UNIVERSIDADE D
COIMBRA

Francisco Martins Ferreira

Anonymizing Private Information
From Noise to Data

Dissertação no âmbito do Mestrado em Engenharia Informática, ramo de Sistemas Inteligentes orientada pelo Professor Doutor Nuno Lourenço, Professor Doutor Bruno Cabral e Professor Doutor João Paulo Fernandes apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia.

Junho de 2021

This page is intentionally left blank.

Faculty of Sciences and Technology

Department of Informatics Engineering

Anonymizing Private Information

From Noise to Data

Francisco Martins Ferreira

Dissertation in the context of the Master in Informatics Engineering, Specialization in Intelligent Systems advised by Prof. Nuno Lourenço, Prof. Bruno Cabral and Professor João Paulo Fernandes and presented to the Faculty of Sciences and Technology / Department of Informatics Engineering.

June 2021

1 2 9 0

UNIVERSIDADE D
COIMBRA

This page is intentionally left blank.

Abstract

In the Information Age data has become more important for all types of organizations. The information carried by large datasets habilitates the creation of intelligent systems that overcome inefficiencies and create a safer and better quality of life. Because of this, organizations have come to see data as a competitive advantage.

Fraud Detection solutions are one example of intelligent systems that are highly dependent on having access to large amounts of data. These solutions receive information about monetary transactions and classify them as legitimate or fraudulent in real time. This field has benefitted from higher availability of data, allowing the application of Machine Learning (ML) algorithms that leverage the information in datasets to finding fraudulent activity in real-time.

In a context of systematic gathering of information, privacy dictates how data can be used and shared, in order to protect the information of users and organizations. In order to retain the utility of data, a growing amount of effort has been dedicated to creating and exploring avenues for privacy conscious data sharing.

Generating synthetic datasets that carry the same information as real data allows for the creation of ML solutions while respecting the limitations placed on data usage. In this work, we introduce Duo-GAN and DW-GAN as frameworks for synthetic data generation that learn the specificities of financial transactions data and generate fictitious data that keeps the utility of the original collections of data. Both these frameworks use two generators, one for generating fraudulent instances and one for generating legitimate instances. This allows each generator to learn the distribution for each class, avoiding the problems created by highly unbalanced data. Duo-GAN achieves positive results, in some instances achieving a disparity of only 4% in F_1 score between classifiers trained with synthetic data and classifiers trained with real data and both tested on the same real data. DW-GAN presents positive results too with disparity of 3% in F_1 score in the same conditions.

Keywords

Machine Learning, Generative Adversarial Networks, Synthetic Data, Tabular Data, Fraud Detection

This page is intentionally left blank.

Resumo

Na Idade da Informação os dados tornaram-se mais importantes para todos os tipos de organizações. A informação contida pelos grandes datasets permite a criação de sistemas inteligentes que ultrapassam ineficiências e criam qualidade de vida melhor e mais segura. Devido a isto, as organizações começaram a ver os dados com uma vantagem competitiva.

As soluções de Deteção de Fraude são exemplos de sistemas inteligentes que dependem do acesso a grandes quantidades de dados. Estas soluções recebem informação relativas a transações monetárias e atribuem classificações de legítimas ou fraudulentas em tempo real. Este é um dos campos que beneficiou da maior disponibilidade de dados, sendo capaz de aplicar algoritmos de Machine Learning que utilizam a informação contida nos datasets para detetar atividade fraudulenta em tempo real.

Num contexto de agregação sistemática de informação, a privacidade dita como os dados podem ser utilizados e partilhados, com o objetivo de proteger a informação dos utilizadores de sistemas e de organizações. De forma a reter a utilidade dos dados, uma quantidade crescente de esforço tem sido dispendido em criar e explorar avenidas para a partilha de dados respeitando a privacidade.

A geração de dados sintéticos que contém a mesma informação que os dados reais permite a criação de soluções de Machine Learning (ML) mantendo o respeito pelas limitações colocadas sobre a utilização de dados.

Neste trabalho introduzimos Duo-GAN e DW-GAN como frameworks para geração de dados sintéticos que aprendem as especificidades dos dados de transações financeiras e geram dados fictícios que retém a utilidade das coleções de dados originais. Ambos os frameworks utilizam dois geradores, um para gerar instâncias fraudulentas e outro para gerar instâncias legítimas. Isto permite que cada gerador aprenda a distribuição de cada uma das classes, evitando assim os problemas criados por datasets desiquilibrados. O Duo-GAN atinge resultados positivos, em certos casos atingindo uma disparidade de apenas 4% no F_1 score entre classificadores treinados com dados sintéticos e classificadores treinados com dados reais, e ambos testados nos mesmos dados reais. O DW-GAN também apresenta resultados positivos, com disparidade de 3% no F_1 score para as mesmas condições.

Palavras-Chave

Machine Learning, Generative Adversarial Networks,Dados Sintéticos, Dados Tabulares, Deteção de Fraude

This page is intentionally left blank.

Acknowledgements

Thank you to my loving parents and sisters, my beautiful girlfriend and the boys.

I'd also like to extend a thank you to my advisors for the help and guidance.

This work is partially funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020. and by the CMU|Portugal project CAMELOT (POCI-01-0247-FEDER-045915).

This page is intentionally left blank.

Contents

This page is intentionally left blank.

Acronyms

- **ANN** Artificial Neural Networks. 8, 9, 10
- **DP** Differential Privacy. 2, 15, 17, 18
- **FFNN** Feed Forward Neural Networks. 8, 9, 16, 17
- **GAN** Generative Adversarial Networks. 2, 3, 10, 11, 12, 15, 16, 17, 18, 19, 21, 22, 24, 26, 49, 50, 51
- **LSTM** Long Short Term Memory. 10, 16
- **MCA** Main Components Analysis. 28
- **ML** Machine Learning. iii, v, 2, 5, 6, 25, 37, 50
- **PATE** Private Aggregation of Teacher Ensembles. 17, 18
- **PRC** Precision Recall Curve. 13
- **RNN** Recurrent Neural Networks. 10
- **VAE** Variational AutoEncoders. 9, 10, 15, 18
- **VGM** Varitional Gaussian Mixture. xiii, 16, 17
- **WGAN** Wasserstein Generative Adversarial Networks. 18, 22, 49, 50

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

In recent years, the increase in online commerce and the growing number of monetary transactions fostered extensive data collection. The gathered 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 [2]. Fraudsters' techniques are usually transversally applicable through industries and services, making a specific fraud detection solution of one organisation adequate for other organisations with similar characteristics.

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.

Research communities and other organizations have allocated a growing amount of effort on making privacy-respecting data release possible so that the utility of the data is not lost [10, 40, 46].

1.1 Motivation

The interdisciplinary project CAMELOT is led by the Feedzai company and involves the Carnegie Mellon University, Universidade de Coimbra, Faculdade de Ciências da Universidade de Lisboa and Instituto Superior Técnico. In the CAMELOT context, we are developing a framework for the transfer of information while respecting individuals' and organizations' privacy. The Feedzai company works on fraud detection, and it has clients operating in several industries, with some clients that are even direct competitors. The transversality of fraud techniques could give Feedzai a competitive advantage by employing their data between different clients. However, due to tight contractual obligations regarding the data, Feedzai is prohibited from using client data on different projects. Two problems arise from this: the *cold start problem* where the classification models do not have sufficient information to make accurate predictions; the gathering of data that allows the solutions to perform at an acceptable threshold is a lengthy and costly process that leads to the delay of deployment of the solutions. In this context, we are researching how we can transfer knowledge from one client to another in the same industry while adhering

to strong privacy standards, more concretely how to release data and retain utility.

One approach for achieving responsible data publication is embedding privacy into datasets. By manipulating the information in the records of each dataset, it is possible to guarantee higher degrees of confidentiality. The state-of-the-art approach to privacy is Differential Privacy (DP), originally introduced by Dwork et al. [17], where controlled amounts of noise are added to the records in ways that minimize privacy loss. Phan et al. [34] have observed that it is possible to maintain training efficiency and model quality while applying DP. The authors in [47] observed that DP is used with the assumption that data is not correlated, however correlation in datasets is expected which weakens privacy and leads to unexpected data leakage.

Another option is the creation of artificial data. 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 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 [32, 43, 46]. One generative model that gained relevance in recent years in creating synthetic samples is the Generative Adversarial Networks (GAN). GANs usually combine two neural networks called a Generator and a 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 synthetic samples and the actual input data. This adversarial training process allows for the generation of quality samples after the training process.

The main objective of this work is the creation of a framework for the creation of synthetic data. This synthetic data would then be used for the development of fraud detection solutions without compromising the legal obligations that data is subject to. Creating these solutions requires high quality data, and because of this the data generation employed must ensure that the data resembles the original data. For this we propose Duo-GAN, an architecture that allows the generation of quality synthetic data by employing a generator for each class, learning the class conditional distribution and countering the effects of the lack of balance that is usually found in datasets for fraud detection.

We demonstrate how Duo-GAN is capable of capturing the distributions and the correlations present in the original data, and that Duo-GAN is capable of producing good synthetic data, that when used for training classifiers and tested on real data shows performance gaps ranging from 15% to 4% in F_1 score to the results of classifiers trained with the original data. DW-GAN is also shown to be able to capture the distributions and correlations present in the real data, and DW-GAN data when used to train classifiers obtains gaps of F_1 score performance in the range of 5% to 3%, however for datasets with categorical features the performance gap goes up to $\sim$ 35%. Beyond that, we show that our proposed approaches' data is more effective for training classifiers than data generated by a Single GAN.

1.2 Contributions

This work has resulted in several contributions, which are listed below.

- A compreensive literature review on the field of synthetic data generation focusing particularly on Generative Adversarial Networks and tabular data. In this review, we analyze several works on this field using different architectures ranging from Bayesian Networks to Variational AutoEncoderss and Generative Adversarial Networks, while discussing their results and shortcomings.
- A framework for the evaluation of synthetic data that includes theoretical evaluation and utility validation. We propose a metric for measuring the distance between samples in the synthetic dataset and the real one as well as analysing intrinsic qualities of the data such as the statistical distribution of values and correlations.
- Duo-GAN, a GAN based framework for generating quality synthetic data designed to deal with the specifications of fraud detection data, mainly the unbalanced nature of these datasets. This approach is based on using a generator for each class which allows the generators to only learn the distribution of their given class.

1.3 Document Structure

This document is organized as follows: Chapter 2 introduces the relevant concepts for this work; Chapter 3 presents a literature review on the field of synthetic data generation; In Chapter 4 we present Duo-GAN, our approach for generating quality synthetic data; Chapter 5 details our experimental setup such as datasets, how we evaluate the quality of the generated data and the experimental design; in Chapter 6 the results of the experiments are exposed and discussed; Chapter 7 concludes this work with a reflection on the main points from this study and lays down the future work.

This page is intentionally left blank.

Chapter 2

Background

This chapter introduces the relevant concepts for this work. This includes the concepts on Machine Learning, and Privacy. After this introduction, the current literature on synthetic data generation as a method for private data release will be presented and analyzed.

Machine Learning

Machine Learning (ML) algorithms give computers the ability to make decisions by relying on observed data as learning experience. These algorithms are based on mathematical models that are capable of improving their decisions based on their exposure to more data. Their recent success relies on the unprecedented availability of data and computing resources [38].

There is also diversity in ML algorithms usefulness, as different models target different problems, such as:

- **classification problems**, where data instances are to be assigned to previously established class labels.
- **regression problems** that have the goal of finding a mapping function between input and output.
- **generation of data** where new pieces of data that resemble the original data are created.

In this section ML concepts relevant to this work will be presented.

2.0.1 Types of Learning

According to the problem and the available data, the learning methodology changes. The relevant paradigms for this work are:

Supervised Learning

In this type of learning, the datasets' target labels are known and necessary for training [37]. The standard training algorithm for supervised learning is based on iteratively delivering

data samples to a model and the corresponding desired label, called ground truth. Then the model will calculate its output for the given sample, and based on the error, calculated using a loss function between the output and the ground truth, the model will readjust itself. After training, it is expected that the model will be capable of correctly assigning class labels to the input samples.

Unsupervised Learning

Unlike supervised learning, unsupervised learning does not need knowledge of a target label [37]. Many times this paradigm is used for finding groups with distinct characteristics in the dataset instead of assigning rigid classifications.

2.0.2 Types of models

Discriminative Models

Discriminative Models learn the posterior $p(y|x)$, with x as the input vector and y the class labels or learn a direct map from input x to the class labels [30]. In simpler terms, this means that these models attempt to learn the decision boundary between different classes in a dataset. Decision Trees and Logistic Regression are examples of discriminative models.

Generative Models

Generative Models learn the distribution of data, such as $p(x)$, and $p(x, y)$, with x being the input vector and y as the class label when applicable. This means they learn the data distribution of classes. In classification problems, these models use the learned distribution with the Bayes rule,$P(A|B) = \frac{P(B|A)P(B)}{P(B)}$, to calculate $p(y|x)$ and picking the most likely label y [30]. Generative models' knowledge of data distribution also habilitates them to create new instances of data that follow the same distribution. The family of generative models include Variational AutoEncoders and Generative Adversarial Networks.

2.0.3 Architectures and Algorithms

In ML there are several well known architectures and algorithms to create intelligent models. In this section different ML models are presented.

Bayesian Networks

Bayesian Networks represent a set of variables and their conditional dependencies through directed acyclical graphs. Friedman et al. [19] define a Bayesian Network for U, a random set of variables, as a pair $B = (G, \theta)$ where G is a directed acyclical graph whose vertices correspond to the random variables $X_1, \ldots, X_n$, and whose edges represent direct dependencies between the variables. G assumes each variable X_i is independent of its nondescendants given its parent in G. The second parameter θ represents the set of parameters that quantifies the network. For each possible value of x_i of X_i and Π_{x_i} of Π_{X_i}, where Π_{X_i} represents the set of parents of X_i in G, the network contains the parameter

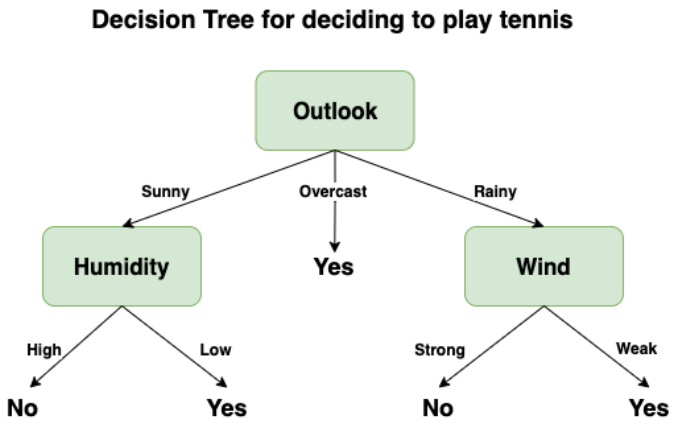


Figure 2.1: Decision Tree to decide on playing Tennis based on the weather. The records for the construction of this tree had 4 features, Outlook, Temperature, Humidity and Wind. If we look at the branch on the left we see that the second node is Humidity but on the right it is Wind, being that for different values in the same feature, the most informative feature may be different. Temperature is not a node on the tree, meaning that not all features are used for building a decision tree

$\theta_{x_i,\Pi_{x_i}} = P_B(x_i|\Pi_{x_i})$. A Bayesian Network defines the joint probability distribution over U given by Equation 2.1.

$$P_B(X_1, \ldots, X_n) = \Pi_{i=1}^{n} P_B(X_1|\Pi_{X_i}) \tag{2.1}$$

In order to classify, the network computes the Bayes rule for the possible classes and chooses the most likely. Zhang et al. [46] propose a methodology for using these Networks for generative purposes.

Logistic Regression

A logistic regression uses the Logistic Equation to solve binary problems by modeling the probabilities of possible outcomes. The output of the model is calculated by $p(y = 1|x, w) = \frac{1}{1+e^{-w^T x}}$ with x as the input vector and parameters w. These parameters are trained in order to minimize the difference between the output of the model and the desired output.

Decision Trees

Decision Trees are classification models that create a clear path for classification or regression. Decision trees are composed of nodes, representing data features, with branches, representing the possible values, or intervals of values, for said features. The nodes in the tree are sorted by their discriminative power, however nodes at the same height are not necessarily the same feature, two nodes at the same depth may represent a different feature as seen in Figure 2.1. In order to obtain a classification for input vector x, the tree must be transversed from top to bottom until finding a node with no branches. In order to choose the right feature for each node several metrics can be used, such as the Gini Index or cross-entropy.

Random Forests

Random Forests are an ensemble classifier consisting of a large number of Decision Trees. Individual trees are built with recourse to bootstrapping a dataset, and using a smaller arbitrary set of features than the original data. Each tree is then built to fit the bootstrapped dataset. This process is repeated n times to generate the forest. For classification, the Random Forrest is given the feature vector x as input and each tree makes their classification as a vote. Classification is decided by the class label that has received more votes.

Adaptative Boosting

Adaptative Boosting(AdaBoost) introduced by Freund and Schapire [18] is an ensemble method for classification. It relies on shallow Decision Trees, called weak learners, to create a strong learner that aggregates the output of the weak learners to make classifications. In AdaBoost each weak learner has a different weight on the final aggregation, based on how it performs on training data. If each weak learner is slightly better than guessing then the strong learner will converge to the desired solution.

XGBoost

Similarly to AdaBoost, XGBoost is an ensemble method using Decision Trees as weak learners, it is also an extension over Gradient Boosting, where instead of an aggregate vote, each weak learner contributes a small part for classification. On top of Gradient Boosting, XGBoost applies regularization, which is intended to curb the tendency decision trees show to overfit their training data.

Feed Forward Neural Networks

Feed Forward Neural Networks (FFNN) are one of the simplest Artificial Neural Networks (ANN) architectures, and a supervised learning model. These are composed of at least three layers, an input layer, an hidden layer and an output layer, the structure of a FFNN can be seen in Figure 2.2. The neurons between layers are usually fully connected. The output of a neuron is the sum of all previous neurons multiplied by the weights that each neuron has assigned to each previous neuron summed to a bias values, and then followed by an activation function φ, that can be either linear or non-linear. The output y of a neuron is calculated by the Equation 2.2, where n is the number of inputs into the neuron, w is the weights assigned to previous neurons, x is the value of the inputs into the neuron, and b is the neuron's bias.

$$y = \varphi(\sum_{i=1}^{n} w_{ki} x_i + b) \tag{2.2}$$

The output y of the neurons in the output layer is the predicted value of the input vector x. The model's loss can be calculated as $l(t, y)$ with t as the desired output. The objective of the model is to minimize the loss function by tuning w and b.

Adjusting the weights is a two step process, the model first calculates the gradients of the loss function with respect to the FFNN weights, and then updates the weights according to the gradients. The most common tuning algorithm for ANNs is backpropagation.

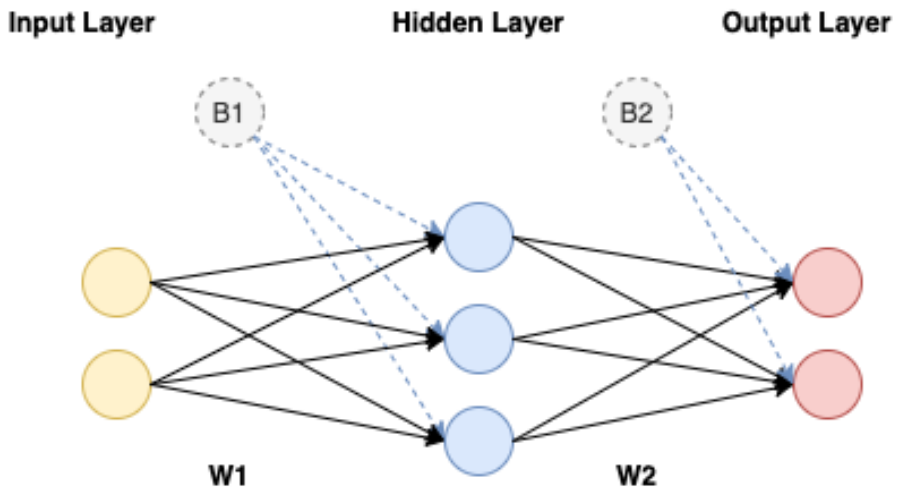


Figure 2.2: Feed Forward Neural Network with an input layer, one hidden layer and one output layer, including a bias

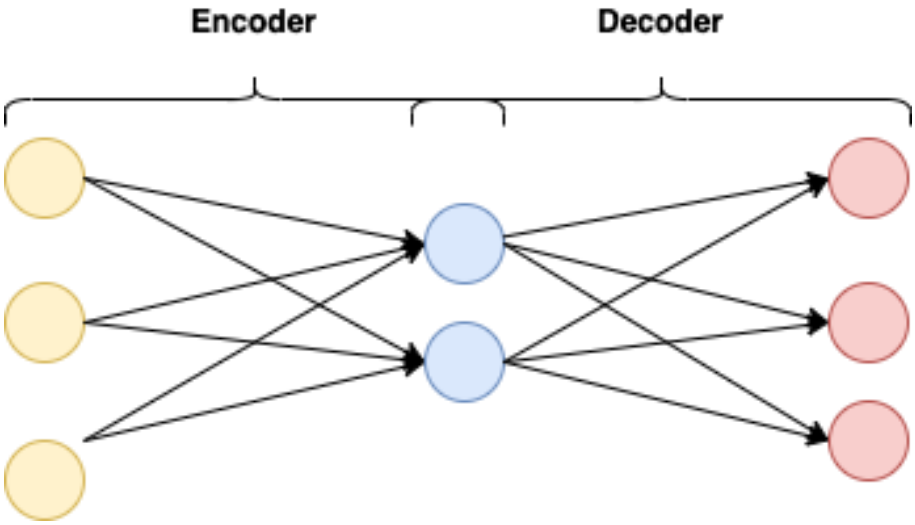


Figure 2.3: Representation of an AutoEncoder

AutoEncoders

AutoEncoders are a type of ANN composed of two elements, an Encoder and a Decoder. The Encoder will embed the input data into a lower dimensionality feature vector in the latent space, while the Decoder takes lower dimensionality feature vector and reconstructs it to the size of the original input data, the point where Encoder and Decoder meet is called the bottleneck. AutoEncoders are trained so that the Decoder is able to reconstruct the Encoded form of the input data. The goal of this model is to learn a data representation with lower dimensionality and simultaneously learn how to decode the lower dimensionality representation of data. With AutoEncoders being a type of ANN they are trained similarly to FFNNs, but minimizing the reconstruction loss, using the input data as the desired output.

Variational AutoEncoders (VAE) are an alternative type of AutoEncoder. VAE leverage the possibility that from one random variable z with one distribution we can create another

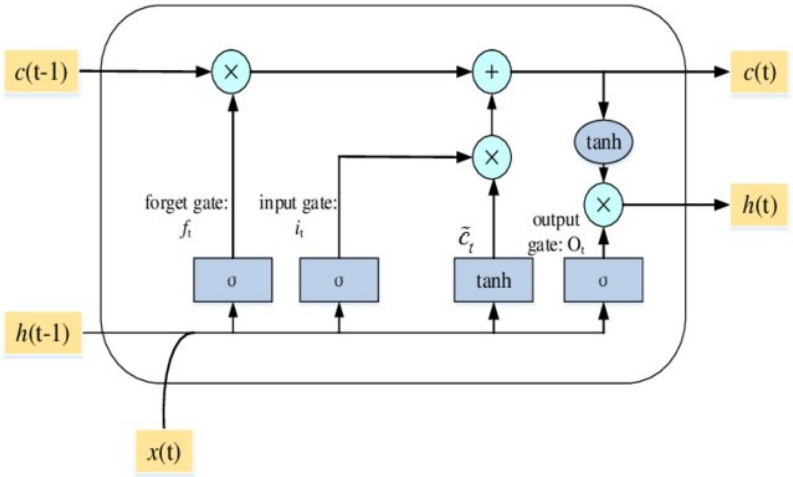


Figure 2.4: LSTM Unit structure. C is the Cell state,x is the input vector and h is the output of the cell. Source: [45]

random variable with a different distribution [15]. In this architecture the bottleneck does not represent a point in the latent space, but a probabilistic distribution over the latent space. Input data is encoded into as a distribution over the latent space, after this a point is sampled from the given distribution. The Decoder will decode the input into the original dimensionality. Similarly to the AutoEncoders, VAE are trained to minimize reconstruction error.

Long Short Term Memory Networks

Recurrent Neural Networks (RNN) are a class of ANN with a sense of temporal shift and memory. The output of a neuron in a RNN is not only influenced by its inputs and their assigned weights but also their previous outputs.

Long Short Term Memory (LSTM) Networks are a class of RNN that are designed for memorizing information for arbitrarily sized input sequences. These Networks are based on the LSTM unit. Figure 2.4 shows the structure of a LSTM unit.

These networks can be trained using optimization algorithms such as *GradientDescent* combined with *backpropagation through time*.

Generative Adversarial Networks

Generative Adversarial Networks (GAN)'s are a Machine Learning architecture for training generative models proposed in 2014 by Goodfellow et al. [20]. This architecture is composed of two models, a generator G and a discriminator D. Figure 2.5 presents the structure of the training process of a GAN.

The Generator G input is an arbitrary fixed-length vector z composed of randomly sampled values from a given distribution, usually Gaussian or uniform, and the output is a sample with the same structure of the ones in the real data. After training it is expected that G produces samples that contain close to the same information of the ones in the real data.

The Discriminator model, D, receives samples of data and outputs a single scalar, corre-

sponding to a classification of real or synthetic. It is trained with samples from the original data and also samples that have been output by G.

Both models are trained simultaneously. G generates a batch of samples, and along with a batch of real samples, they are provided to D. The Discriminator then classifies each of the samples into real or synthetic and the results are backpropagated in D while the parameters of G are frozen. After this, the Generator generates samples once again and uses them to update itself based on the output of the newly updated Discriminator for the samples while D's parameters are frozen. The training algorithm of the GAN is presented in Algorithm 1. G is trained to minimize $log(1 - D(G(z)))$, which means minimizing the classification error of samples as real or synthetic. D is trained so that $D(x)$, the probability of the sample x belonging to the original data and not to synthetic data, is correctly predicted.

Algorithm 1: Generative Adversarial Network training algorithm
G is the generator model with parameters θ_g, D is the discriminator model with parameters θ_d, O is the original dataset, n is the number of training epochs, b is the batch size for training, k is the number of training iterations in an epoch resulting from number of samples in O divided by b

for *1..n* **do**
  for *1..k* **do**
    Sample b samples from O
    Generate b samples by generating noise z and obtaining $G(z)$
    Update D by ascending it stochastic gradient:
    $\nabla_{\theta_d} \frac{1}{b} \sum_{i=1}^{b} [log(D(x_i)) + log(1 - D(G(z_i)))]$
  end
  Generate b samples by generating noise z and obtaining $G(z)$
  Update G by descending its stochastic gradient:
  $\nabla_{\theta_g} \frac{1}{b} \sum_{i=1}^{b} log(1 - D(G(z_i)))$
end
Result: We can discard D and keep G as a trained generator

The two models are adversarial since the success of one model depends on the failure of the other. In the case that G creates a sample that is not recognized as synthetic by D, its parameters will not change much, or not at all. However, D's parameters will be modified a substantial amount. In the opposite case, the change of parameters will also be opposite. Both models play a minimax game with value function $V(D, G)$:

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[logD(x)] + E_{z \sim p_z(z)}[log(1 - G(D(x))] \tag{2.3}$$

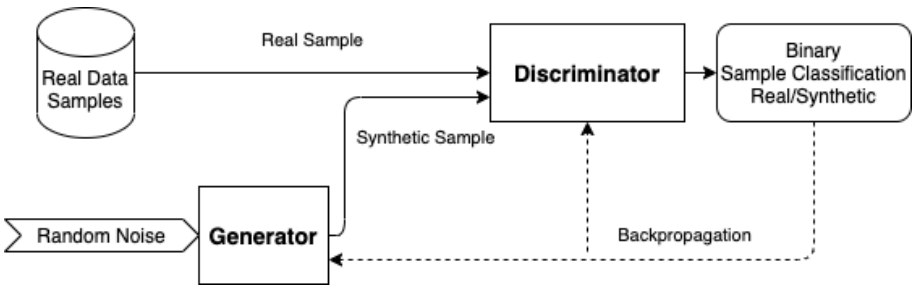


Figure 2.5: Structure of a Generative Adversarial Network.

