# FairGAN: Fairness-aware Generative Adversarial Networks

Depeng Xu, Shuhan Yuan, Lu Zhang, Xintao Wu
University of Arkansas, Fayetteville, AR, USA
Email: {depengxu, sy005, lz006, xintaowu}@uark.edu

***Abstract*—Fairness-aware learning is increasingly important in data mining. Discrimination prevention aims to prevent discrimination in the training data before it is used to conduct predictive analysis. In this paper, we focus on fair data generation that ensures the generated data is discrimination free. Inspired by generative adversarial networks (GAN), we present fairness-aware generative adversarial networks, called FairGAN, which are able to learn a generator producing fair data and also preserving good data utility. Compared with the naive fair data generation models, FairGAN further ensures the classifiers which are trained on generated data can achieve fair classification on real data. Experiments on a real dataset show the effectiveness of FairGAN.**

## I. Introduction

Discrimination refers to unjustified distinctions in decisions against individuals based on their membership in a certain group. Currently, many organizations or institutes adopt machine learning models trained on historical data to automatically make decisions, including hiring, lending and policing [1]. However, many studies have shown that machine learning models have biased performance against the *protected group* [2], [3]. In principle, if a dataset has discrimination against the protected group, the predictive model simply trained on the dataset will incur discrimination.

Many approaches aim to mitigate discrimination from historical datasets. A general requirement of modifying datasets is to preserve the data utility while removing the discrimination. Some methods mainly modify the labels of the dataset [4], [5]. Some methods also revise the attributes of data other than the label, such as the Preferential Sampling [6] and the Disparate Impact Removal [7].

In this work, instead of removing the discrimination from the existing dataset, we focus on generating fair data. Generative adversarial networks (GAN) have demonstrated impressive performance on modeling the real data distribution and generating high quality synthetic data that are similar to real data [8], [9]. After generating high quality synthetic data, many approaches can adopt the synthetic dataset to conduct predictive analysis instead of using the real data, especially when the real data is very limited [10]. However, due to high similarity between the real data and synthetic data, if the real data incur discrimination, the synthetic data can also incur discrimination. The following predictive analysis which is based on the synthetic data can be subject to discrimination. Throughout the paper, for ease of representation, we assume that there is only one protected attribute, which is a binary attribute associated with the domain values of the unprotected group and the protected group. We also assume there is one binary decision attribute associated with the domain values of the positive decision and the negative decision. Formally, let $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}, \mathcal{S}\}$ be a historical dataset where $\mathcal{X} \in \mathbb{R}^n$ is the unprotected attributes, $\mathcal{Y} \in \{0,1\}$ is the decision, and $\mathcal{S} \in \{0,1\}$ is the protected attribute. We aim to generate a fair dataset $\hat{\mathcal{D}} = \{\hat{\mathcal{X}}, \hat{\mathcal{Y}}, \hat{\mathcal{S}}\}$. In principle, the generated fair data $\hat{\mathcal{D}}$ should meet following requirements: 1) **data utility** which indicates the generated data should preserve the general relationship between attributes and decision in the real data; 2) **data fairness** which indicates there is no discrimination in the generated data; 3) **classification utility** which indicates classifiers trained on the generated data should achieve high accuracy when deployed for decision prediction of future real data; 4) **classification fairness** which indicates classifiers trained on the generated data should not incur discrimination when predicting on real data.

