IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 21, 2021, accepted November 2, 2021, date of publication November 8, 2021, date of current version November 12, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3126656*

# When Two Are Better Than One: Synthesizing Heavily Unbalanced Data

**FRANCISCO FERREIRA[1], NUNO LOURENÇO[1], BRUNO CABRAL[1], AND JOÃO PAULO FERNANDES[2]**

[1]CISUC, DEI, University of Coimbra, 3030-290 Coimbra, Portugal
[2]LIACC, DEI, Faculdade de Engenharia da Universidade do Porto, 4200-465 Porto, Portugal

Corresponding author: Nuno Lourenço (naml@dei.uc.pt)

This work was supported in part by the National Funds through the FCT—Foundation for Science and Technology, I.P., within the scope of the project Centre for Informatics and Systems of the University of Coimbra (CISUC) under Grant UID/CEC/00326/2020; in part by the European Social Fund, through the Regional Operational Program Centro 2020; and in part by the Carnegie Mellon University (CMU)|Portugal Project autonomiC plAtform for MachinE Learning using anOnymized daTa (CAMELOT) under Grant POCI-01-0247-FEDER-045915.

**ABSTRACT** Nowadays, data is king and if treated and used properly it promises to give organizations a competitive edge over rivals by enabling them to develop and design Intelligent Systems to improve their services. However, they need to fully comply with not only ethical but also regulatory obligations, where, e.g., privacy (strictly) needs to be respected when using or sharing data, thus protecting both the interests of users and organizations. Fraud Detection systems are examples of such systems where Machine Learning algorithms leverage information to classify financial transactions as legitimate or illicit. The data used to create these solutions is usually highly structured and contains categorical and continuous features characterised by complex distributions. One of the main challenges of fraud detection is concerned with the scarcity of fraudulent instances which results in highly unbalanced datasets. Additionally, privacy is crucial, and it is usually forbidden, or not possible, to share the data of organizations and individuals for creating or improving models. In this paper we propose a framework for private data sharing based on synthetic data generation using *Generative Adversarial Networks (GAN)* that learns the specificities of financial transactions data and generates fictitious data that keeps the utility of the original datasets. Our proposal, called Duo-GAN, uses two GAN generators to handle the data imbalance problem, one generator for fraudulent instances and the other for legitimate instances. With this approach, we observed, at most, a 5% disparity in F1 scores between classifiers trained and tested with actual data and the ones trained with synthetic data and tested with actual data.

**INDEX TERMS** Fraud detection, generative adversarial networks, privacy, machine learning, synthetic data generation, tabular data.

## I. INTRODUCTION

In recent years, the growing number of monetary transactions fostered extensive data collection. These datasets contain information about individuals' and organisations' spending patterns, which upheld the development of Machine Learning based solutions to analyse transactions and detect financial fraud in real-time [1].

Fraudsters' techniques are usually transversely applicable through industries and services, making a specific fraud detection solution of one organisation adequate for other organisations with similar characteristics. However, companies are subject to tight regulations concerning data privacy, either enforced through existing laws or service contracts. Consequently, sensitive information such as social security numbers and credit card numbers make sharing and using these financial datasets challenging, even between departments of the same company. To address this issue, companies usually have to go through a laborious anonymization process, select non-private information, validate it with legal teams, and convince their clients that the data is not used for any undisclosed purposes. Two problems arise from this: the *cold start problem* where the classification models do not have sufficient information to make accurate predictions; and the delayed deployment of solutions in order to gather the necessary data to have the models at an acceptable quality

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

threshold. To minimise these problems, researchers from both the academia and industry made some progress on how to share sensitive data without revealing it, i.e., keeping it private, and that, at the same time, the utility of the data is not lost [2]–[5].

One way to achieve this goal is employing Differential Privacy [6] or other anonymization techniques. The authors of [7] observed that Differential Privacy is used with the assumption that data is not correlated. However, correlation in datasets is expected and desirable, which weakens privacy and leads to unexpected data leakage. One other issue with this approach is that adding noise to datasets leads to information loss, which results in a decrease of the data utility. Other possible alternative is to use Federated Learning to train models locally, and then sharing the results of the training/computation [8], [9].

Another alternative to anonymization of sensitive information is data synthesis. Synthetic datasets are composed of samples with information that is not natural, i.e., they are artificially generated instead of being collected but exhibit properties similar to those found on the original data. If the resemblance is high, one can use the synthetic data to learn the real dataset's underlying statistical properties employing Machine Learning and statistical tools without ever having to look at the actual data. Synthetic data generation would help handle the privacy concerns that Machine Learning (ML) practitioners and Data Scientists must face.

One way to generate synthetic data, whilst respecting privacy constraints, is to use generative models, which can capture the distribution of training data and generate new artificial instances that maintain the utility of the original data [2], [10], [11]. One generative model that gained relevance in recent years in creating synthetic samples is the Generative Adversarial Networks (GAN) [12]. GANs usually combine two deep neural networks called Generator and Discriminator. Both undergo adversarial training where these networks are confronted in a zero-sum game between them. The Generator creates fake samples based on an input distribution, aiming at deceiving the Discriminator. On the other hand, the Discriminator's goal is to learn to distinguish between the fake (i.e., synthetic) samples and the actual input data.

GANs have shown impressive generative abilities specially in the field of computer vision [13]–[15]. These remarkable results have inspired researchers to pursue the use of GANs beyond image generation, namely for tabular data, with relative success [4], [16]–[18]. Building generative models for tabular data raises several challenges. The information is usually organised in a two dimensional array, where the lines are observations of a certain phenomenon and the columns are the attributes (i.e., features) that were measured for each observation. The attributes can have different data types, such as from numerical, categorical, time, text, geo-location data. Additionally, and to make things more arduous, the attributes can follow different probabilistic distributions, with multiple modes and long tails [16], [19].

In spite of the recent efforts to use GANs in tabular data, their application to synthetic data generation of highly unbalanced datasets remains mostly unexplored. Our work aims to address this by designing, implementing, and validating a Generative Adversarial Networks framework for generating valuable synthetic tabular data to the domain of fraud detection. In concrete, we developed a generative model called Duo-GAN that uses two GANs, one to generate synthetic positive samples and the other to generate synthetic negative samples. This architecture aims at overcoming the problem of having highly imbalanced datasets, as it is standard on the fraud detection domain, where the positive examples (i.e., fraudlent instances) are usually scarce. The results show that Duo-GAN can capture the data's underlying distributions, keeping the correlations that existed between features in the original data. We also show that classification models trained on synthetic data produced by our framework outperform classifiers trained on data generated by a single GAN Generator Model. Moreover, our proposed approach generates synthetic data that can be used to train classifiers and achieve comparable results to models trained on the actual data.

The main contributions of this work are:

- Proposal of a novel architecture called Duo-GAN to generate synthetic data for highly unbalanced datasets;
- Methodology to show that Duo-GAN is able to generate synthetic data with the same distribution of the original data without overfitting to it;
- Validation of the utility of the synthetic data generated by using it to train Machine Learning models. these models are then tested in real world data, and we show that there is almost no loss in performance when using synthetic data to build the models.