GAN can be extended to a conditional model if both G and D are conditioned on some extra information e [29]. This information can be class labels. The extra information e is now also used as input for both G and D. The models still play a minimax game but now the values function $V(D, G)$ is as follows:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x|e)] + E_{z\sim p_z(z)}[log(1 - G(D(x|e))] \tag{2.4}$$

2.0.4 Evaluating models

Machine Learning algorithms need to be evaluated in order to guarantee they achieve their goals. In classification problems evaluation is based on comparing the ground truth labels included in the dataset to the labels attributed by the algorithm. In our context, we will work with binary classification problems, so the metrics used are specific for this binary context. These metrics are based on the concept of positive and negative variable, which in the context of this work, a positive value means a transaction is fraudulent, and negative means a transaction is legitimate.

When a classification model identifies a record as positive and the ground truth is also a positive value, then it is considered a True Positive TP. If the classification falsely identifies a record as positive it is considered a False Positive FP. When the classification model correctly labels a record as negative, it is considered a True Negative TN. If the classification falsely classifies a record it is a False Negative FN.

The traditional method for calculating performance is accuracy, defined as

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{2.5}$$

Accuracy is however a poor metric for unbalanced datasets, looking at a scenario where records are 95% positive, a model that classifies every instance as positive has 95% accuracy however it is not useful.

The metrics used to evaluate the classifiers, both rely on Precision and Recall. Recall calculates the proportion of correctly identified positive instances, calculated as follows:

$$recall = \frac{TP}{TP + FN} \tag{2.6}$$

Precision identifies how many positive records identified are actually positive, it is calculated as follows:

$$precision = \frac{TP}{TP + FP} \tag{2.7}$$

F_1 score

The F_1 score is the harmonic mean between precision and recall calculated by

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} \tag{2.8}$$

Precision and recall are inherently more informative metrics, especially in unbalanced contexts, which leads to the F_1 score being a more valid indicator of classifier behavior for unbalanced datasets [21].

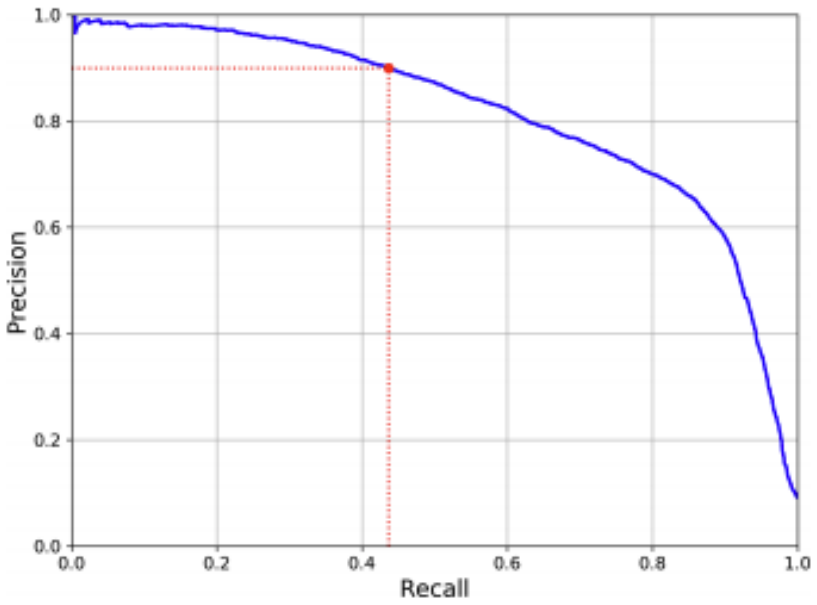


Figure 2.6: Precision Recall Curve. Calculating the area under the blue curve gives us the Precision Recall Area Under the Curve. Source: [21]

Precision Recall Curve

The Precision Recall Curve (PRC) shows the tradeoff between precision and recall for different classification thresholds. By plotting the precision and recall at different threshold values and calculating the area under the curve, we can obtain a single value for quantifying, classifiers performance. In Figure 2.6 we can see an example of a Precision Recall Curve.

This page is intentionally left blank.

Chapter 3

Related Work

This work intends to create a framework for the creation of quality synthetic data. With this goal in mind, this Chapter presents a review of the relevant literature into synthetic data generation.

Zhang et al. [46] propose PrivBayes as a method for private data release. PrivBayes is an algorithm for creating differentially private Bayesian Networks capable of synthesizing private high dimensional data. This algorithm calculates a set of low-dimensional marginals P of a dataset D to be injected with noise, subsequently a low-degree Bayesian Network, using the noisy marginals, approximates the distribution of D. The use of noise in the marginals P ensures Differential Privacy (DP)[17]. By adding noise to the low-dimensional marginals instead of the high-dimensional datasets, PrivBayes attempts to avoid the curse of dimensionality. For the lowest privacy budgets tested, classifiers trained on PrivBayes generated data obtained poor results when compared to classifiers trained on real data, however for larger privacy budgets, the results got significantly better.

Variational AutoEncoders have also been researched as a solution to the problem at hand. These architectures have mostly been researched with image generation in mind [22, 41], however some studies on tabular data have also been developed [24, 26] with satisfying results. These architectures offer the advantage of being able to generate conditional samples, that is, generating samples with a fixed value for a variable or groups of variables, this would allow for balancing datasets. There has been some work with successful results into incorporating DP into VAE [9].

Generative Adversarial Networks rose to prominence for their successful results on synthetic image generation. Radford et al. [35] have demonstrated how a GAN based framework can generate visual data that replicates the original dataset distribution, introducing Deep Convolutional GAN, DCGAN. This work also explores vector arithmetic on the input vectors. By averaging the outputs of samples of similar visual concepts such as "woman" or "smiling" we can obtain the input vector that generates such features, allowing for manipulation of the output to obtain different visual concepts, as can be seen in Figure 3.1. This shows that a GAN is able to learn complex concepts such as "glasses" and reproduce them or remove them.

There are however some common issues with using GANs as generator models. Mode collapse is the most widely reported problem in the literature. It happens when a GAN only outputs samples belonging to a small number of modes in the original distribution. This happens because the GAN Generator discovers some points in the data distribution which the Discriminator has trouble identifying as real or synthetic, leading the Generator

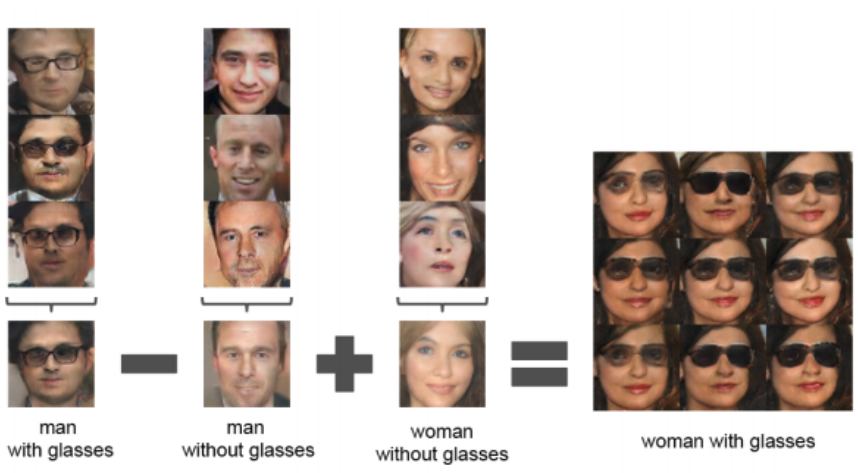


Figure 3.1: Vector arithmetic for visual concepts. The input vectors of each image in the column are averaged. Arithmetic was then performed, producing the vector Y, which is fed into the Generator and produces the shown image. Source: [35]

to only produce samples belonging to that small distribution in order to minimize its loss. There have been several works that propose solutions for mode collapse [8, 36, 39].

Research has also explored the utility of GANs for tabular data. There are several differences between image data, where GAN have most been employed, and tabular data. Tabular data regularly contains features with continuous, binary and discrete values in the same tables, also with mixed data types. These factors make the generation of tabular data a non-trivial task [44].

One approach proposed for generating synthetic tabular data is table-GAN [32]. This method is based on the aforementioned DCGAN, and uses a Convolutional Neural Network for the Discriminator and a Neural Network with de-convolutional layers. The Generator creates synthetic records that contain all the features of the data, but the label is assigned by a Classifier, a network that is trained with the ground truth label of the real data. Before training the GAN, the data is processed to be compatible with the model, the numerical features are scaled to [-1,1], and the categorical features are attributed a value in the [-1,1] range. These transformations to the data are reverted after the generation of the samples. The results in the work show that classifiers trained on synthetic data generated by table-GAN obtain comparable performance to classifiers trained on real data.

Other approach proposed to synthesize tabular data was introduced by Xu and Veeramachaneni [42] called Tabular GAN (TGAN). TGAN uses an LSTM as the Generator and a fully connected FFNN as the Discriminator. In order to deal with the specificities of tabular data the authors use pre-processing techniques. For binary features noise is added to the values, categorical features are converted to one-hot encoding and numerical features undergo mode-specific normalization. In order to apply this method for a feature, a Varitional Gaussian Mixture (VGM) [4] is trained to learn the distribution and each value in the feature vector is encoded according to the distribution. More details on the specificities of mode-specific normalization can be found in Figure 3.2. These transformations are all reversed after the generation process in order to return samples that are structurally identical to the real data samples.

The work compared the utility retained in the data by different synthesizing methods by comparing different classifiers performance when trained with synthetic data. TGAN

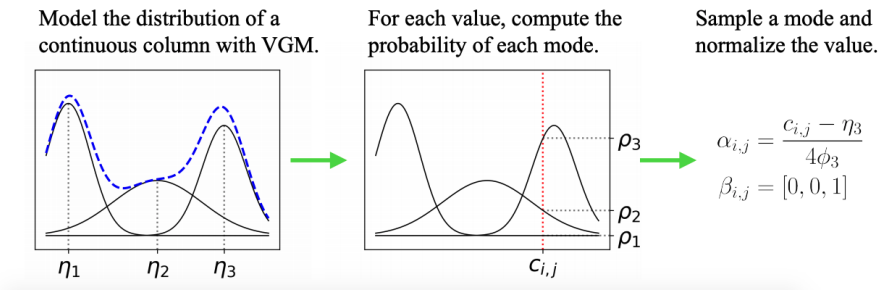


Figure 3.2: Mode-specific normalization. $c_{i,j}$ is the jth value of the i feature vector, and with a VGM modelling the distribution of the feature values, modes η_n and standard deviations ϕ_n. For $c_{i,j}$ we compute the mode that is most likely to have produced it. Based on this we compute $\alpha_{i,j}$ and $\beta_{i,j}$ and then encode them as shown above. Source: [44]

outperformed all methods using accuracy and the F1 metric, in all instances, having a performance gap of 5.7% to classifiers trained on real data. Building upon the positive results of TGAN, Xu et al. [44] introduced Conditional TGAN (CTGAN). This model is a conditional GAN that uses a fully connected FFNN as Generator while using another fully connected FFNN as Discriminator and applies the same pre-processing techniques as TGAN. The results of this work are equally positive to the ones presented in the work that introduced TGAN.

Similar to fraud detection, the medical field is also extremely limited when sharing health records. Choi et al. [10] propose medGAN to circumvent these limitations. This method achieves positive results, with human doctors being unable to distinguish synthetic samples from real samples, except for several outliers. medGAN however is susceptible to membership inference attacks, when the attacker can deduce if a specific patient was used for training the model if it has significant amounts of knowledge about him. This method is also limited in the types of variables it can reproduce, those being binary and count variables.

The Post Processing Theorem, as defined in Theorem 1, is at the base of several approaches for ensuring that GANs are DP-compliant. This theorem allows to move the burden of privacy exclusively to the Discriminator of the GAN, since the Generator has no contact with the training data.

Theorem 1 (Post Processing)**.** *Let M be an (ε, δ)-differentially private algorithm and let $f : O \to O^{'}$ where $O^{'}$ is any arbitrary space. Then $f \circ M$ is (ε, δ)-differentially private*

Jordon et al. [23] introduces Private Aggregation of Teacher Ensembles (PATE)-GAN, which builds on the PATE framework introduced in [31] and applies it to the GAN architecture. In PATE-GAN the dataset is partitioned into k disjoint sets and the Discriminator D is replaced by k teacher-discriminators. In each training iteration the teachers receive samples from the generator G and instances of real data from their partition, and are trained to classify them correctly as real or synthetic. The Student S then receives samples generated by G and is trained to minimize its error compared to the teacher vote Aggregation with added noise. While G is trained to minimize the error in relation to the output of S. The Precision Recall Area Under the Curve (PRAUC) obtained by synthetic datasets is significantly lower than the same metric for real datasets. For example for the Credit Card dataset, classifier models trained on real data obtained an average PRAUC of

~ 0.70 while the models trained on synthetic data obtained an average of ~ 0.33 PRAUC. In low-dimensional datasets PATE-GAN has similar performance to a non-private GAN however when dealing with higher dimensionality datasets its performance decreases significantly. The authors believe this to be due to the fact that S is trained exclusively on synthetic samples, which makes it so that in the early iterations of training the generated samples are less likely to be close to the original dataset distribution. S is limited to being trained by generated samples because otherwise it would compromise DP.

Other works have built upon the Post Processing Theorem and proposed using a privacy accountant to guarantee DP in synthetic data. Liu et al. [27] introduce PPGAN which is composed of a standard GAN architecture coupled with Moments Accountant for the privacy budget applied to the Discriminator. The results show that for smaller privacy(20 or under) budgets PPGAN has problems generating quality data. In [40] the authors propose DPCGAN which employs the Renyi Differential Privacy Accountant. In this framework once again the privacy accountant is only applied to the Discriminator. The authors obtained satisfying results for a single-digit privacy budget, although it was for image data, tabular data remains to be tested.

Arjovsky et al. [1] have proposed Wasserstein GAN (WGAN) as a way to improve the training process and provide better data generation capabilities. In this model, the GAN Discriminator is turned into a Critic that quantifies the difference between the real distribution and the synthetic one. This is achieved by training a neural network that calculates the Wasserstein Distance, the Critic, between the two distributions, real and generated, which will provide the Generator with a meaningful metric of how it behaves, how far the distribution of the generated data is from the real distribution, and then attempting to minimize the distance between the distributions, intsead of minimizing the probability of the generated data being classified as fake which is a more arbitrary metric. The authors claim that using a Critic avoids mode collapse, and lends more versatility to the architectures and models that can be used as Generator. In [3] WTGAN is introduced, as an adaptation of the aforementioned TGAN to a Wasserstein Generative Adversarial Networks (WGAN) model. The work does not show specific scores in classification taks, but shows that the synthetic data generated by WTGAN used for classification tasks has similar scores to classifiers trained on real data.

Discussion

Several works that aim at creating Differentially Private datasets ,such as [28, 34], were omitted from the above section since using synthetic datasets satisfies the privacy requirements for this work. The loss of information resulting from enforcing DP makes the GAN models that enforce it, such as PPGAN and PATEGAN, not be considered further in this work.

After this review we can take some conclusions about the synthetic data generation field. The most promising results are either VAE-based [9] or GAN-based [42**?**]. These works have shown good synthetic data utility. The most adequate approach seems to be a GAN-based framework for data synthesis, because of the promising results presented in this section, and the large breadth of work that has been done with the goal of generating data that retains utility.

The processing of data before using it for training generative models is an important step for assuring that the data that will result from generator models is of high quality. Looking at the works that target the generation of synthetic data we can see that TGAN and all

the methods that are extensions of it use much more sofisticated data processing than the one used by table-GAN, mainly for categorical features.

This work's main contribution is applying the research that has been done and described above in the synthetic data generation field and apply it to the fraud detection field. Despite a large amount of work using GANs for synthetic data generation and some for specific industries, such as the aforementioned medGAN, there is no work focusing on fraud detection data. This industry requires solutions tailored to it, given the specificities of the data, such as the different types of features and the small number of fraudulent instances compared to the number of legitimate instances.

This page is intentionally left blank.

Chapter 4

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 non-tabular data [44]. It is common for tabular datasets to have columns that have non-gaussian distributions, which may lead to vanishing gradients in normalization 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 present class, which becomes too similar to the dominant class. This poor representation has consequences on classifying tasks, given that the positive instances are of poor quality and too similar to the negative ones, it becomes a difficult task for classifiers to differentiate between the two different classes.

4.1 Duo-GAN

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 4.1. 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.

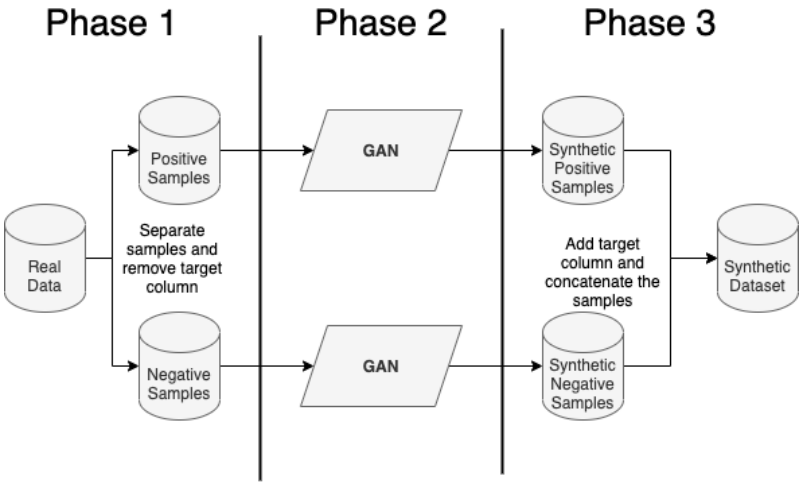


Figure 4.1: Duo-GAN - Proposed Generative Model for accurate synthetic data generation of heavily unbalanced datasets.

4.2 DW-GAN

Keeping in mind the problems that are inherent to the GAN architectures, such as mode collapse and unstable training, we propose a solution that swaps out the GAN and instead uses WGAN as the Generator Model. We have named it DW-GAN, **D**uo **W**asserstein **GAN**. Since WGAN are a slight modification over the common GAN architecture, this change does not present any need for adaptation of the approach we have described, which means the description in the above section still apply for DW-GAN.

Chapter 5

Experimental Study

In this chapter we detail the methodology used for our work, as well as the experimental design used to gauge the capabilities of the proposed approach.

5.1 Methodology

The experimental pipeline is detailed in Figure 5.1, and consists of three main steps: 1) data generation; 2) validation of the synthetic data; 3) synthetic data utility validation.

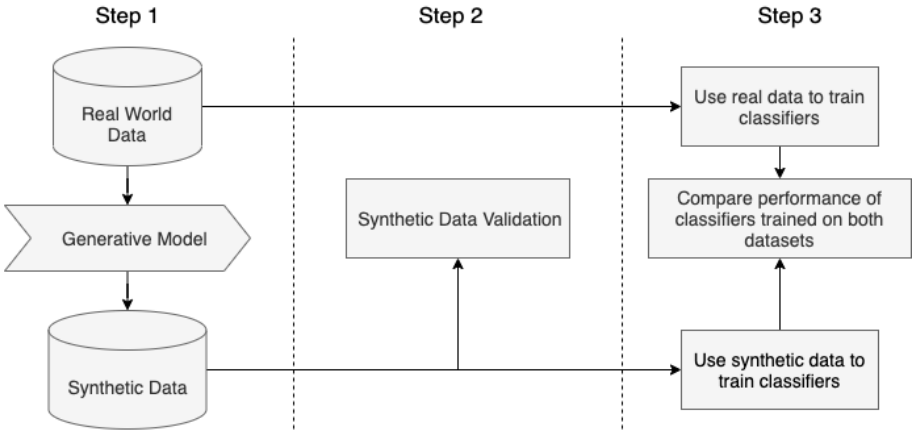


Figure 5.1: Methodology pipeline.

In Step 1, we use real-world, highly unbalanced, public datasets to train models to create synthetic data, following the strategy we propose, Duo-GAN, and then generate a synthetic dataset using the trained models. We detail this stage in section 5.1.1.

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 5.1.2 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 extracted from the real dataset. Comparing the

behaviour 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 5.1.3 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 see which approach produces data with higher quality.

5.1.1 Synthetic Data Generation

As sketched in Step 1 of Figure 5.1 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, T_{test} is set aside until Step 3, described in Section 5.1.3. 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 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 actual dataset.

5.1.2 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 5.1, we aim 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 as defined in:

$$singledivergence = \sum_{i=1}^{n} neq(d[i], s[i])) \tag{5.1}$$

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 (5.1), 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 5.2

$$Divergence = \sum_{i=1}^{j} \frac{1}{j} min(divergences[i])) \tag{5.2}$$

where j is the number of samples, and $divergences[i]$ is a vector of *singledivergence* 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.

This metric gives us two 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 matrixes for both the original and the synthetic datasets. The method selection is dependent on the type of the features. For datasets that include categorical features *Spearman Rank Correlation* will be used, and *Pearson Correlation* for datasets that do not.

5.1.3 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 5.1 illustrates this process. Datasets used for training:

- 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. 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 and Multi-Layer Perceptron using the default parameters defined in the library [33].

5.2 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.

5.2.1 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 [16] 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. The features in this dataset have complex distributions, per example the age feature as can be seen in Figure 5.2.

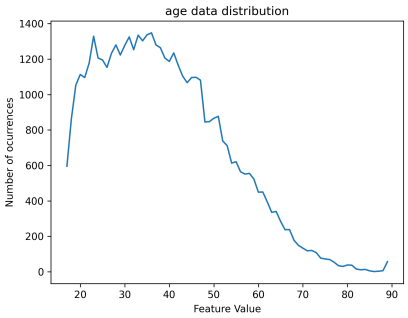


Figure 5.2: Distribution of values for the age feature in the Adult dataset.

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 [5, 6, 7, 11, 12, 13, 14, 25] 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.

5.2.2 Single and Double Generator Architectures

This experiments aims 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 this experiments we follow the steps layed down in 5.1 to generate and evaluate the quality of the data.

The following experiments will be executed:

1. **Single GAN and Duo-GAN** For this experiment we will use TGAN [42] as the single generator, and Duo-GAN as described in Chapter 4 with TGAN as the generators. We will test this approach on both the **Adult** and the **Credit Card Fraud Detection** datasets. The configuration for TGAN used can be seen in Table 5.1.

Table 5.1: Configuration used for TGAN.

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

2. **Single WGAN and DW-GAN** For this experiment we will use WTGAN [3] as the single generator, while using DW-GAN as described in Section 4.2 with WTGAN as the generators. We will test this approach on both the **Adult** and the **Credit Card Fraud Detection** datasets. The configuration for WTGAN is the same as the one used for TGAN, shown in Table 5.1.

5.2.3 Feature Engineering

The engineering of features that add information to datasets, and augment the performance of intelligent systems that detect transaction fraud has become an essential part of the development of these solutions. We need to take this into account when applying Duo-GAN to datasets.

The main question in regards to Duo-GAN and feature engineering is whether data generation is better applied before or after the feature engineering process. Figure 5.3 displays the creation of an augmented dataset.

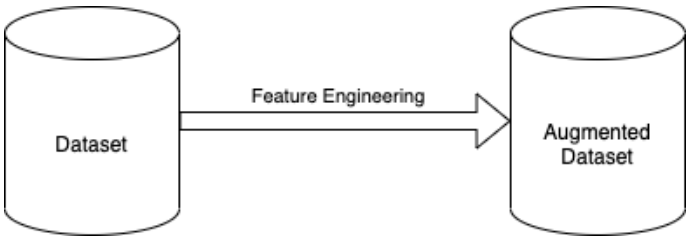


Figure 5.3: Creation of an augmented dataset using feature engineering.

In Figure 5.4 we can see the pipeline of generating synthetic data and then applying a feature engineering routine. For this approach to be successful it requires that Duo-GAN has the capability of faithfully reproducing the underlying characteristics in the data to allow feature engineering to create information rich features.

In Figure 5.5 we can see the timeline of generating a synthetic dataset based on an already

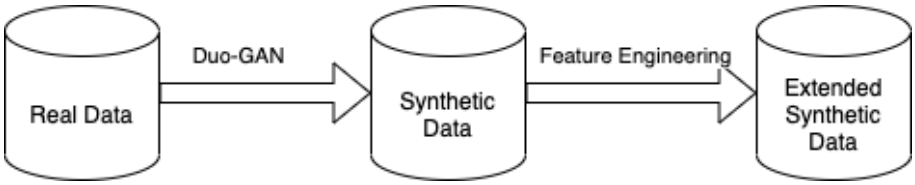


Figure 5.4: Creation of an extended synthetic dataset with feature engineering after synthetic data generation.

extended real dataset. Using Duo-GAN to create a synthetic dataset that already includes engineered features poses its own set of challenges as well. Not only will it have to generate data with more features, these extra features may have complex distribution that are harder to replicate.

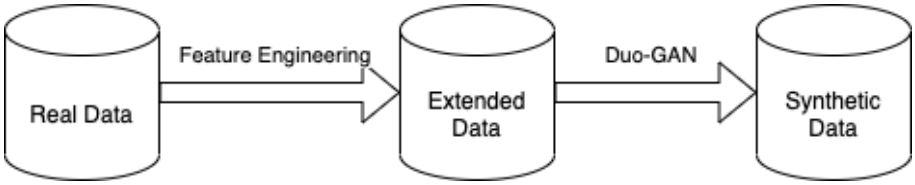


Figure 5.5: Creation of an augmented dataset using feature engineering and then creating an extended synthetic dataset.

In this experiment we aim to perform a preliminary analysis of the relationship between the artificial data generated by Duo-GAN and feature engineering. For this we create a feature engineering routine that will be applied to the real and synthetic datasets. It will allow us to analyze how much utility is added by the inclusion of feature engineering in the process and look into where it better impacts the quality of generated data.

We will take the **Adult** dataset and extend it with the 3 first components resulting from a Main Components Analysis (MCA) applied to the dataset, named p1,p2 and p3. We will use Duo-GAN and DW-GAN as the generator models, and the baseline for this experiment will be the classification results on the extended real dataset.

This page is intentionally left blank.

