dimensions set to 50. The dropout ratio was set to 0.2 on both discriminator's and generator's hidden layers. Batch size is 64 and number of epochs was set to 100. In terms of activation function rectified linear unit (ReLU) was used for the hidden layers where sigmoid for the output layer of discriminator and tanh for the ouptut layer of the generator. Adam optimiser was selected for the training [34].

Table I: SDG-GAN hyperparameters settings

Hyperparameters	**Value**
Learning Rate	$1x10^{-4}$
Optimiser	$Adam$
Epochs	100
Batch Size	64
Generator Layers	$(Noise, 128), (128, 64), (64, datasize)$
Discriminator Layers	$(datasize, 128), (128, 64), (64, 32), (32, 1)$
Activation function	$ReLU$
Noise Distribution	$N(0, 1)$
Noise	50

V. Experimental Design

To evaluate SDG-GAN as an oversampling method to tackle binary classification problems in imbalanced data, we compared the performance of the classification algorithms when combined with SDG-GAN and other state-of-the-art oversampling methods, e.g. SMOTE [7], ADASYN [8] and B-SMOTE [9], and other GAN-based oversampling architectures, e.g. cGAN.

In Section V-A, we introduce the publicly available datasets used as part of the evaluation process. In Section VII, we apply our method to the real-world gambling dataset provided by our industrial partners, examining money laundering risk in online gambling.The following hypotheses need to be met to describe our method as successful:

- H_1: The use of SDG-GAN to augment imbalanced datasets will improve the algorithmic performance in baseline experiments on the benchmark imbalanced datasets.
- H_2: The use of SDG-GAN will improve the algorithmic performance of classification algorithms in the real-world gambling dataset.

H_1 and H_2 are tested by combining the original and synthetic datasets with the four classification algorithms, i.e. LR, RF, XGBoost and MLP, in Section VI.

A. Benchmark Datasets

We evaluated our method on the different benchmark imbalanced datasets presented in Table II. The IR was defined as the imbalance ratio between the minority and majority classes. We used data from different sectors to examine the range of applications for our method. We selected the Credit Card Fraud Dataset from Kaggle [35] and the Pima Diabetes [36] and Breast Cancer Wisconsin (Diagnostic) Datasets from the UCI Machine Learning Repository [37], an online resource containing several datasets for machine learning purposes.

The rationale behind using the benchmark datasets was so that the results of this study could be easily compared to similar studies carried out previously and in the future. Moreover, it was decided that all datasets should describe a binary classification problem and contain numeric features to be in the same format as our gambling data.

The Credit Card Fraud Dataset contains transactions made by credit cards in September 2013 by European cardholders. It presents transactions that occurred over two days, with 492 frauds out of the 2,492 transactions. The Wisconsin Breast Cancer Dataset includes features computed from a digitised image of a fine needle aspirate of a breast mass. The features describe characteristics of the cell nuclei present in the image.

Table II: UCI datasets. There are three different sectors (B = business, L= life sciences). Number of features, number of instances, imbalance ratio

ID	Data Set	Sector	#Features	#Instances	IR
1	Credit Card Fraud	B	30	2,492	1:4.07
2	PIMA Diabetes	L	8	768	1:1.87
3	Breast Cancer	L	30	569	1:1.68

Table III: Real-world Gambling Dataset

ID	Data Set	Sector	#Features	#Instances	IR
1	Gambling Fraud	B	31	4,700	1:2.97

The purpose of the dataset is to classify a diagnosis as positive or negative. The PIMA Diabetes Dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. Its objective is to diagnostically predict whether a patient has diabetes based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database; in particular, all patients are females at least 21 years old of Pima Indian heritage.

Note that before these datasets were used, their attribute values were scaled to be in interval $[0, 1]$ by the min-max method to make the range of all attributes the same, preventing any one of them from dominating the others due to its scale. This reduced the range of values that the generator had to produce as well. With regard to implementation, all standard oversampling method tests were implemented using the '$imblearn.over\_sampling$' module in Python. We used the default hyperparameter settings for SMOTE and its variants, i.e. $kneighbours = 5$. For cGAN, we primarily used the hyperparameter settings that we set for SDG-GAN method, as seen in Table I.

VI. Results

For each dataset, we present the classification results observed after 10 runs for each oversampling technique and classification algorithm. The results in this section represent the average scores during those 10 runs. Similar to the process of [12], we split the data into testing and training sets. The training set included 80% of the total population of the samples of each class and the testing set the other 20% of the data. The data were shuffled to ensure reliable distribution in the sets.