The remainder of the paper is organised as follows. Section II discusses the related work. In Section III we detail the architecture of Duo-GAN, and in Section IV describes the methodology followed to validate our approach. Section V presents and discusses the experimental results obtained, and in Section VI draws the main conclusions and points towards future work.

## II. RELATED WORK

In this section, we look into the work developed in the field of synthetic data generation. There have been several research efforts targeting the generation of synthetic data using approaches such as classification and regression trees [20], Bayesian Networks [2], and copulas [21]. However, the nature of the data, be it the typology of features, their distributions, or privacy limitations, limits the success of these approaches for data reproduction.

Other researchers have used Variational Auto Encoders as a possible solution to address the problem of synthetic data generation. The most successful domain of application for these models is synthetic image generation [22], [23], but some studies on tabular data have

also been developed [24], [25]. Tabular data regularly contains features with continuous, binary, and discrete values and complex distributions. These factors make the generation of tabular data a non-trivial task [17].

Generative Adversarial Networks architectures are perhaps the most well-known generative model nowadays due to their success in the domain of synthetic image generation [12], [26]. It does not come as a surprise that researchers have started to apply them to other domains, namely tabular data generation.

In [16] the authors propose TGAN, a GAN based data synthesizer architecture to generate tabular data. TGAN uses one-hot-encoding for modeling categorical features and Gaussian Mixture Models (GMM) for encoding continuous features to avoid saturating gradients using normalisation. The Generator in TGAN is a Long Short-Term Memory (LSTM), and the Discriminator is a Feed Forward Neural Network. CTGAN [17] was introduced as an extension over TGAN, using a conditional Generator and Discriminator. In CTGAN, both the Generator and Discriminator are Feed Forward Neural Networks. Both TGAN and CTGAN showed positive results when testing the classifiers trained on synthetic data on the actual data.

Similar to fraud detection, the medical field is bound to high privacy regulations, making use and sharing health records amongst research communities hard. MedGAN [4] is proposed to circumvent these limitations. This method achieves positive results, with human doctors unable to distinguish synthetic samples from actual samples, except for several outliers.

Having already registered relative success on several domains, by handling tabular data diversity for those specific domains, these GAN-based approaches still struggle with unbalanced data [27], [28] present in the fraud detection domain where the data points for legitimate transactions overshadow fraudulent ones [29].

## III. PROPOSED APPROACH: DUO-GAN

In this section we present Duo-GAN, our approach for sharing and using data for monetary fraud detection whilst preserving privacy and assuring a comparable success rate of fraud detection. Our goal is to generate synthetic data that exhibits the same characteristics, patterns and distributions of the original data without exposing private information.

When working in the domain of fraud detection, two main challenges arise: i) we have to work with tabular data that contains features of different data types, such as binary, discrete and continuous, and depicts diverse kinds of distributions; ii) the data points for legitimate transactions strongly outweigh fraudulent ones, creating a highly unbalanced dataset.

Concerning the first challenge, the characteristics of the data alone are known to impose a significant amount of difficulties regarding the generation of tabular data [17]. It is common for tabular datasets to have columns that have non-gaussian distributions, which may lead to vanishing gradients in normalisation processes or multi-modal distributions that are difficult to model accurately.

Regarding the second challenge, problems emerge when we have highly unbalanced datasets, as the computational models will struggle to generate samples that maintain the distributions and relationships between features. This emerges from an over-exposure to one class while under-exposing the model to the other class. This will lead the model to capture the distributions present in the dominant class because it is largely exposed to it, while training instances from the less represented class are so few that they fail to have an impact on the parameters of the model. This leads to poor representation of the data mainly in the less represented class, whose synthetic samples will become too similar to the dominant class. This will result in severe consequences on classification tasks, given that the instances of the underrepresented class are of poor quality and too similar to the dominant ones, making it difficult for classifiers to differentiate between the two different classes.

To generate more faithful synthetic datasets, we introduce Duo-GAN, a Generative Model using two GANs: one for positive, i.e., fraudulent instances, and another for negative instances. This setup allows each generator to learn the class conditional distributions, as well as the relationships in each class, in place of learning the distribution and relationships of the whole data. This allows for the creation of more faithful samples for each respective class, mainly improving the quality of under-represented classes, i.e. case the fraudulent instances. With more faithful data for the positive class, classifier models should be able to differentiate better between fraudulent and legitimate instances of data.

The architecture of Duo-GAN is depicted in Figure 1. The process for generating synthetic datasets starts, in Phase 1, with dividing the original dataset into a positive dataset and a negative dataset and removing the target column from each one. In Phase 2, we feed the datasets of positive samples and negative samples to two GANs, which will learn the characteristics of the samples that compose each dataset. In Phase 3, we generate a positive synthetic dataset using the GAN trained on the original positive dataset and a negative dataset from the GAN trained on the original negative dataset. After this, we add the target column for each of the synthetic datasets and merge them to create a full synthetic dataset.

## IV. EXPERIMENTAL STUDY

In this chapter we detail the methodology and the experimental settings used to validate Duo-GAN. Firstly, we need to assess if our approach in generating novel samples that resemble the ones from the original dataset, without being an explicit copy. Then we need to evaluate how useful the synthetic samples are to train a Machine Learning model, i.e., we need to evaluate the utility of the data.

### A. METHODOLOGY

The experimental pipeline is detailed in Figure 2, and consists of three main steps: 1) data generation; 2) validation of the

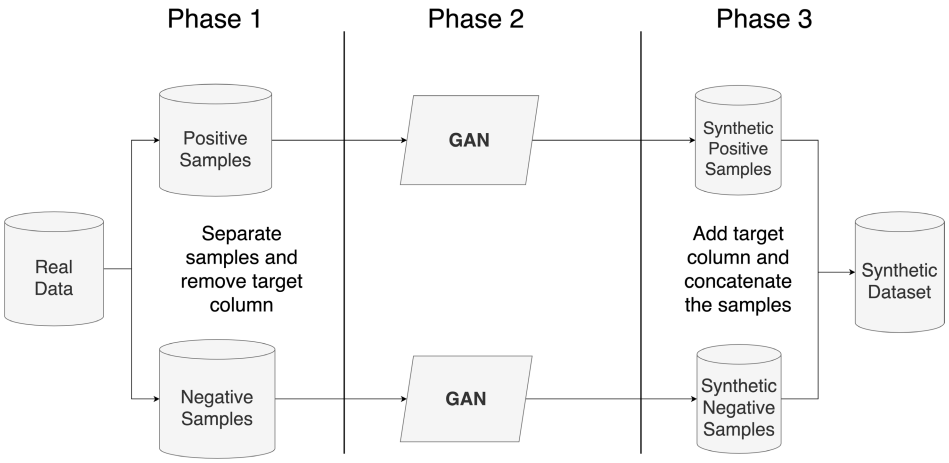



**FIGURE 1.** Duo-GAN - proposed generative model for accurate synthetic data generation of heavily unbalanced datasets.

synthetic data; 3) synthetic data utility validation. Essentially, this step abstracts the model described in Figure 1.

