



Article

TabFairGAN: Fair Tabular Data Generation with Generative Adversarial Networks

Amirarsalan Rajabi ¹ and Ozlem Ozmen Garibay ^{1,2,*}

¹ Department of Computer Science, University of Central Florida, Orlando, FL 32816, USA; amirarsalan@knights.ucf.edu

² Department of Industrial Engineering and Management Systems, University of Central Florida, Orlando, FL 32816, USA

* Correspondence: ozlem@ucf.edu

Abstract: With the increasing reliance on automated decision making, the issue of algorithmic fairness has gained increasing importance. In this paper, we propose a Generative Adversarial Network for tabular data generation. The model includes two phases of training. In the first phase, the model is trained to accurately generate synthetic data similar to the reference dataset. In the second phase we modify the value function to add fairness constraint, and continue training the network to generate data that is both accurate and fair. We test our results in both cases of unconstrained, and constrained fair data generation. We show that using a fairly simple architecture and applying quantile transformation of numerical attributes the model achieves promising performance. In the unconstrained case, i.e., when the model is only trained in the first phase and is only meant to generate accurate data following the same joint probability distribution of the real data, the results show that the model beats the state-of-the-art GANs proposed in the literature to produce synthetic tabular data. Furthermore, in the constrained case in which the first phase of training is followed by the second phase, we train the network and test it on four datasets studied in the fairness literature and compare our results with another state-of-the-art pre-processing method, and present the promising results that it achieves. Comparing to other studies utilizing GANs for fair data generation, our model is comparably more stable by using only one critic, and also by avoiding major problems of original GAN model, such as mode-dropping and non-convergence.

Keywords: fairness in artificial intelligence; generative adversarial networks; fair data generation



Citation: Rajabi, A.; Garibay, O.O. TabFairGAN: Fair Tabular Data Generation with Generative Adversarial Networks. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 488–501. <https://doi.org/10.3390/make4020022>

Academic Editor: Isaac Triguero

Received: 12 April 2022

Accepted: 13 May 2022

Published: 16 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial intelligence has gained paramount importance in the contemporary human life. With an ever-growing body of research and increasing processing capacity of computers, machine learning systems are being adopted by many firms and institutions for decision-making. Various industries such as insurance companies, financial institutions, and healthcare providers rely on automated decision making by machine learning models, making fairness-aware learning crucial since many of these automated decisions could have major impacts on the lives of individuals. There are numerous evidence suggesting that bias exists in AI systems. One well known example is Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), which is a decision making system deployed by the US criminal justice system to assess the likelihood of a criminal defendant's recidivism (re-offending). It is shown that COMPAS is biased against African American defendants [1]. Another example is a Google's targeted advertising that was found to have shown the high paid jobs significantly more to males than females [2].

The existence of such bias and unfair classifications in AI systems has led the research community to pay attention to the problem of bias in AI. There are different approaches to improve fairness existing in the AI fairness literature. Let $D = \{X, S, Y\}$ be a labelled

dataset, where $X \in \mathbb{R}^n$ are the unprotected attributes, S is the protected attribute, and Y is the decision. From a legal perspective, protected attribute is the attribute identified by law, based on which it is illegal to discriminate [3], e.g., gender or race. The proposed fairness enforcement methods in the literature could be categorised into three main classes of pre-process methods, in-process methods, and post-process methods. Pre-process methods include modifying the training data before feeding the data into machine learning algorithm. For instance, in one study [4], four methods are presented to remove bias including suppression which is to remove attributes highly correlated with protected attributes S , massaging the dataset which is to change labels (Y) of some objects in the dataset, and reweighing that involves assigning weights to different instances in the dataset. These are preliminary and simpler methods that result in more fair predictions, but entail a higher fairness-utility cost. In other words fairness is achieved at the expense of accuracy. Another preprocessing method proposed in the literature is the work of Feldman et al. [5] in which a repairment mechanism is proposed to modify the unprotected attributes (X) and achieve fairness with higher accuracy comparing to the aforementioned methods. This method will be discussed in more detail in Section 5.2 as the baseline method. In-process approaches involve modifying the learning algorithm to achieve fairness during training [3]. These methods mostly include modifying the objective functions or adding regularization terms to the cost function. For example, Kamishima et al. [6] proposes adding a regularization term to the objective function which penalize mutual information between the protected attributes and the classifier predictions. Finally, post-process mechanisms include modifying the final decisions of the classifiers. For instance, Hardt et al. [7] propose a method to modify the final classification scores in order to enhance equalized odds.

The emergence of unfairness in AI systems is mostly attributed to: (1) direct bias existing in the historical datasets being used to train the algorithms, (2) bias caused by missing data, (3) bias caused by proxy attributes, where bias against the minority population is present in non-protected attributes, and (4) bias resulting from algorithm objective functions, where the aggregate accuracy of the whole population is sought and therefore the algorithm might disregard the minority group for the sake of majority [3]. Since historical datasets are a major source of discrimination in AI, we focus on generating unbiased datasets to achieve fairness.

There is a rich and growing literature on generative models. The main idea behind a generative model is to capture the probabilistic distribution that could generate data similar to a reference dataset [8]. Broadly speaking, generative models could be divided into two main classes of models [8]: energy-based models such as Boltzmann Machines [9] and cost function-based models such as autoencoders and generative adversarial networks (GANs) [10]. GANs address some deficiencies in traditional generative models, and are shown to excel in various tasks comparing to other generative models such as in image generation [11] and video generation [12].