We develop fairness-aware generative adversarial networks (FairGAN) for fair data generation. Besides generating synthetic samples that match the distribution of real data, we also aim to prevent discrimination in the generated dataset. In pariticular, FairGAN consists of one generator and two discriminators. The generator generates fake samples $\{\hat{\mathcal{X}}, \hat{\mathcal{Y}}\}$ conditioned on the protected attribute $\mathcal{S}$. One discriminator aims to ensure the generated data $\{\hat{\mathcal{X}}, \hat{\mathcal{Y}}, \hat{\mathcal{S}}\}$ close to the real data $\{\mathcal{X}, \mathcal{Y}, \mathcal{S}\}$ while the other discriminator aims to ensure there are no correlation between $\hat{\mathcal{X}}$ and $\hat{\mathcal{S}}$ and no correlation between $\hat{\mathcal{Y}}$ and $\hat{\mathcal{S}}$. Note that $\hat{\mathcal{S}} = \mathcal{S}$ since the generator is conditioned on $\mathcal{S}$. FairGAN generates revised unprotected attributes $\hat{\mathcal{X}}$ and decision $\hat{\mathcal{Y}}$ given the protected attribute $\hat{\mathcal{S}}$ ($\mathcal{S}$) and achieves $\hat{\mathcal{X}} \perp\!\!\!\perp \mathcal{S}$ and $\hat{\mathcal{Y}} \perp\!\!\!\perp \mathcal{S}$. Therefore, the generated data can meet requirements of data fairness and classification fairness. The experimental results show that FairGAN can achieve fair data generation with good data utility and free from discrimination and the classifiers trained on the synthetic datasets can achieve fair classification on the real data with high accuracy.

## II. Related Work

In fairness-aware learning, discrimination prevention aims to remove discrimination by modifying the biased data and/or the predictive algorithms built on the data. Many approaches have been proposed for constructing discrimination-free classifiers, which can be broadly classified into three categories:

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

the pre-process approaches that modify the training data to remove discriminatory effect before conducting predictive analytics [4], [7], [11]–[13], the in-process approaches that enforce fairness to classifiers by introducing constraints or regularization terms to the objective functions [14], [15], and the post-process approaches that directly change the predicted labels [16], [17].

The pre-process approaches that modify the training data are widely studied. The fundamental assumption of the pre-process methods is that, once a classifier is trained on a discrimination-free dataset, the prediction made by the classifier will also be discrimination free [5]. Research in [18] proposed a causal graph based approach that removes discrimination based on the block set and ensures that there is no discrimination in any meaningful partition. For the in-process approaches, some tweak or regularizers are applied to the classifier to penalize discriminatory prediction during the training process. In principle, preventing discrimination when training a classifier consists of balancing two contrasting objectives: maximizing the accuracy of the extracted predictive model and minimizing the number of predictions that are discriminatory. Research in [12] proposed a predictive model for maximizing utility subject to the fair constraint that achieves both statistical parity and individual fairness, i.e., similar individuals should be treated similarly.

Recently, several studies have been proposed to remove discrimination through adversarial training. Research in [19] incorporated an adversarial model to learn a discrimination free representation. Based on that, research in [2] studies how the choice of data for the adversarial training affects the fairness. Studies in [20], [21] further proposed various adversarial objectives to achieve different levels of group fairness including demographic parity, equalized odds and equal opportunity. This paper studies how to generate a discrimination free dataset while still preserving the data generation utility. Fair data generation is in line with the pre-process approaches. The classical pre-process methods like Massaging [5] cannot remove disparate treatment and disparate impact, and the certifying framework [7], which can remove the disparate impact, can only apply on numerical attributes. On the contrary, FairGAN can remove the disparate treatment and disparate impact from both numerical and categorical data. Meanwhile, compared with the pre-process methods, FairGAN can generate more data for training predictive models, especially when the original training data is very limited.

## III. Preliminary

### A. Fairness and Discrimination

**Definition 1** (Statistical Parity/Fairness in a Labeled Dataset)**.** Given a labeled dataset $\mathcal{D}$, the property of statistical parity or fairness in the labeled dataset is defined as:

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

The discrimination in a labeled dataset w.r.t the protected attribute $\mathcal{S}$ is evaluated by the risk difference: $disc(\mathcal{D}) = P(y=1|s=1) - P(y=1|s=0)$.