In Step 1, we use real-world, highly unbalanced, public datasets to train models to create synthetic data using Duo-GAN, and then generate a synthetic dataset using the trained models. We detail this stage in section IV-A1.

In Step 2, our generated dataset goes through an evaluation process to estimate the synthetic data's utility. This evaluation includes analysis of underlying information of data, measured using the distributions of feature values and correlations. In Section IV-A2 the specifics of this step are explained.

Finally, in Step 3, two groups of classifiers are trained, one with a real dataset, and the other with a synthetic dataset created by a generator model trained with the real data. Both groups are then tested on a test set of real examples. Comparing the behavior of the group trained on real data with the group trained on synthetic records allows us to quantify the loss of performance resulting from using synthetic data. Section IV-A3 contains a more detailed explanation of this step.

As well as using groups of classifiers trained with real data, a standard method of generating synthetic data composed of a single GAN will also create a synthetic dataset to train a group of classifiers. Comparing the performance of classifiers trained on synthetic datasets with different origins allows us to validate the effectiveness of our approach approach in synthesizing high quality data.

### 1) SYNTHETIC DATA GENERATION

As sketched in Step 1 of Figure 2 we train the GAN model and generate the synthetic samples. In this step we divide the dataset into two sets, $T_{train}$ and $T_{test}$, with the training set being made up of 70% the records, while the test set contains the other 30%. $T_{train}$ is used for training both generator models (Duo-GAN and Single GAN) and $T_{test}$ is set aside until Step 3 (described in Section IV-A3). Records that include unknown values are removed from the data. There is no need for additional processing of the data at this stage, given that both models are capable of handling datasets with both numerical and categorical features. The training process runs until we reach a specific value of the loss function or a maximum number of epochs. The loss function considers the performance of both the Generator and the Discriminator. When the training finishes, we take the Generator from the GAN models and use it to create synthetic samples. To guarantee that the real and synthetic datasets are somewhat comparable, we ensure that the synthetic dataset contains the same number of positive and negative instances as the real dataset, i.e., we ensure that the synthetic and the real datasets are similar.

### 2) SYNTHETIC DATA VALIDATION

The main goal when validating the synthetic data is to analyse and anticipate its utility. As depicted in Step 2 from Figure 2, we need to understand and verify if the generative model can create synthetic samples that keep the characteristics of the real data. We start by randomly sampling 5000 instances from both the original dataset and the synthetic dataset. Afterwards, we calculate a *singledivergence* score between each data instance defined as:

$$Single\ Divergence = \sum_{i=1}^{n} neq(d[i], s[i])) \tag{1}$$

where $n$ is the number of features, $d[i]$ is the value of feature $i$ for sample $d$ of the original dataset, $s[i]$ is the value of feature $i$ for sample $s$ from the synthetic dataset, and $neq$ is a function that returns 1 if the values are different and 0 if they are equal.

After computing all the pairwise distances using (1), we obtain the smallest value for each of the synthetic instances and calculate the average minimum distance to actual instances in the dataset, as described in Equation 2

$$Divergence = \sum_{i=1}^{j} \frac{1}{j} min(Single\ Divergence[i])) \tag{2}$$

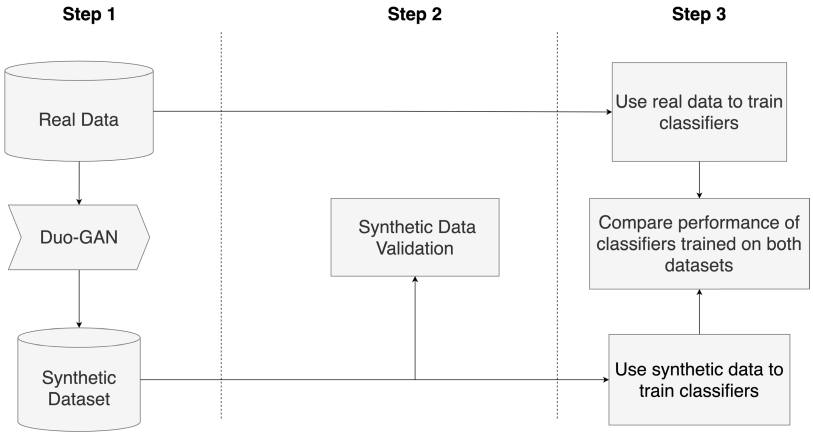



**FIGURE 2.** Methodology pipeline.

where $j$ is the number of samples, and *Single Divergence*[$i$] is a vector of *SingleDivergences* between instance $i$ and the instances in the real dataset.

We call this the *Divergence* score. The lower the *Divergence* value, the closer the resemblance between the two datasets.

The results of this metric are important, since they provide two relevant insights into the data: a lower *Divergence* value indicates samples that more closely resemble the ones present in the real dataset meaning we can expect higher utility; and an insight into the privacy of the synthetic dataset, since it allows us to see if any record is a copy of an original record if they have a *singledivergence* value of 0 in relation to real records.

Additionally, we rely on statistical tools to compare the distribution of each feature's values in the synthetic data. First, for each feature in the datasets we compare the distributions by creating an histogram of values. Then we perform a correlation analysis by creating correlation matrices for both the original and the synthetic datasets. The correlation method used depends on the type of the features being analysed. For the datasets that include categorical features *Spearman Rank Correlation* will be used, and for the datasets with only real valued features we will use *Pearson Correlation*.

#### 3) UTILITY VALIDATION

Synthetic data must keep as much utility as possible to produce classifiers that can be as good as the classifiers created using real data. To evaluate data utility, we will analyse how effective are the ML models when trained with the synthetic data and tested on real data. Afterwards, we will compare the performance of the same models using real-world data. Step 3 of Figure 2 illustrates this process. The datasets used for training are:

- $T_{train}$ - Dataset composed of real-world samples;
- $S$ - Dataset composed of synthetic samples.

After training, we use the $T_{test}$ dataset to evaluate each model. We use the $F_1$ score to assess the classifiers performance since it is better tailored to deal with unbalanced data.

Measuring the gap in performance between the models trained in the real data and those trained with synthetic data allows us to quantify how capable the generated data is of replacing actual data for training models and, as such, to validate our proposal. This evaluation procedure will assess how mutually compatible the real-world and the synthetic datasets are.

The models used for the classification task were the *scikit-learn* library implementations of the XGBoost, AdaBoost, Decision Trees using the default parameters defined in the library [30].

### B. EXPERIMENTAL DESIGN

In this section we detail the datasets used in our experiments, and after, we explain the structure of each experiment, including generator models and datasets used, along with their objectives.

#### 1) DATASETS

To test the ability of the proposed approach to generate synthetic data, we use two datasets that reflect the characteristics of data present on financial transactions and whose details we will describe next Both datasets are suited for binary classification, the same way fraud detection is a binary classification problem since a transaction is either fraudulent or legitimate.