In the SDG-GAN, given an imbalanced training dataset, we first calculated the imbalance difference between the classes in the dataset. Then, a set of noise vectors with a dimension of 50 was used as the input for the generator. We trained the network generator by optimising the generator using the loss equation (5). Real and synthetic data were then used as input for the discriminator D to output a probability value for evaluating the authenticity of the input data. Finally, the simulation samples generated through SDG-GAN were bonded with the original samples to enhance and balance the training dataset, which was then fed into the machine learning model for training.

A. Results of Benchmark Datasets

Table IV, Table V and Table VII show the results observed for the three imbalanced public datasets of Credit Card Fraud, Breast Cancer and Pima Diabetes. We compare the five oversampling techniques in combination with four classification algorithms. The performance of each classification method is measured in terms of recall, precision and F1 score.

For the Credit Card Fraud Dataset in Table IV, the highest F1 score was achieved when SDG-GAN was combined with RF for a score of 91.31%. In Table V for the Breast Cancer Dataset, cGAN combined with XGBoost outperformed the rest of the methods with F1 score of 91.95%. Similarly with the Credit Card, in the Pima Diabetes Dataset, SDG-GAN in combination with RF produced the best results with an F1 score of 70.80% as Table VII indicates. This was a significant improvement of $\approx$ 5% compared to when no oversampling was used and an improvement of $\approx$ 2% than the second-best combination between MLP and ADASYN. Another observation that could be drawn from the results was that when the standard oversampling techniques were used i.e. SMOTE, ADASYN, there was a drastic improvement in the classification of the minority class with better overall recall compared to precision (in the majority of cases). However, simultaneously, there was a huge drop in the classification accuracy of the majority class. This was supported by the increase of the recall score in the Credit Card Fraud Dataset prior to the use of any oversampling method; on average, the recall was 85% and the precision 94%. When SMOTE was used, the recall score increased significantly, while the precision decreased. However, this was not the case when SDG-GAN was used for oversampling, whereby we saw a more robust improvement in the classification metrics, as Table IV and Table VII show.

The mean rankings of the F1 score per classifier across all datasets are presented in Table VIII. No one oversampling technique performed best across all classification methods and datasets. However, the SDG-GAN method performed consistently well and managed to achieve the highest overall mean rank score (2.6).

Among the oversampling methods, SMOTE produced the second-best results, outperforming ADASYN and B-SMOTE. This indicated that the more recent variations of SMOTE do not necessarily outperform their predecessor, mirroring previous findings in the credit scoring literature [31]. Considering the mean ranking results from Table VIII, we could address the second hypothesis, H_1, stating that the use of SDG-GAN improves the classification performance in experiments on benchmark imbalanced datasets.

VII. SDG-GAN in Online Gambling

After the success of our proposed method on the benchmark datasets, we applied our SDG-GAN technique for generating synthetic players' data to tackle the imbalanced class in the

Table IV: Credit Card detection results: recall, precision and F1 measure

Algorithms	Metrics	W/O	SMOTE	ADASYN	B-SMOTE	cGAN	SDG-GAN
LR	Recall	0.8142	0.8571	0.8667	0.8495	0.8144	0.8090
	Precision	**1.0000**	0.9545	0.7865	0.9320	0.9875	0.9863
	F1	0.8975	0.9032	0.8246	0.8888	0.8926	0.8889
RF	Recall	0.8984	0.8694	0.9288	0.8894	0.8453	0.9208
	Precision	0.9170	0.9586	0.8309	0.9106	0.9647	0.9055
	F1	0.9076	0.9116	0.8771	0.8999	0.9010	**0.9131**
XGB	Recall	0.8973	0.9163	0.9087	0.8776	0.8559	0.9053
	Precision	0.9112	0.8959	0.8787	0.8600	0.9694	0.9122
	F1	0.9042	0.9060	0.8935	0.8687	0.9091	0.9087
MLP	Recall	0.8191	0.8830	0.9087	0.8761	0.8454	**0.9487**
	Precision	0.9390	0.8384	0.8536	0.9082	0.9879	0.8315
	F1	0.8750	0.8601	0.8803	0.8919	0.9111	0.8862