The original GAN consists of two networks, generator and discriminator [10]. The two networks play a minimax game. The generator takes a latent random variable Z as input and generates a sample $G(Z)$, that is similar to the real data. The discriminator, on the other hand, is fed with both real and generated samples, and its task is to correctly classify the input sample as real or generated. Over time if the networks have enough capacity, they are trained together and ideally optimized to reach an equilibrium state in which the generator produces data from the exact targeted distribution and the discriminator gives the real and generated samples an equal probability of 0.5. The work in [10] shows that training the discriminator to optimality is equal to minimizing Jensen–Shannon divergence [13]. The work of Arjovsky et al. develops Wasserstein GANs, where a critic replaces the discriminator, and minimizing Earth-mover’s distance is used instead of minimizing Jensen–Shannon divergence [14]. They show that WGAN could address some common training problems attributed to GANs, such as requirement to maintain a careful balance during training as well as mode dropping [15].

In recent studies adversarial training has been used to remove discrimination. One such study, for example, by formulating the model as a minimax problem, proposes an adversarial learning framework that could learn representations of data that are discrimination-free and do not contain explicit information about the protected attribute [16]. Other adversarial objectives are proposed by the works of [17,18] to achieve group fairness measures such as demographic parity and equality of odds. The application of generative adversarial networks for fairness in tabular datasets is not discussed enough in the literature, but has recently attracted the attention of the research community. For instance, the work of Sattigeri et al. [19] proposes an approach to generate image datasets such that demographic fairness in the generated dataset is imposed. In their work Xu et al. [20] design a GAN that produces discrimination free tabular datasets. Their network includes one generator and two discriminators. The generator is adopted from [21] and produces fake pairs of data (\hat{X}, \hat{Y}) following the conditional distribution $P_G(X, Y|S)$, where S is the protected attribute. One discriminator's task is to ensure the generator produces data with good accuracy, and the second discriminator ensures the generator produces fair data.

In this paper, we propose a Wasserstein GAN, TabFairGAN, that can produce high quality tabular data with the same joint distribution as the original tabular dataset. In Section 2, we discuss the fairness measure: demographic parity and discrimination score. In Section 3, we introduce the model architecture, data transformation, value functions, and the training process of the model. In Section 4, we compare the results of TabFairGAN with two other state-of-the-art GANs for tabular data generation, namely TGAN [22] and CTGAN [23]. In Section 5, we show how the model could be used for fair synthetic data generation and test the model on four real dataset. We compare the results of our model with the method developed by [5], which is another pre-process methods to enforce fairness. Finally in Section 5.4, we explore the fairness–accuracy trade-off. This work has two main contributions. We show that in the case of no constraints present (no fairness), the model is able to produce high quality synthetic data, competing with the state-of-the-art GANs designed for tabular data generation. This is achieved by quantile transformation of numerical attributes, enabling us to achieve high accuracy with a simple network architecture. Second contribution is producing high quality fair synthetic data, by adding a fairness constraint in the loss function of the generator. Comparing our model to previous application of GANs for fair tabular data generation, the model is more stable based on two merits: (1) the proposed model is a Wasserstein GAN which is shown to improve original GAN model in terms of some common GAN pitfalls, such as mode-dropping phenomena [15], and (2) the model only uses one critic instead of two [20] or three [24] discriminators.

2. Discrimination Score

Among the most frequently practiced fairness metrics specified in legal notions and the literature is *demographic parity* or statistical parity/fairness. The goal of demographic fairness is to ensure that the overall proportion of members with respect to the protected group receiving a positive decision is identical. Let $D = \{X, S, Y\}$ be a labelled dataset, where $X \in \mathbb{R}^n$ is the unprotected attributes, S is the protected attribute, and Y is the decision. In this paper, we consider the binary case, and for notational convenience we assume that the protected attribute S takes two values, where $S = 0$ represents the underprivileged minority class, and $S = 1$ represents the privileged majority class. For instance, in a binary racial discrimination study the value 0 will be assigned to “African-American”, whereas 1 is assigned to “White”. We also assign 1 to Y for a successful decision (for instance an admission to a higher education institution), and assign 0 to Y for an unsuccessful decision (rejection). Demographic fairness for the labeled dataset is defined as follows [7]:

$$P(y = 1|s = 1) = P(y = 1|s = 0) \quad (1)$$

In this context, demographic parity is defined by the difference between the conditional probability and its marginal. We define the discrimination with respect to the protected

attribute S by *discrimination score* (DS) and calculate it by: $DS = P(y = 1|s = 1) - P(y = 1|s = 0)$. A similar measure could be obtained for a labeled dataset D and a classifier $f : (X, S) \rightarrow Y$ where the discrimination score for the classifier f with respect to protected attribute S can be obtained by:

$$P(\hat{y} = 1|x, s = 1) - P(\hat{y} = 1|x, s = 0) \quad (2)$$

3. Model Description

3.1. Tabular Dataset Representation and Transformation

A tabular dataset contains N_C numerical columns $\{c_1, \dots, c_{N_C}\}$ and N_D categorical columns $\{d_1, \dots, d_{N_D}\}$. In this model, categorical columns are transformed and represented by one-hot vectors. Representing numerical columns on the other hand is non-trivial due to certain properties of numerical columns. One such property is that numerical columns are often sampled from multi-modal distributions. Some models such as [21] use min-max normalization to normalize and transform numerical columns. The work of Xu et al. [23] proposes a more complex process, namely a mode-specific normalization using variational Gaussian mixture models (VGM) to estimate the number of modes and fit a Gaussian mixture model to each numerical column. In our model, each numerical column is transformed using a quantile transformation [25]:

$$c'_i = \Phi^{-1}(F(c_i)) \quad (3)$$

where c_i is the i th numerical feature, F is the CDF (cumulative distribution function) of the feature c_i , and Φ is the CDF of a uniform distribution. After transforming numerical and discrete columns, the representation of each transformed row of the data is as follows:

$$\mathbf{r} = c'_1 \oplus \dots \oplus c'_{N_C} \oplus d'_1 \oplus \dots \oplus d'_{N_D} \quad (4)$$

$$l_i = \dim(d'_i) \quad (5)$$

$$l_w = \dim(\mathbf{r}) \quad (6)$$

where c'_i represents the i th numerical column, d'_i denotes the one-hot encoded vector of the i th categorical columns, and \oplus is the symbol denoting concatenation of vectors. Furthermore, l_i shows the dimension of the i th discrete column's one-hot encoding vector and l_w shows the dimension of \mathbf{r} .