Chapter 6

Results

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

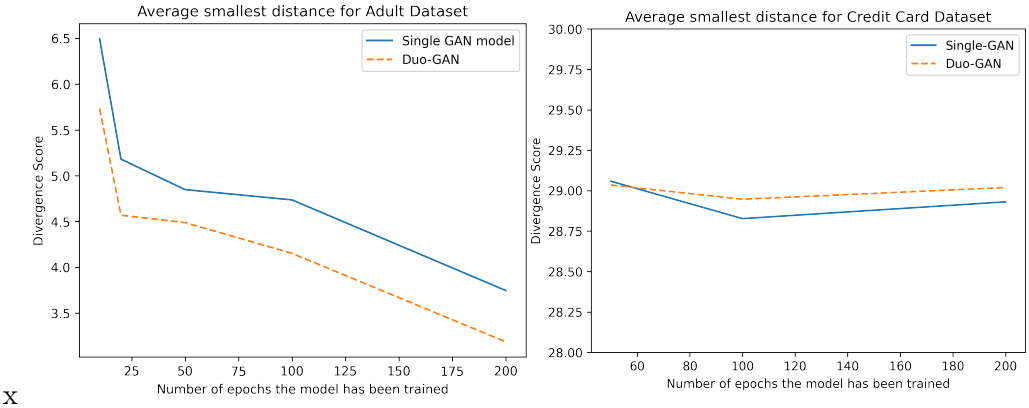
The experiments were conducted for 10, 20, 50, 100 and 200 epochs of training for the shown approaches, however for some training epochs the single generator models did not generate positive samples so for some experiments only the results where the single generator created positive samples are shown. The Appendix contains complimentary information regarding the results of the experiments.

6.1 Single GAN and Duo-GAN

Synthetic Data validation

In Figure 6.1 we show the results of the *Divergence* score over the number of epochs that the model was trained for the Adult (Figure 6.1a) and the Credit Card Fraud Detection (Figure 6.1b) 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 behaviour of the approaches, we can see 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. 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 assumption over which one presents a more desirable behaviour. 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 of them. Concerning Duo-GAN (dashed line), we can generate positive and negative instances much earlier. Another interesting aspect is that the *Divergence*

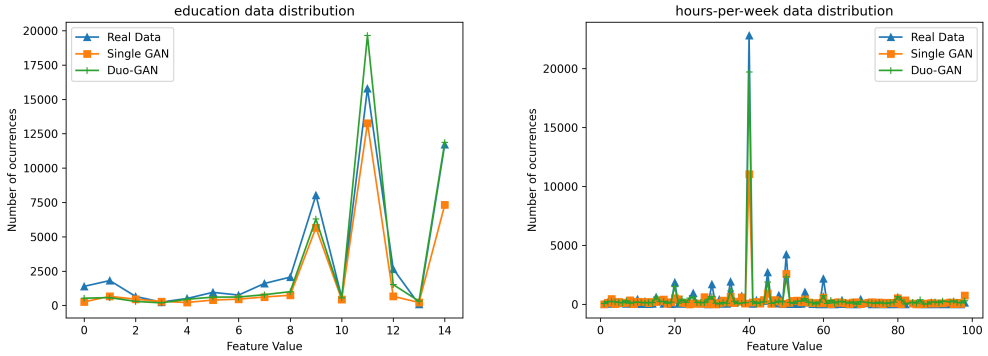


x

(a) Adult Dataset

(b) Credit card Dataset

Figure 6.1: Evolution of the *Divergence* score for the Adult and Credit Card Fraud Detection datasets. Lower values indicate an high resemblance between the original dataset and the synthetic data.



(a) Distribution of values for education categorical feature.

(b) Distribution of values for hours-per-week continuous feature.

Figure 6.2: Comparison of the distribution of features between the different approaches and the Adult dataset

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 memorising 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. Figure 6.2 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 6.3 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. Figures 6.4 and 6.5 present the correlations matrix results for Adult and the Credit Card Fraud Detection,

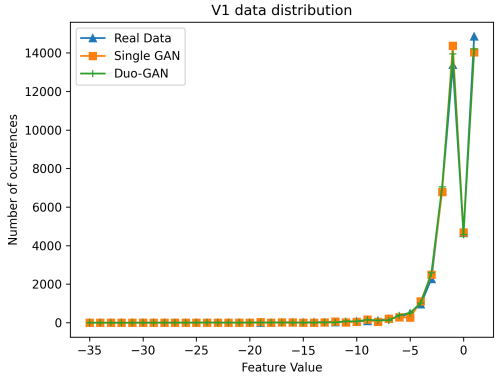


Figure 6.3: Distribution of values for the V1 continuous feature for the Credit Card Fraud Detection

Table 6.1: 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.

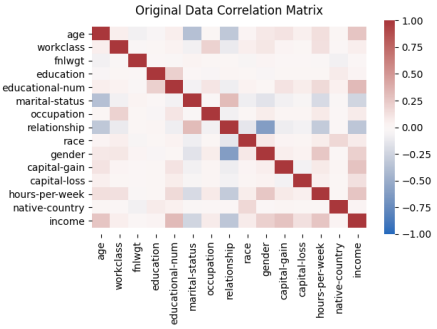
Classifier	**Approach**	**10 Epochs**	**20 Epochs**	**50 Epochs**	**100 Epochs**	**200 Epochs**	**Real Data Score**
Adaboost	Single GAN	0.0461	0.0187	0.0771	0.2396	0.3098	0.6858
	Duo-GAN	0.5961	0.6424	**0.6481**	0.6400	0.6460	
DecisionTree	Single GAN	0.2353	0.2386	0.0771	0.2396	0.3098	0.6238
	Duo-GAN	0.5732	0.5884	**0.6292**	0.6009	0.5361	
MLP	Single GAN	0.1866	0.1863	0.0771	0.2396	0.3098	0.0895
	Duo-GAN	0.5783	**0.6065**	0.0254	0.5806	0.5558	
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	

respectively. Looking at the results Figures 6.4 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 6.4a we can see a medium to high correlation (~ 0.6). Looking at the same cell in Figure 6.4c we can see that the correlation still exists but to a small degree (~ 0.4). However, looking at the correlation value between age and income in Figure 6.4b 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.

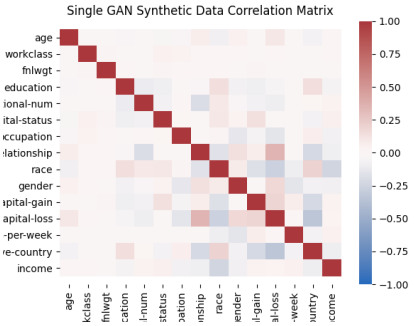
When looking at the correlation results for the Credit Card Fraud Detection (Figure 6.5) the differences between Duo-GAN and the Single GAN are even more significant. A perusal of Figure 6.5c reveals that our proposal can capture most of the existing correlations in the Original dataset (Figure 6.5a). On the contrary, the Single GAN (Figure 6.5b) cannot capture any of the existing correlations. When using the Single GAN approach, we lose all the existing correlations between the features.

Synthetic Data Utility Evaluation

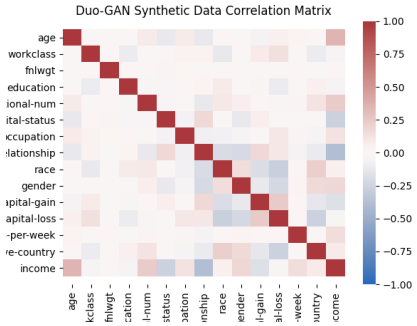
In Tables 6.1 and 6.4 we present the F_1 scores obtained during classification, and in Tables 6.2 and 6.5 we present the Precision-Recall Area Under the Curve. In general, it is possible to see that Duo-GAN obtains the best results for both problems in all of the classifiers. One



(a) Original

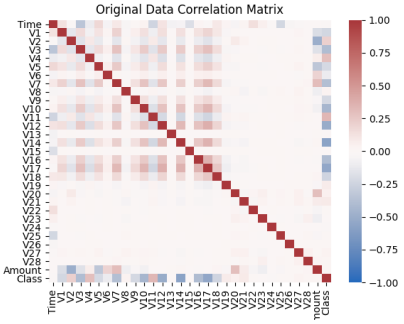


(b) Single GAN

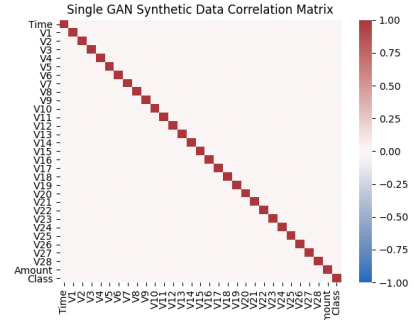


(c) Duo-GAN

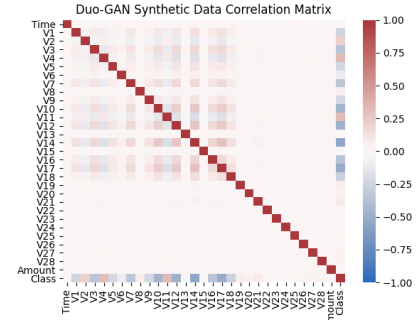
Figure 6.4: Correlation matrices between the features for the Adult dataset. The synthetic datasets used are the ones obtained from training the generator models for 50 epochs.



(a) Original



(b) Single GAN



(c) Duo-GAN

Figure 6.5: Correlation matrices between the features for the Credit Card Fraud Detection dataset.The synthetic datasets used are the ones obtained from training the generator models for 50 epochs.

Table 6.2: Precision Recal 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	
MLP	Single GAN	0.2549	0.3093	0.1770	0.2700	0.4615	0.6168
	Duo-GAN	0.5941	0.6217	**0.6316**	0.5348	0.5586	
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 6.3: Precision and Recal 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
MLP	Single GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.7530	1.0000
		Positive	0.9878	0.0237
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 6.4: 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	
MLP	Single GAN	0.0000	0.0000	0.0045	0.8638
	Duo-GAN	0.8720	**0.8765**	**0.8765**	
XGBoost	Single GAN	0.0000	0.0000	0.0045	0.9062
	Duo-GAN	**0.8636**	0.8496	0.7687	

Table 6.5: Precision Recal 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	
MLP	Single GAN	0.0168	0.0540	0.0056	0.8723
	Duo-GAN	0.7951	**0.7972**	0.7937	
XGBoost	Single GAN	0.0168	0.0540	0.0056	0.8993
	Duo-GAN	**0.8526**	0.8446	0.8487	

Table 6.6: Precision and Recal 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
MLP	Single GAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
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

other interesting aspect is the observable degeneration in the quality of data for generator models trained during longer periods. This can be seen in the results for the models trained for 100 or 200 epochs obtain 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 6.1, and 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 6.4), the most significant difference is about 0.14, and the smallest is about 0.05.

Tables 1 and 6.6 we can see the Precision and Recall 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 precsion of 0.0615 for the Positive class when trained with data from a Single GAN generator, with Duo-GAN as the generator model, the precision for the Positive class is 0.7562. The gap in performance is bigger for the Credit Card dataset. For XGBoost the recall for the Positive class and a Single GAN generator model is 0.0362, for Duo-GAN as the generator model the recall for the same class is 0.8188.

6.2 Single WGAN and DW-GAN

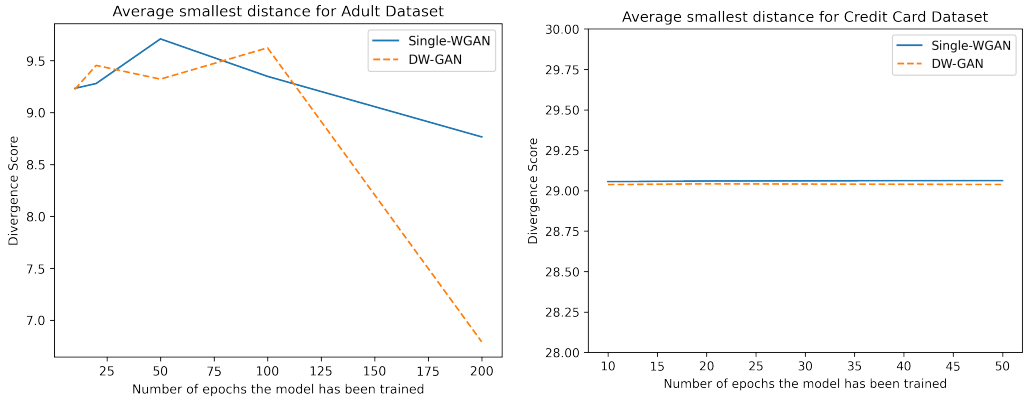
Synthetic Data validation

The results for the *Divergence* score are displayed in Figure 6.6.

The Generator Models scores for the Adult Dataset present an interesting behaviour. While models trained with smaller numbers of epochs present similar behaviour, DW-GAN outperforms a Single WGAN after longer training periods. We can also notice that both models in this experiment still attain worse performances than both models in the previous experiment (Section 6.1), with the best *Divergence* score in this experiment being ~ 7, while the worst scores in the previous experiment are already lower, starting at ~ 6.5. For the Credit Card dataset the relation between the scores is clearer, DW-GAN is consistent in getting lower *Divergence* than the Single WGAN model, however, similarly to the previous experiment (Figure 6.1b), the difference between scores is small enough that does not allow us to make an inference of which model captures the data better.

These scores for the *Divergence* allow us to conclude that both these models are not memorising the original data but that they are producing their own set of samples.

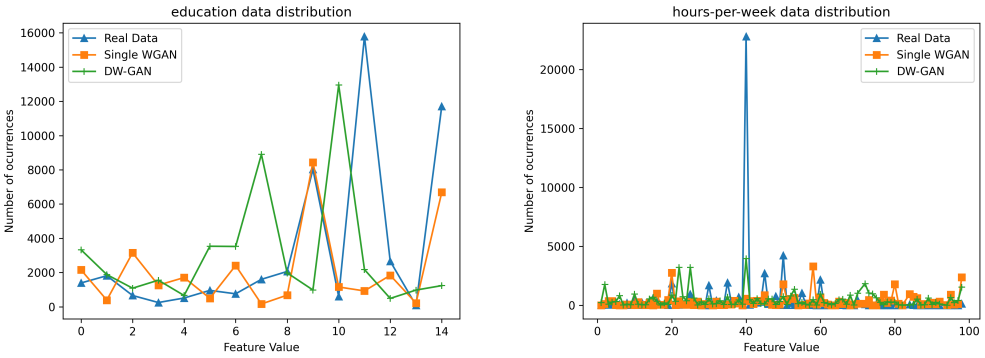
The next step for assessing the quality of the data is comparing the distribution of values for the real dataset with the synthetic datasets distributions. For the Adult dataset, in Figure 6.7, we can see the comparison for categorical (Figure 6.7a) and numerical (Figure 6.7b) features. The figures show that the generated features' distributions are not similar, with the peaks of the original distribution rarely aligning with the peaks in the generated data. This shows us that the generator models are not capable of accurately capturing the distribution of the real data. For the Credit Card dataset, which the results can be seen in Figure 6.8, we observe the opposite to what happened with the Adult dataset, since both



(a) Adult Dataset

(b) Credit card Dataset

Figure 6.6: Evolution of the *Divergence* score for the Adult and Credit Card Fraud Detection datasets. Lower values indicate an high resemblance between the original dataset and the synthetic data.



(a) Distribution of values for education categorical feature.

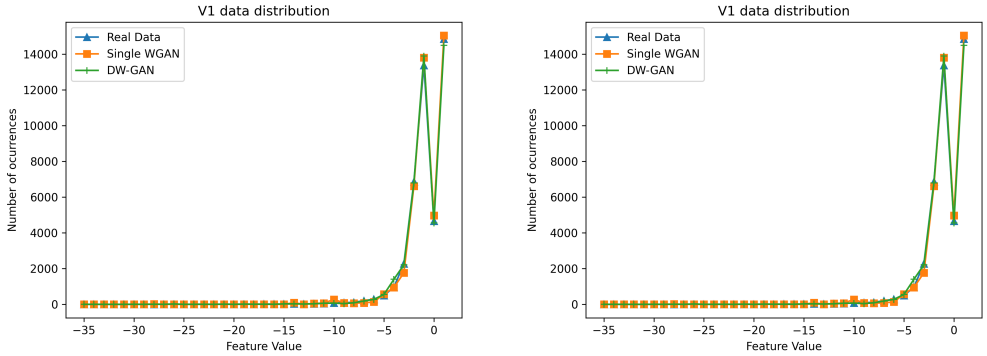
(b) Distribution of values for hours-per-week continuous feature.

Figure 6.7: Comparison of the distribution of features between the different approaches and the Adult dataset

approaches are capable of closely replicating the distribution of the original dataset.

It is important to assess that the generator models are maintaining the correlations present in the original data, due to the importance these relations between features have on the creation of ML models. In Figure 6.9 we can see the correlation matrixes for the original Adult dataset as well as the synthetic datasets produced by both generator models. In this experiment both synthetic datasets do not have similar correlation matrixes to the original. One example is the relation between age and marital-status. In the original dataset, the correlation value is ~ -0.3 while in both synthetic datasets the correlation value is ~ 0. However DW-GAN seems to be able to capture more faithfully the correlations existent in the original data. One such example is the relation between income and education-num, that has a value of ~ 0.3 in the original data. While in the Single WGAN dataset the same relation has a value of ~ -0.1, the DW-GAN dataset shows a value of ~ 0.2, a much closer value to the real data.

For the Credit Card dataset the behaviour is different. As can be seen in Figure 6.10b the Single WGAN dataset is incapable of producing any correlation between features. In opposition, the DW-GAN dataset correlation matrix, depicted in Figure 6.10c, while not

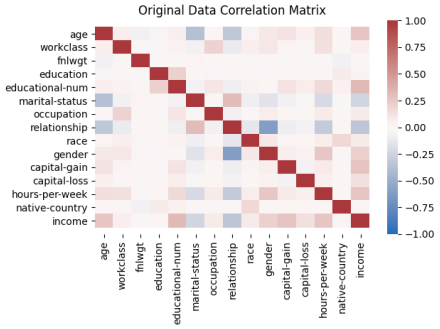


(a) Distribution of values for V1 continuous feature.

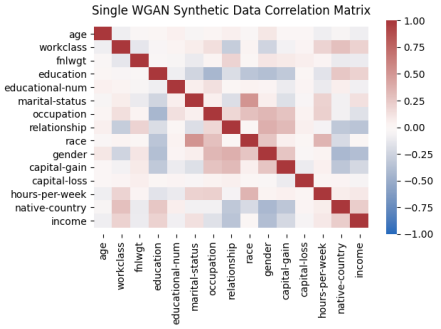
(b) Distribution of values for V26 continuous feature.

Figure 6.8: Comparison of the distribution of features between the different approaches and the Credit Card dataset

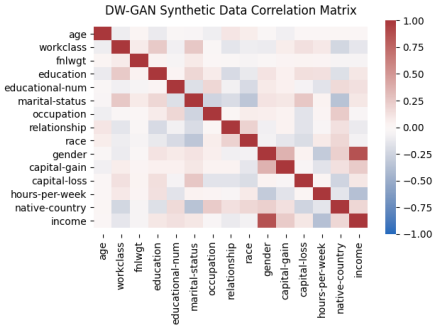
completely matching the original, depicted in Figure 6.10a, it still resembles it. Looking at the Class feature, per example, we can see that the relations with other features are similar in the original and the DW-GAN datasets.



(a) Original



(b) Single WGAN



(c) DW-GAN

Figure 6.9: Correlation matrices between the features for the Adult dataset.

Synthetic Data Utility Evaluation

In Tables 6.7, 6.8, 6.10 and 6.11 the results for this experiment are shown.

One pattern that is clear is DW-GAN generally outperforming the Single WGAN model on both problems. However the utility retained is different for both datasets. In the Adult

Table 6.7: 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 WGAN	0.0329	0.3557	0.0110	0.0211	0.0851	0.6858
	DW-GAN	0.3409	0.3257	0.3874	0.3881	**0.4614**	
DecisionTree	Single WGAN	0.2739	0.3816	0.2424	0.1451	0.1634	0.6238
	DW-GAN	0.3335	0.3331	0.3808	0.3324	**0.4032**	
MLP	Single WGAN	0.0000	**0.3851**	0.0000	0.0000	0.3799	0.0895
	DW-GAN	0.0000	0.3308	0.0445	0.0631	0.0073	
XGBoost	Single WGAN	0.1120	0.2280	0.0197	0.0446	0.0198	0.7143
	DW-GAN	0.3964	0.3157	0.3870	0.3892	**0.4500**	

Table 6.8: Precision Recal 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 WGAN	0.1806	0.1691	0.1719	0.1704	0.3346	0.8107
	DW-GAN	0.3928	0.2065	0.1936	0.1723	**0.5283**	
DecisionTree	Single WGAN	0.3609	0.5508	0.3344	0.2490	0.2656	0.6685
	DW-GAN	0.5117	0.4896	**0.6018**	0.5321	0.4768	
MLP	Single WGAN	0.6214	0.6129	0.6214	0.6214	0.6051	0.6168
	DW-GAN	0.6214	0.5419	**0.6248**	0.6219	0.6230	
XGBoost	Single WGAN	0.1861	0.1638	0.1824	0.1666	0.3575	0.8346
	DW-GAN	0.2626	0.2383	0.2013	0.4255	**0.4961**	

Table 6.9: Precision and Recal breakdown by class for data generated by model trained for 200 epochs for the Adult dataset.

Classifier	Generator	Class	Precision	Recall
Adaboost	Single WGAN	Negative	0.7521	0.9694
		Positive	0.3494	0.0489
	DW-GAN	Negative	0.8195	0.9694
		Positive	0.4189	0.4959
DecisionTree	Single WGAN	Negative	0.7312	0.8324
		Positive	0.1520	0.0894
	DW-GAN	Negative	0.7979	0.8324
		Positive	0.4051	0.3921
MLP	Single WGAN	Negative	0.0000	0.0000
		Positive	0.2478	0.9807
	DW-GAN	Negative	0.7489	0.0000
		Positive	1.0000	0.0021
XGBoost	Single WGAN	Negative	0.7496	0.9961
		Positive	0.4521	0.0097
	DW-GAN	Negative	0.8163	0.9961
		Positive	0.4326	0.4683

Table 6.10: F1-score for machine learning models trained on real data and synthetic data for the Credit Card Fraud Detection dataset. The best results attained with models trained with synthetic data are highlighted in bold.

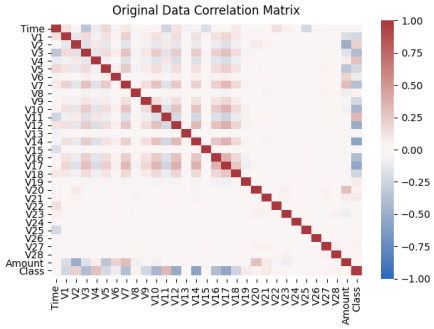
Classifier	**Approach**	**50 Epochs**	**100 Epochs**	**200 Epochs**	**Real Data Score**
Adaboost	Single WGAN	0.0000	0.0144	0.0000	0.8571
	DW-GAN	0.8716	0.8819	**0.8854**	
DecisionTree	Single WGAN	0.2633	0.1825	0.1684	0.7817
	DW-GAN	0.8075	**0.8425**	0.8413	
MLP	Single WGAN	0.0000	0.0000	**0.0181**	0.8638
	DW-GAN	0.0000	0.0000	0.0000	
XGBoost	Single WGAN	0.0000	0.0000	0.1074	0.9062
	DW-GAN	0.8516	**0.8730**	0.8606	

Table 6.11: Precision-Recall Area Under the Curve for machine learning models trained on real data and synthetic data for the Credit Card Fraud Detection dataset. The best results attained with models trained with synthetic data are highlighted in bold.

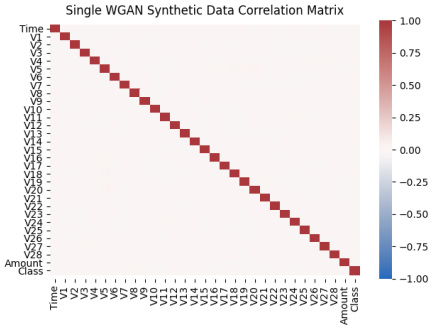
Classifier	**Approach**	**50 Epochs**	**100 Epochs**	**200 Epochs**	**Real Data Score**
Adaboost	Single WGAN	0.0047	0.0291	0.0047	0.8879
	DW-GAN	**0.8370**	0.8318	0.8352	
DecisionTree	Single WGAN	0.3203	0.2419	0.2105	0.7832
	DW-GAN	0.8100	0.8499	**0.8500**	
MLP	Single WGAN	0.0047	0.0047	0.0150	0.8723
	DW-GAN	**0.7851**	0.7846	0.7828	
XGBoost	Single WGAN	0.0075	0.0786	0.2312	0.8993
	DW-GAN	**0.8583**	0.8455	0.8440	

Table 6.12: Precision and Recal breakdown by class for data generated by model trained for 50 epochs for the Credit Card dataset.

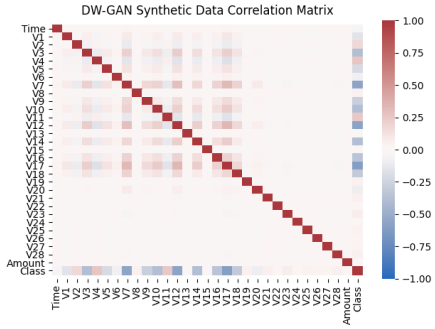
Classifier	Generator	Class	Precision	Recall
Adaboost	Single WGAN	Negative	0.9908	0.9999
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9982	0.9999
		Positive	0.9737	0.8043
DecisionTree	Single WGAN	Negative	0.9951	0.9811
		Positive	0.1902	0.4783
	DW-GAN	Negative	0.9979	0.9811
		Positive	0.9469	0.7754
MLP	Single WGAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
XGBoost	Single WGAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9981	1.0000
		Positive	0.9646	0.7899



(a) Original



(b) Single WGAN



(c) DW-GAN

Figure 6.10: Correlation matrices between the features for the Credit Card Fraud Detection dataset.

dataset F_1 scores, displayed in Table 6.7, show a difference in utility, when compared to classifiers trained on real data, for XGBoost ranging from 0.4 to 0.26, which is a significant loss of performance. However for the Credit Card dataset, we observe a very positive retention of utility. Table 6.10 shows the F_1 scores for classifiers trained on synthetic data and real data. For some of the classifiers the synthetic data ends up outperforming slightly the real data even, but for XGBoost the loss in performance is significantly low, ranging from 0.06 to 0.03. The difference in performance for both datasets might be explained by WGAN being worse at generating categorical data.

We can compare the classifier behaviour for the specific classes. In Table 6.9 we see the behavior of classifiers according to class and the generator model for the Adult dataset. Using XGBoost as an example we can see that for negative instances, there is a slight improvement between a single WGAN and DW-GAN in the precision scores. A much bigger difference is between the positive classes, while precision remains the same between generator models, the recall improves by ~ 0.45. For the Credit Card dataset, shown on Table 6.12 there is no improvement in the results of negative instances, however for the positive instances experience a significant improvement, as can be seen for the XGBoost results.

6.3 Feature Engineering

In this section we present and discuss the results obtained following the experimental setup detailed in Section 5.2.3

6.3.1 Pre Feature Engineering

In this experiment we create a synthetic dataset and then apply a feature engineering routine to it, and compare it to real data that has been subjected to the same routine.

Synthetic Data validation

Figure 6.11 shows the evolution of the *Divergence* score for the number of training epochs. As we can see there is a tendency for the *Divergence* score to go down as the training time gets longer for both approaches. Furthermore, the overperformance of Duo-GAN over DW-GAN is clear, while DW-GAN's scores range from ~ 13 to ~ 10, Duo-GAN's range from ~ 9 to ~ 6. This serves as an indication that data generated by Duo-GAN resembles the original data more than DW-GAN.