**Definition 2** (Statistical Parity/Fairness in a Classifier)**.** Given a labeled dataset $\mathcal{D}$ and a classifier $\eta : \mathcal{X} \rightarrow \mathcal{Y}$, the property of statistical parity or fairness in a classifier is defined as:

$$P(\eta(\mathbf{x})=1|s=1)=P(\eta(\mathbf{x})=1|s=0)$$

We can then derive the *discrimination in a classifier* in terms of risk difference as $disc(\eta) = P(\eta(\mathbf{x}) = 1|s = 1) - P(\eta(\mathbf{x}) = 1|s = 0)$.

The classification fairness on a dataset is achieved if both the *disparate treatment* and *disparate impact* are removed from the data. To remove the disparate treatment, the classifier cannot use the protected attribute to make decisions. As for the disparate impact, research in [7] proposed the concept of $\epsilon$-fairness to examine the potential disparate impact.

**Definition 3** ($\epsilon$-fairness [7])**.** A dataset $\mathcal{D} = (\mathcal{X}, \mathcal{Y}, \mathcal{S})$ is said to be $\epsilon$-fair if for any classification algorithm $f : \mathcal{X} \rightarrow \mathcal{S}$

$$BER(f(\mathcal{X}), \mathcal{S}) > \epsilon$$

with empirical probabilities estimated from $\mathcal{D}$, where $BER$ (balanced error rate) is defined as

$$BER(f(\mathcal{X}), \mathcal{S}) = \frac{P[f(\mathcal{X})=0|\mathcal{S}=1] + P[f(\mathcal{X})=1|\mathcal{S}=0]}{2}.$$

$BER$ indicates the average class-conditioned error of $f$ on distribution $\mathcal{D}$ over the pair $(\mathcal{X}, \mathcal{S})$.

The $\epsilon$-fairness quantifies the fairness of data through the error rate of predicting the protected attribute $\mathcal{S}$ given the unprotected attributes $\mathcal{X}$. If the error rate is low, it means $\mathcal{S}$ is predictable by $\mathcal{X}$. In the fair data generation scenario, for a classifier trained on the synthetic dataset and tested on the real dataset, the classification fairness is achieved if disparate impact in terms of the real protected attribute is removed from the synthetic dataset, i.e. $\hat{\mathcal{X}} \perp\!\!\!\perp \mathcal{S}$.

### B. Generative Adversarial Network

Generative adversarial nets (GAN) are generative models that consist of two components: a generator $G$ and a discriminator $D$. Typically, both $G$ and $D$ are multilayer neural networks. $G(\mathbf{z})$ generates fake samples from a prior distribution $P_{\mathbf{z}}$ on a noise variable $\mathbf{z}$ and learns a generative distribution $P_G$ to match the real data distribution $P_{\text{data}}$. The discriminative component $D$ is a binary classifier that predicts whether an input is real data $\mathbf{x}$ or fake data generated from $G(\mathbf{z})$. The objective function of $D$ is defined as:

$$\max_{D} \quad \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))], \quad (1)$$

where $D(\cdot)$ outputs the probability that $\cdot$ is from the real data rather than the generated fake data. In order to make the generative distribution $P_G$ close to the real data distribution $P_{\text{data}}$, $G$ is trained by fooling the discriminator unable to distinguish the generated data from the real data. Thus, the objective function of $G$ is defined as:

$$\min_{G} \quad \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))]. \quad (2)$$

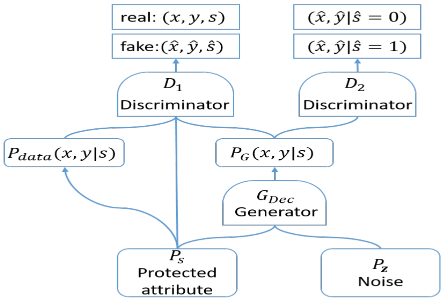

Figure 1: The Structure of FairGAN

Minimization of Equation 2 ensures that the discriminator is fooled by $G(\mathbf{z})$ and $D$ predicts high probability that $G(\mathbf{z})$ is real data.