1) The **Adult** dataset [31] contains data extracted from a census database. It comprises eight discrete features (e.g., gender, relationship status, work class) and six continuous features (e.g., age, capital gain, capital loss number of years of education) describing a person. The goal is to predict whether the income of a person will exceed \$50K dollars a year. The dataset has around 45 thousand instances, with only 24% of them belonging to the positive class, indicating income over \$50K dollars a year.
2) Understanding if a transaction is fraudulent is important for credit card companies to protect clients from unsolicited purchases. The **Credit Card Fraud Detection** dataset [32]–[38] contains 284807 transactions

made by European citizen cardholders in a period of two days in September of 2013. It contains 30 features that describe a certain transaction, where 28 of them result from a Principal Components Analysis (PCA) transformation, one is `Time` which corresponds to the amount of time elapsed since the first transaction on the dataset. The last one is `Amount`, which corresponds to the monetary value of the transaction. The goal is to use predict whether a transaction is fraudulent or genuine. Some changes to the dataset were made in order to reduce the lack of balance and aiding with reducing the run time of experiments. For that purpose we sample all 492 positive instances of fraud and then randomly sample 49508 instances from the remaining negative instances, still leaving the dataset highly unbalanced at just under 1% of the transactions recorded as fraudulent instances.

#### 2) PARAMETERS SETTINGS

The experiments describe above aim at assessing if a double GAN architecture is capable of creating synthetic data that retains the utility present in real data, while also comparing the performance of a double GAN architecture to a single GAN architecture.

In what concerns the GAN model adopted for our experiments, we selected the TGAN implementation described in [16] and publicly available in Github,[1] with the parameter settings described in Table 1. To keep any biases and contamination introduced by any specific parameterisation from affecting the analysis of our results, the numerical settings were selected taking into account the recommendations detailed in [16]. We also keep the structure and architecture of the original TGAN model. TGAN uses a generator based on LSTM Recurrent Neural Network (RNN) to generate the values of the columns (i.e. features) in the table, taking into account the values in the previous ones. Having a RNN as a generator allows us to capture and maintain the relationships between the different columns. Regarding the discriminator, it is a fully connected artificial network that receives as many inputs as there are features in the problem being tackled. To address the issues of having features with a multi-modal distribution, TGAN relies on a GMM that clusters the values of the numerical variables.

All the models were trained using the following computational resources: Intel(R) Xeon(R) CPU @ 2.20GHz, 13GB RAM, 1 NVIDIA Tesla K80.

## V. RESULTS

This section details the results of the experimental design we detailed in the last section. For each experiment we start by performing the synthetic data validation where we show that Duo-GAN is not memorizing the real-world dataset, but rather it is capturing its underlying patterns. Then we analyze the synthetic data utility by training the classifiers and testing them on real data.

[1]Github Repository: https://github.com/sdv-dev/TGAN

**TABLE 1. Configuration used for each TGAN in our proposal [16].**

| Parameter | Value |
|---|---|
| Learning rate | 0.001 |
| L2 Norm | 0.00001 |
| Random noise vector | 200 |
| Noise upper bound | 0.2 |
| Batch size | 200 |
| Optimizer | Adam |
| LSTM Hidden State size | 50 |
| Generator fully connected layer size | 64 |
| Discriminator number of layers | 2 |
| Discriminator hidden layer size | 100 |
| Discriminator steps | 2 |

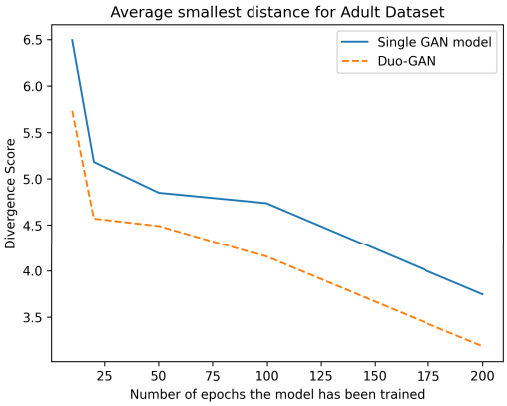



**FIGURE 3. Divergence score for adult dataset.**

The experiments were conducted for 10, 20, 50, 100 and 200 epochs of training of both SingleGAN and Duo-GAN. However some training epochs the SingleGAN models did not generate positive samples during the epochs that they were allowed to train. Therefore, for some experiments only the results where the single generator created positive samples are shown.

### A. SYNTHETIC DATA VALIDATION

In Figure 3 and Figure 4 we show the results of the *Divergence* score over the number of epochs that the model was trained for the Adult (Figure 3) and the Credit Card Fraud Detection (Figure 4) datasets. The lower the *Divergence* value, the higher the resemblance between the original and synthetic data.

The results seen for the Adult dataset show that with both approaches, the *Divergence* score decreases as the number of epochs increases. This result indicates that both approaches are capturing the patterns that exist in the dataset. However, when we compare the relative behavior of the approaches, we can see some differences. In particular, looking at the *Divergence* curve of Duo-GAN, it is possible to see that it attains lower values, which means that it can better capture the original dataset's properties.