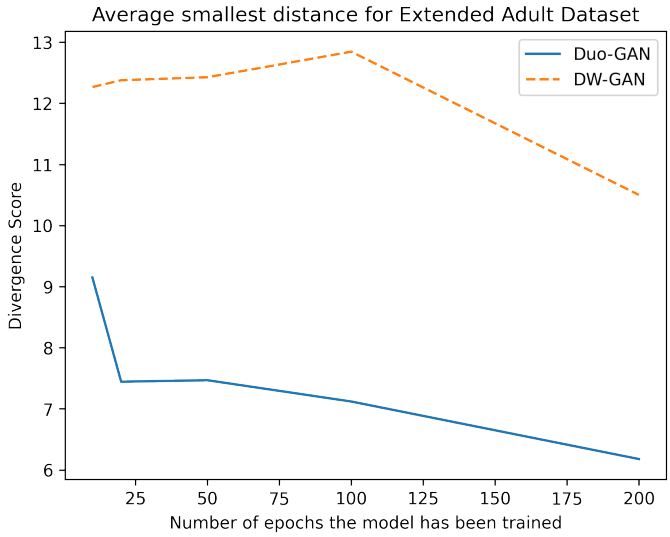
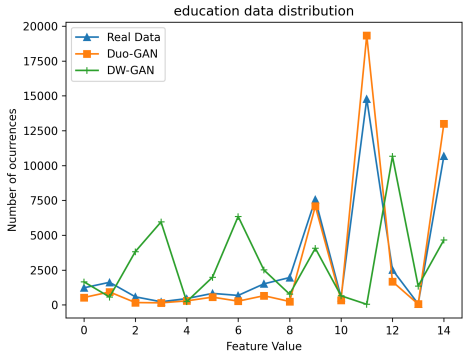


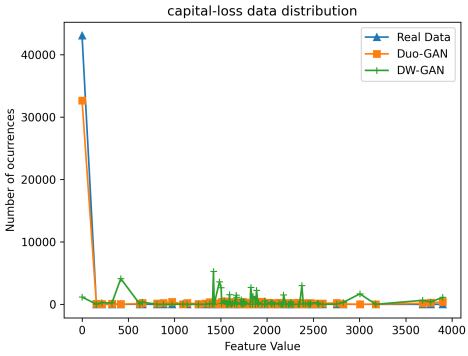
Figure 6.11: Evolution of the *Divergence* score for the Extended Adult dataset with synthetic data generated before feature engineering.

After looking at the *Divergence* score, we compare the distributions of the original data to the ones present in the synthetic datasets created by our generator models. In Figure 6.12, we can see that Duo-GAN is better able to capture the patterns present in the original data, than DW-GAN. These results are consistent with the previous experiments done on the Adult Dataset (Section 6.1 and Section 6.2).

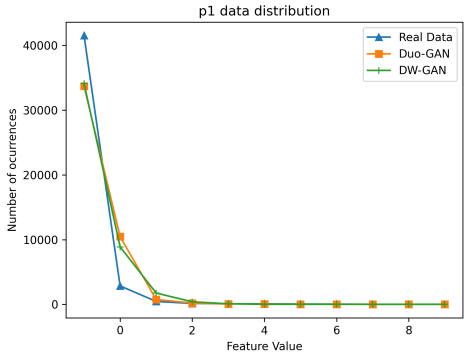
Looking at the correlation matrixes of the original and synthetic datasets, pictured in Figure 6.13, and especifically looking at the engineered features (p1,p2 and p3) we can see how and if executing a feature engineering routine on synthetic datasets maintains the utility of feature engineering on real data. Looking at the relationship between the engineered features and the continuous feature fnlwgt is a good display of the differences in the matrixes accross the different datasets. In the real data the value of the relation between fnlwgt and p1,p2 and p3 is ~ 0.7,~ 0.7,~ 1 respectively, Duo-GAN has values of ~ 0.3,~ 0.3,~ 0.8, while DW-GAN's values are ~ 0.4,~ 0.6,~ 0. While DW-GAN's values are closer to the ones present in real data for two of the features the relation with the strongest value in real data is not captured in the DW-GAN dataset, Duo-GAN on the other hand has correlation values further away in the p1 and p2 features however the larger pattern of relation between the engineered features and fnlwgt is similar to the one in the original dataset.



(a) Distribution of values for education categorical feature.



(b) Distribution of values for capital-loss continuous feature.



(c) Distribution of values for engineered p1 continuous feature.

Figure 6.12: Comparison of the distribution of features between the different approaches and the Extended Adult dataset

Synthetic Data Utility Evaluation

The first assessment that must be done in this context is asserting whether or not the feature engineering routine is adding utility to the original dataset. If we compare the values for real data present in this experiment, present in Table 6.13, with the ones from experiments not including feature engineering, such as the ones present in Table 6.1, we can see that the F_1 Score for most classifiers improved slightly. For example the score for XGBoost went from 0.7143 to 0.7799. The same pattern is visible for the Precision Recall Area Under the Curve, as can be seen in Tables 6.14 and 6.2.

Comparing the results between the data subjected to feature engineering and raw data allow us to see how the generator models are able to cope with these techniques. Comparing Duo-GAN's results in this scenario, in Table 6.13, with the ones in the scenario without feature engineering, in Table 6.1, we can see that there is no clear improvement in the scores. Comparing DW-GAN's data performance in this scenario to the performance with raw data, shown on Table 6.7, we can see that once again there is not improvement in performance, except for the 100 epochs data which experienced a consistent improvement, per example the XGBoost performance in the F_1 score went from 0.3892 to 0.4642.

Table 6.15 shows a breakdown of the precision and recall for each class according to the generator, as well as the value for the same metrics for data with no exposure to feature engineering. Looking at the scores for XGBoost we can see there is no clear improvement.

Table 6.13: F1-score for machine learning models trained on real data and synthetic data for the Extended Adult dataset. The best results attained with models trained with synthetic data are highlighted in bold.

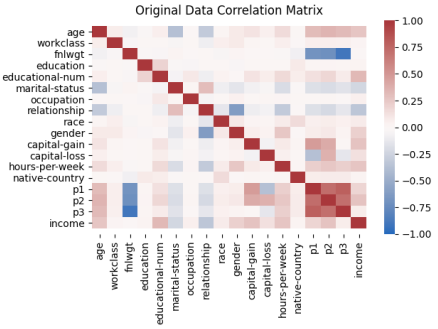
Classifier	**Approach**	**10 Epochs**	**20 Epochs**	**50 Epochs**	**100 Epochs**	**200 Epochs**	**Real Data Score**
Adaboost	Duo-GAN	0.6133	0.6262	0.6494	0.6288	0.6025	0.6960
	DW-GAN	0.1171	0.1623	0.2219	0.4711	0.4058	
DecisionTree	Duo-GAN	0.5409	0.6123	0.6108	0.5777	0.5119	0.6228
	DW-GAN	0.3437	0.3734	0.3902	0.4688	0.3994	
MLP	Duo-GAN	0.0006	0.0035	0.0058	0.0041	0.3927	0.3302
	DW-GAN	0.0000	0.0000	0.3289	0.1451	0.3210	
XGBoost	Duo-GAN	0.6535	0.6438	0.6505	0.6548	0.5748	0.7799
	DW-GAN	0.2206	0.3073	0.3760	0.4642	0.4352	

Table 6.14: Precision-Recall Area Under the Curve for machine learning models trained on real data and synthetic data for the Extended Adult dataset. The best results attained with models trained with synthetic data are highlighted in bold.

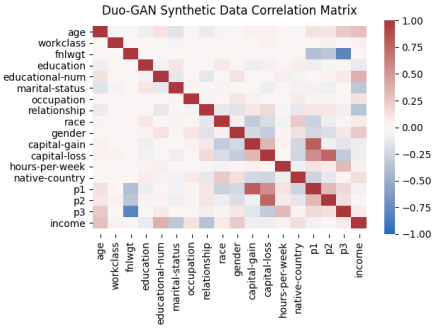
Classifier	**Approach**	**10 Epochs**	**20 Epochs**	**50 Epochs**	**100 Epochs**	**200 Epochs**	**Real Data Score**
Adaboost	Duo-GAN	0.7180	0.7180	0.7150	0.6862	0.6286	0.8167
	DW-GAN	0.2709	0.2014	0.4025	0.5254	0.2936	
DecisionTree	Duo-GAN	0.6306	0.6586	0.6573	0.6308	0.5729	0.6697
	DW-GAN	0.4308	0.5534	0.5980	0.6116	0.5959	
MLP	Duo-GAN	0.6259	0.6264	0.6270	0.6265	0.6123	0.5354
	DW-GAN	0.6257	0.6257	0.5657	0.6036	0.4619	
XGBoost	Duo-GAN	0.7287	0.7384	0.7337	0.7094	0.5713	0.8932
	DW-GAN	0.2623	0.3804	0.5164	0.4507	0.5466	

Table 6.15: Precision and Recal breakdown by class for data generated by model trained for 50 epochs for the Extended Adult dataset.

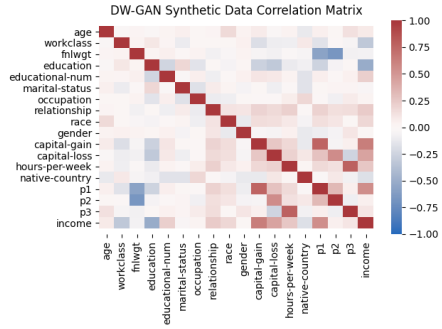
Classifier	Generator	Class	Precision	Recall	Raw data Precision	Raw data Recall
Adaboost	DW-GAN	Negative	0.7735	0.9552	0.8195	0.9694
		Positive	0.5565	0.1674	0.4189	0.4959
	Duo-GAN	Negative	0.9147	0.9552	0.9078	0.8504
		Positive	0.5440	0.7837	0.5623	0.7576
DecisionTree	DW-GAN	Negative	0.9260	0.1640	0.7979	0.8324
		Positive	0.2786	0.9610	0.4051	0.3921
	Duo-GAN	Negative	0.8967	0.1640	0.8967	0.7133
		Positive	0.5356	0.7301	0.5415	0.7283
MLP	DW-GAN	Negative	0.0000	0.0000	0.7489	0.0000
		Positive	0.2515	1.0000	1.0000	0.0021
	Duo-GAN	Negative	0.7518	0.0000	0.7530	1.0000
		Positive	1.0000	0.0176	0.9878	0.0237
XGBoost	DW-GAN	Negative	0.8006	0.9712	0.8163	0.9961
		Positive	0.7658	0.2799	0.4326	0.4683
	Duo-GAN	Negative	0.9141	0.9712	0.9106	0.8199
		Positive	0.5801	0.7729	0.6055	0.7562



(a) Original



(b) Duo-GAN



(c) DW-GAN

Figure 6.13: Correlation matrixes for different approaches and the Extended Adult dataset.

Despite the precision for positive instances generated by DW-GAN growing, the recall gets smaller, for Negative instances both metrics scores go down. For Duo-GAN generated data there is a noticeable growth on the recall for the negative class and a slight growth in precision, for positive instances precision grows ~ 0.2 while the recall goes down ~ 0.2.

6.3.2 Post Feature Engineering

In this experiment we applied a feature engineering routine to the real dataset, and then we took the extended dataset and used it as input for a generator model, that outputs a synthetic dataset that includes engineered features.

In Figure 6.14 we can see the evolution of the *Divergence* score for the number of training epochs. Similarly to previous experiments, the *Divergence* score trends down as the training time get longer for both models. DW-GAN's scores range from ~ 13 to ~ 10, while Duo-GAN's go from ~ 9 to ~ 6. This shows a clear overperformance of Duo-GAN, indicating that data generated by it will resemble the original data than data produced by DW-GAN.

The distribution of values in datasets can be seen in Figure 6.15. We can see that Duo-GAN is able to capture the general pattern of the distribution of the original features in the real data. DW-GAN once again, same as in Section 6.2, struggled to reproduce the distribution of the original dataset. In regard to the engineered features, which has its distribution depicted in Figure 6.15c, both approaches were capable of almost precisely reproducing it.

The correlation matrixes of the synthetic datasets, when compared to the correlation matrix of the original dataset, will give us insights into how our approaches are able to generate

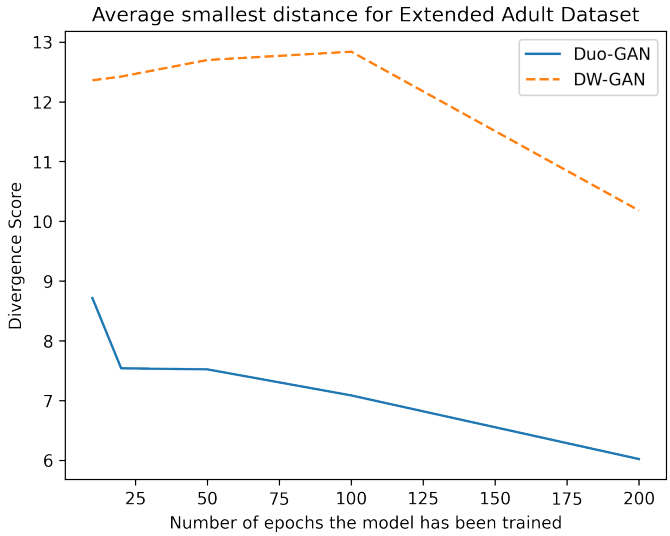


Figure 6.14: Evolution of the *Divergence* score for the Extended Adult dataset with synthetic data generated after feature engineering.

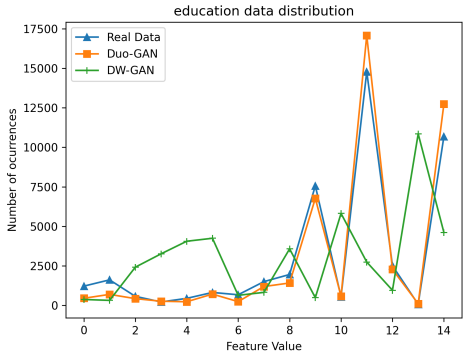
Table 6.16: F1-score for machine learning models trained on real data and synthetic data for the Extended Adult dataset. The best results attained with models trained with synthetic data are highlighted in bold.

Classifier	**Approach**	**10 Epochs**	**20 Epochs**	**50 Epochs**	**100 Epochs**	**200 Epochs**	**Real Data Score**
Adaboost	Duo-GAN	0.6416	0.6481	0.6462	0.6401	0.6239	0.6960
	DW-GAN	0.3601	0.2452	0.1523	0.2931	0.5955	
DecisionTree	Duo-GAN	0.5360	0.6005	0.6049	0.6148	0.5503	0.6228
	DW-GAN	0.5752	0.5080	0.3522	0.3532	0.5955	
MLP	Duo-GAN	0.0994	0.0514	0.0591	0.0209	0.3628	0.3302
	DW-GAN	0.0000	0.0000	0.1005	0.0180	0.0000	
XGBoost	Duo-GAN	0.6475	0.6639	0.6599	0.6499	0.6210	0.7799
	DW-GAN	0.4508	0.3829	0.2598	0.2171	0.5955	

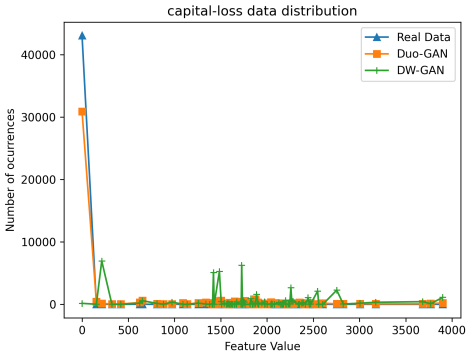
data that includes engineered features. Looking at the correlation matrix of the dataset generated by DW-GAN in Figure 6.16c, we can see that DW-GAN is capable of capturing some of the relations present in the original data such as the relation between education-number and income. However, the DW-GAN dataset has relations that do not exist in the original data, such as capital-gains and occupation, or gender and education-num. The relations observed in the real data relating to the engineered features are not present in the DW-GAN dataset. The dataset resulting from Duo-GAN misses the relations relating to the engineered features as well. Duo-GAN also creates relations in its data that are not present in the original dataset, such as the negative relations between native-country and capital-loss and between gender and capital-gain, both of which do not exist in the real data. Some relations are kept, even if lower values, such as the relation between marital-status and relationship which has a value of ~ 0.4 in the original data and a value of ~ 0.2 in the dataset synthesized by Duo-GAN.

Synthetic Data Utility Evaluation

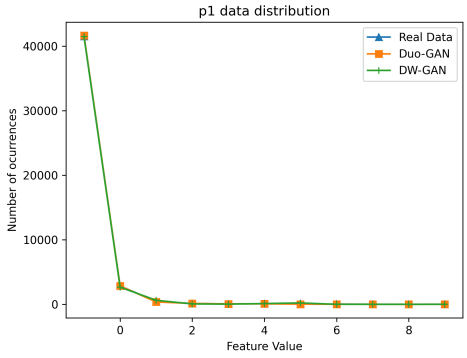
The results of this experiment need to be evaluated on two parameters, how they compare to the performance on real data, and how it compares to situations where feature engineering was not part of the process in order to measure how effective Duo-GAN is at generating data that includes engineered features.



(a) Distribution of values for education categorical feature.



(b) Distribution of values for capital-loss continuous feature.



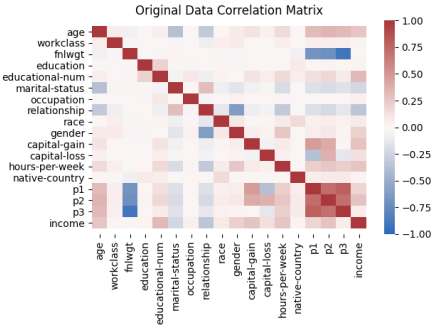
(c) Distribution of values for engineered p1 continuous feature.

Figure 6.15: Comparison of the distribution of features between the different approaches and the Extended Adult dataset

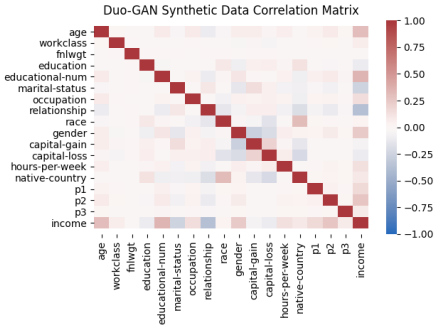
When comparing Duo-GAN with the real data scores, in Table 6.16, we can see that for the XGBoost classifier the loss in utility ranges from ~ 0.15 to ~ 0.11. DW-GAN data shows much larger drop utility, while for XGBoost the real data score is 0.7799, with DW-GAN data the scores range from 0.2171 to 0.5955.

Looking at the relation with situations without feature engineering, Duo-GAN data did not have better results. The results for XGBoost compared to the ones shown on Table 6.1, show no clear improvement over the results. However DW-GAN does show improvement, as we can see if we compare the results for XGBoost in Table 6.7. The best result with no engineered features in the dataset is ~ 0.45, while the best result in this context was 0.5955, which translates into a significant improvement.

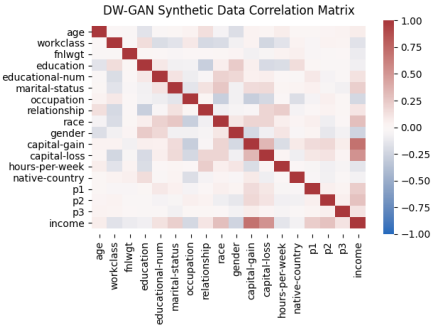
When looking at the precision and recall results for each class, shown in Table 6.18, we can see that there is no discernible pattern in the change of results. The precision for positive instances generated by DW-GAN has shown significant growth, however the recall went down. For the same generator the change of performance in the negative instances is negligible. For data generated by Duo-GAN there is improvement for both recall and precision in negative instances, however for the positive ones, while recall went up by ~ 0.05, the precision went down by nearly the same amount.



(a) Original



(b) Duo-GAN



(c) DW-GAN

Figure 6.16: Correlation matrixes for different approaches and the Extended Adult dataset.

Table 6.17: Precision-Recall Area Under the Curve for machine learning models trained on real data and synthetic data for the Extended 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	Duo-GAN	0.7127	0.7414	0.7314	0.7147	0.7011	0.8167
	DW-GAN	0.5924	0.4233	0.3424	0.4254	0.6731	
DecisionTree	Duo-GAN	0.5919	0.6480	0.6633	0.6633	0.6069	0.6697
	DW-GAN	0.6665	0.6243	0.5321	0.5447	0.6731	
MLP	Duo-GAN	0.6193	0.6258	0.6279	0.6297	0.5679	0.5354
	DW-GAN	0.6257	0.6257	0.6176	0.6294	0.5012	
XGBoost	Duo-GAN	0.7055	0.7485	0.7398	0.7183	0.6560	0.8932
	DW-GAN	0.5749	0.5534	0.4030	0.3541	0.5244	

6.4 Discussion

The results presented in this chapter allow us to make a number of assertions over the proposed methodology, experiment design and the proposed approaches.

Our proposed data validation, in Section 5.1.2, needs to be tuned in order to be a better predictor of the performance of data in the utility evaluation step. The divergence metric is not a reliable predictor of performance. In Figure 6.1a we see the value of divergence getting lower with more training epochs, however the classification performance deteriorates after 50 epochs of training. For the Credit Card dataset, we see that the divergence values tend to be high, ~ 29 out of a maximum of 31, however synthetic data generated by our proposed approaches still obtained performance close to the utility on the real data. This disparity between the performance and the *divergence* scores can be caused by the way it deals with continuous features since for these features it is harder to obtain an exact match.

Table 6.18: Precision and Recal breakdown by class for data generated by model trained for 50 epochs for the Extended Adult dataset.

Classifier	Generator	Class	Precision	Recall	Raw data Precision	Raw data Recall
Adaboost	DW-GAN	Negative	0.7740	0.9931	0.8195	0.9694
		Positive	0.8699	0.1372	0.4189	0.4959
	Duo-GAN	Negative	0.9107	0.9931	0.9078	0.8504
		Positive	0.5498	0.7699	0.5623	0.7576
DecisionTree	DW-GAN	Negative	0.7308	0.5396	0.7979	0.8324
		Positive	0.2296	0.4083	0.4051	0.3921
	Duo-GAN	Negative	0.9019	0.5396	0.8967	0.7133
		Positive	0.5088	0.7556	0.5415	0.7283
MLP	DW-GAN	Negative	0.7549	0.9996	0.7489	0.0000
		Positive	0.9667	0.0340	1.0000	0.0021
	Duo-GAN	Negative	0.7533	0.9996	0.7530	1.0000
		Positive	0.9885	0.0252	0.9878	0.0237
XGBoost	DW-GAN	Negative	0.7810	0.9953	0.8163	0.9961
		Positive	0.9233	0.1694	0.4326	0.4683
	Duo-GAN	Negative	0.9256	0.9953	0.9106	0.8199
		Positive	0.5593	0.8121	0.6055	0.7562

The experiments did not allow us to make any meaningful assertions about where in the pipeline of synthetic data generation should feature engineering be included. The feature engineering routine did not change the performance enough to draw conclusions. This may be the cause for none of the approaches experiencing significant change in performance with feature engineering, either for feature engineering before or after the generation of data.

Looking at the proposed approaches we can see that both of them outperformed 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 performance in classifying tasks compared to classifiers trained on real data. The difference in performance for the positive instances highlights how a double generator model creates better quality data by improving the quality of the positive records. We can also see that a traditional GAN approach, Duo-GAN, is superior to a WGAN approach, DW-GAN, especially for the Adult dataset which includes categorical features. This may be due to the usage of Wasserstein distance as a loss function for the Generator because calculating the distance between distributions that include categorical values is not as clear as it is for exclusively numerical datasets.

Our proposed approaches finished training with 200 epochs in around 30 minutes for the Adult dataset, and 60 minutes for the Credit Card datasets. 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.

Chapter 7

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 anonymisation procedures. These procedures raise some issues, namely the delay in the deployment of solutions and the fact that the anonymisation might remove important details hindering accurate predictions. Synthetic datasets that contain the same statistical properties, such as distributions and correlations between features, can help organizations speed up their development process for effective solutions.

In this work we propose Duo-GAN as a framework for generating synthetic datasets. This framework is designed taking into account the specific limitations of the fraud detection datasets, namely the variety in the typology of features that is natural to tabular data and the high imbalance of the datasets, where fraudulent instances tend to be rare. 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. The GAN used in our framework are based on TGAN[42], since this model has mechanisms for dealing with the different types of features. We also introduce DW-GAN, a framework similar to Duo-GAN that trades the traditional GAN for WGAN.

To validate our approaches we created a testing scheme that includes a theorethical validation of data, as well as a measurement of its utility. We introduce a metric to measure the divergence between the real and synthetic datasets as well as look at the distributions of features and at the correlation matrixes and compare them with the ones in real data. This process should allow us to predict the utility present in synthetic datasets, however the *divergence* score we introduce is not a good indicator of utility due to being too strict for continuous features. To measure the utility retained by synthetic data we use it to train a group of classifiers and then test it on real data. We also compare the performance to classifiers trained on real data and tested on the same data.

The results from the experimental setup described in this work show that our proposed approach not only outperforms single GAN generator models, it also generates high quality synthetic datasets that show performance close to the performance of real datasets. Specifically, the best model trained with synthetic data generated by Duo-GAN obtains classification performance with a gap of 4% in F_1 score, while a Single-GAN model gen-

erates data that has close to 0 in F_1 score in classification tasks, performing particularly poorly in identifying positive samples. DW-GAN has similar performance to Duo-GAN in the Credit Card dataset, however the performance decays for the Adult dataset, which contains categorical features with the smallest gap in perfomance being $\sim 43\%$.

Future Work

As mentioned above, this work presented some shortfalls namely with theoretical validation of data and considering feature engineering. Future work will aim to solve these issues. The *divergence* metric should be modified to better accomodate continuous features. With regards to feature engineering there is a need to create a feature engineering routine that adds bigger utility to data before evaluating where feature engineering should take place in the synthetic data generation pipeline.

The results for Duo-GAN are positive and indicate that synthetic data generation can help speed up the development of fraud detection solutions. In the future, we will study how datasets containing both real and synthetic data perform, essentially measuring the compatibility between data, that will allow for the integration of real data in the development of solutions continually while it is being aggregated. The compatibility of data also opens the door for data augmentation, namely artificially balancing the datasets. Conditional GAN are also promising for the purposes of data augmentation, because they allow the sampling of data with specific charactheristics, helping to solve problems found in the classifiers. Given this in the future we will test different GAN models inside the Duo-GAN architecture as generators, including Conditional GAN and models that focus on Differential Privacy.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[2] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)*, pages 1–9, 2017. doi: 10.1109/ICCNI.2017.8123782.

[3] Arjen P. de Vries Bauke Brenninkmeijer, Youri Hille. On the generation and evaluation of synthetic tabular data using gans. 2019.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[5] Fabrizio Carcillo, Andrea Dal Pozzolo, Yann-Aël Le Borgne, Olivier Caelen, Yannis Mazzer, and Gianluca Bontempi. Scarff : a scalable framework for streaming credit card fraud detection with spark. *Information Fusion*, 41, 09 2017. doi: 10.1016/j.inffus.2017.09.005.

[6] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, and Gianluca Bontempi. Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization. *CoRR*, abs/1804.07481, 2018. URL `http://arxiv.org/abs/1804.07481`.

[7] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, Yacine Kessaci, Frédéric Oblé, and Gianluca Bontempi. Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 05 2019. doi: 10.1016/j.ins.2019.05.042.

[8] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks, 2017.

[9] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaafar, and Haojin Zhu. Differentially private data generative models. *CoRR*, abs/1812.02274, 2018. URL `http://arxiv.org/abs/1812.02274`.

[10] Edward Choi, Siddharth Biswal, Bradley A. Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete electronic health records using generative adversarial networks. *CoRR*, abs/1703.06490, 2017. URL `http://arxiv.org/abs/1703.06490`.