3.2. Network Structure

While traditional GANs suffer from problems such as non-convergence and mode-collapse, the work of [15] developed Wasserstein GANs which improve training of GANs to some extent, and replace the discriminator with a critic. The network designed in this model is a WGAN with gradient penalty [26]. The WGAN value function using the Kantorovich-Rubinstein duality [27] is as follows [26]:

$$\min_G \max_{C \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim P_{data}(\mathbf{x})} [C(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim P_z(\mathbf{z})} [C(G(\mathbf{z}))] \quad (7)$$

where \mathcal{C} is the set of 1-Lipschitz functions. The generator receives a latent variable Z from a standard multivariate normal distribution and produces a sample data point which is then forwarded to the critic. Once the critic and the generator are trained together, the generator will produce data close to the real data.

The generator includes a fully-connected first layer with ReLU activation function. The second hidden layer of the generator network is then formed by concatenation of multiple vectors that could form data similar to transformed original data. For the numerical variables, a fully connected layer of $FC_{l_w \rightarrow N_C}$, with a ReLU activation is implemented. For nodes that are supposed to produce discrete columns, multiple fully connected layers of $FC_{l_w \rightarrow l_i}$, with Gumbel softmax [28] activation are used in order to produce the one-hot

vectors (d'_i). The resulting nodes are then concatenated to produce data similar to the transformed original data (with the same dimension of l_w), which is then fed to the critic network. The structure of the critic network is simple and includes 2 fully connected layers with Leaky ReLu activation functions.

The generator network's architecture is formally described as:

$$\begin{cases} h_0 = z \\ h_1 = \text{ReLu}(\text{FC}_{l_w \rightarrow l_w}(h_0)) \\ h_2 = \text{ReLu}(\text{FC}_{l_w \rightarrow N_C}(h_1)) \oplus \text{gumbel}_{0.2}(\text{FC}_{l_w \rightarrow l_1}(h_1)) \oplus \\ \text{gumbel}_{0.2}(\text{FC}_{l_w \rightarrow l_2}(h_1)) \oplus \dots \oplus \text{gumbel}_{0.2}(\text{FC}_{l_w \rightarrow l_{N_D}}(h_1)) \end{cases} \quad (8)$$

where z denotes latent vector, $\text{FC}_{a \rightarrow b}$ denotes a fully connected layer with input size a and output size b , $\text{ReLu}(x)$ shows applying a ReLu activation on x , and $\text{gumbel}_\tau(x)$ denotes applying Gumbel softmax with parameter τ on a vector x , and \oplus denotes concatenation of vectors.

The critic network's architecture is formally described as:

$$\begin{cases} h_0 = x \\ h_1 = \text{LeakyReLu}_{0.01}(\text{FC}_{l_w \rightarrow l_w}(h_0)) \\ h_2 = \text{LeakyReLu}_{0.01}(\text{FC}_{l_w \rightarrow l_w}(h_1)) \end{cases} \quad (9)$$

where x denotes input to the critic (output of the generator or transformed real data), $\text{LeakyReLu}_\tau(x)$ denotes applying Leaky ReLu activation function [29] with slope τ on x . Figure 1 shows the architecture of the model.

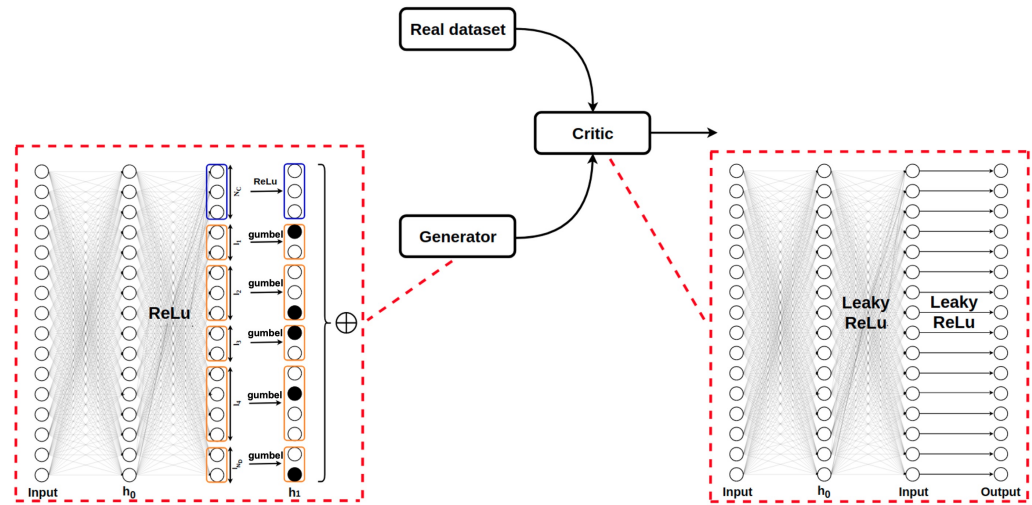


Figure 1. Model architecture. The generator consists of an initial fully connected layer with ReLu activation function, and a second layer which uses ReLu for numerical attributes generation and gumbel-softmax to form one-hot representations of categorical attributes. The final data are then produced by concatenating all attributes in the last layer of the generator. The critic consists of fully-connected layers with LeakyReLu activation function.