Table V: Breast Cancer detection results: recall, precision and F1 measure

Algorithm	Metrics	W/O	SMOTE	ADASYN	B-SMOTE	cGAN	SDG-GAN
LR	Recall	0.8095	0.8888	0.9048	0.9302	0.9067	0.8837
	Precision	0.9189	0.9302	0.8636	0.8888	0.8863	0.9500
	F1	0.8608	0.9091	0.8837	0.9090	0.8966	0.9157
RF	Recall	0.8604	0.9069	0.8666	0.9069	0.8604	0.8809
	Precision	0.8809	0.8478	0.9069	0.8667	0.9024	0.8604
	F1	0.8706	0.8764	0.8863	0.8863	0.8809	0.8706
XGB	Recall	0.8524	0.9262	0.9143	0.9119	0.9524	0.8571
	Precision	0.8802	0.8282	0.8426	0.8567	0.8889	0.9767
	F1	0.8637	0.8742	0.8754	0.8821	**0.9195**	0.9130
MLP	Recall	0.8604	0.9069	**0.9381**	0.9302	0.9069	0.8604
	Precision	**0.9487**	0.8863	0.7940	0.8888	0.8863	0.9737
	F1	0.9024	0.8965	0.8578	0.9090	0.8965	0.9137

real world fraud detection gambling dataset. As mentioned in Table III we have 4,700 instances in the dataset from which 1,200 are described as high risk for money laundering. We compare our results with the existing system that our partners have in place with overall F1 score 84.7%.

We used SDG-GAN as part of the supervised learning framework for oversampling the minority class. Similar to the benchmark dataset case experiments, we evaluated the effectiveness of our approach for practical applications against the standard oversampling techniques and a GAN-based approach introduced in this paper. Table IX presents the classification performance results for the gambling fraud dataset.

Similar to the experiments on the Credit Card Fraud and Diabetes Datasets, the performance of SDG-GAN was superior, with the highest F1 measure and precision at 89.73% and 89.43%, respectively. As Table IX shows, SDG-GAN combined with XGBoost and RF outperformed the other oversampling techniques. However, when combined with LR, we did not expect it to improve the classification performance. Overall, the SDG-GAN results showed it can effectively estimate even complex data distributions. Furthermore, the results from Table IX supported the final hypothesis, H_2, stating that the use of SDG-GAN could improve the identification rate of risk of money laundering in online gambling.

Comparing the new classification results with SDG-GAN and the rule-based system, there was a significant F1 score improvement of around 5%. Overall, with our oversampling method, we managed to reduce the number of both false

Table VII: Pima Diabetes Dataset results: recall, precision and F1 measure

Algorithm	Metrics	W/O	SMOTE	ADASYN	B-SMOTE	cGAN	SDG-GAN
LR	Recall	0.6181	0.7455	0.7272	0.7091	0.6727	0.6727
	Precision	**0.6939**	0.5694	0.5479	0.5652	0.6851	0.6379
	F1	0.6538	0.6456	0.6250	0.6290	0.6788	0.6549
RF	Recall	0.6727	0.7636	0.7454	0.6727	0.6545	0.7272
	Precision	0.6379	0.6086	0.5775	0.6271	0.6101	0.6897
	F1	0.6548	0.6774	0.6507	0.6491	0.6315	**0.7080**
XGB	Recall	0.6727	0.7091	0.7636	0.7455	0.6727	0.6545
	Precision	0.5781	0.6094	0.5753	0.5775	0.6066	0.5902
	F1	0.6218	0.6555	0.6562	0.6508	0.6379	0.6207
MLP	Recall	0.6182	0.7818	**0.7818**	0.7091	0.6727	0.6727
	Precision	0.6938	0.6142	0.5890	0.6094	0.6491	0.6852
	F1	0.6538	0.6880	0.6718	0.6555	0.6607	0.6788

Table VIII: Summary Rank Results For F1 score

	Overall	Classifier			
Method	Mean Rank	LR	RF	XGB	MLP
SDG-GAN	**2.6**	2.7	2.3	3.3	2.0
W/O	4.3	3.3	4.0	5.0	4.7
SMOTE	2.9	2.0	2.7	3.3	3.7
B-SMOTE	3.6	4.0	3.7	3.7	3.0
ADASYN	4.3	5.3	4.0	3.7	4.3
cGAN	3.1	2.7	4.3	2.0	4.0

positives and false negatives compared to the other techniques and enhanced the ability of the classification algorithm to distinguish the AML and Normal groups' classes.

VIII. Conclusion

In this paper, we introduced SDG-GAN, an architecture based on GANs for generating synthetic data. Our method was compared against popular oversampling techniques i.e. SMOTE, B-SMOTE and ADASYN as well as other adversarial network architecture that has been used for generating new data i.e. cGANs. We evaluated the ability of SDG-GAN to produce high-quality synthetic data by comparing the algorithmic performance of four machine learning classification algorithms when combined with our method on three public imbalanced datasets and a real-world gambling fraud dataset. We found that the SDG-GAN oversampling compared favourably to the other oversampling methods and achieved the highest overall rank, as Table VIII shows. Our method outperformed SMOTE, ADASYN, B-SMOTE and cGAN on three out of the four examined imbalanced datasets, with the best performance achieved when it was combined with RF in two out of the three experiments.