[11] Andrea Dal Pozzolo, Olivier Caelen, Yann-Aël Le Borgne, Serge Waterschoot, and Gianluca Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41:4915–4928, 08 2014. doi: 10.1016/j.eswa.2014.02.026.

[12] Andrea Dal Pozzolo, Olivier Caelen, Reid Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. 12 2015. doi: 10.1109/SSCI.2015.33.

[13] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–14, 09 2017. doi: 10.1109/TNNLS.2017.2736643.

[14] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–14, 09 2017. doi: 10.1109/TNNLS.2017.2736643.

[15] Carl Doersch. Tutorial on variational autoencoders, 2021.

[16] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[17] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

[18] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL `http://www.sciencedirect.com/science/article/pii/S002200009791504X`.

[19] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 11 1997. doi: 10.1023/A:1007465528199.

[20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[21] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 1st edition, 2017. ISBN 1491962291.

[22] Huaibo Huang, Zhihang Li, Ran He, Zhenan Sun, and Tieniu Tan. Introvae: Introspective variational autoencoders for photographic image synthesis. *CoRR*, abs/1807.06358, 2018. URL `http://arxiv.org/abs/1807.06358`.

[23] J. Jordon, Jinsung Yoon, and M. V. D. Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *ICLR*, 2019.

[24] Ally Salim Jr. Synthetic patient generation: A deep learning approach using variational autoencoders. *CoRR*, abs/1808.06444, 2018. URL `http://arxiv.org/abs/1808.06444`.

[25] Bertrand Lebichot, Yann-Aël Le Borgne, Liyun He, Frédéric Oblé, and Gianluca Bontempi. *Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection*, pages 78–88. 01 2019. ISBN 978-3-030-16840-7. doi: 10.1007/978-3-030-16841-4_8.

[26] S. Li, B. Tai, and Y. Huang. Evaluating variational autoencoder as a private data release mechanism for tabular data. In *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 198–1988, 2019. doi: 10.1109/PRDC47002.2019.00050.

[27] Yi Liu, Jialiang Peng, James Jian Qiao Yu, and Yi Wu. PPGAN: privacy-preserving generative adversarial network. *CoRR*, abs/1910.02007, 2019. URL `http://arxiv.org/abs/1910.02007`.

[28] Mohammad Malekzadeh, Richard G. Clegg, and Hamed Haddadi. Replacement autoencoder: A privacy-preserving algorithm for sensory data analysis. *CoRR*, abs/1710.06564, 2017. URL `http://arxiv.org/abs/1710.06564`.

[29] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL `http://arxiv.org/abs/1411.1784`.

[30] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, pages 841–848. MIT Press, 2002. URL `https://proceedings.neurips.cc/paper/2001/file/7b7a53e239400a13bd6be6c91c4f6c4e-Paper.pdf`.

[31] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data, 2017.

[32] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083, Jun 2018. ISSN 2150-8097. doi: 10.14778/3231751.3231757. URL `http://dx.doi.org/10.14778/3231751.3231757`.

[33] 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. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[34] NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction, 2016. URL `https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12174`.

[35] A. Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.

[36] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016.

[37] R. Sathya and Annamma Abraham. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2, 02 2013. doi: 10.14569/IJARAI.2013.020206.

[38] Osvaldo Simeone. A very brief introduction to machine learning with applications to communication systems. *CoRR*, abs/1808.02342, 2018. URL `http://arxiv.org/abs/1808.02342`.

[39] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning, 2017.

[40] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation, 2020.

[41] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 835–851, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46478-7.

[42] Lei Xu and Kalyan Veeramachaneni. Synthesizing tabular data using generative adversarial networks. *CoRR*, abs/1811.11264, 2018. URL `http://arxiv.org/abs/1811.11264`.

[43] Lei Xu and Kalyan Veeramachaneni. Synthesizing tabular data using generative adversarial networks, 2018.

[44] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. *CoRR*, abs/1907.00503, 2019. URL `http://arxiv.org/abs/1907.00503`.

[45] Xiaofeng Yuan, Lin Li, and Yalin Wang. Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. *IEEE Transactions on Industrial Informatics*, PP:1–1, 02 2019. doi: 10.1109/TII.2019.2902129.

[46] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4), October 2017. ISSN 0362-5915. doi: 10.1145/3134428. URL `https://doi.org/10.1145/3134428`.

[47] Tao Zhang, Tianqing Zhu, Renping Liu, and Wanlei Zhou. Correlated data in differential privacy: Definition and analysis, 2020.

Appendices

This page is intentionally left blank.

Appendix

Single GAN and Duo-GAN

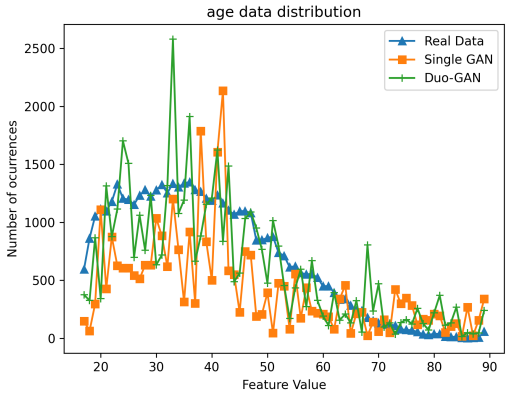


Figure 1: Distribution of values for the age continuous feature of the Adult dataset.

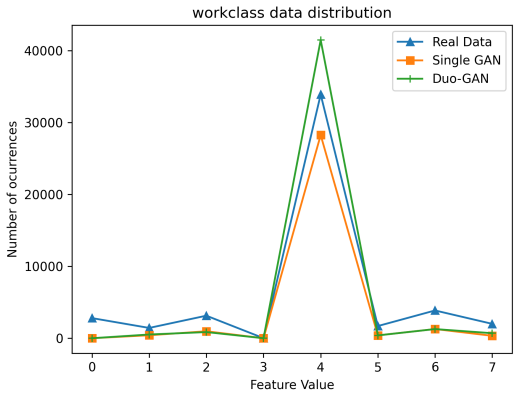


Figure 2: Distribution of values for the workclass categorical feature of the Adult dataset.

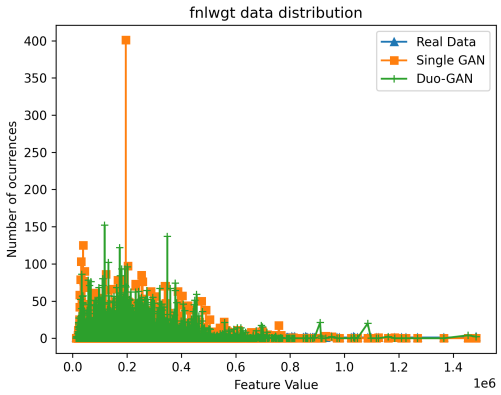


Figure 3: Distribution of values for the fnlwgt continuous feature of the Adult dataset.

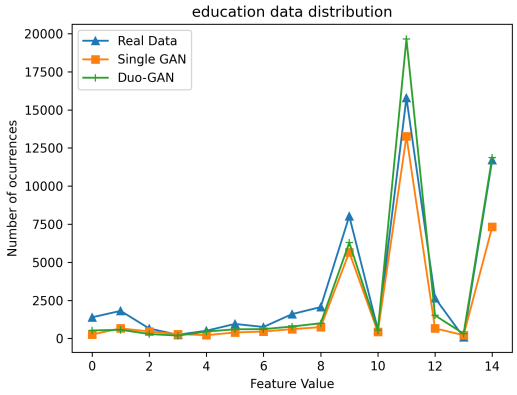


Figure 4: Distribution of values for the education categorical feature of the Adult dataset.

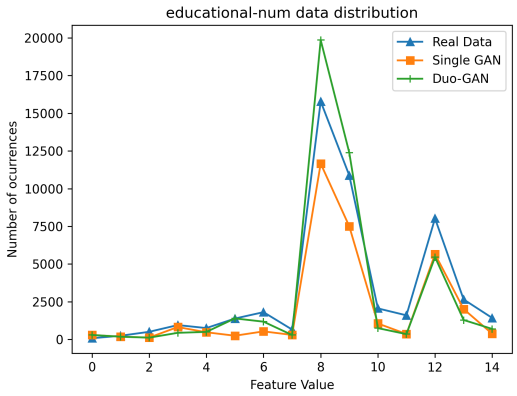


Figure 5: Distribution of values for the educational-num categorical feature of the Adult dataset.

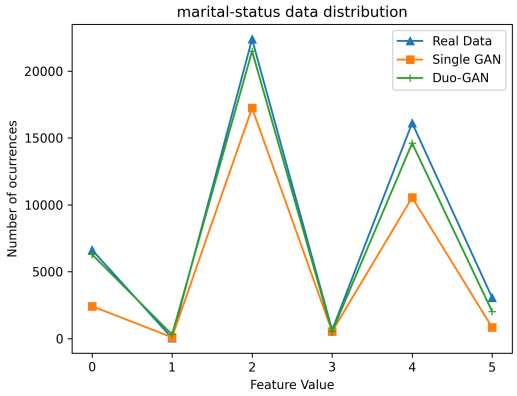


Figure 6: Distribution of values for the marital-status categorical feature of the Adult dataset.

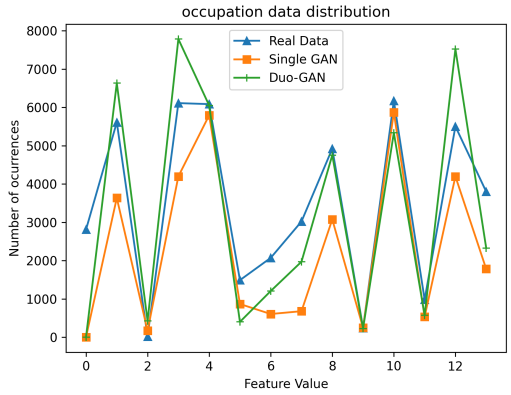


Figure 7: Distribution of values for the occupation categorical feature of the Adult dataset.

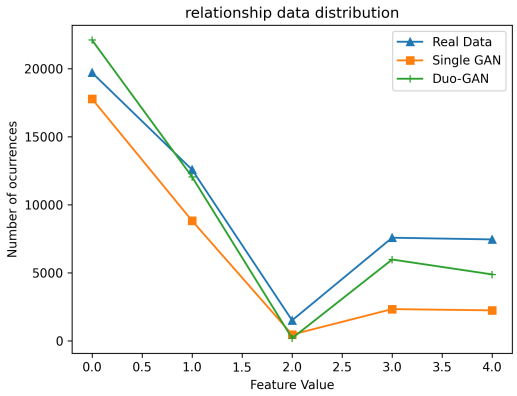


Figure 8: Distribution of values for the relationship categorical feature of the Adult dataset.

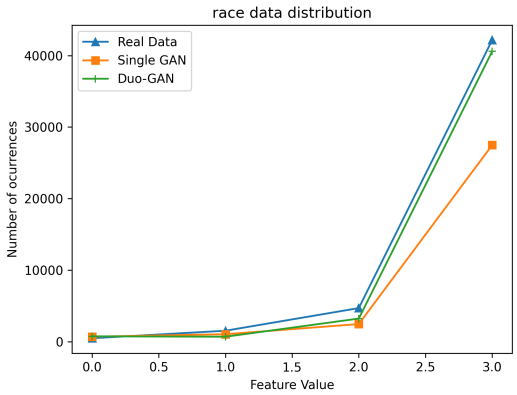


Figure 9: Distribution of values for the race categorical feature of the Adult dataset.

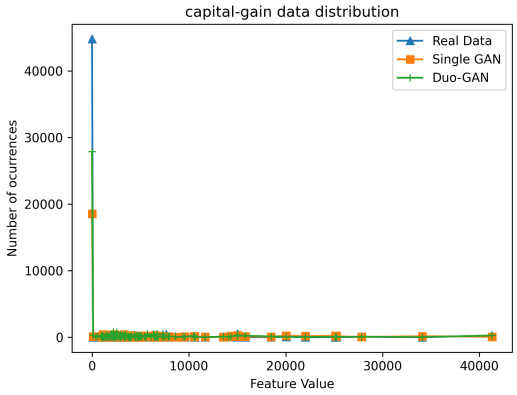


Figure 10: Distribution of values for the capital-gain continuous feature of the Adult dataset.

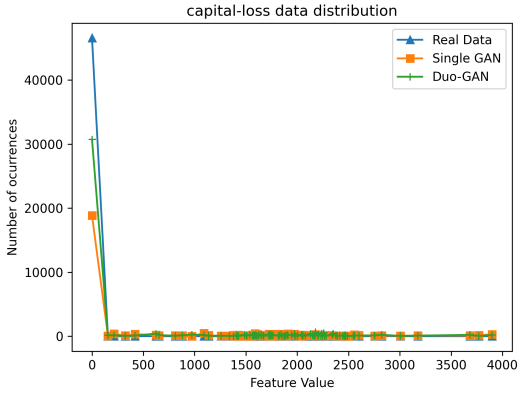


Figure 11: Distribution of values for the capital-loss continuous feature of the Adult dataset.

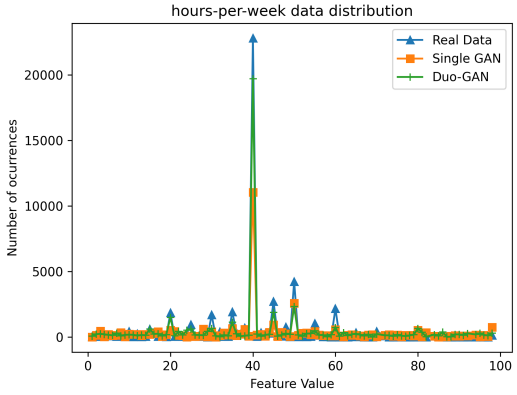


Figure 12: Distribution of values for the hours-per-week continuous feature of the Adult dataset.

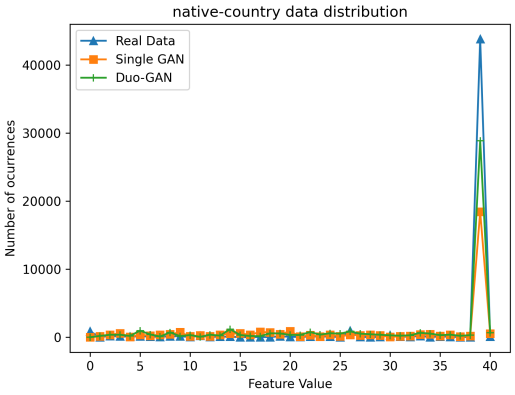


Figure 13: Distribution of values for the native-country categorical feature of the Adult dataset.

Table 1: Precision and Recal breakdown by class for data generated by model trained for 10 epochs for the Adult dataset.

Adaboost	Single GAN	Negative	0.7475	0.9688
		Positive	0.2173	0.0258
	Duo-GAN	Negative	0.8944	0.9688
		Positive	0.5015	0.7348
DecisionTree	Single GAN	Negative	0.7577	0.8383
		Positive	0.2959	0.2022
	Duo-GAN	Negative	0.9068	0.8383
		Positive	0.4649	0.7872
MLP	Single GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.7534	1.0000
		Positive	0.9570	0.0261
XGBoost	Single GAN	Negative	0.7465	0.9547
		Positive	0.2055	0.0349
	Duo-GAN	Negative	0.9286	0.9547
		Positive	0.4854	0.8397

Table 2: Precision and Recal breakdown by class for data generated by model trained for 20 epochs for the Adult dataset.

Adaboost	Single GAN	Negative	0.7487	0.9913
		Positive	0.2727	0.0097
	Duo-GAN	Negative	0.9112	0.9913
		Positive	0.5504	0.7714
DecisionTree	Single GAN	Negative	0.7614	0.8861
		Positive	0.3385	0.1735
	Duo-GAN	Negative	0.8939	0.8861
		Positive	0.4918	0.7371
MLP	Single GAN	Negative	0.7487	1.0000
		Positive	1.0000	0.0009
	Duo-GAN	Negative	0.7549	1.0000
		Positive	0.9667	0.0340
XGBoost	Single GAN	Negative	0.7529	0.9913
		Positive	0.5510	0.0317
	Duo-GAN	Negative	0.9151	0.9913
		Positive	0.5611	0.7805

Table 3: Precision and Recal breakdown by class for data generated by model trained for 100 epochs for the Adult dataset.

Adaboost	Single GAN	Negative	0.7584	0.8442
		Positive	0.3009	0.1996
	Duo-GAN	Negative	0.9071	0.8442
		Positive	0.5540	0.7576
DecisionTree	Single GAN	Negative	0.7431	0.6398
		Positive	0.2417	0.3417
	Duo-GAN	Negative	0.8831	0.6398
		Positive	0.5220	0.6896
MLP	Single GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.7486	1.0000
		Positive	1.0000	0.0003
XGBoost	Single GAN	Negative	0.7541	0.8114
		Positive	0.2746	0.2125
	Duo-GAN	Negative	0.9110	0.8114
		Positive	0.5516	0.7705

Table 4: Precision and Recal breakdown by class for data generated by model trained for 200 epochs for the Adult dataset.

Adaboost	Single GAN	Negative	0.8026	0.9188
		Positive	0.5752	0.3274
	Duo-GAN	Negative	0.8947	0.9188
		Positive	0.5952	0.7063
DecisionTree	Single GAN	Negative	0.8095	0.8030
		Positive	0.4273	0.4376
	Duo-GAN	Negative	0.8517	0.8030
		Positive	0.5098	0.5785
MLP	Single GAN	Negative	0.0000	0.0000
		Positive	0.2514	0.9997
	Duo-GAN	Negative	0.0194	0.0000
		Positive	0.2459	0.9704
XGBoost	Single GAN	Negative	0.7802	0.9260
		Positive	0.5040	0.2236
	Duo-GAN	Negative	0.8773	0.9260
		Positive	0.5623	0.6550

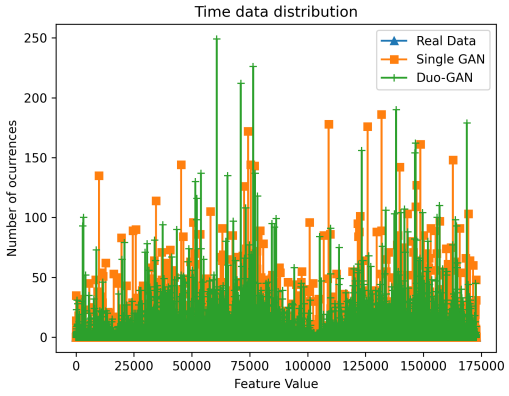


Figure 14: Distribution of values for the Time continuous feature of the Credit Card dataset.

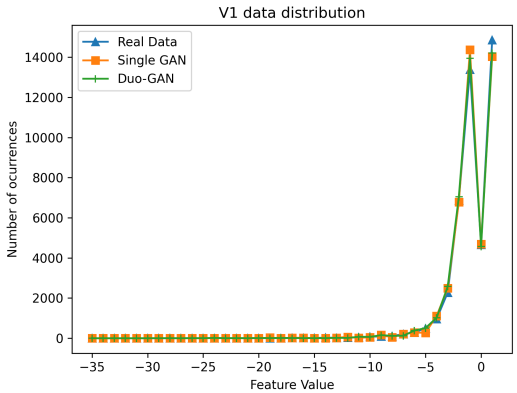


Figure 15: Distribution of values for the V1 continuous feature of the Credit Card dataset.

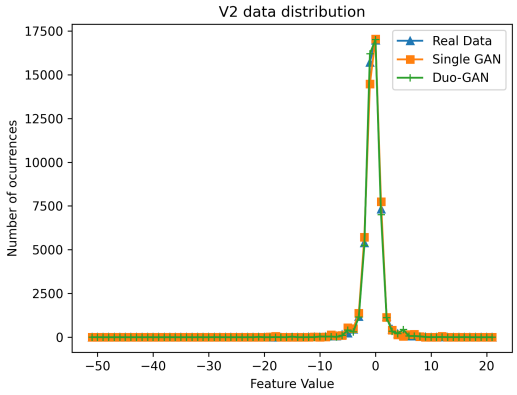


Figure 16: Distribution of values for the V2 continuous feature of the Credit Card dataset.

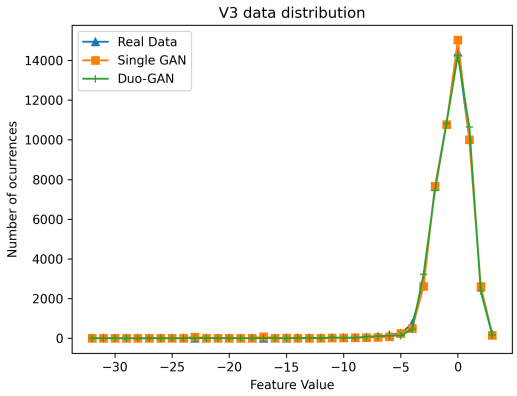


Figure 17: Distribution of values for the V3 continuous feature of the Credit Card dataset.

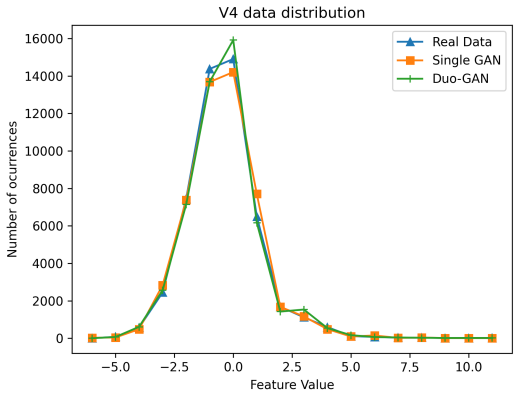


Figure 18: Distribution of values for the V4 continuous feature of the Credit Card dataset.

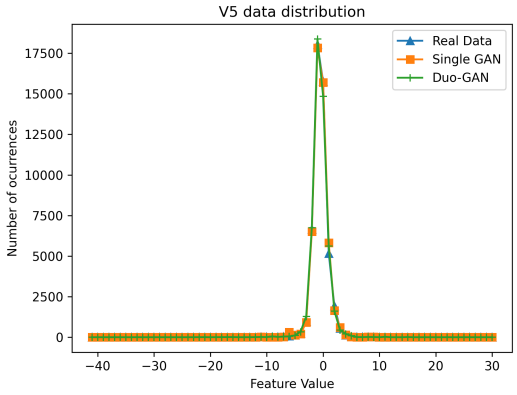


Figure 19: Distribution of values for the V5 continuous feature of the Credit Card dataset.

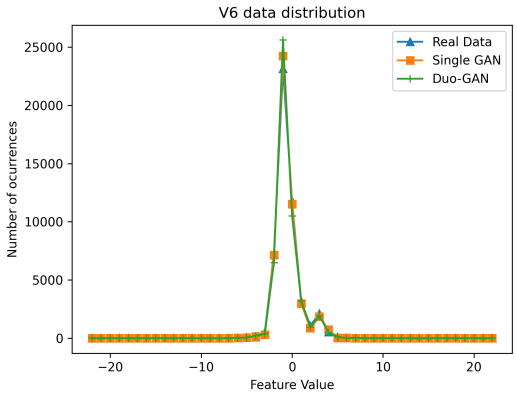


Figure 20: Distribution of values for the V6 continuous feature of the Credit Card dataset.

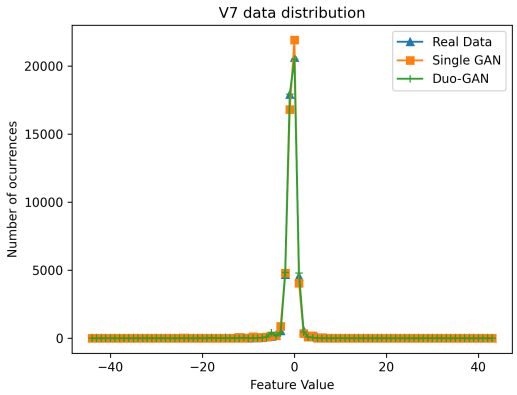


Figure 21: Distribution of values for the V7 continuous feature of the Credit Card dataset.

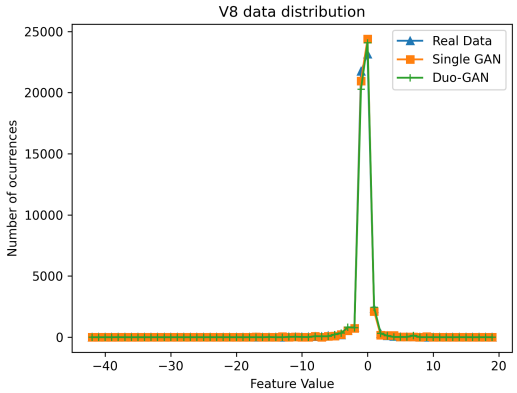


Figure 22: Distribution of values for the V8 continuous feature of the Credit Card dataset.

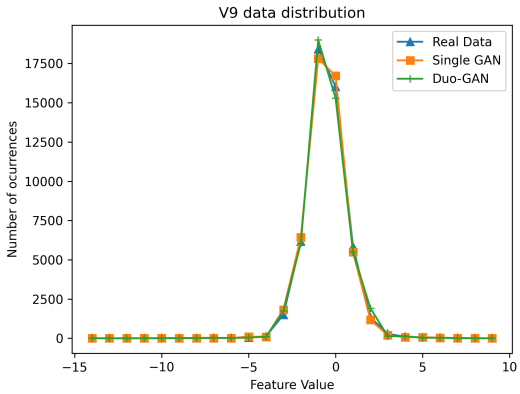


Figure 23: Distribution of values for the V9 continuous feature of the Credit Card dataset.

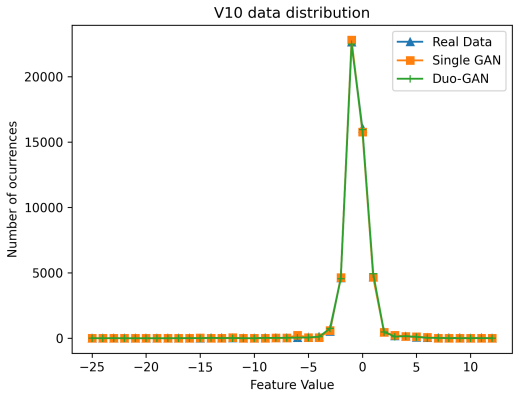


Figure 24: Distribution of values for the V10 continuous feature of the Credit Card dataset.

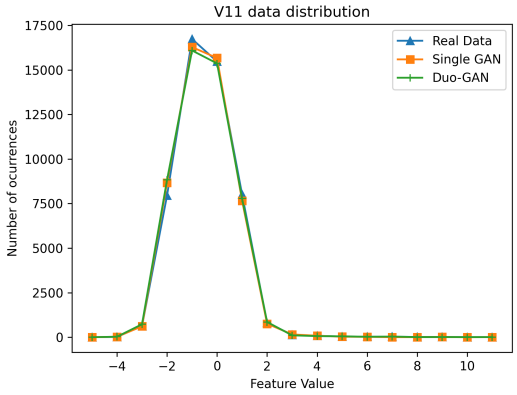


Figure 25: Distribution of values for the V11 continuous feature of the Credit Card dataset.

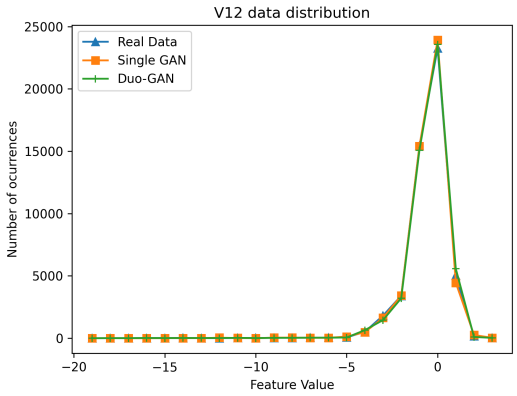


Figure 26: Distribution of values for the V12 continuous feature of the Credit Card dataset.