Overall, GAN is formalized as a minimax game $\min_G \max_D V(G, D)$ with the value function:

$$V(G, D) = \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim P_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))]. \tag{3}$$

**GAN for discrete data generation.** The generator of a regular GAN cannot generate discrete samples [8]. In order to tackle this limitation, medGAN incorporates an autoencoder in a regular GAN model to generate high-dimensional discrete variables [10]. A basic autoencoder consists of an encoder $Enc$ and a decoder $Dec$. The objective function of the autoencoder is to make the reconstructed input $\mathbf{x}'$ close to the original input $\mathbf{x}$:

$$\mathcal{L}_{AE} = ||\mathbf{x}' - \mathbf{x}||_2^2, \tag{4}$$

where $\mathbf{x}' = Dec(Enc(\mathbf{x}))$.

To generate the dataset which contains discrete attributes, the generator $G_{Dec}$ in medGAN consists of two components, the generator $G$ and the decoder $Dec$. The generator $G$ is trained to generate the salient representations. The decoder $Dec$ from autoencoder seeks to construct the synthetic data from the salient representations $Dec(G(\mathbf{z}))$. Hence, the generator of medGAN $G_{Dec}(\mathbf{z})$ is defined as: $G_{Dec}(\mathbf{z}) = Dec(G(\mathbf{z}))$, where $\mathbf{z}$ is a noise variable. The discriminator $D$ aims to distinguish whether the input is from real data or $Dec(G(\mathbf{z}))$. The generator $G_{Dec}$ can be viewed as a regular generator $G$ with extra hidden layers that maps continuous salient representations to discrete samples.

# IV. FairGAN

## A. Problem Statement

Given a dataset $\{\mathcal{X}, \mathcal{Y}, \mathcal{S}\} \sim P_{\text{data}}$, FairGAN aims to generate a fair dataset $\{\hat{\mathcal{X}}, \hat{\mathcal{Y}}, \hat{\mathcal{S}}\} \sim P_G$ which achieves the *statistical parity* w.r.t the protected attribute $\hat{\mathcal{S}}$, i.e., $P(\hat{y} = 1|\hat{s} = 1) = P(\hat{y} = 1|\hat{s} = 0)$. Meanwhile, our goal is to ensure that given a generated dataset $\{\hat{\mathcal{X}}, \hat{\mathcal{Y}}\}$ as training samples, a classification model seeks an accurate function $\eta : \hat{\mathcal{X}} \to \hat{\mathcal{Y}}$ while satisfying fair classification with respect to $\mathcal{S}$ on the real dataset, i.e., $P(\eta(\mathbf{x}) = 1|s = 1) = P(\eta(\mathbf{x}) = 1|s = 0)$.

## B. Model

FairGAN consists of one generator $G_{Dec}$ and two discriminators $D_1$ and $D_2$. We adopt the revised generator from medGAN [10] to generate both discrete and continuous data. Figure 1 shows the structure of FairGAN. In FairGAN, every generated sample has a corresponding value of the protected attribute $s \sim P_{\text{data}}(s)$. The generator $G_{Dec}$ generates a fake pair $(\hat{\mathbf{x}}, \hat{y})$ following the conditional distribution $P_G(\mathbf{x}, y|s)$. The fake pair $(\hat{\mathbf{x}}, \hat{y})$ is generated by a noise variable $\mathbf{z}$ given the protected attribute $s$, namely,

$$(\hat{\mathbf{x}}, \hat{y}) = G_{Dec}(\mathbf{z}, s) = Dec(G(\mathbf{z}, s)), \mathbf{z} \sim P_{\mathbf{z}}(\mathbf{z}), \tag{5}$$

where $P_{\mathbf{z}}(\mathbf{z})$ is a prior distribution. Hence, the generated fake sample $(\hat{\mathbf{x}}, \hat{y}, \hat{s})$ is from the joint distribution $P_G(\mathbf{x}, y, s) = P_G(\mathbf{x}, y|s)P_G(s)$, where $P_G(s) = P_{\text{data}}(s)$. The discriminator $D_1$ is trained to distinguish between the real data from $P_{\text{data}}(\mathbf{x}, y, s)$ and the generated fake data from $P_G(\mathbf{x}, y, s)$.