3.3. Training

In this section, we introduce the loss functions for the critic network and generator network of the developed WGAN. The overall process of training the model includes two phases. Phase I of training only focuses on training the model such that the generator could generate data with a joint probability distribution similar to that of the real data. Phase II of training further trains the generator to produce samples which have a joint probability

distribution similar to that of real data and is also fair, with respect to discrimination score (DS) defined in Section 2.

3.3.1. Phase I: Training for Accuracy

In the first phase, the generator and critic are trained with respect to their value functions. Critic's loss function with gradient penalty is [26]:

$$V_C = \mathbb{E}_{\hat{\mathbf{x}} \sim P_g} [C(\hat{\mathbf{x}})] - \mathbb{E}_{\mathbf{x} \sim P_r} [C(\mathbf{x})] + \lambda \mathbb{E}_{\bar{\mathbf{x}} \sim P_{\bar{\mathbf{x}}}} [(\|\nabla_{\bar{\mathbf{x}}} C(\bar{\mathbf{x}})\|_2 - 1)^2] \quad (10)$$

where P_r and P_g are real data distribution and generated data distribution, respectively. Note that the third term is the gradient penalty to enforce the Lipschitz constraint, and λ is the gradient penalty coefficient. $P_{\bar{\mathbf{x}}}$ is implicitly defined sampling uniformly along straight lines between pairs of points sampled from the data distribution P_r and the generator distribution P_g [26].

The loss function for the generator network in Phase I of training is also as follows:

$$V_G = - \mathbb{E}_{\hat{\mathbf{x}} \sim P_g} [C(\hat{\mathbf{x}})] \quad (11)$$

3.3.2. Phase II: Training for Fairness and Accuracy

In the second phase of training, the fairness constraint is enforced on the generator to produce fair data. Similar to the definitions in Section 2, let $\hat{D} = \{\hat{X}, \hat{Y}, \hat{S}\}$ be a batch of generated data, i.e., \hat{X} are the generated unprotected attributes of the data, \hat{Y} is the generated decision with $\hat{Y} = 1$ being the successful and favorable value for the decision (e.g., having an income of >50 K for an adult in the adult income dataset), and \hat{S} being the generated protected attribute with $\hat{S} = 0$ representing the unprivileged minority group. The new loss function for the generator in Phase II of training is as follows:

$$V_G = - \mathbb{E}_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{s}}) \sim P_g} [C(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{s}})] - \lambda_f \left(\mathbb{E}_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{s}}) \sim P_g} [\hat{\mathbf{y}} | \hat{\mathbf{s}} = 0] - \mathbb{E}_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{s}}) \sim P_g} [\hat{\mathbf{y}} | \hat{\mathbf{s}} = 1] \right) \quad (12)$$

With the above loss function for the generator, the model aims to generate a fair dataset $\{\hat{X}, \hat{Y}, \hat{S}\} \sim P_g$ which achieves the demographic fairness with respect to the protected attribute \hat{S} in the generated samples, by minimizing discrimination score in the generated data $P(\hat{Y} | \hat{S} = 1) - P(\hat{Y} | \hat{S} = 0)$. λ_f is the discrimination penalty coefficient. The goal in this phase of training is to train the generator to generate synthetic data which is both similar to the real data $\hat{D} \sim D$, and the generated data are fair based on demographic fairness measure. In the ideal case, the generator would produce synthetic data such that $\hat{Y} \perp \hat{S}$. After training is done, the samples are generated and inverse transformed to the original data format. The formal procedure of training the model is shown in Algorithm 1.

Algorithm 1 training algorithm for the proposed WGAN. We use $n_{crit} = 4$, batch size of 256, $\lambda_p = 10$, Adam optimizer with $\alpha = 0.0002$, $\beta_1 = 0.5$, and $\beta_2 = 0.999$.

```

1: for  $T_1$  do
2:   for  $t = 1, \dots, n_{crit}$  do
3:     Sample batch  $m$   $D(x, y, s) \sim P_r$  and  $z \sim P(z)$  and  $\epsilon \sim U[0, 1]$ 
4:      $\hat{D} = (\hat{x}, \hat{s}, \hat{y}) \leftarrow G_\theta(z)$ 
5:      $\bar{D} \leftarrow \epsilon(D) + (1 - \epsilon)(\hat{D})$ 
6:     Update the critic by descending the gradient:
7:      $\nabla_w \frac{1}{m} \sum_{i=1}^m C_w(\bar{D}) - C_w(D) + \lambda_p (\|\nabla_{\bar{D}} C_w(\bar{D})\|_2 - 1)^2$ 
8:   end for
9:   Sample a batch  $m$   $z \sim P(z)$ 
10:  Update the generator by descending the gradient:
11:   $\nabla_\theta \frac{1}{m} \sum_{i=1}^m -(C_w(G_\theta(z)))$ 
12: end for
13: for  $T_2$  do
14:   for  $t = 1, \dots, n_{crit}$  do
15:     Sample batch  $m$   $D(x, y, s) \sim P_r$  and  $z \sim P(z)$  and  $\epsilon \sim U[0, 1]$ 
16:      $\hat{D} = (\hat{x}, \hat{s}, \hat{y}) \leftarrow G_\theta(z)$ 
17:      $\bar{D} \leftarrow \epsilon(D) + (1 - \epsilon)(\hat{D})$ 
18:     Update the critic by descending the gradient:
19:      $\nabla_w \frac{1}{m} \sum_{i=1}^m C_w(\bar{D}) - C_w(D) + \lambda_p (\|\nabla_{\bar{D}} C_w(\bar{D})\|_2 - 1)^2$ 
20:   end for
21:   sample a batch  $m$   $\hat{D} = \hat{x}, \hat{s}, \hat{y} \sim P(G_\theta(z))$ 
22:   Update the generator by descending the gradient:
23:    $\nabla_\theta \frac{1}{m} \sum_{i=1}^m -C_w(\hat{D}) - \lambda_f \left( \frac{|D_{s=0, y=1}|}{|D_{s=0}|} - \frac{|D_{s=1, y=1}|}{|D_{s=1}|} \right)$ 
24: end for