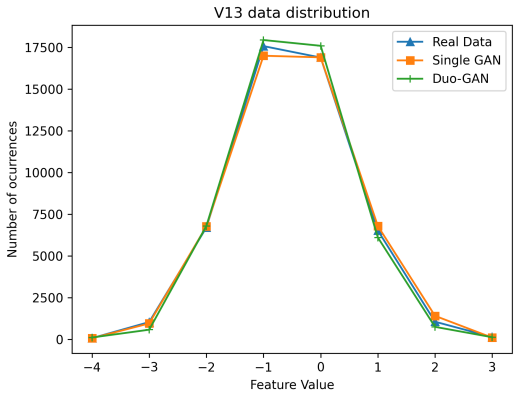


Figure 27: Distribution of values for the V13 continuous feature of the Credit Card dataset.

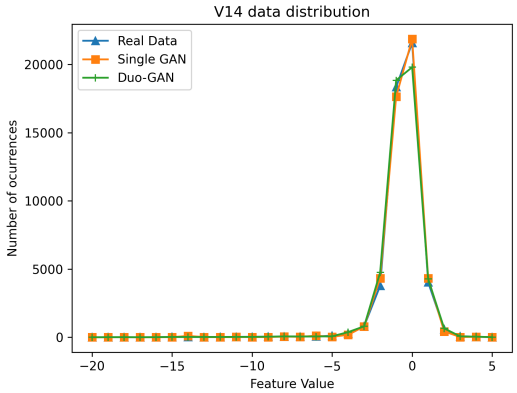


Figure 28: Distribution of values for the V14 continuous feature of the Credit Card dataset.

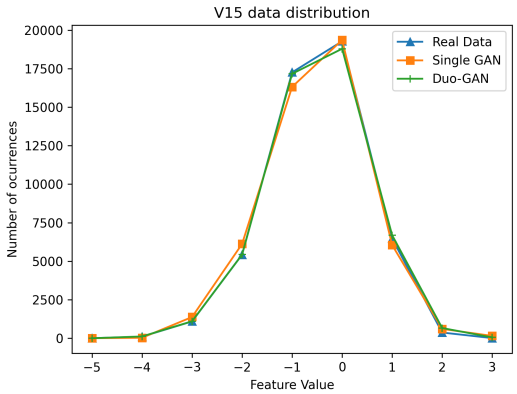


Figure 29: Distribution of values for the V15 continuous feature of the Credit Card dataset.

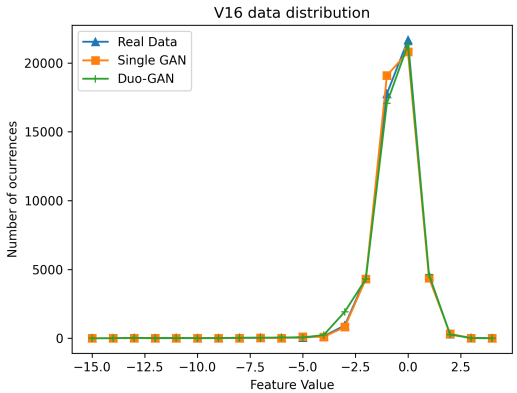


Figure 30: Distribution of values for the V16 continuous feature of the Credit Card dataset.

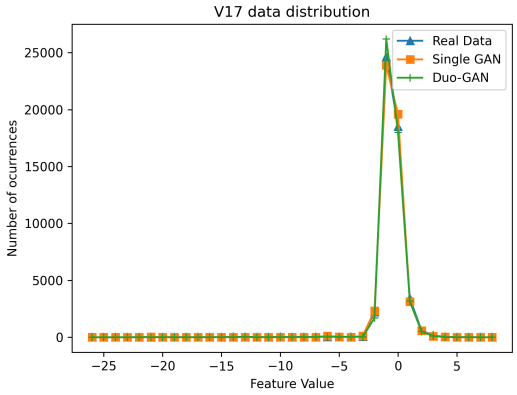


Figure 31: Distribution of values for the V17 continuous feature of the Credit Card dataset.

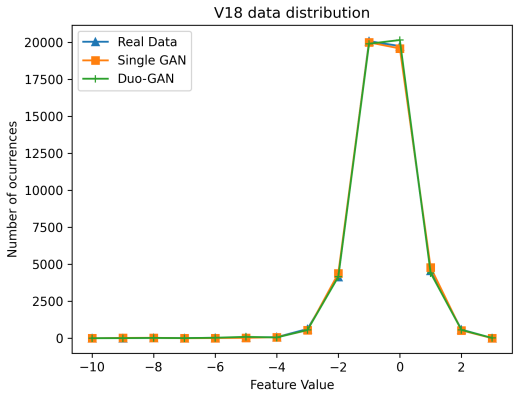


Figure 32: Distribution of values for the V18 continuous feature of the Credit Card dataset.

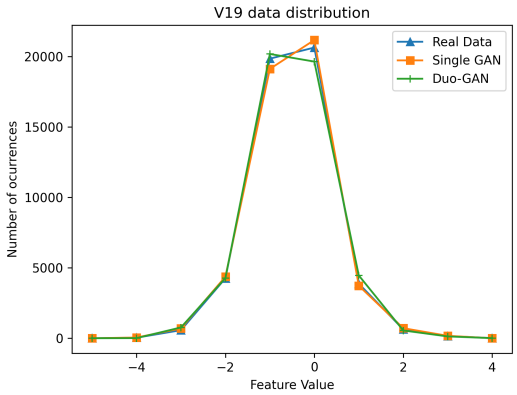


Figure 33: Distribution of values for the V19 continuous feature of the Credit Card dataset.

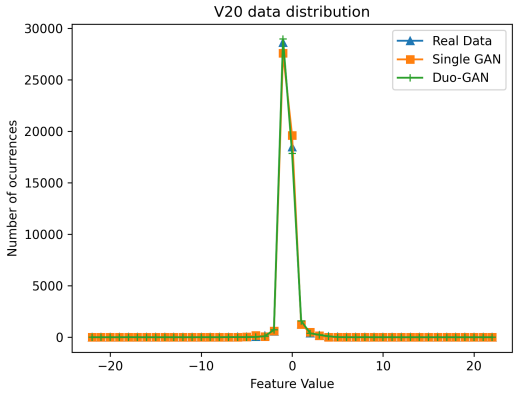


Figure 34: Distribution of values for the V20 continuous feature of the Credit Card dataset.

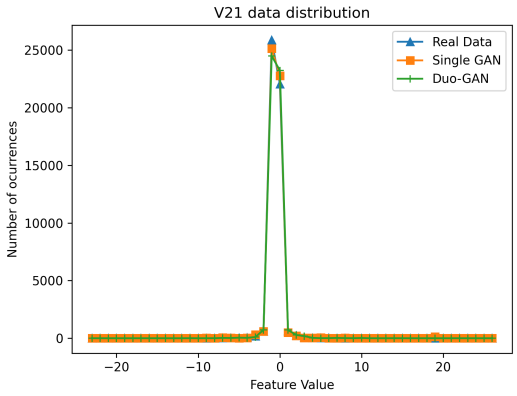


Figure 35: Distribution of values for the V21 continuous feature of the Credit Card dataset.

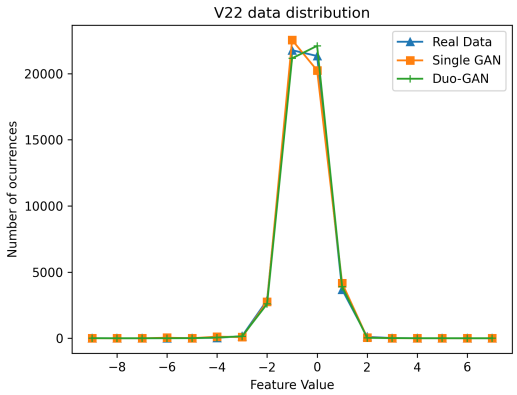


Figure 36: Distribution of values for the V22 continuous feature of the Credit Card dataset.

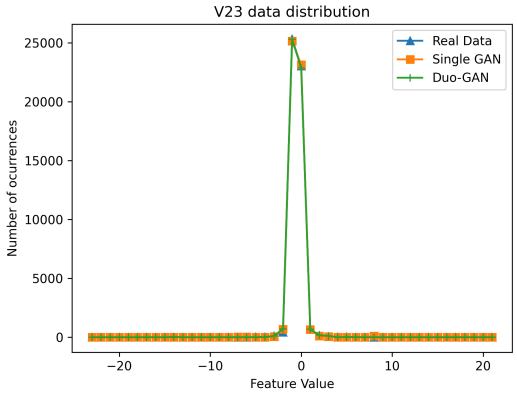


Figure 37: Distribution of values for the V23 continuous feature of the Credit Card dataset.

Figure 38: Distribution of values for the V24 continuous feature of the Credit Card dataset.

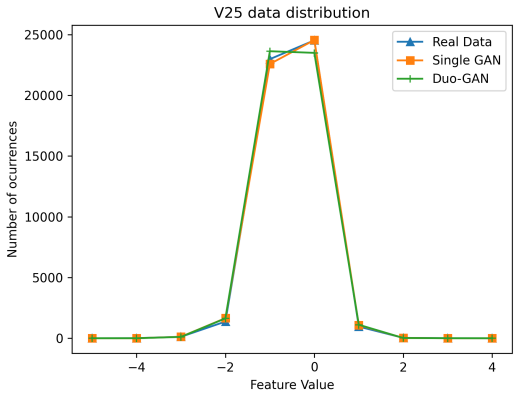


Figure 39: Distribution of values for the V25 continuous feature of the Credit Card dataset.

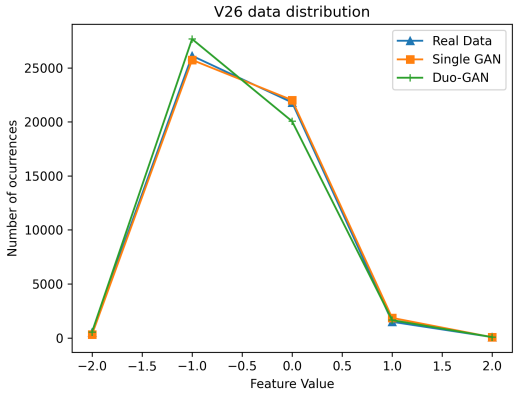


Figure 40: Distribution of values for the V26 continuous feature of the Credit Card dataset.

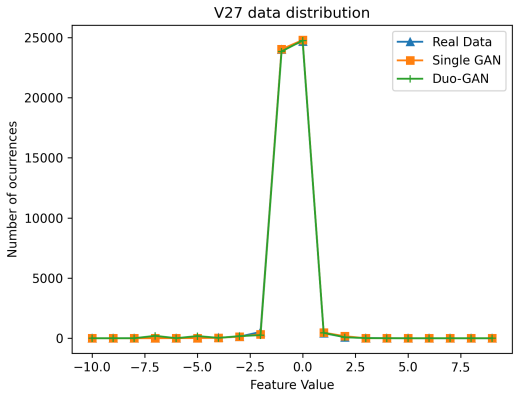


Figure 41: Distribution of values for the V27 continuous feature of the Credit Card dataset.

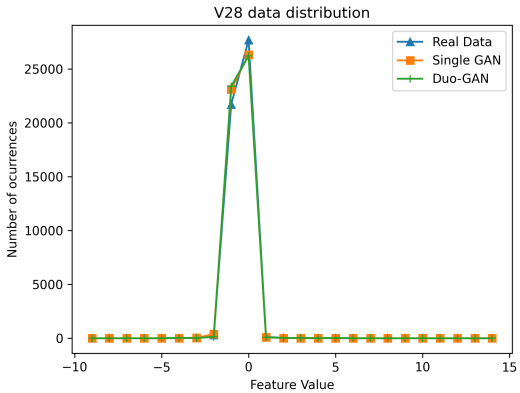


Figure 42: Distribution of values for the V28 continuous feature of the Credit Card dataset.

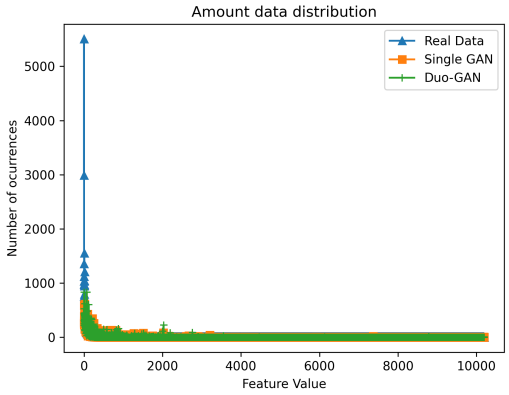


Figure 43: Distribution of values for the Amount continuous feature of the Credit Card dataset.

Table 5: Precision and Recal breakdown by class for data generated by model trained for 100 epochs for the Credit Card dataset.

Adaboost	Single GAN	Negative	0.9908	0.9993
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.9982	0.9993
		Positive	0.8538	0.8043
DecisionTree	Single GAN	Negative	0.9912	0.9972
		Positive	0.1458	0.0507
	Duo-GAN	Negative	0.9984	0.9972
		Positive	0.7500	0.8261
MLP	Single GAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
XGBoost	Single GAN	Negative	0.9908	0.9997
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.9983	0.9997
		Positive	0.8626	0.8188

Table 6: Precision and Recal breakdown by class for data generated by model trained for 200 epochs for the Credit Card dataset.

Adaboost	Single GAN	Negative	0.9762	0.1764
		Positive	0.0060	0.5362
	Duo-GAN	Negative	0.9983	0.1764
		Positive	0.8129	0.8188
DecisionTree	Single GAN	Negative	0.9888	0.3432
		Positive	0.0081	0.5797
	Duo-GAN	Negative	0.9986	0.3432
		Positive	0.5021	0.8551
MLP	Single GAN	Negative	0.9909	0.9983
		Positive	0.0385	0.0072
	Duo-GAN	Negative	0.9908	0.9983
		Positive	0.0000	0.0000
XGBoost	Single GAN	Negative	0.9769	0.2985
		Positive	0.0032	0.2391
	Duo-GAN	Negative	0.9986	0.2985
		Positive	0.6964	0.8478

Single WGAN and DW-GAN

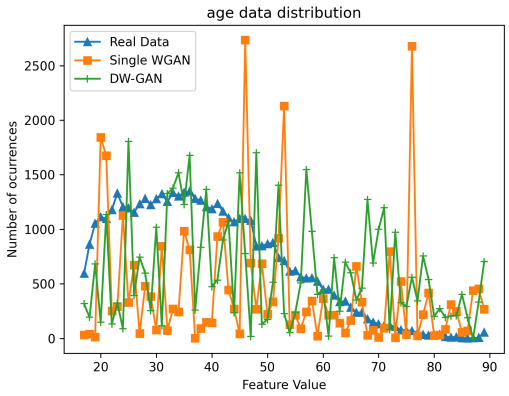


Figure 44: Distribution of values for the age continuous feature of the Adult dataset.

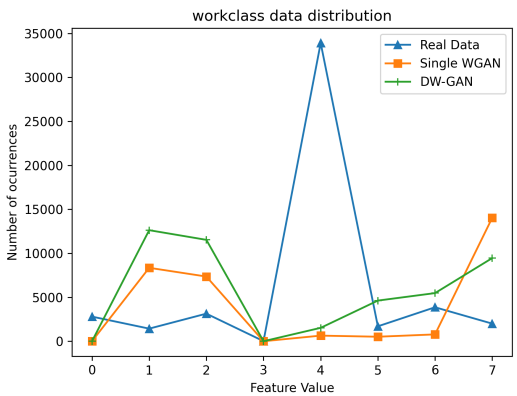


Figure 45: Distribution of values for the workclass categorical feature of the Adult dataset.

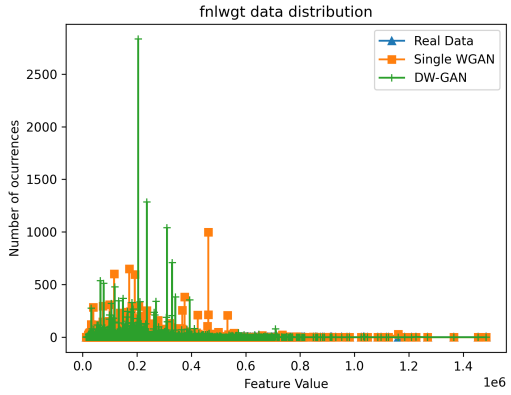


Figure 46: Distribution of values for the fnlwgt continuous feature of the Adult dataset.

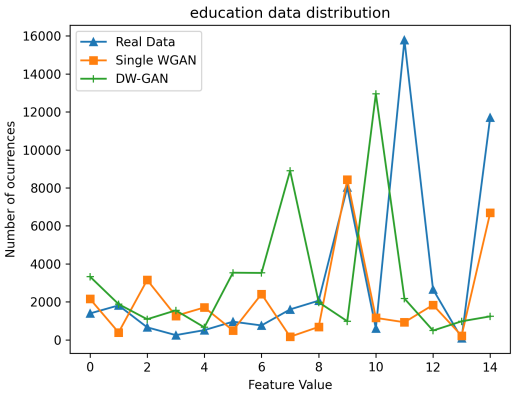


Figure 47: Distribution of values for the education categorical feature of the Adult dataset.

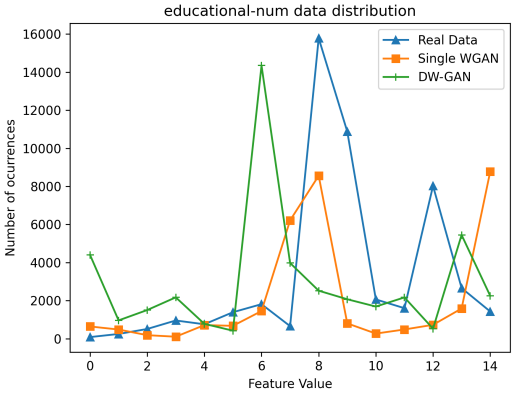


Figure 48: Distribution of values for the educational-num categorical feature of the Adult dataset.

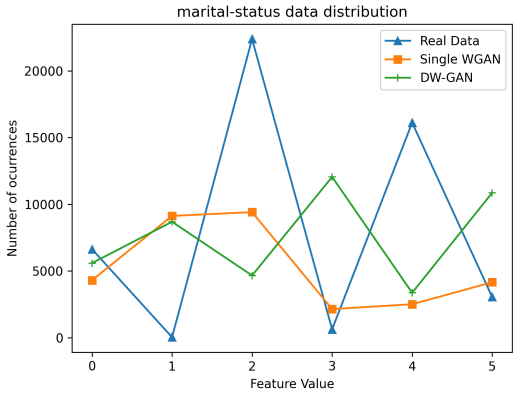


Figure 49: Distribution of values for the marital-status categorical feature of the Adult dataset.

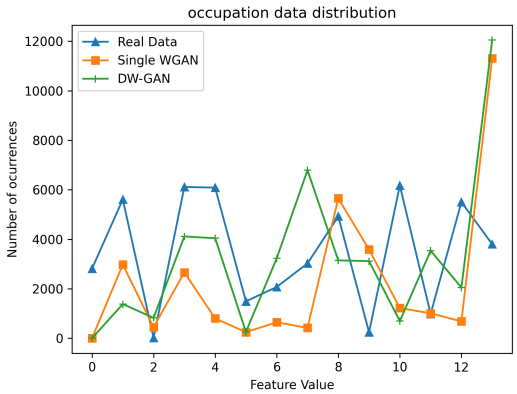


Figure 50: Distribution of values for the occupation categorical feature of the Adult dataset.

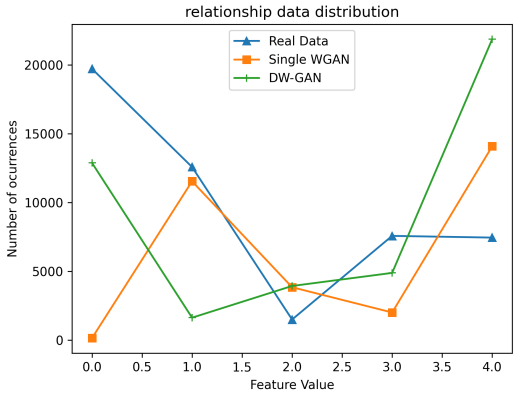


Figure 51: Distribution of values for the relationship categorical feature of the Adult dataset.

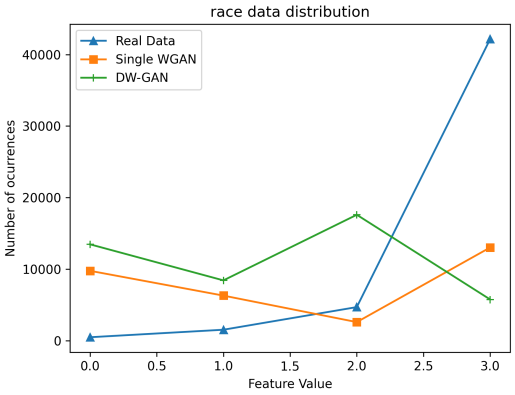


Figure 52: Distribution of values for the race categorical feature of the Adult dataset.

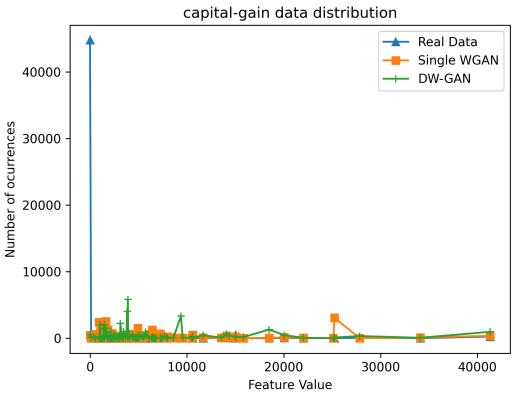


Figure 53: Distribution of values for the capital-gain continuous feature of the Adult dataset.

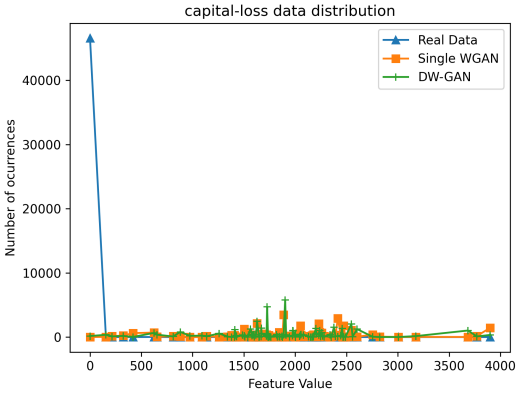


Figure 54: Distribution of values for the capital-loss continuous feature of the Adult dataset.

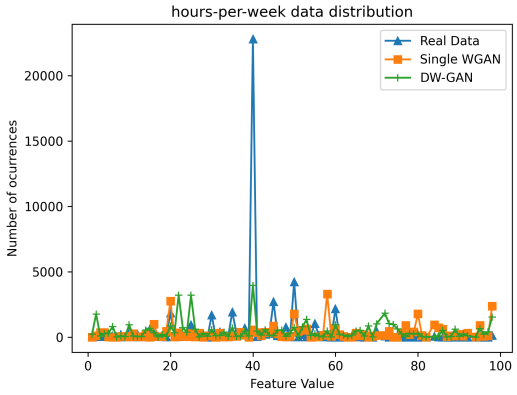


Figure 55: Distribution of values for the hours-per-week continuous feature of the Adult dataset.

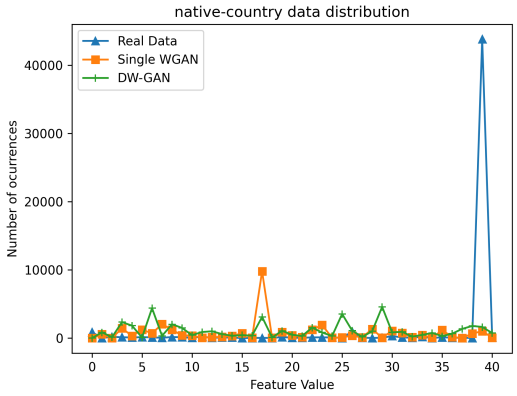


Figure 56: Distribution of values for the native-country categorical feature of the Adult dataset.

Table 7: Precision and Recal breakdown by class for data generated by model trained for 10 epochs for the Adult dataset.

Adaboost	Single WGAN	Negative	0.7412	0.9444
		Positive	0.1003	0.0185
	DW-GAN	Negative	0.7810	0.9444
		Positive	0.3798	0.3051
DecisionTree	Single WGAN	Negative	0.7850	0.6871
		Positive	0.3209	0.4399
	DW-GAN	Negative	0.5375	0.6871
		Positive	0.2173	0.7433
MLP	Single WGAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
XGBoost	Single WGAN	Negative	0.7364	0.8530
		Positive	0.1724	0.0911
	DW-GAN	Negative	0.8164	0.8530
		Positive	0.2555	0.9593

Table 8: Precision and Recal breakdown by class for data generated by model trained for 20 epochs for the Adult dataset.

Adaboost	Single WGAN	Negative	0.5639	0.0677
		Positive	0.2333	0.8441
	DW-GAN	Negative	0.6900	0.0677
		Positive	0.2337	0.7122
DecisionTree	Single WGAN	Negative	0.7777	0.2853
		Positive	0.2625	0.7573
	DW-GAN	Negative	0.6556	0.2853
		Positive	0.1855	0.4314
MLP	Single WGAN	Negative	0.7486	1.0000
		Positive	1.0000	0.0003
	DW-GAN	Negative	0.7709	1.0000
		Positive	0.9004	0.1193
XGBoost	Single WGAN	Negative	0.6056	0.3041
		Positive	0.1654	0.4106
	DW-GAN	Negative	0.6606	0.3041
		Positive	0.2185	0.6316

Table 9: Precision and Recal breakdown by class for data generated by model trained for 50 epochs for the Adult dataset.

Adaboost	Single WGAN	Negative	0.7388	0.9443
		Positive	0.0374	0.0064
	DW-GAN	Negative	0.7500	0.9443
		Positive	0.2515	0.9903
DecisionTree	Single WGAN	Negative	0.7482	0.7792
		Positive	0.2504	0.2195
	DW-GAN	Negative	0.5559	0.7792
		Positive	0.2472	0.9616
MLP	Single WGAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.7551	1.0000
		Positive	0.9675	0.0349
XGBoost	Single WGAN	Negative	0.7440	0.9654
		Positive	0.0977	0.0111
	DW-GAN	Negative	0.6935	0.9654
		Positive	0.2507	0.9821

Table 10: Precision and Recal breakdown by class for data generated by model trained for 100 epochs for the Adult dataset.