Meanwhile, in order to make the generated dataset achieve fairness, a constraint is applied to the generated samples, which aims to keep $P_G(\mathbf{x}, y|s = 1) = P_G(\mathbf{x}, y|s = 0)$. Therefore, another discriminator $D_2$ is incorporated into the FairGAN model and trained to distinguish the two categories of generated samples, $P_G(\mathbf{x}, y|s = 1)$ and $P_G(\mathbf{x}, y|s = 0)$.

The value function of the minimax game is described as:

$$\min_{G_{Dec}} \max_{D_1, D_2} V(G_{Dec}, D_1, D_2) = V_1(G_{Dec}, D_1) + \lambda V_2(G_{Dec}, D_2), \tag{6}$$

where

$$\begin{aligned} &V_1(G_{Dec}, D_1) \\ &= \mathbb{E}_{s\sim P_{\text{data}}(s),(\mathbf{x},y)\sim P_{\text{data}}(\mathbf{x},y|s)}[\log D_1(\mathbf{x}, y, s)] \\ &+ \mathbb{E}_{\hat{s}\sim P_G(s),(\hat{\mathbf{x}},\hat{y})\sim P_G(\mathbf{x},y|s)}[\log(1 - D_1(\hat{\mathbf{x}}, \hat{y}, \hat{s}))], \end{aligned} \tag{7}$$

$$\begin{aligned} V_2(G_{Dec}, D_2) = &\mathbb{E}_{(\hat{\mathbf{x}},\hat{y})\sim P_G(\mathbf{x},y|s=1)}[\log D_2(\hat{\mathbf{x}}, \hat{y})] \\ &+ \mathbb{E}_{(\hat{\mathbf{x}},\hat{y})\sim P_G(\mathbf{x},y|s=0)}[\log(1 - D_2(\hat{\mathbf{x}}, \hat{y}))], \end{aligned} \tag{8}$$

and $\lambda$ is a hyperparameter that specifies a trade off between utility and fairness of data generation.

The first value function $V_1$ is similar to a conditional GAN model [22], where the generator $G$ seeks to learn the joint distribution $P_G(\mathbf{x}, y, s)$ over real data $P_{\text{data}}(\mathbf{x}, y, s)$ by first drawing $\hat{s}$ from $P_G(s)$ and then drawing $\{\hat{\mathbf{x}}, \hat{y}\}$ from $P_G(\mathbf{x}, y|s)$ given a noise variable. Note that in the generated sample $\{\hat{\mathbf{x}}, \hat{y}, \hat{s}\}$, the protected attribute $\hat{s} = s$ is due to the generator conditioning on $s$ to generate $\{\hat{\mathbf{x}}, \hat{y}\}$. The second value function $V_2$ aims to make the generated samples not encode any information supporting to predict the value of protected attribute $s$. Therefore, $D_2$ is trained to correctly predict $s$ given a generated sample while the generator $G$ aims to fool the discriminator $D_2$. Once the generated sample $\{\hat{\mathbf{x}}, \hat{y}\}$ cannot be used to predict the protected attribute $\hat{s}$ ($s$), the correlation between $\{\hat{\mathbf{x}}, \hat{y}\}$ and $s$ is removed, i.e., $\{\hat{\mathbf{x}}, \hat{y}\} \perp\!\!\!\perp s$. FairGAN can ensure that the generated samples do not have the disparate impact.