```

4. Experiment: Only Phase I (No Fairness)

In this section, we evaluate the effectiveness of the model in producing synthetic data similar to data coming from a known probability distribution. We show that the model is able to generate synthetic data similar to the reference dataset and compare our results with two state-of-the-art GAN models for tabular datasets, namely TGAN [22] and CTGAN [23]. TGAN generates relational tables by clustering numerical variables to deal with multi-modal distributions and adding noise and KL divergence into the loss function to generate discrete features. In CTGAN, mode-specific normalization is applied to numerical values and the generator works conditionally in order to overcome the imbalance in training data. We evaluate the models on the UCI Adult Income Dataset <http://archive.ics.uci.edu/ml/datasets/adult> accessed on 10 January 2022. The task we are trying to achieve is as follows: given a dataset $D = \{X, S, Y\} \sim P_{data}$, generate a dataset $\hat{D}_{syn} = \{\hat{X}, \hat{S}, \hat{Y}\} \sim P_{syn}$ s.t. $P_{syn} \sim P_{data}$. We are not seeking to achieve fairness in this section, and we solely seek to generate data following the same distribution as real data to achieve data utility (accuracy).

To compare data utility among generated datasets among different models, we evaluate the performance of using synthetic data as training data for machine learning. At first, the real dataset is divided into two parts: D_{train} and D_{test} . The Adult dataset contains a total of 48,842 rows. 90% of the data were assigned to D_{train} and the rest 10% were assigned to D_{test} . Next, each classifier is trained on the training set D_{train} for 300 epochs three times. With each training, the trained models are used to generate their corresponding synthetic data D_{syn} . Three machine learning classifiers are then chosen and trained on each generated D_{syn} , tested on D_{test} , and the accuracy and F1 score of classification are recorded. The classifiers used are a Decision Tree Classifier (DTC), Logistic Regression (LR), and a Multi Layer Perceptron (MLP). Table 1 reports the results of classification, and compares the results with the case that a classifier is trained on the original D_{train} , and tested on D_{test} (reporting the means and standard deviations of evaluation metrics). The results show

that TabFairGAN and CTGAN outperform TGAN in all cases. TabFairGAN outperforms CTGAN with a DT Classifier. With a LR classifier, the performance of TabFairGAN and CTGAN is identical with respect to accuracy, and TabFairGAN performs slightly better than CTGAN with respect to F1 score. With an MLP classifier, CTGAN performs slightly better than TabFairGAN with respect to accuracy, while TabFairGAN outperforms CTGAN with respect to F1 score. These results display the effectiveness of TabFairGAN with respect to generating data identical to real tabular data.

Table 1. Comparing the results TabFairGAN for accurate data generation with TGAN and CTGAN models.

Classifier	DTC		LR		MLP	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
Original Data	0.811 ± 0.001	0.606 ± 0.002	0.798 ± 0.000	0.378 ± 0.000	0.780 ± 0.051	0.488 ± 0.075
TabFairGan	0.783 ± 0.001	0.544 ± 0.002	0.794 ± 0.020	0.239 ± 0.012	0.778 ± 0.045	0.405 ± 0.174
TGAN	0.661 ± 0.013	0.503 ± 0.012	0.765 ± 0.010	0.170 ± 0.008	0.623 ± 0.197	0.376 ± 0.159
CTGAN	0.777 ± 0.003	0.482 ± 0.004	0.794 ± 0.023	0.232 ± 0.012	0.784 ± 0.007	0.305 ± 0.104

5. Experiments: Fair Data Generation and Data Utility (Training with both Phase I and Phase II)

In the second set of experiments, the effectiveness of the model in generating data which is both similar to the reference dataset and also fair is evaluated, and the tradeoff between machine learning efficacy and fairness is investigated. We will experiment with four datasets to test the fairness/utility tradeoff of the model. The four datasets and their attributes are first introduced. All four datasets used in experiments are studied in the literature of algorithmic fairness [3]. Next, we introduce the baseline method with which the results of TabFairGAN are compared. The results are presented and compared in Table 2.

5.1. Datasets

The first dataset is UCI Adult Dataset. This dataset is based on 1994 US census data and contains 48,842 rows with attributes such as age, sex, occupation, and education level. for each person, and the target variable indicates whether that individual has an income that exceeds \$50K per year. In our experiments, we consider the protected attribute to be sex ($S = \text{"Sex"}$, $Y = \text{"Income"}$).

The second dataset used in the experiments is the Bank Marketing Data Set [30]. This dataset contains information about a direct marketing campaign of a Portuguese banking institution. Each row of the dataset contains attributes about an individual such as age, job, marital status, housing, duration of that call, and the target variable determines whether that individual subscribed a term deposit or not. The dataset contains 45,211 records. Similar to [31], we have considered age to be the protected attribute (a young individual has a higher chance of being labeled as “yes” to subscribe a term deposit). In order to have a binary protected attribute, we set a cut-off value of 25 and an age of more than 25 is considered “older”, while an age of less than or equal to 25 is considered “younger” ($S = \text{"Age"}$, $Y = \text{"Subscribed"}$).