Adaboost	Single WGAN	Negative	0.7442	0.9656
		Positive	0.1051	0.0120
	DW-GAN	Negative	0.7203	0.9656
		Positive	0.2503	0.9531
DecisionTree	Single WGAN	Negative	0.7021	0.5143
		Positive	0.1952	0.3505
	DW-GAN	Negative	0.2216	0.5143
		Positive	0.2169	0.8095
MLP	Single WGAN	Negative	0.7489	1.0000
		Positive	1.0000	0.0021
	DW-GAN	Negative	0.7568	1.0000
		Positive	0.9618	0.0443
XGBoost	Single WGAN	Negative	0.7299	0.8789
		Positive	0.0821	0.0322
	DW-GAN	Negative	0.7818	0.8789
		Positive	0.2547	0.9232

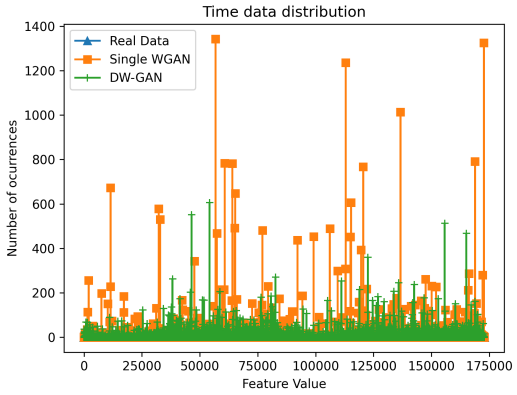


Figure 57: Distribution of values for the Time continuous feature of the Credit Card dataset.

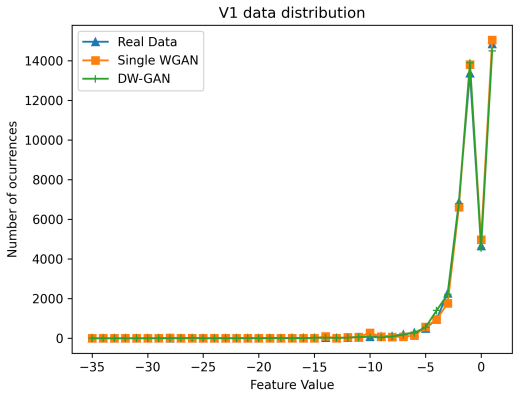


Figure 58: Distribution of values for the V1 continuous feature of the Credit Card dataset.

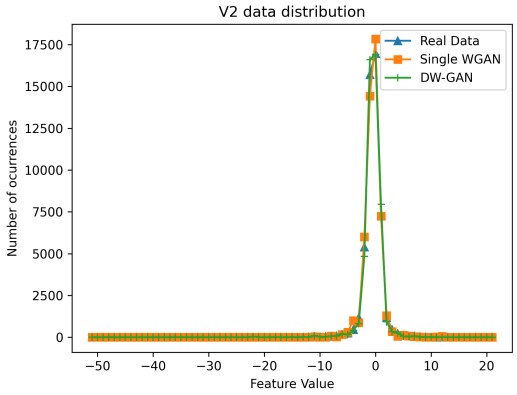


Figure 59: Distribution of values for the V2 continuous feature of the Credit Card dataset.

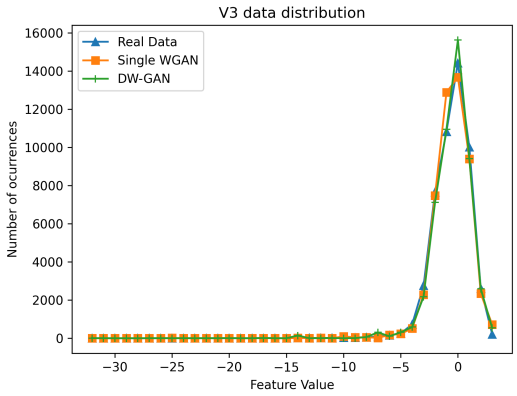


Figure 60: Distribution of values for the V3 continuous feature of the Credit Card dataset.

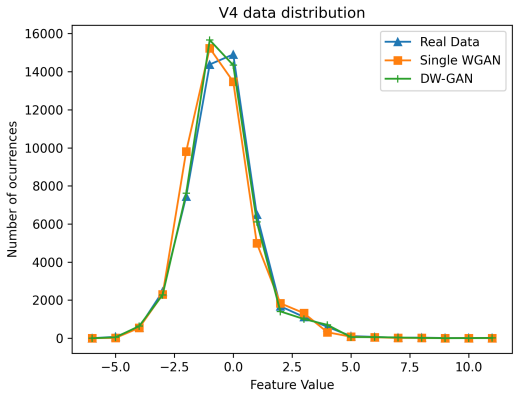


Figure 61: Distribution of values for the V4 continuous feature of the Credit Card dataset.

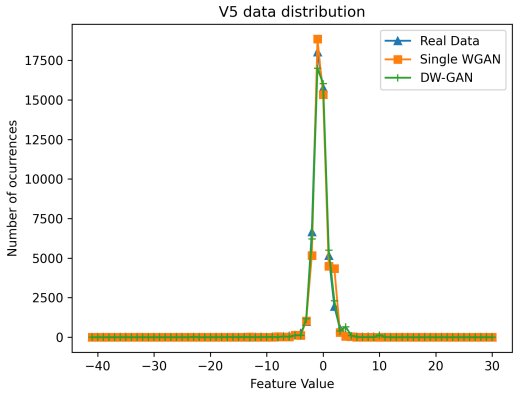


Figure 62: Distribution of values for the V5 continuous feature of the Credit Card dataset.

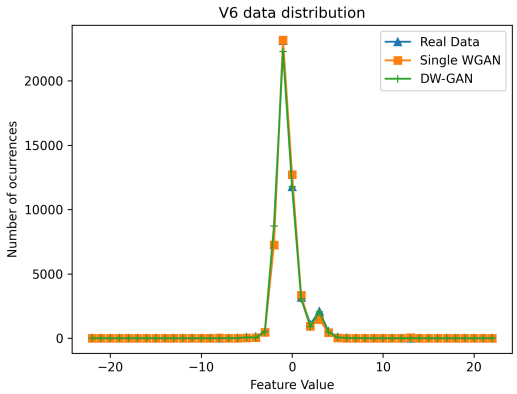


Figure 63: Distribution of values for the V6 continuous feature of the Credit Card dataset.

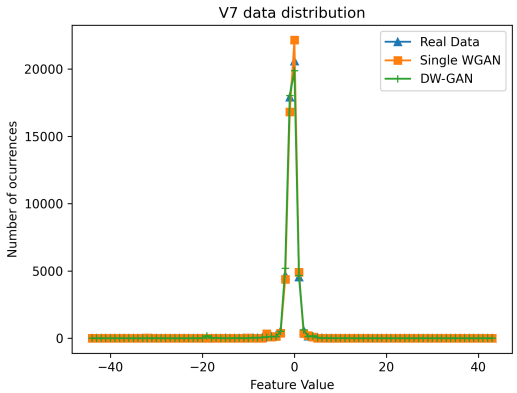


Figure 64: Distribution of values for the V7 continuous feature of the Credit Card dataset.

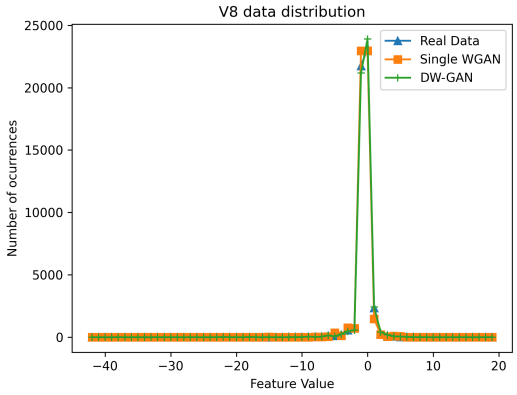


Figure 65: Distribution of values for the V8 continuous feature of the Credit Card dataset.

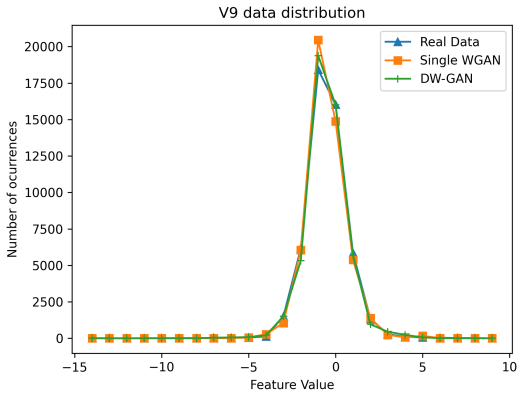


Figure 66: Distribution of values for the V9 continuous feature of the Credit Card dataset.

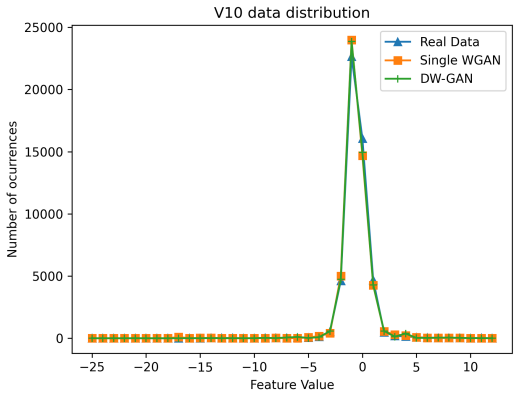


Figure 67: Distribution of values for the V10 continuous feature of the Credit Card dataset.

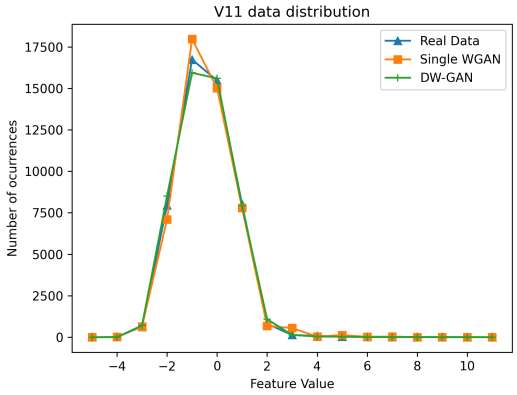


Figure 68: Distribution of values for the V11 continuous feature of the Credit Card dataset.

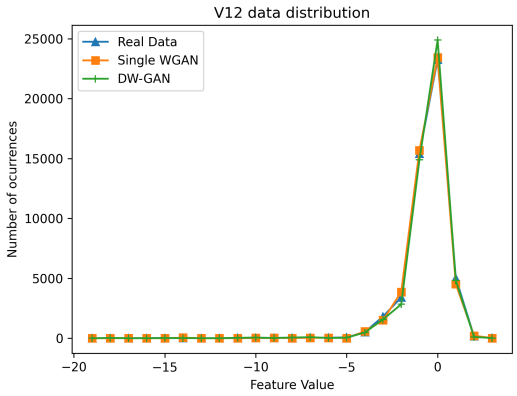


Figure 69: Distribution of values for the V12 continuous feature of the Credit Card dataset.

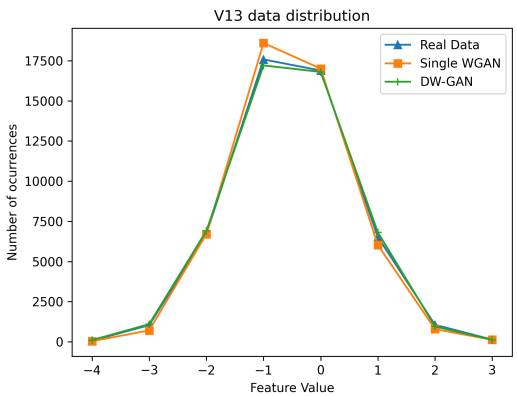


Figure 70: Distribution of values for the V13 continuous feature of the Credit Card dataset.

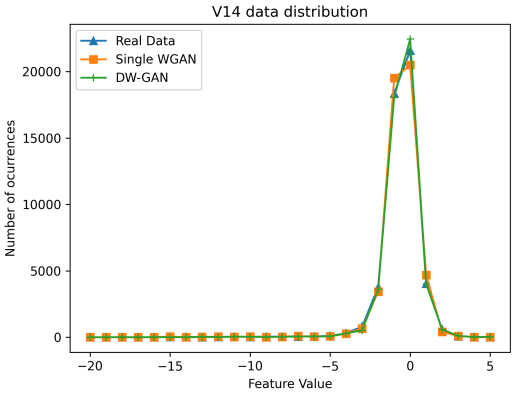


Figure 71: Distribution of values for the V14 continuous feature of the Credit Card dataset.

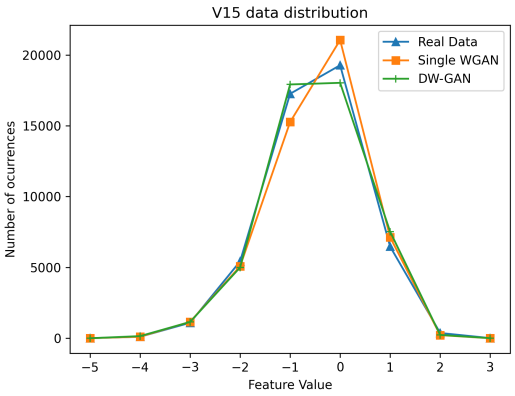


Figure 72: Distribution of values for the V15 continuous feature of the Credit Card dataset.

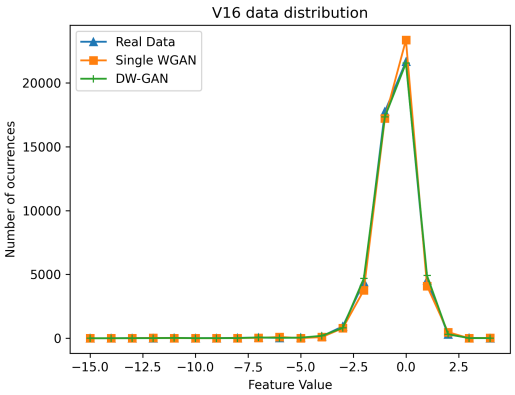


Figure 73: Distribution of values for the V16 continuous feature of the Credit Card dataset.

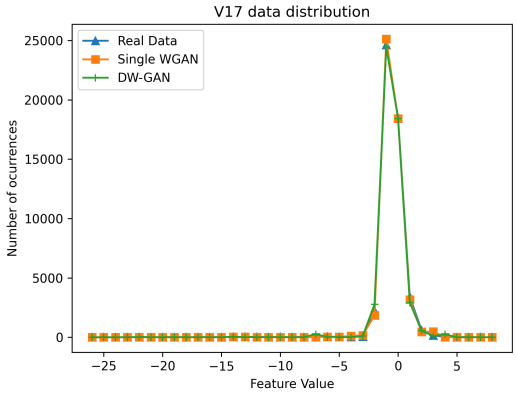


Figure 74: Distribution of values for the V17 continuous feature of the Credit Card dataset.

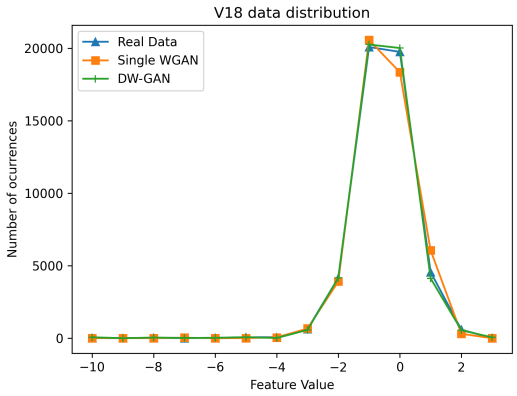


Figure 75: Distribution of values for the V18 continuous feature of the Credit Card dataset.

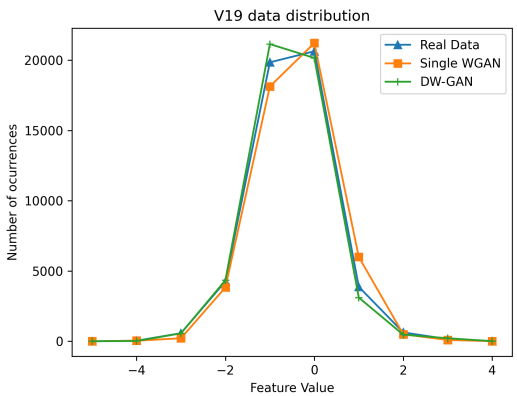


Figure 76: Distribution of values for the V19 continuous feature of the Credit Card dataset.

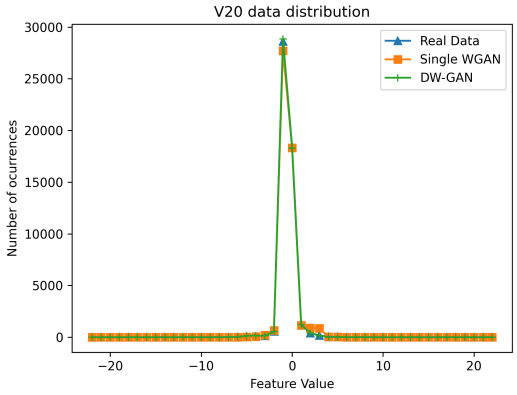


Figure 77: Distribution of values for the V20 continuous feature of the Credit Card dataset.

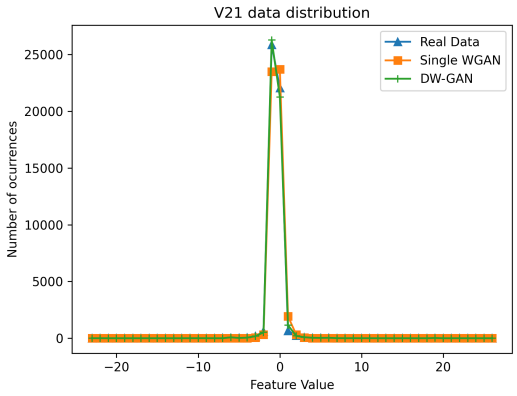


Figure 78: Distribution of values for the V21 continuous feature of the Credit Card dataset.

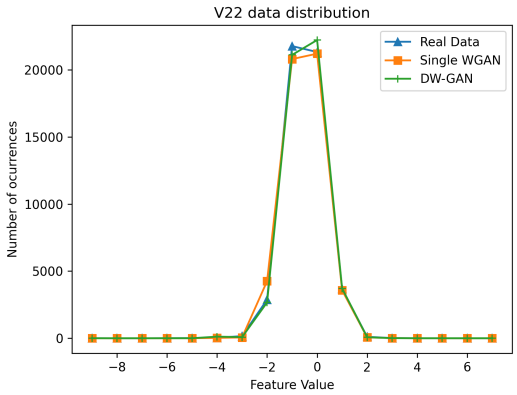


Figure 79: Distribution of values for the V22 continuous feature of the Credit Card dataset.

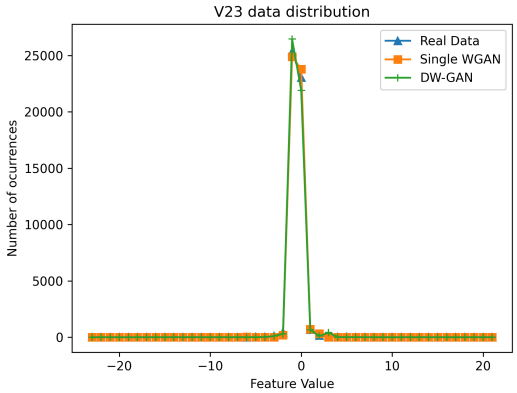


Figure 80: Distribution of values for the V23 continuous feature of the Credit Card dataset.

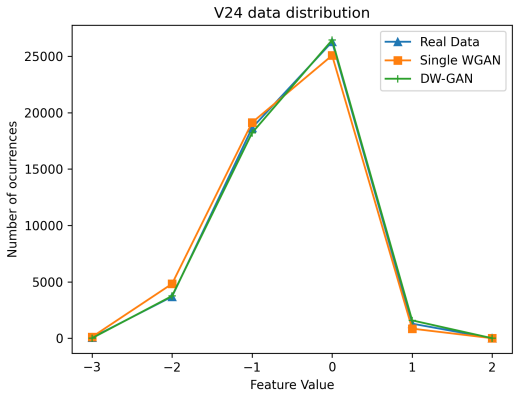


Figure 81: Distribution of values for the V24 continuous feature of the Credit Card dataset.

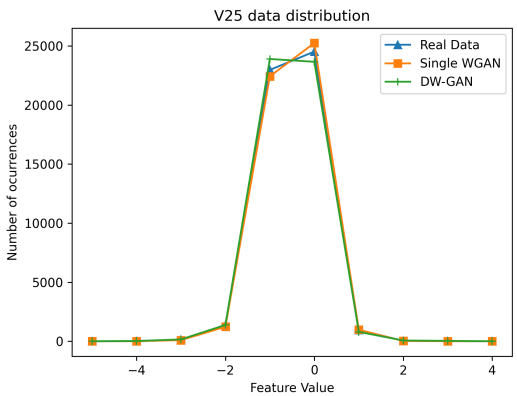


Figure 82: Distribution of values for the V25 continuous feature of the Credit Card dataset.

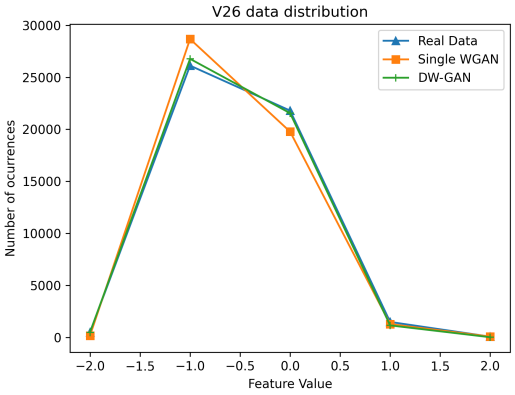


Figure 83: Distribution of values for the V26 continuous feature of the Credit Card dataset.

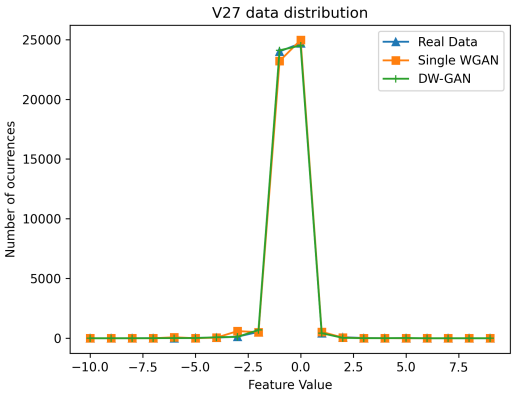


Figure 84: Distribution of values for the V27 continuous feature of the Credit Card dataset.

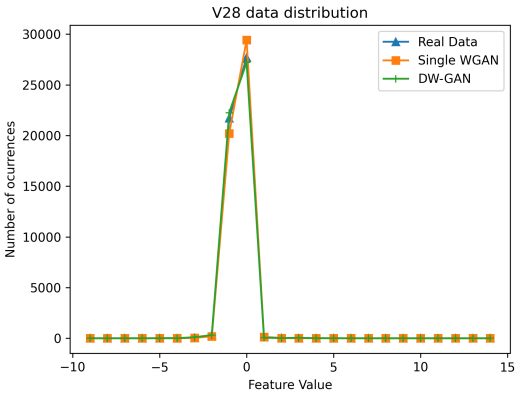


Figure 85: Distribution of values for the V28 continuous feature of the Credit Card dataset.

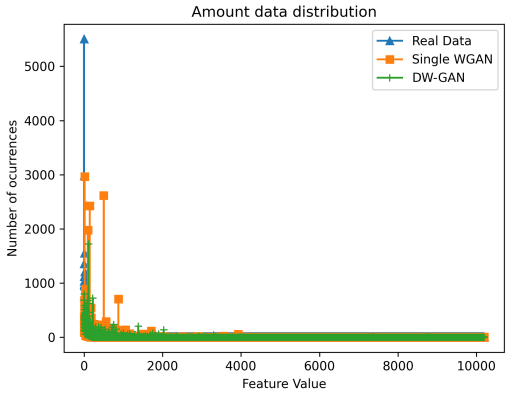


Figure 86: Distribution of values for the Amount continuous feature of the Credit Card dataset.

Table 11: Precision and Recal breakdown by class for data generated by model trained for 10 epochs for the Credit Card dataset.

Adaboost	Single WGAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9983	1.0000
		Positive	0.9113	0.8188
DecisionTree	Single WGAN	Negative	0.9941	0.9784
		Positive	0.1394	0.3768
	DW-GAN	Negative	0.9980	0.9784
		Positive	0.8308	0.7826
MLP	Single WGAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
XGBoost	Single WGAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9982	1.0000
		Positive	0.9174	0.8043

Table 12: Precision and Recal breakdown by class for data generated by model trained for 20 epochs for the Credit Card dataset.

Adaboost	Single WGAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9981	1.0000
		Positive	0.9649	0.7971
DecisionTree	Single WGAN	Negative	0.9934	0.9856
		Positive	0.1575	0.2899
	DW-GAN	Negative	0.9979	0.9856
		Positive	0.9217	0.7681
MLP	Single WGAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
XGBoost	Single WGAN	Negative	0.9908	1.0000
		Positive	0.0000	0.0000
	DW-GAN	Negative	0.9981	1.0000
		Positive	0.9565	0.7971

Pre Feature Engineering

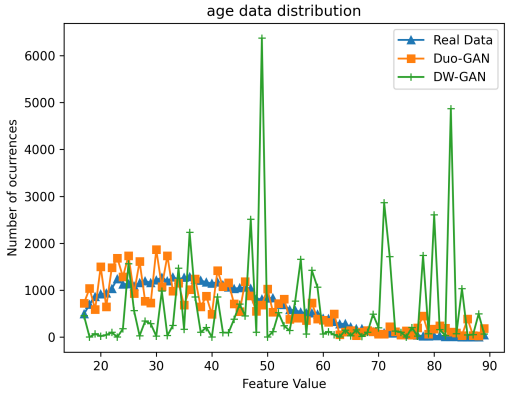


Figure 87: Distribution of values for the age continuous feature of the Extended Adult dataset.

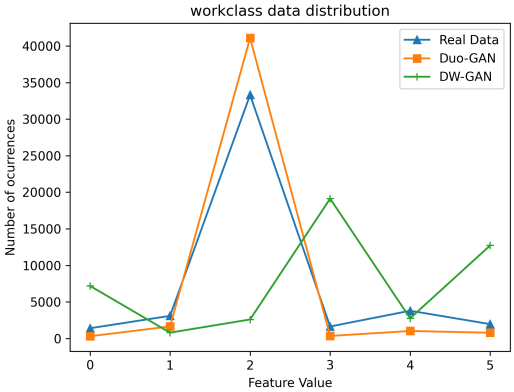


Figure 88: Distribution of values for the workclass categorical feature of the Extended Adult dataset.

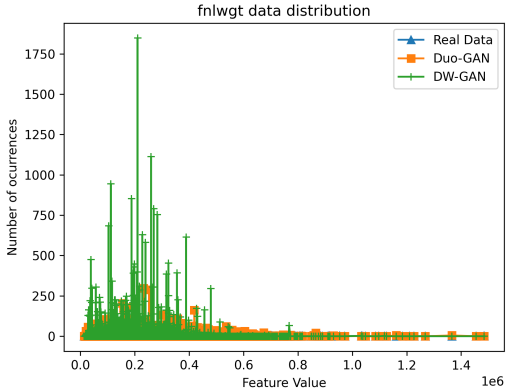


Figure 89: Distribution of values for the fnlwgt continuous feature of the Extended Adult dataset.

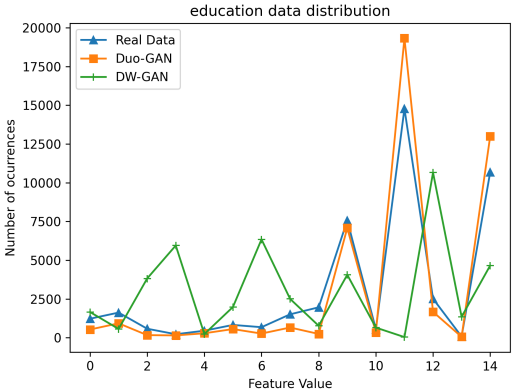


Figure 90: Distribution of values for the education categorical feature of the Extended Adult dataset.