In the real-world gambling dataset, the application of SDG-GAN helped improve the identification rate by improving the F1 score by 5% compared to the rule-based system and around 0.4% compared to the other oversampling techniques.

References

[1] C. Charitou, A. d. Garcez, and S. Dragicevic, "Semi-supervised gans for fraud detection," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[2] G. E. Batista, A. C. Carvalho, and M. C. Monard, "Applying one-sided selection to unbalanced datasets," in *Mexican International Conference on Artificial Intelligence*. Springer, 2000, pp. 315–325.

[3] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.

[4] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[5] R. Batuwita and V. Palade, "Class imbalance learning methods for support vector machines," 2013.

[6] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Icml*, vol. 97. Citeseer, 1997, pp. 179–186.

[7] 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, 2002.

[8] 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)*. IEEE, 2008, pp. 1322–1328.

[9] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*. Springer, 2005, pp. 878–887.

[10] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Systems with applications*, vol. 91, pp. 464–471, 2018.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[12] F. H. K. d. S. Tanaka and C. Aranha, "Data augmentation using gans," *arXiv preprint arXiv:1904.09135*, 2019.

Table IX: Gambling Dataset results

Algorithm	Metrics	W/O	SMOTE	ADASYN	B-SMOTE	cGAN	SDG-GAN
LR	Recall	0.6842	0.8907	0.8745	0.9109	0.6541	0.7206
	Precision	0.8942	0.8209	0.8120	0.7840	0.8788	0.8900
	F1	0.7752	0.8544	0.8421	0.8427	0.7500	0.7964
RF	Recall	0.9245	0.9338	0.9249	0.9367	0.9245	0.9004
	Precision	0.8546	0.8389	0.8328	0.8223	0.8556	**0.8943**
	F1	0.8881	0.8838	0.8764	0.8761	0.8887	**0.8973**
XGB	Recall	0.9195	0.9449	0.9492	0.9576	0.8923	0.9322
	Precision	0.8645	0.8479	0.8327	0.8278	0.8722	0.8627
	F1	0.8912	0.8938	0.8871	0.8880	0.8821	0.8961
MLP	Recall	0.8189	0.9671	0.9588	**0.9712**	0.8213	0.8601
	Precision	0.8805	0.7655	0.7767	0.7540	0.8816	0.8636
	F1	0.8486	0.8545	0.8582	0.8489	0.8807	0.8619

[13] K. S. Ngwenduna and R. Mbuvha, "Alleviating class imbalance in actuarial applications using generative adversarial networks," *Risks*, vol. 9, no. 3, p. 49, 2021.

[14] H. Alhakbani, "Handling class imbalance using swarm intelligence techniques, hybrid data and algorithmic level solutions," Ph.D. dissertation, Goldsmiths, University of London, 2019.

[15] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.

[16] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*. Springer, 2015.

[17] C. X. Ling and C. Li, "Data mining for direct marketing: Problems and solutions." in *Kdd*, vol. 98, 1998, pp. 73–79.

[18] N. V. Chawla, "Data mining for imbalanced datasets: An overview," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 875–886.

[19] S. Ganguly and S. Sadaoui, "Classification of imbalanced auction fraud data," in *Canadian Conference on Artificial Intelligence*. Springer, 2017, pp. 84–89.

[20] G. E. Batista, A. L. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: a case study." in *WOB*, 2003, pp. 10–18.

[21] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[22] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong gan: Continual learning for conditional image generation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2759–2768.

[23] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1316–1324.

[24] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, 2019, pp. 7335–7345.

[25] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019.

[26] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[27] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[28] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, 2016, pp. 2234–2242.

[29] J. Li, A. Madry, J. Peebles, and L. Schmidt, "Towards understanding the dynamics of generative adversarial networks," *arXiv preprint arXiv:1706.09884*, 2017.

[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[31] J. Engelmann and S. Lessmann, "Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning," *arXiv preprint arXiv:2008.09202*, 2020.

[32] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[33] M. Zheng, T. Li, R. Zhu, Y. Tang, M. Tang, L. Lin, and Z. Ma, "Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification," *Information Sciences*, vol. 512, pp. 1009–1023, 2020.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: a realistic modeling and a novel learning strategy," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3784–3797, 2017.

[36] J. W. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*. American Medical Informatics Association, 1988, p. 261.

[37] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.