For the decoder $Dec$ to convert the representations to data samples, FairGAN first pre-trains the autoencoder model. The decoder then can generate samples given the representation from $G(\mathbf{z}, s)$. Meanwhile, since the autoencoder is pre-trained by the original dataset that may contain discrimination information, we further fine-tune the decoder $Dec$ to remove the discrimination information when optimizing $G$. The procedure of training FairGAN is shown in Algorithm 1. FairGAN first pretrains the autoencoder (from Line 1 to 4). For training the generator $G_{Dec}$ and discriminators $D_1$ and $D_2$, FairGAN first samples a batch of real data and a batch of fake data to train $G_{Dec}$ and $D_1$ (from Line 6 to 9) and then applies the fair constraint to train $G_{Dec}$ and $D_2$ (from Line 10 to 12).

**Algorithm 1** The procedure of training FairGAN.

```
1:  for number of pre-training iterations do
2:      Sample a batch of $m$ examples $(\mathbf{x}, y, s) \sim P_{\text{data}}(\mathbf{x}, y, s)$
3:      Update Autoencoder by the loss function in Eq. 4.
4:  end for
5:  for number of training iterations do
6:      Sample a batch of $m$ examples $(\mathbf{x}, y, s) \sim P_{\text{data}}(\mathbf{x}, y, s)$
7:      Sample a batch of $m$ examples $(\hat{\mathbf{x}}, \hat{y}, \hat{s}) \sim P_G(\mathbf{x}, y, s)$ from generator $G_{Dec}(\mathbf{z}, s)$ by first drawing $s \sim P_G(s)$ and noise samples $\mathbf{z} \sim P_{\mathbf{z}}(\mathbf{z})$
8:      Update $D_1$ by Eq. 7;
9:      Update $G_{Dec}$ by Eq. 7;
10:     Sample a batch of $m$ examples $(\hat{\mathbf{x}}, \hat{y}|\hat{s} = 1) \sim P_G(\mathbf{x}, y|s = 1)$ and sample another batch of $m$ examples $(\hat{\mathbf{x}}, \hat{y}|\hat{s} = 0) \sim P_G(\mathbf{x}, y|s = 0)$
11:     Update $D_2$ by by Eq. 8;
12:     Update $G_{Dec}$ by Eq. 8;
13: end for
```

### *C. NaïveFairGAN*

In this subsection, we present a naive approach which can only achieve fair data generation (disparate treatment) but cannot achieve fair classification (disparate impact).

To mitigate the disparate treatment, a straightforward approach is to remove $\mathcal{S}$ from the dataset. Hence, if a GAN model ensures the generated samples have the same distribution as the real data with unprotected attributes and decision, i.e., $P_G(\mathbf{x}, y) = P_{\text{data}}(\mathbf{x}, y)$, and randomly assigns the values of $\hat{\mathcal{S}}$ with only preserving the ratio of protected group to unprotected group the same as the real data, the completely generated dataset could achieve the statistical parity in the dataset. Because there is no additional fair constraint in data generation, the NaïveFairGAN model is a regular GAN model which consists of one generator and one discriminator. The value function of NaïveFairGAN is defined as:

$$\min_{G_{Dec}} \max_{D} V(G_{Dec}, D) = \mathbb{E}_{\mathbf{x},y \sim P_{\text{data}}(\mathbf{x},y)}[\log D_1(\mathbf{x}, y)] + \mathbb{E}_{\hat{\mathbf{x}},\hat{y} \sim P_G(\mathbf{x},y)}[\log(1 - D_1(\hat{\mathbf{x}}, \hat{y}))].$$

In principle, NaïveFairGAN achieves the fair data generation by *randomly generating* the protected attribute $\hat{\mathcal{S}}$. However, due to the property $P_G(\mathbf{x}) = P_{\text{data}}(\mathbf{x})$, the disparate impact caused by the correlation between generated unprotected attributes $\hat{\mathcal{X}}$ and the real protected attribute $\mathcal{S}$ is not removed. The classifier trained on the generated dataset cannot achieve fair prediction when tested on real data.