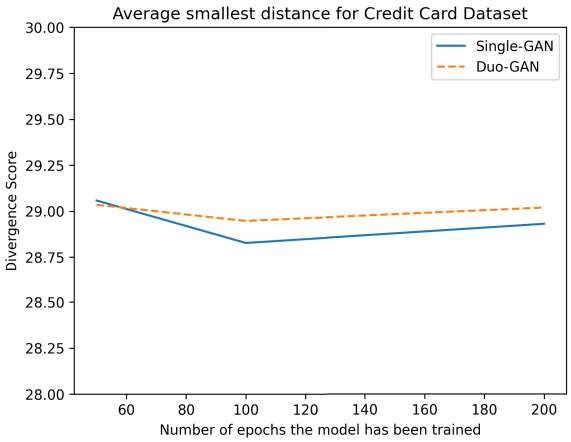



**FIGURE 4.** Divergence score credit card dataset.

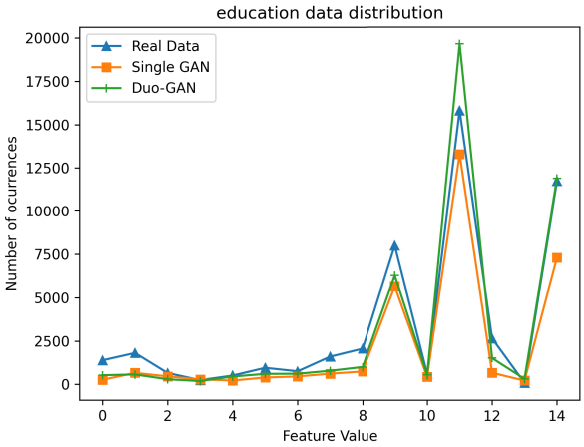



**FIGURE 5.** Distribution of values for education categorical feature.

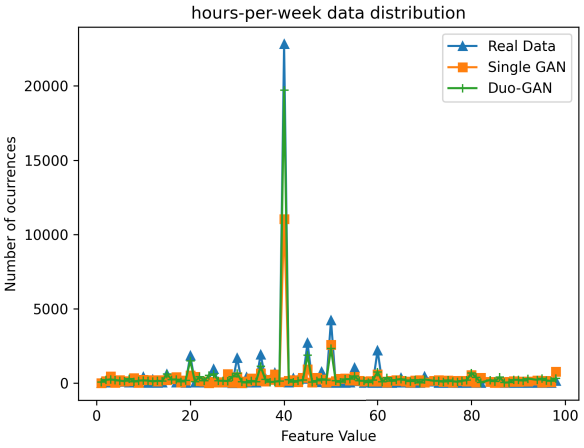



**FIGURE 6.** Distribution of values for hours-per-week continuous feature.

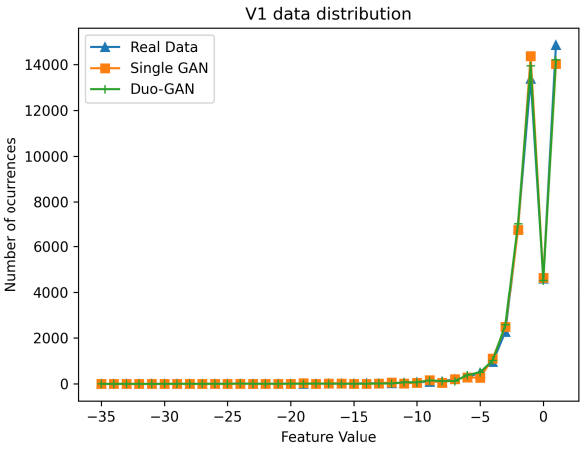



**FIGURE 7.** Distribution of values for the V1 continuous feature for the credit card fraud detection.

Concerning the Credit Card Fraud Detection dataset, we can see that that there are noticeable differences between both approaches. The first difference is the smaller range of variation in the *Divergence* values. While Duo-GAN seems to present higher values, the difference between the scores is very small (∼0.2) so we cannot make any conclusion over which one presents a more desirable behavior. The second one is that for the Single GAN model, we can only generate positive, i.e., fraudulent, instances after 50 epochs. Given that the number of positive instances in the dataset is small (less than 1%), the model will rarely take acquaintance with them during training. Concerning Duo-GAN (dashed line), we can generate positive and negative instances much earlier. Another interesting aspect is that the *Divergence* score is higher for both approaches, with values of around 29 when the maximum possible is 31. This result is understandable because all features in this dataset are continuous, making it harder to have an exact match for these features.

For both datasets we can conclude that both generator models are not simply memorizing the original data, but instead replicating patterns learned from the real-word datasets.

In order to continue assessing the theoretical quality of the data, we look to the distribution of values for the real datasets compared to the generator models. Figures 5,6 presents the results for two features for the Adult dataset, one categorical and the other continuous. The results show that the approaches can capture the general distribution of the features.

Figure 7 shows the distribution of the feature V1 of the Credit Card Fraud Detection. In this case, it is possible to see that we can capture the real-world data distribution without having an exact match between the samples in the datasets.

Finally, it is important to see if the generator models can generate synthetic datasets that keep the correlations that exist with the feature in the real-world data. Figure 8 presents the correlation matrices for the Adult dataset and Figure 9 presents them for the Credit Card dataset. Looking at the results for the adult dataset, Duo-GAN can better capture the existing correlations between the variables than the Single GAN approach. For example, let us consider the relationship between age and income. Looking at the cell that shows the correlation between these two variables in Figure 8a we can see a medium to high correlation (∼0.6). Looking at the same cell in Figure 8c we can see that the correlation still exists but to a small degree (∼0.4). However, looking at the correlation

**TABLE 2.** F1-score for machine learning models trained on real data and synthetic data for the Adult dataset. The best results attained with models trained with synthetic data are highlighted in bold.

| Classifier | Approach | 10 Epochs | 20 Epochs | 50 Epochs | 100 Epochs | 200 Epochs | Real Data Score |
|---|---|---|---|---|---|---|---|
| Adaboost | Single GAN | 0.0461 | 0.0187 | 0.0771 | 0.2396 | 0.3098 | 0.6858 |
| | Duo-GAN | 0.5961 | 0.6424 | **0.6481** | 0.6400 | 0.6460 | |
| DecisionTree | Single GAN | 0.2353 | 0.2386 | 0.0771 | 0.2396 | 0.3098 | 0.6238 |
| | Duo-GAN | 0.5732 | 0.5884 | **0.6292** | 0.6009 | 0.5361 | |
| XGBoost | Single GAN | 0.0596 | 0.0599 | 0.0771 | 0.2396 | 0.3098 | 0.7143 |
| | Duo-GAN | 0.6152 | 0.6529 | **0.6739** | 0.6429 | 0.6051 | |

**TABLE 3.** F1-score for machine learning models trained on real data and synthetic data for the credit card dataset. The best results attained with models trained with synthetic data are highlighted in bold.

| Classifier | Approach | 50 Epochs | 100 Epochs | 200 Epochs | Real Data Score |
|---|---|---|---|---|---|
| Adaboost | Single GAN | 0.0000 | 0.0000 | 0.0045 | 0.8571 |
| | Duo-GAN | **0.8528** | 0.8346 | 0.7740 | |
| DecisionTree | Single GAN | 0.0000 | 0.0000 | 0.0045 | 0.7817 |
| | Duo-GAN | 0.5808 | **0.7904** | 0.7024 | |
| XGBoost | Single GAN | 0.0000 | 0.0000 | 0.0045 | 0.9062 |
| | Duo-GAN | **0.8636** | 0.8496 | 0.7687 | |

**TABLE 4.** Precision recall area under the curve for machine learning models trained on real data and synthetic data for the Adult dataset. The best results attained with models trained with synthetic data are highlighted in bold.

| Classifier | Approach | 10 Epochs | 20 Epochs | 50 Epochs | 100 Epochs | 200 Epochs | Real Data Score |
|---|---|---|---|---|---|---|---|
| Adaboost | Single GAN | 0.2803 | 0.3671 | 0.1770 | 0.2700 | 0.4615 | 0.8107 |
| | Duo-GAN | 0.6468 | **0.6998** | 0.6976 | 0.6874 | 0.6711 | |
| DecisionTree | Single GAN | 0.3427 | 0.3615 | 0.1770 | 0.2700 | 0.4615 | 0.6685 |
| | Duo-GAN | 0.6442 | 0.6480 | **0.6749** | 0.6515 | 0.5920 | |
| XGBoost | Single GAN | 0.2499 | 0.3898 | 0.1770 | 0.2700 | 0.4615 | 0.8346 |
| | Duo-GAN | 0.6373 | 0.7193 | **0.7300** | 0.6851 | 0.6084 | |

**TABLE 5.** Precision recall area under the curve for machine learning models trained on real data and synthetic data for the credit card dataset. The best results attained with models trained with synthetic data are highlighted in bold.