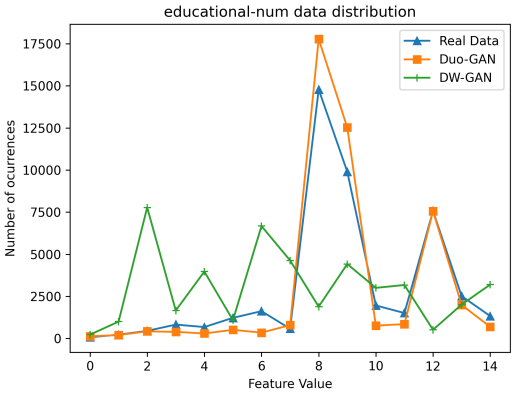


Figure 91: Distribution of values for the educational-num categorical feature of the Extended Adult dataset.

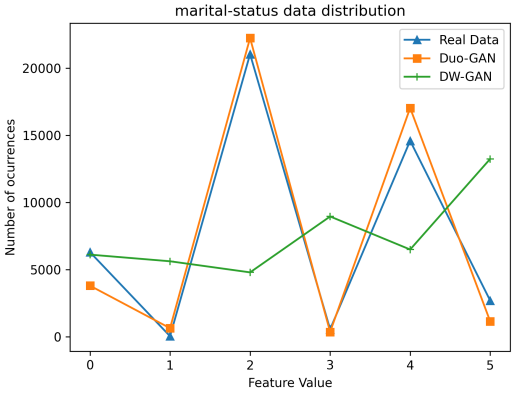


Figure 92: Distribution of values for the marital-status categorical feature of the Extended Adult dataset.

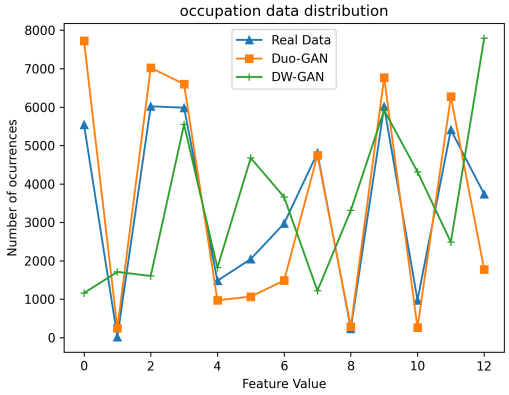


Figure 93: Distribution of values for the occupation categorical feature of the Extended Adult dataset.

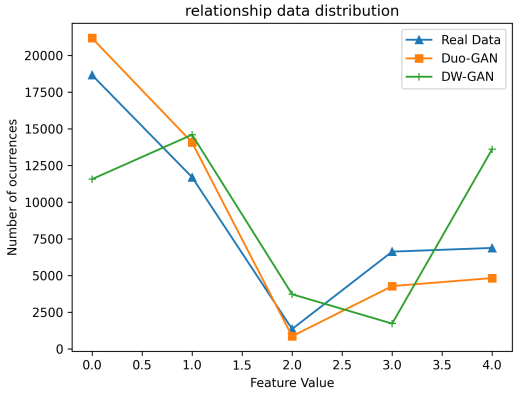


Figure 94: Distribution of values for the relationship categorical feature of the Extended Adult dataset.

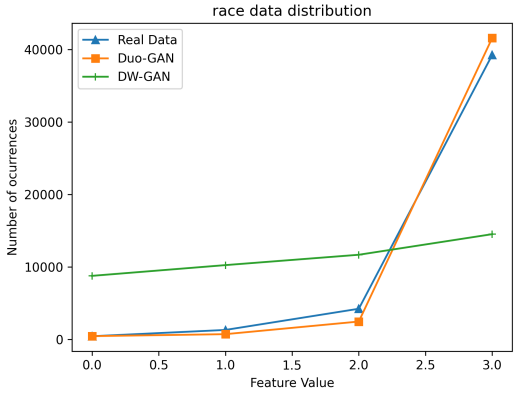


Figure 95: Distribution of values for the race categorical feature of the Extended Adult dataset.

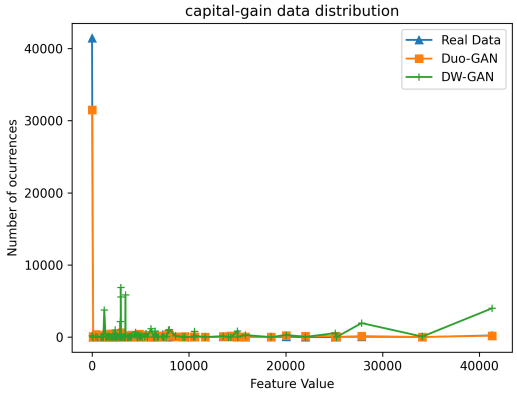


Figure 96: Distribution of values for the capital-gain continuous feature of the Extended Adult dataset.

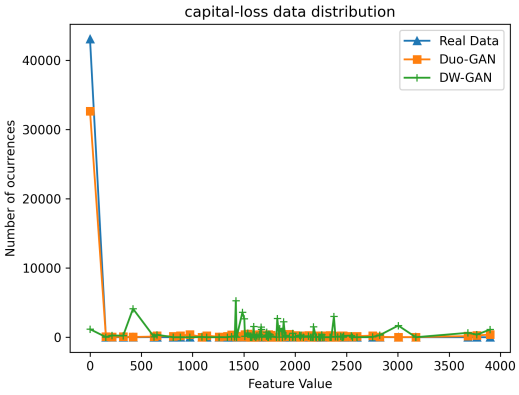


Figure 97: Distribution of values for the capital-loss continuous feature of the Extended Adult dataset.

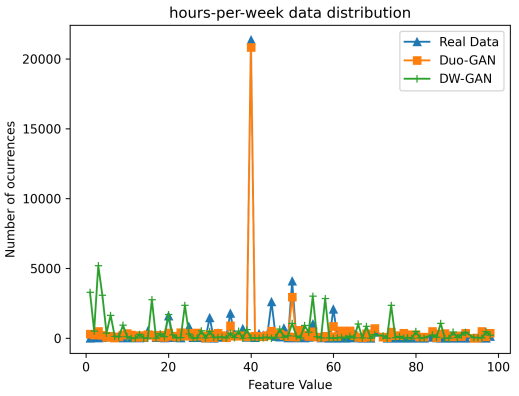


Figure 98: Distribution of values for the hours-per-week continuous feature of the Extended Adult dataset.

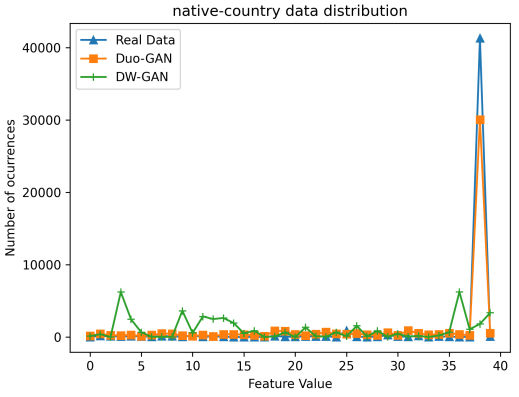


Figure 99: Distribution of values for the native-country categorical feature of the Extended Adult dataset.

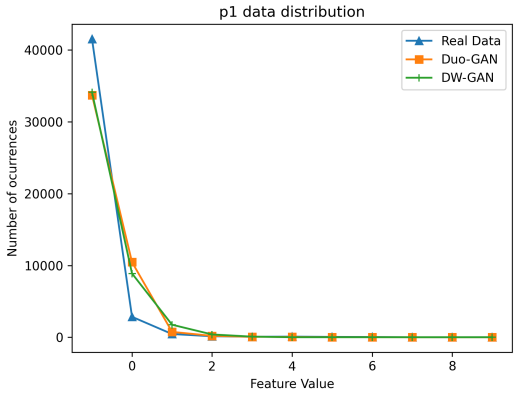


Figure 100: Distribution of values for the p1 continuous feature of the Extended Adult dataset.

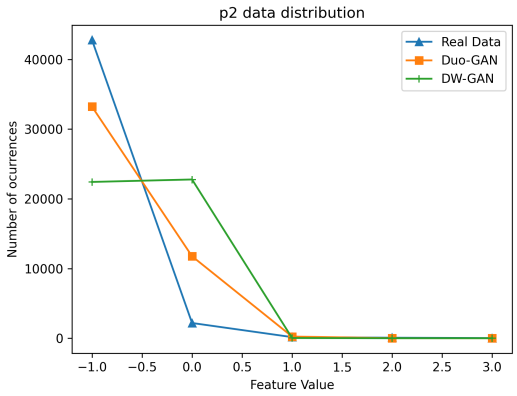


Figure 101: Distribution of values for the p2 continuous feature of the Extended Adult dataset.

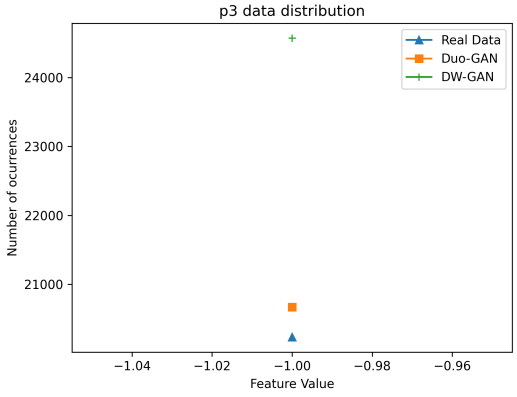


Figure 102: Distribution of values for the p3 continuous feature of the Extended Adult dataset.

Table 13: Precision and Recal breakdown by class for data generated by model trained for 10 epochs for the Extended Adult dataset.

Adaboost	DW-GAN	Negative	0.7524	0.9775
		Positive	0.3887	0.0425
	Duo-GAN	Negative	0.8999	0.9775
		Positive	0.5897	0.7251
DecisionTree	DW-GAN	Negative	0.6815	0.5427
		Positive	0.1526	0.2450
	Duo-GAN	Negative	0.7724	0.5427
		Positive	0.4748	0.1823
MLP	DW-GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.7489	1.0000
		Positive	1.0000	0.0021
XGBoost	DW-GAN	Negative	0.7619	0.8929
		Positive	0.3469	0.1694
	Duo-GAN	Negative	0.9144	0.8929
		Positive	0.5836	0.7732

Table 14: Precision and Recal breakdown by class for data generated by model trained for 20 epochs for the Extended Adult dataset.

Adaboost	DW-GAN	Negative	0.7191	0.6891
		Positive	0.1770	0.1990
	Duo-GAN	Negative	0.8819	0.6891
		Positive	0.6432	0.6498
DecisionTree	DW-GAN	Negative	0.7757	0.8892
		Positive	0.4159	0.2348
	Duo-GAN	Negative	0.8856	0.8892
		Positive	0.5682	0.6826
MLP	DW-GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.7499	1.0000
		Positive	1.0000	0.0073
XGBoost	DW-GAN	Negative	0.7791	0.9043
		Positive	0.4539	0.2368
	Duo-GAN	Negative	0.8820	0.9043
		Positive	0.6694	0.6445

Table 15: Precision and Recal breakdown by class for data generated by model trained for 100 epochs for the Extended Adult dataset.

Adaboost	DW-GAN	Negative	0.8936	0.3714
		Positive	0.3170	0.8684
	Duo-GAN	Negative	0.8859	0.3714
		Positive	0.5841	0.6791
DecisionTree	DW-GAN	Negative	0.5477	0.0130
		Positive	0.2479	0.9681
	Duo-GAN	Negative	0.8573	0.0130
		Positive	0.5829	0.5730
MLP	DW-GAN	Negative	0.7549	0.9996
		Positive	0.9669	0.0343
	Duo-GAN	Negative	0.7485	0.9996
		Positive	0.0000	0.0000
XGBoost	DW-GAN	Negative	0.8878	0.3833
		Positive	0.3180	0.8558
	Duo-GAN	Negative	0.8861	0.3833
		Positive	0.6316	0.6674

Table 16: Precision and Recal breakdown by class for data generated by model trained for 200 epochs for the Extended Adult dataset.

Adaboost	DW-GAN	Negative	0.8912	0.0339
		Positive	0.2557	0.9877
	Duo-GAN	Negative	0.8816	0.0339
		Positive	0.5407	0.6776
DecisionTree	DW-GAN	Negative	0.7915	0.1065
		Positive	0.2563	0.9165
	Duo-GAN	Negative	0.8460	0.1065
		Positive	0.4805	0.5703
MLP	DW-GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.0000	1.0000
		Positive	0.2490	0.9868
XGBoost	DW-GAN	Negative	0.8324	0.1516
		Positive	0.2647	0.9091
	Duo-GAN	Negative	0.8696	0.1516
		Positive	0.5333	0.6375

Post Feature Engineering

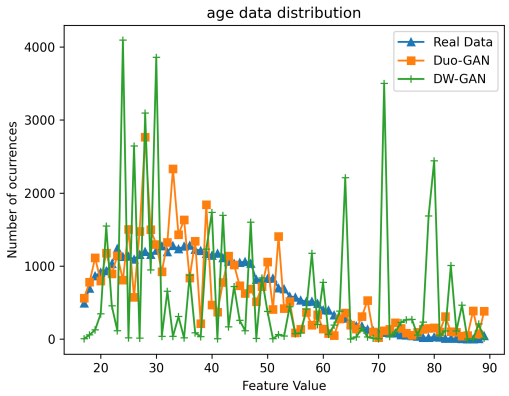


Figure 103: Distribution of values for the age continuous feature of the Extended Adult dataset.

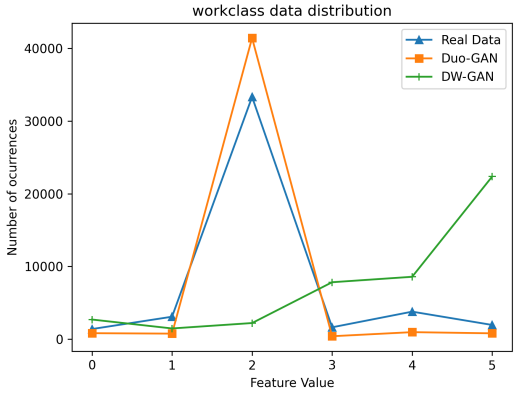


Figure 104: Distribution of values for the workclass categorical feature of the Extended Adult dataset.

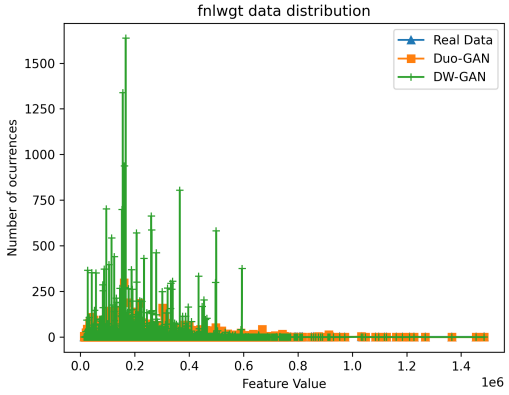


Figure 105: Distribution of values for the fnlwgt continuous feature of the Extended Adult dataset.

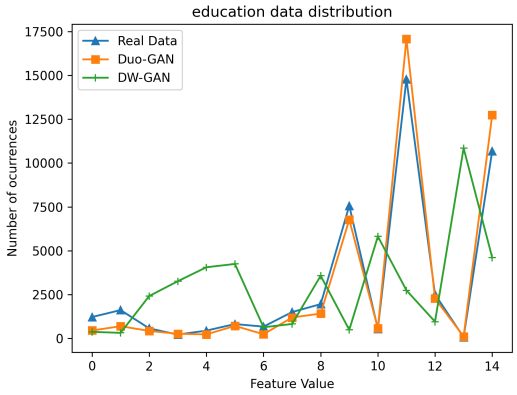


Figure 106: Distribution of values for the education categorical feature of the Extended Adult dataset.

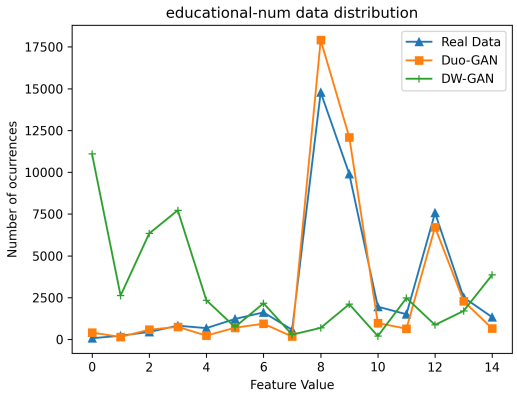


Figure 107: Distribution of values for the educational-num categorical feature of the Extended Adult dataset.

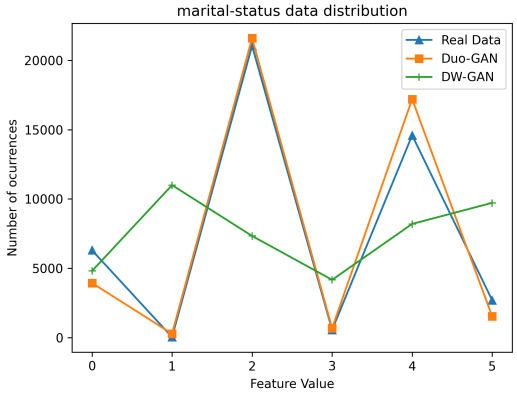


Figure 108: Distribution of values for the marital-status categorical feature of the Extended Adult dataset.

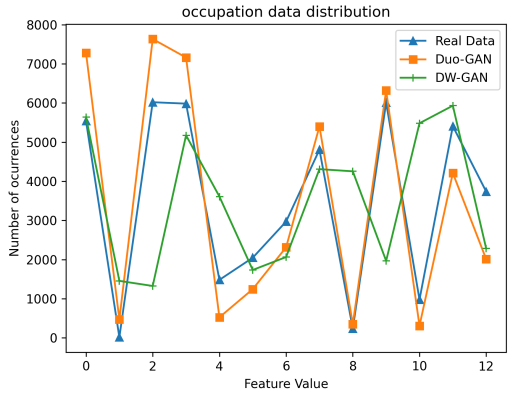


Figure 109: Distribution of values for the occupation categorical feature of the Extended Adult dataset.

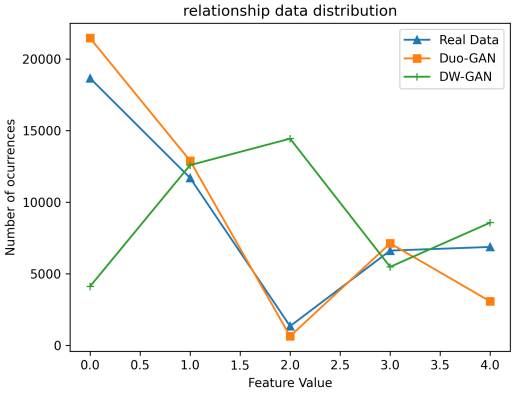


Figure 110: Distribution of values for the relationship categorical feature of the Extended Adult dataset.

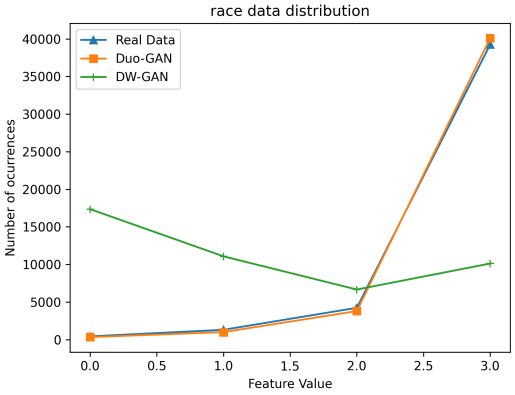


Figure 111: Distribution of values for the race categorical feature of the Extended Adult dataset.

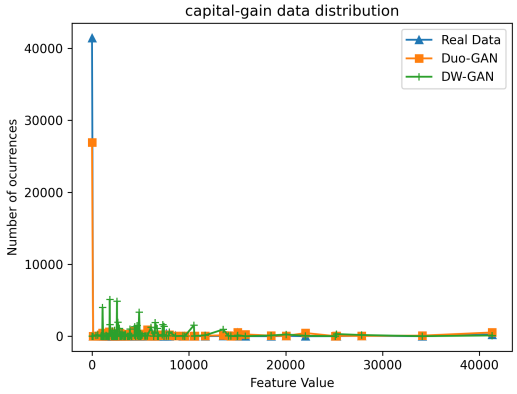


Figure 112: Distribution of values for the capital-gain continuous feature of the Extended Adult dataset.

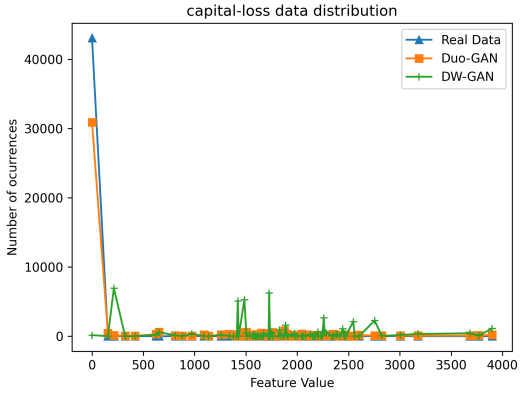


Figure 113: Distribution of values for the capital-loss continuous feature of the Extended Adult dataset.

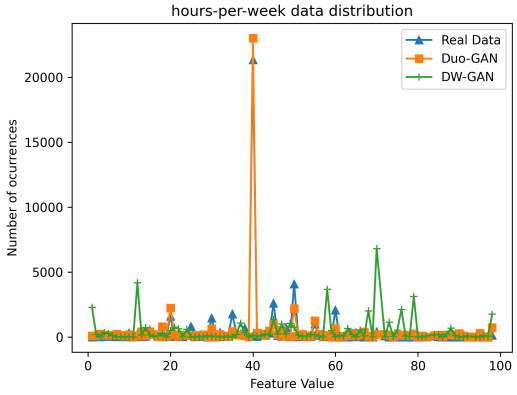


Figure 114: Distribution of values for the hours-per-week continuous feature of the Extended Adult dataset.

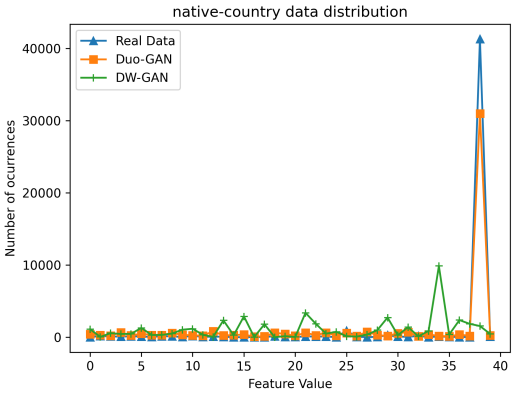


Figure 115: Distribution of values for the native-country categorical feature of the Extended Adult dataset.

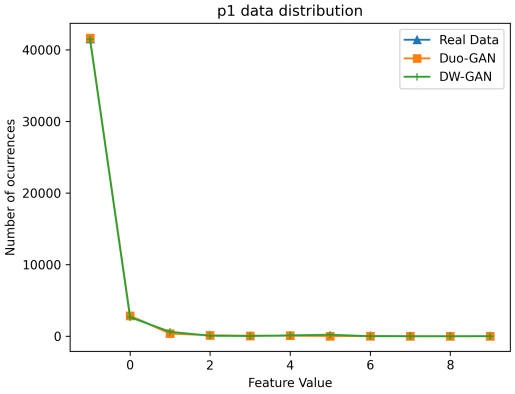


Figure 116: Distribution of values for the p1 continuous feature of the Extended Adult dataset.

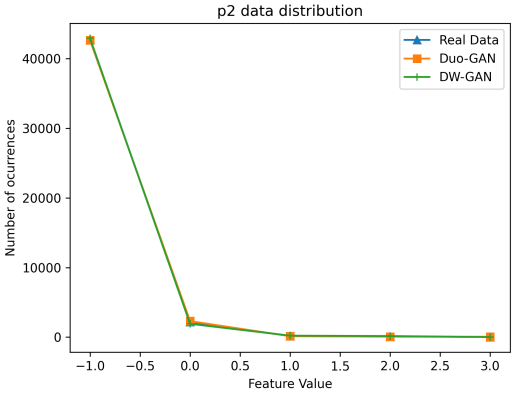


Figure 117: Distribution of values for the p2 continuous feature of the Extended Adult dataset.

Table 17: Precision and Recal breakdown by class for data generated by model trained for 10 epochs for the Extended Adult dataset.

Adaboost	DW-GAN	Negative	0.7926	0.9828
		Positive	0.8205	0.2345
	Duo-GAN	Negative	0.8908	0.9828
		Positive	0.6049	0.6905
DecisionTree	DW-GAN	Negative	0.9434	0.6091
		Positive	0.4337	0.8913
	Duo-GAN	Negative	0.8740	0.6091
		Positive	0.5247	0.6568
MLP	DW-GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.7509	1.0000
		Positive	1.0000	0.0129
XGBoost	DW-GAN	Negative	0.8080	0.9855
		Positive	0.8755	0.3030
	Duo-GAN	Negative	0.8964	0.9855
		Positive	0.6035	0.7098

Table 18: Precision and Recal breakdown by class for data generated by model trained for 20 epochs for the Extended Adult dataset.

Adaboost	DW-GAN	Negative	0.7754	0.9919
		Positive	0.8579	0.1451
	Duo-GAN	Negative	0.9107	0.9919
		Positive	0.5498	0.7699
DecisionTree	DW-GAN	Negative	0.8798	0.6604
		Positive	0.4198	0.7315
	Duo-GAN	Negative	0.9001	0.6604
		Positive	0.5052	0.7515
MLP	DW-GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.7564	1.0000
		Positive	0.9664	0.0422
XGBoost	DW-GAN	Negative	0.7896	0.9382
		Positive	0.5819	0.2562
	Duo-GAN	Negative	0.9256	0.9382
		Positive	0.5593	0.8121

Table 19: Precision and Recal breakdown by class for data generated by model trained for 100 epochs for the Extended Adult dataset.

Adaboost	DW-GAN	Negative	0.7804	0.9656
		Positive	0.6513	0.1911
	Duo-GAN	Negative	0.8994	0.9656
		Positive	0.5654	0.7298
DecisionTree	DW-GAN	Negative	0.7597	0.0657
		Positive	0.2523	0.9382
	Duo-GAN	Negative	0.8943	0.0657
		Positive	0.5295	0.7242
MLP	DW-GAN	Negative	0.7512	1.0000
		Positive	1.0000	0.0141
	Duo-GAN	Negative	0.7505	1.0000
		Positive	1.0000	0.0106
XGBoost	DW-GAN	Negative	0.7866	0.7663
		Positive	0.3541	0.3813
	Duo-GAN	Negative	0.9155	0.7663
		Positive	0.5563	0.7828

Table 20: Precision and Recal breakdown by class for data generated by model trained for 200 epochs for the Extended Adult dataset.

Adaboost	DW-GAN	Negative	0.9295	0.6619
		Positive	0.4581	0.8505
	Duo-GAN	Negative	0.8762	0.6619
		Positive	0.6507	0.6272
DecisionTree	DW-GAN	Negative	0.9295	0.6619
		Positive	0.4581	0.8505
	Duo-GAN	Negative	0.8217	0.6619
		Positive	0.5154	0.4449
MLP	DW-GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
	Duo-GAN	Negative	0.7485	1.0000
		Positive	0.0000	0.0000
XGBoost	DW-GAN	Negative	0.9295	0.6619
		Positive	0.4581	0.8505
	Duo-GAN	Negative	0.8706	0.6619
		Positive	0.6231	0.6128