## V. Experiments

### *A. Experimental Setup*

**Baselines.** To evaluate the effectiveness of FairGAN, we compare the performance of FairGAN with the regular **GAN** model and **NaïveFairGAN** model. GAN aims to generate the synthetic samples that have the same distribution as the real data, i.e., $P_G(\mathbf{x}, y, s) = P_{\text{data}}(\mathbf{x}, y, s)$. The regular GAN model cannot achieve fair data generation. We adopt GAN as a baseline to evaluate the utility of data generation.

In this paper, we don't compare with the pre-process methods, because the classical methods like Massaging cannot remove disparate treatment and disparate impact [6]. Although the certifying framework proposed algorithms to remove disparate impact, they only work on numerical attributes [7].

**Datasets.** We evaluate FairGAN and baselines on the UCI Adult income dataset which contains 48,842 instances [23]. The decision indicates whether the income is higher than \$50k per year, and the protected attribute is gender. Each instance in the dataset consists of 14 attributes. We convert each attribute to a one-hot vector and combine all of them to a feature vector with 57 dimensions.

In our experiments, besides adopting the original Adult dataset, we also generate four types of synthetic data, **SYN1-GAN** that is generated by a regular GAN model, **SYN2-NFGAN** that is generated by NaïveFairGAN, and **SYN3-FairGAN** that is generated by FairGAN with $\lambda = 1$. For each type of synthetic data, we generate five datasets to evaluate the data fairness and classification fairness. We then report the mean and stand deviation of evaluation results. The sizes of the synthetic datasets are same as the real dataset.

**Implementation Details.** We first pretrain the autoencoder for 200 epochs. Both the encoder $Enc$ and the decoder $Dec$ have one hidden layer with the dimension size as 128. The generator $G$ is a feedforward neural network with two hidden layers, each having 128 dimensions. The discriminator $D$ is also a feedforward network with two hidden layers where the first layer has 256 dimensions and the second layer has 128 dimensions. FairGAN is first trained without the fair constraint for 2,000 epochs ($D_1$ and $G_{Dec}$) and then trained with the fair constraint for another 2,000 epochs ($D_1$, $D_2$ and $G_{Dec}$). The regular GAN and NaïveFairGAN are trained for 2,000 epochs. We adopt Adam [24] with the learning rate as 0.001 for stochastic optimization.

### *B. Fair Data Generation*

We evaluate FairGAN on data generation from two perspectives, fairness and utility. Fairness is to check whether FairGAN can generate fair data, while the utility is to check whether FairGAN can learn the distribution of real data.

**Fairness.** We adopt the *risk difference in a labeled dataset* ($disc(\mathcal{D}) = P(y = 1|s = 1) - P(y = 1|s = 0)$) as the

Table I: Risk differences of real and synthetic datasets

| | Real Data | SYN1-GAN | SYN2-NFGAN | SYN3-FairGAN |
|---|---|---|---|---|
| $disc(\mathcal{D})$ | 0.1989 | 0.1798±0.0026 | 0.0025±0.0007 | 0.0411±0.0295 |

metric to compare the performance of different GAN models on fair data generation. Table I shows the risk differences in the real and synthetic datasets. The risk difference in the Adult dataset is 0.1989, which indicates discrimination against female. The SYN-GAN, which is trained to be close to the real dataset, has the similar risk difference to the real dataset. On the contrary, SYN2-NFGAN and SYN3-FairGAN have lower risk differences than the real dataset. In particular, SYN2-NFGAN has extremely small risk differences. This is because the protected attribute of SYN2-NFGAN is independently assigned, i.e., $\hat{y} \perp\!\!\!\perp \hat{s}$. Hence, the synthetic dataset from SYN2-NFGAN is free from disparate treatment. FairGAN prevents the disparate treatment by generating revised $\hat{y}$ to make $\hat{y} \perp\!\!\!\perp \hat{s}$. The risk difference of SYN3-FairGAN is 0.0411, which shows the effectiveness of FairGAN on fair data generation.