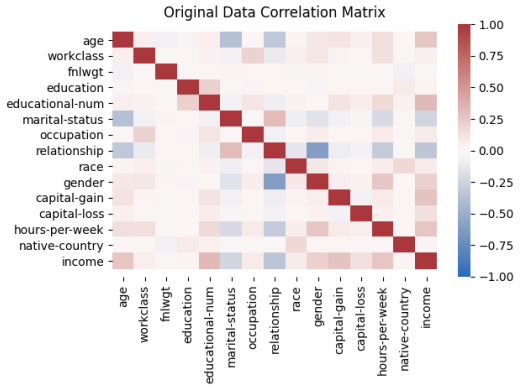
| Classifier | Approach | 50 Epochs | 100 Epochs | 200 Epochs | Real Data Score |
|---|---|---|---|---|---|
| Adaboost | Single GAN | 0.0168 | 0.0540 | 0.0056 | 0.8879 |
| | Duo-GAN | **0.8346** | 0.8236 | 0.8318 | |
| DecisionTree | Single GAN | 0.0168 | 0.0540 | 0.0056 | 0.7832 |
| | Duo-GAN | 0.6403 | **0.7933** | 0.7262 | |
| XGBoost | Single GAN | 0.0168 | 0.0540 | 0.0056 | 0.8993 |
| | Duo-GAN | **0.8526** | 0.8446 | 0.8487 | |

value between age and income in Figure 8b we can see that the value is 0.0, indicating no correlation between these two variables. The same pattern is visible for other pairs such as educational-num and income, marital-status and income, relationship and marital-status.

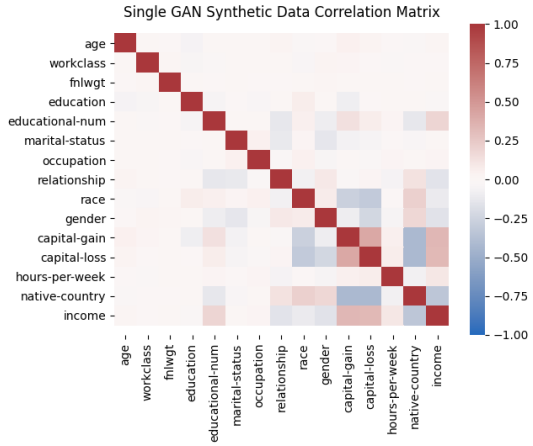
When looking at the correlation results for the Credit Card Fraud Detection the differences between Duo-GAN and the Single GAN are even more accentuated. A perusal of Figure 9c reveals that our proposal can capture most of the existing correlations in the Original dataset (Figure 9a). On the contrary, the Single GAN (Figure 9b) cannot capture any of the existing correlations. When using the Single GAN approach, we lose all the existing correlations between the features.

### B. SYNTHETIC DATA UTILITY EVALUATION

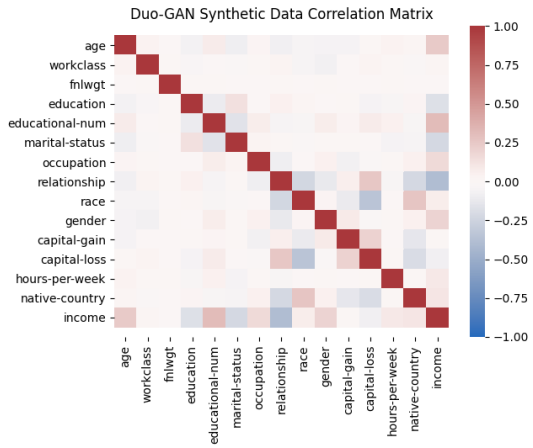
In Tables 2 and 3 we present the $F_1$ scores obtained during classification, and in Tables 4 and 5 we present the Precision-Recall Area Under the Curve. In general, it is possible to




(a) Correlation Matrix for the Real Dataset



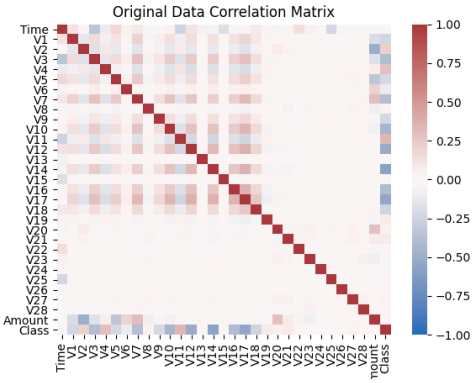

(b) Correlation matrix for the Synthetic Adult Dataset generated by a Single GAN



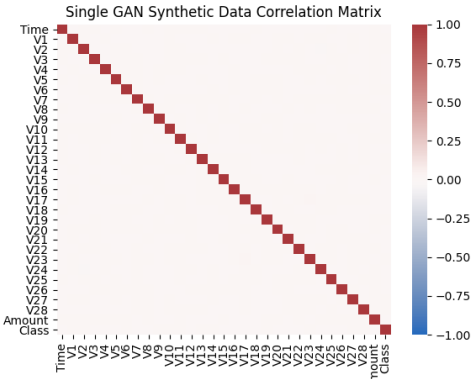

(c) Correlation matrix for the Synthetic Adult Dataset generated by Duo-GAN

**FIGURE 8.** Correlation analysis for the adult dataset using real dataset, the single and Duo-GAN synthetic dataset.

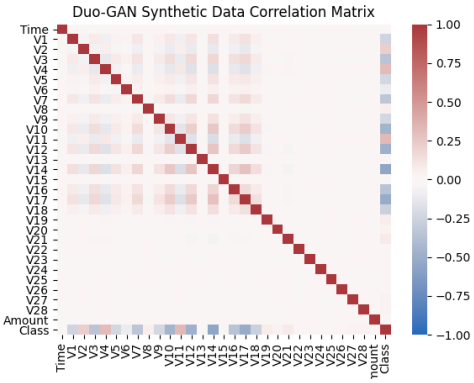
see that Duo-GAN obtains the best results for both problems for all of the classifiers. Another interesting aspect is the observable degeneration in the quality of data for generator models trained for longer periods. This can be seen in the results for the models trained for 100 or 200 epochs obtain




(a) Correlation Matrix for the Real Dataset




(b) Correlation matrix for the Synthetic Credit Card Dataset generated by a Single GAN




(c) Correlation matrix for the Synthetic Credit Card Dataset generated by Duo-GAN

**FIGURE 9. Correlation analysis for the credit card dataset using real dataset, the single and Duo-GAN synthetic dataset.**

worse performance than models trained for shorter periods of time. This result might indicate that the generative models are learning properties that do not exist in the original data. Our approach's results are positive, particularly in the Credit Card dataset, given its unbalanced nature.