Table 2. Comparing the results of TabFairGAN for fair data generation with CRDI. Each number in the table reports the average and standard deviation over 5 trainings.

Original Data				TabFairGAN				CRDI			
Dataset	Orig. Acc.	F1 Orig.	DS Data	DS Gen.	Acc. Gen.	F1 Gen.	DS Classifier	DS Rep.	Acc. Rep.	F1 Rep.	DS Classifier
Adult	0.816 ± 0.005	0.619 ± 0.013	0.195	0.009 ± 0.027	0.773 ± 0.013	0.536 ± 0.022	0.082 ± 0.038	0.165 ± 0.048	0.793 ± 0.011	0.558 ± 0.029	0.121 ± 0.024
Bank	0.879 ± 0.004	0.491 ± 0.020	0.126	0.001 ± 0.011	0.854 ± 0.004	0.373 ± 0.024	0.060 ± 0.056	0.122 ± 0.004	0.776 ± 0.004	0.384 ± 0.011	0.050 ± 0.017
COMPAS	0.903 ± 0.007	0.914 ± 0.007	0.258	0.009 ± 0.102	0.860 ± 0.040	0.876 ± 0.033	0.208 ± 0.072	0.119 ± 0.128	0.893 ± 0.021	0.906 ± 0.020	0.205 ± 0.055
Law School	0.854 ± 0.008	0.918 ± 0.005	0.302	0.024 ± 0.036	0.847 ± 0.020	0.916 ± 0.012	0.153 ± 0.072	0.233 ± 0.103	0.892 ± 0.004	0.941 ± 0.002	0.289 ± 0.057

The third dataset used in this section is the ProPublica dataset from the COMPAS risk assessment system [32]. This dataset contains information about defendants from Broward County, and contains attributes about defendants such as their ethnicity, language, marital status, and sex, and for each individual a score showing the likelihood of recidivism (re-offending). In this experiments we used a modified version of the dataset. First, attributes such as FirstName, LastName, MiddleName, CASE_ID, and DateOfBirth are removed. Studies have shown that this dataset is biased against African Americans [1]. Therefore, ethnicity is chosen to be the protected attribute for this study. Only African American and Caucasian individuals are kept and the rest are dropped. The target variable in this dataset is a risk decile score provided by the COMPAS system, showing the likelihood of that individual to re-offend, which ranges from 1 to 10. The final modified dataset contains 16,267 records with 16 features. To make the target variable binary, a cut-off value of 5 is considered and individuals with a decile score of less than 5 are considered “Low_Chance”, while the rest are considered “High_Chance”. ($S = \text{“Ethnicity”}$, $Y = \text{“Recidivism_Chance”}$).

The last dataset used in experiments is the Law School Admission Council which was made by conducting a survey across 162 law schools in the United States [33]. This dataset contains information on 21,790 law students such as their GPA (grade-point average), LSAT score, race, and the target variable is whether the student had a high FYA (first year average grade). Similar to other studies (such as [34]), we have considered race to be the protected attribute. We only considered individuals with “Black” or “White” race. The modified data contain 19,567 records. ($S = \text{“Race”}$, $Y = \text{“FYA”}$). The discrimination score (DS) of all datasets are reported in Table 2.

5.2. Baseline Model: Certifying and Removing Disparate Impact

In their work Feldman et al. [5] proposed a method to modify a dataset to remove bias and preserve relevant information in the data. In dataset $D = \{X, S, Y\}$, given the protected attribute S and a *single* numerical attribute X , let $X_s = Pr(X|S = s)$ denote the marginal distribution on X conditioned on $S = s$. Considering $F_s : X_s \rightarrow [0, 1]$ the cumulative distribution function for values $x \in X_s$, they define a “median” distribution A in terms of its quantile function $F_A^{-1} : F_A^{-1}(u) = \text{median}_{s \in S} F_s^{-1}(u)$. They then propose a repair algorithm which creates \bar{X} , such that for all $x \in X_s$ the corresponding $\bar{x} = F_A^{-1}(F_s(x))$. To control the trade-off between fairness and accuracy, they define and calculate λ – *partial repair* by:

$$\bar{F}_s^{-1} = (1 - \lambda)F_s^{-1} + \lambda(F_A)^{-1} \quad (13)$$

The result of such partial repair procedure is a dataset $\bar{D} = \{\bar{X}, S, Y\}$ which is more fair and preserves relevant information for classification task. We call this method CRDI henceforth.

5.3. Results

The goal in this section is to train the proposed network on datasets and produce similar data that is also fair with respect to protected attributes defined for each dataset. The process is as follows: The models are first trained on each dataset. As mentioned in Section 3.3, training of the network includes two phases: in the first phase, the network is only trained for accuracy for a certain number of epochs, and then in the second phase, the loss function of generator is modified and the network gets trained for accuracy and fairness. Once the training is finished, the generator of the network is used to produce synthetic data D_{syn} . We also generated repaired datasets using CRDI method described in Section 5.2 to compare our results with. For each model, we train five times and report the means and standard deviations of evaluation results in Table 2.

The generated data D_{syn} is then evaluated from two perspective: fairness and utility. To evaluate the fairness of D_{syn} , we adopt discrimination score (DS): $DS = P(y = 1|s = 1) - P(y = 1|s = 0)$. Looking into Table 2, the results show that comparing with CRDI, TabFairGAN could more effectively produce datasets s.t. demographic parity in the gen-

erated data are almost removed. The demographic parity of the produced datasets by TabFairGAN, beat the repaired datasets produced by CRDI.