It is essential to compare the results obtained with those of the classifiers trained and tested in the actual data because it allows us to quantify how much utility is lost when using synthetic data. Looking at the results presented in Table 2, and

**TABLE 6. Precision and recall breakdown by class for data generated by model trained for 50 epochs for the Adult dataset.**

| Classifier | Generator | Class | Precision | Recall |
|---|---|---|---|---|
| Adaboost | Single GAN | Negative | 0.7245 | 0.8504 |
| | | Positive | 0.0777 | 0.0375 |
| | Duo-GAN | Negative | 0.9078 | 0.8504 |
| | | Positive | 0.5623 | 0.7576 |
| DecisionTree | Single GAN | Negative | 0.7179 | 0.7133 |
| | | Positive | 0.1628 | 0.1659 |
| | Duo-GAN | Negative | 0.8967 | 0.7133 |
| | | Positive | 0.5415 | 0.7283 |
| XGBoost | Single GAN | Negative | 0.7222 | 0.8199 |
| | | Positive | 0.1030 | 0.0615 |
| | Duo-GAN | Negative | 0.9106 | 0.8199 |
| | | Positive | 0.6055 | 0.7562 |

**TABLE 7. Precision and recall breakdown by class for data generated by model trained for 50 epochs for the credit card dataset.**

| Classifier | Generator | Class | Precision | Recall |
|---|---|---|---|---|
| Adaboost | Single GAN | Negative | 0.9908 | 1.0000 |
| | | Positive | 0.0000 | 0.0000 |
| | Duo-GAN | Negative | 0.9983 | 1.0000 |
| | | Positive | 0.8485 | 0.8116 |
| DecisionTree | Single GAN | Negative | 0.9918 | 0.9872 |
| | | Positive | 0.0777 | 0.1159 |
| | Duo-GAN | Negative | 0.9985 | 0.9872 |
| | | Positive | 0.7582 | 0.8406 |
| XGBoost | Single GAN | Negative | 0.9911 | 0.9997 |
| | | Positive | 0.5556 | 0.0362 |
| | Duo-GAN | Negative | 0.9983 | 0.9997 |
| | | Positive | 0.9113 | 0.8188 |

considering the best performing classifier, i.e. the XGBoost, we can see that the most considerable difference in F1-score is 0.12, and the lowest difference is about 0.05. In what concerns the best performing classifier for the Credit Card dataset (Table 3), the most significant difference is about 0.14, and the smallest is about 0.05.

Tables 6 and 7 detail the results of the different classifiers, showing the Precision and Recall Area Under Curve scores by class for data generated by models trained for 50 epochs for both datasets. In these tables we can see how Duo-GAN has better performance especially for the Positive class. For the Adult dataset we can see that XGBoost achieves precision of 0.0615 for the Positive class when trained with data from a Single GAN generator. However, when we use our approach, Duo-GAN, the precision results for the Positive class improve to 0.7562. Looking at the results obtained in the Credit Card dataset, we can see an increase in the performance gap. Looking at the results of the XGBoost model, the recall for the Positive class and a Single GAN generator model is 0.0362, whilst for Duo-GAN as the generator model the recall for the same class is 0.8188.

Looking at the results obtained by the proposed approach, we can see that both of them outperformed the Single GAN generator models. Having a model learn the class conditional distribution for both positive and negative instances creates better quality data which is reflected mainly when

it comes to the utility of the data, which is reflected on the classification results. The difference in performance for the positive instances highlights how a double generator model creates better quality data by improving the quality of the underrepresented class.

It is also important to refer that our model takes around 30 minutes to train a model for 200 epochs the Adult dataset and 60 minutes for the Credit Card dataset. Since both datasets have roughly the same number of rows this indicates that the number of features of the dataset is what drives the training time up.

## VI. CONCLUSION

With the increased volume of monetary transactions in online commerce, more and more companies, regardless of the industry they belong to or the service they provide, rely on ML techniques to automatically process large volumes of data. However, companies are subject to tight regulations regarding data privacy and have to follow strict anonymization procedures. These procedures raise some issues, namely the delay in the deployment of solutions and the fact that the anonymization might remove important details hindering accurate predictions. Synthetic datasets that contain the same statistical properties, such as distributions and correlations between features, can help organisations speed up their development process for effective solutions.

In this work we propose Duo-GAN as a framework for generating highly unbalanced synthetic datasets. Duo-GAN is composed of two GANs, one that generates positive records, and one that generates negative ones. This allows each of the GAN to learn the class conditional distribution for each of the classes which counters the over exposure to negative records that happens in single GAN generator models, that leads to poor quality positive records which in turn results in poor performance in classification tasks.

To validate our approaches we designed an experimental methodology that includes the validation of synthetic data, as well as a measurement of its utility. In concrete, we measure the divergence between the real and synthetic datasets, we analyse the distribution and the correlation matrices of features and compare them with the ones in real data. To measure the utility retained by synthetic data we use it to train a group of classifiers and then test them on real data. We also compare the performance to classifiers trained and test on real data.

The results show that our approach, Duo-GAN not only outperforms single GAN generator models, but also generates high quality synthetic datasets that that allow the development of ML models that attain a performance similar to the ML models trained on real datasets. Specifically, the best model trained with synthetic data generated by Duo-GAN obtains classification performance with a gap of 5% in $F_1$ score.

In what concerns future work, we intended to perform a study on how different configurations of Duo-GAN impact not only the quality and utility of the synthetic data, but also the time and computational resources needed to train the models. Moreover, given that the *Divergence* metric has some issues dealing with the continuous features, we will research possible modifications and improvements to it. Another direction of this work is concerned with the comparison of our proposal with others from the literature, e.g., [9].

## REFERENCES

[1] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *Proc. Int. Conf. Comput. Netw. Informat. (ICCNI)*, 2017, pp. 1–9.
[2] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private data release via Bayesian networks," *ACM Trans. Database Syst.*, vol. 42, no. 4, pp. 1–41, Oct. 2017.
[3] R. Torkzadehmahani, P. Kairouz, and B. Paten, "DP-CGAN: Differentially private synthetic data and label generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Mar. 2020, pp. 1–7.
[4] E. Choi, S. Biswal, B. A. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete electronic health records using generative adversarial networks," *CoRR*, vol. abs/1703.06490, pp. 1–20, Mar. 2017.
[5] K. Bhanot, M. Qi, J. S. Erickson, I. Guyon, and K. P. Bennett, "The problem of fairness in synthetic healthcare data," *Entropy*, vol. 23, no. 9, p. 1165, Sep. 2021.
[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory Cryptography*, S. Halevi and T. Rabin, Eds. Berlin, Germany: Springer, 2006, pp. 265–284.
[7] T. Zhang, T. Zhu, R. Liu, and W. Zhou, "Correlated data in differential privacy: Definition and analysis," *Concurrency Comput., Pract. Exper.*, vol. 2020, Sep. 2020, Art. no. e6015.
[8] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Mach. Intell.*, vol. 2, no. 6, pp. 305–311, Jun. 2020.
[9] M. Rasouli, T. Sun, and R. Rajagopal, "FedGAN: Federated generative adversarial networks for distributed data," 2020, *arXiv:2006.07228*.
[10] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," 2018, *arXiv:1811.11264*.
[11] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," *Proc. VLDB Endowment*, vol. 11, no. 10, pp. 1071–1083, Jun. 2018.
[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014, *arXiv:1406.2661*.
[13] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.
[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.
[15] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.
[16] L. Xu and K. Veeramachaneni, "Synthesizing tabular data using generative adversarial networks," *CoRR*, vol. abs/1811.11264, pp. 1–12, Nov. 2018.
[17] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," *CoRR*, vol. abs/1907.00503, pp. 1–15, Jul. 2019.
[18] B. Vega-Márquez, C. Rubio-Escudero, J. C. Riquelme, and I. Nepomuceno-Chamorro, "Creation of synthetic data with conditional generative adversarial networks," in *Proc. 14th Int. Conf. Soft Comput. Models Ind. Environ. Appl. (SOCO)*, F. M. Álvarez, A. T. Lora, J. A. S. Muñoz, H. Quintián, and E. Corchado, Eds. Cham, Switzerland: Springer, 2019, pp. 231–240.
[19] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," 2018, *arXiv:1806.03384*.
[20] J. P. Reiter, "Using cart to generate partially synthetic, public use microdata," *J. Off. Statist.*, vol. 21, pp. 441–462, Sep. 2003.
[21] Y. Sun, A. Cuesta-Infante, and K. Veeramachaneni, "Learning vine copula models for synthetic data generation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 5049–5057.
[22] H. Huang, Z. Li, R. He, Z. Sun, and T. Tan, "IntroVAE: Introspective variational autoencoders for photographic image synthesis," *CoRR*, vol. abs/1807.06358, pp. 1–20, Jul. 2018.