To evaluate data utility, we adopt a decision tree classifier with the default parameter setting [35]. For TabFairGAN data, we train the decision tree classifier on D_{syn} , test it on D_{test} , and report the accuracy and F1-score of the classifier. We also train decision tree classifiers on repaired data \bar{D} produced by CRDI, and test on D_{test} and report accuracy and f1-score. Table 2 shows that repaired data \bar{D} produced by CRDI has better data utility for adult dataset, COMPAS dataset, and Law School dataset by less than 5% in all cases, while the accuracy of D_{syn} produced by TabFairGAN is almost 8% higher than that of \bar{D} produced by CRDI.

The last evaluation we perform on the produced datasets is to examine discrimination score (DS) of the classifier: $DS = P(\hat{y} = 1 | s = 1) - P(\hat{y} = 1 | s = 0)$. The results in Table 2 show that discrimination score of the decision tree classifier trained on D_{syn} for Adult dataset and Law School is lower by almost 4% and 13%, respectively, while the discrimination score of the decision tree classifier trained on \bar{D} for Bank dataset and COMPAS dataset is only slightly higher than TabFairGAN by 1% and 0.003%, respectively.

It should be noted that the λ parameter was chosen for CRDI such that the repaired dataset could achieve the best possible fairness metrics. The parameter settings of the models on each dataset is reported in the Appendix A. The results show, while CRDI narrowly beats TabFairGAN in terms of data utility, TabFairGAN beats CRDI in terms of discrimination score in all cases for generated data and in two out of four cases in the generated classifiers. This is attributed to fairness utility trade-off of TabFairGAN governed by λ_f . The case of COMPAS dataset is interesting since none of the models could decrease discrimination score in the classifier much, comparing to the discrimination score in the original dataset. Looking into the data and performing a correlation analysis, risk decile score (target variable) has a high Pearson correlation of 0.757 with one of columns names RecSupervisionLevel which denotes the supervisory status of each individual. This reveals that although the generated dataset D_{syn} has a lower discrimination score of 0.009, disparate impact exists in the dataset, indicating that the discriminatory outcomes are not explicitly caused by the protected attribute, but are also from the proxy unprotected attributes [20].

5.4. Utility and Fairness Trade-Off

To explore the trade-off between utility and fairness of the generated data, we perform the following experiment: λ_f was increased between [0.05, 0.7] in steps of 0.05, and for each value of λ_f the model was trained 170 epochs in phase I and 30 times in the phase II. For each λ_f value, we trained five models and the average of Discrimination Score was recorded for each λ_f . Figure 2 shows the results, plotted along with standard deviation as confidence intervals. We can observe that discrimination score of the generated synthetic datasets (D_{syn}) is decreasing significantly as λ_f decreases. Meanwhile, classifier accuracy layoff, i.e., the reduction in decision tree classifier's accuracy comparing to the case in which the classifier is trained on the real original training dataset (D_{train}), is increasing slightly as λ_f increases.

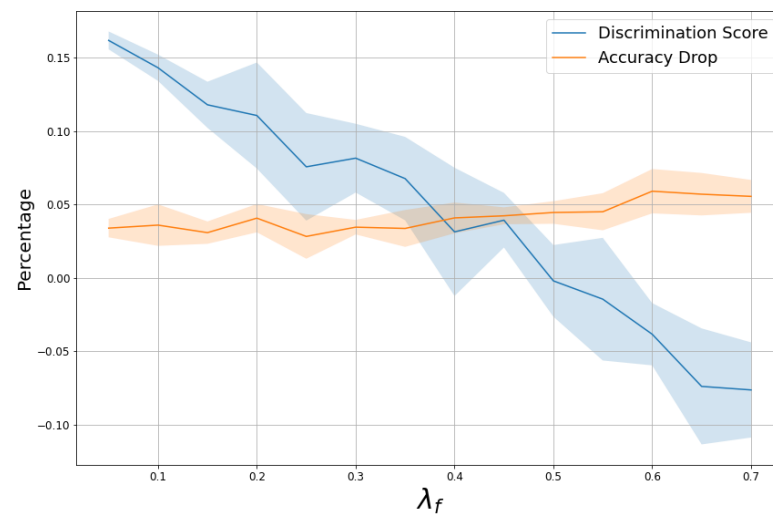


Figure 2. Exploring the trade-off between accuracy and fairness by incremental increasing of parameter λ_f . Each data point is the average over five trainings, with standard deviation of the five training shown as confidence intervals.

6. Conclusions

In this paper, we proposed a Generative Adversarial Network that could generate synthetic data similar to a reference data. We showed that in the case of unconditional tabular data generation, i.e., with no fairness constraints, the model is able to produce data with high quality comparing to other GANs developed for the same purpose. We also showed that by adding a fairness constraint to the generator, the model is able to achieve data generation which improves the demographic parity of the generated data. We tested the model on four datasets studied in the fairness literature and compared our results with the method explained in [5]. As a generative model, GANs have a great potential to be utilized for fair data generation, specially in the case that the real dataset is limited. Our proposed model is able to produce synthetic fair tabular data, addressing both fairness and privacy preservation issues. In future work, we will explore more sophisticated data generation constraints, e.g., considering enforcing other fairness metrics such as equality of odds and equality of opportunity. We also consider exploring and utilizing GANs for fairness in other data types, such as text and image data.

Author Contributions: Conceptualization, A.R. and O.O.G.; methodology, A.R. and O.O.G.; software, A.R.; validation, A.R. and O.O.G.; formal analysis, A.R. and O.O.G.; investigation, A.R. and O.O.G.; resources, O.O.G.; writing—original draft preparation, A.R., O.O.G.; writing—review and editing, A.R. and O.O.G.; visualization, A.R.; supervision, O.O.G.; project administration, O.O.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data and model code for this study are openly available at <https://github.com/amirarsalan90/TabFairGAN> (accessed on 10 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1 reports the models' hyperparameters used in Section 5 experiments.

Table A1. Paramter Configuration for TabFairGAN and CRDI.

	TabFairGAN			CRDI
	T_1	T_2	λ_f	λ
Adult	170	30	0.5	0.999
Bank	195	5	0.75	0.9
COMPAS	c40	30	2.2	0.999
Law School	180	20	2.5	0.999

References

1. Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data* **2017**, *5*, 153–163. [[CrossRef](#)] [[PubMed](#)]
2. Lambrecht, A.; Tucker, C. Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of stem career ads. *Manag. Sci.* **2019**, *65*, 2966–2981. [[CrossRef](#)]
3. Pessach, D.; Shmueli, E. Algorithmic fairness. *arXiv* **2020**, arXiv:2001.09784.
4. Kamiran, F.; Calders, T. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* **2012**, *33*, 1–33. [[CrossRef](#)]
5. Feldman, M.; Friedler, S.A.; Moeller, J.; Scheidegger, C.; Venkatasubramanian, S. Certifying and removing disparate impact. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 259–268.
6. Kamishima, T.; Akaho, S.; Asoh, H.; Sakuma, J. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 35–50.
7. Hardt, M.; Price, E.; Srebro, N. Equality of opportunity in supervised learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3315–3323.
8. Oussidi, A.; Elhassouny, A. Deep generative models: Survey. In Proceedings of the 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 2–4 April 2018; pp. 1–8.
9. Fahlman, S.E.; Hinton, G.E.; Sejnowski, T.J. Massively parallel architectures for AI: NETL, Thistle, and Boltzmann machines. In Proceedings of the National Conference on Artificial Intelligence, AAAI, Washington, DC, USA, 22–26 August 1983.
10. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
11. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096.
12. Vondrick, C.; Pirsiavash, H.; Torralba, A. Generating videos with scene dynamics. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 613–621.
13. Menéndez, M.; Pardo, J.; Pardo, L.; Pardo, M. The jensen-shannon divergence. *J. Frankl. Inst.* **1997**, *334*, 307–318. [[CrossRef](#)]
14. Rubner, Y.; Tomasi, C.; Guibas, L.J. The earth mover’s distance as a metric for image retrieval. *Int. J. Comput. Vis.* **2000**, *40*, 99–121. [[CrossRef](#)]
15. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.
16. Edwards, H.; Storkey, A. Censoring representations with an adversary. *arXiv* **2015**, arXiv:1511.05897.
17. Madras, D.; Creager, E.; Pitassi, T.; Zemel, R. Learning adversarially fair and transferable representations. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 3384–3393.
18. Zhang, B.H.; Lemoine, B.; Mitchell, M. Mitigating unwanted biases with adversarial learning. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, New Orleans, LA, USA, 2–3 February 2018; pp. 335–340.
19. Sattigeri, P.; Hoffman, S.C.; Chenthamarakshan, V.; Varshney, K.R. Fairness GAN: Generating datasets with fairness properties using a generative adversarial network. *IBM J. Res. Dev.* **2019**, *63*, 3:1–3:9. [[CrossRef](#)]
20. Xu, D.; Yuan, S.; Zhang, L.; Wu, X. Fairgan: Fairness-aware generative adversarial networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 570–575.
21. Choi, E.; Biswal, S.; Malin, B.; Duke, J.; Stewart, W.F.; Sun, J. Generating multi-label discrete patient records using generative adversarial networks. In Proceedings of the Machine Learning for Healthcare Conference, Boston, MA, USA, 18–19 August 2017; pp. 286–305.
22. Xu, L.; Veeramachaneni, K. Synthesizing Tabular Data using Generative Adversarial Networks. *arXiv* **2018**, arXiv:1811.11264.
23. Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; Veeramachaneni, K. Modeling Tabular data using Conditional GAN. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 7333–7343.
24. Xu, D.; Yuan, S.; Zhang, L.; Wu, X. Fairgan+: Achieving fair data generation and classification through generative adversarial nets. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 1401–1406.
25. Beasley, T.M.; Erickson, S.; Allison, D.B. Rank-based inverse normal transformations are increasingly used, but are they merited? *Behav. Genet.* **2009**, *39*, 580–595. [[CrossRef](#)] [[PubMed](#)]

26. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved Training of Wasserstein GANs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5769–5779.
27. Villani, C. *Optimal Transport: Old and New*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 338.
28. Jang, E.; Gu, S.; Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv* **2016**, arXiv:1611.01144.
29. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv* **2015**, arXiv:1505.00853.
30. Moro, S.; Cortez, P.; Rita, P. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.* **2014**, *62*, 22–31. [[CrossRef](#)]
31. Zafar, M.B.; Valera, I.; Róriguez, M.G.; Gummadi, K.P. Fairness constraints: Mechanisms for fair classification. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 9–11 May 2017; pp. 962–970.
32. Angwin, J.; Larson, J.; Mattu, S.; Kirchner, L. Machine Bias ProPublica. 2016. Available online: <https://github.com/propublica/compas-analysis> (accessed on 21 July 2021).
33. Wightman, L.F. LSAC National Longitudinal Bar Passage Study. LSAC Research Report Series. Available online: <https://eric.ed.gov/?id=ED469370> (accessed on 20 July 2021).
34. Bechavod, Y.; Ligett, K. Penalizing unfairness in binary classification. *arXiv* **2017**, arXiv:1707.00044.
35. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.