[23] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 835–851.
[24] A. Salim, "Synthetic patient generation: A deep learning approach using variational autoencoders," *CoRR*, vol. abs/1808.06444, pp. 1–11, Aug. 2018.
[25] S.-C. Li, B.-C. Tai, and Y. Huang, "Evaluating variational autoencoder as a private data release mechanism for tabular data," in *Proc. IEEE 24th Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Dec. 2019, p. 198.
[26] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, pp. 1–16, Nov. 2016.
[27] A. Ali-Gombe and E. Elyan, "MFC-GAN: Class-imbalanced dataset classification using multiple fake class generative adversarial network," *Neurocomputing*, vol. 361, pp. 212–221, Oct. 2019.
[28] P. Shamsolmoali, M. Zareapoor, L. Shen, A. H. Sadka, and J. Yang, "Imbalanced data learning by minority class augmentation using capsule adversarial networks," *Neurocomputing*, vol. 459, pp. 481–493, Oct. 2021.
[29] P. Mrozek, J. Panneerselvam, and O. Bagdasar, "Efficient resampling for fraud detection during anonymised credit card transactions with unbalanced datasets," in *Proc. IEEE/ACM 13rd Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2020, pp. 426–433.
[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[31] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Tech. Rep., 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[32] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2015, pp. 159–166.
[33] A. Dal Pozzolo, O. Caelen, Y.-A. L. Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, 2014.
[34] 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 Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018.
[35] F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "SCARFF: A scalable framework for streaming credit card fraud detection with spark," *Inf. Fusion*, vol. 41, pp. 182–194, May 2018.
[36] B. Lebichot, Y.-A. Le Borgne, L. He-Guelton, F. Oblé, and G. Bontempi, "Deep-learning domain adaptation techniques for credit cards fraud detection," in *Recent Advances in Big Data and Deep Learning*, L. Oneto, N. Navarin, A. Sperduti, and D. Anguita, Eds. Cham, Switzerland: Springer, 2020, pp. 78–88.
[37] F. Carcillo, Y.-A. Le Borgne, O. Caelen, and G. Bontempi, "Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization," *Int. J. Data Sci. Anal.*, vol. 5, no. 4, pp. 285–300, Jun. 2018.
[38] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Inf. Sci.*, vol. 557, pp. 317–331, May 2021.

**FRANCISCO FERREIRA** received the B.Sc. and M.Sc. degrees in informatics engineering from the University of Coimbra, in 2018 and 2021, respectively. He is currently a Former Student of the University of Coimbra. His master's thesis work delved on creating machine learning based methodologies for generating synthetic records used for accurately training fraud detection models. His research interests include artificial intelligence, machine learning, computer vision, and information theory.

**NUNO LOURENÇO** received the Ph.D. degree in information science and technology from the Department of Informatics Engineering, University of Coimbra, in 2016. He is the Current coordinator of the Evolutionary and Complex Systems (ECOS) Group. He is a member of the Centre for Informatics and Systems of the University of Coimbra (CISUC), since 2009. Formerly, he was appointed as a Senior Research Officer at the University of Essex, U.K. He is currently an Assistant Professor with the Department of Informatics Engineering, University of Coimbra. He is the Co-Creator of Structured Grammatical Evolution, Probabilistic Grammatical Evolution, and DENSER, a novel approach to automatically design deep artificial neural networks using evolutionary computation. He has authored or coauthored more than 60 articles in journals and top conferences from the evolutionary computation and artificial intelligence areas and he has been involved as a researcher in 13 projects (national and international). His main research interests include bio-inspired algorithms, optimization, and machine learning. He is a member of the Program Committee of GECCO, PPSN, EuroGP; a member of the Steering Committee of EuroGP; and an Executive Board Member of SPECIES. He served as the Chair in the main conferences for the Evolutionary Computation field, namely EuroGP 2020 and 2021 as the Program Chair, and PPSN 2018 and EuroGP 2019 as the Publication Chair.

**BRUNO CABRAL** received the Ph.D. degree (Hons.) from the University of Coimbra (UC), in 2009. He has been an Adjunct Associate Teaching Professor at Carnegie Mellon University (CMU), USA, and was a Faculty of the dual-degree master's in software engineering (MSE). He was the Co-ordinator of the Master Program in Software Engineering at the UC and a member of the Scientific Council of the Department of Informatics Engineering, UC. He is currently an Assistant Professor with the Informatics Engineering Department, UC, teaching courses on software architecture, business intelligence and enterprise systems development. He was and is the PI on several EU and state funded projects, and also by the European Space Agency, Microsoft Research, Novabase, Critical Software, and among others. He has authored over 40 peer-reviewed articles. His current research goal is to create new technology to simplify the development of safer, faster, more resilient and energy efficient systems for processing large amounts of data.

**JOÃO PAULO FERNANDES** graduated (Hons.) in mathematics and computer science from the University of Minho, Portugal, in 2004. He received the Ph.D. degree from the University of Minho, in 2009. He is currently an Associate Professor with the Department of Informatics Engineering, Faculty of Engineering, University of Porto (DEI-FEUP) and a member of the Artificial Intelligence and Computer Science Laboratory (LIACC). He has founded/coordinate(d) and/or is involved in projects and initiatives, such as GreenHub, Green Sw Lab, GreenHaskell, and Sustainable. His research is generally focused on the rigorous analysis and transformation of software, with the general goal of optimizing its non-functional properties while ensuring its functional correctness. His research results have been described in 60+ publications in high quality international conferences and journals, such as the IEEE Transactions on Software Engineering, *EMSE*, *JSS*, ICSME, SANER ou SLE, and include a Data Showcase Special Mention Award at MSR 2019. He has integrated 40+ program committees for international conferences and journals, having been awarded the Outstanding Reviewer Award at SBES 2019.