We further evaluate the $\epsilon$-fairness (disparate impact) by calculating the balanced error rates (BERs) in the real data and SYN3-FairGAN. Because the protected attribute in SYN2-NFGAN is randomly assigned, the real $s$ given $\hat{\mathbf{x}}$ is unknown. The BERs in SYN2-NFGAN cannot be calculated. The BER in the real dataset is 0.1538, which means a classifier can predict $s$ given $\mathbf{x}$ with high accuracy. Hence, there is disparate impact in the real dataset. On the contrary, the BER in SYN3-FairGAN is 0.3862±0.0036, which indicates using the generated $\hat{\mathbf{x}}$ in SYN3-FairGAN to predict the real $s$ has much higher error rate. The disparate impact in SYN3-FairGAN is small. It shows the effectiveness of FairGAN on removal of the disparate impact in terms of the real $s$. Note that we adopt a linear SVM as a classifier to predict $s$.

**Utility.** We then evaluate the data utility of synthetic datasets. In Table II, we evaluate the closeness between each synthetic dataset and the real dataset by calculating the Euclidean distance of joint and conditional probabilities ($P(\mathbf{x}, y)$, $P(\mathbf{x}, y, s)$, and $P(\mathbf{x}, y|s)$). The Euclidean distance is calculated between the estimated probability vectors (probability mass function) on the sample space from the synthetic dataset and the real dataset. A smaller distance indicates better closeness between the real data and the synthetic data. As expected, SYN1-GAN has the smallest distance to the real dataset for joint and conditional probabilities. For synthetic datasets generated by FairGAN and NaïveFairGAN, SYN2-NFGAN has the smallest distance in terms of $||P_{\text{data}}(\mathbf{x}, y) - P_G(\mathbf{x}, y)||_2$ since its objective is $P_G(\mathbf{x}, y) = P_{\text{data}}(\mathbf{x}, y)$, while SYN3-FairGAN has the smallest distance in terms of conditional probability $||P_{\text{data}}(\mathbf{x}, y|s) - P_G(\mathbf{x}, y|s)||_2$ and joint probability $||P_{\text{data}}(\mathbf{x}, y, s) - P_G(\mathbf{x}, y, s)||_2$ since only FairGAN aims to ensure $P_G(\mathbf{x}, y, s) = P_{\text{data}}(\mathbf{x}, y, s)$. Overall, without considering the protected attribute, all the synthetic datasets from FairGAN and NaïveFairGAN models are close to the real dataset. When considering the protected attribute, FairGAN has better performance than NaïveFairGAN. Therefore, after removing disparate impact, FairGAN still achieves good data utility.

Table II: Euclidean distances of joint and conditional probabilities between synthetic datasets and real dataset

| Euclidean Distance | SYN1-GAN | SYN2-NFGAN | SYN3-FairGAN |
|---|---|---|---|
| $\Vert P_{\text{data}}(\mathbf{x}, y)$<br>$-P_G(\mathbf{x}, y)\Vert_2$ | 0.0231±0.0003 | 0.0226±0.0003 | 0.0233±0.0004 |
| $\Vert P_{\text{data}}(\mathbf{x}, y \vert s = 1)$<br>$-P_G(\mathbf{x}, y \vert s = 1)\Vert_2$ | 0.0108±0.0002 | 0.0118±0.0003 | 0.0111±0.0004 |
| $\Vert P_{\text{data}}(\mathbf{x}, y \vert s = 0)$<br>$-P_G(\mathbf{x}, y \vert s = 0)\Vert_2$ | 0.0166±0.0002 | 0.0194±0.0003 | 0.0176±0.0005 |
| $\Vert P_{\text{data}}(\mathbf{x}, y, s)$<br>$-P_G(\mathbf{x}, y, s)\Vert_2$ | 0.0198±0.0002 | 0.0227±0.0003 | 0.0208±0.0005 |

### C. Fair Classification

In this subsection, we adopt the real and synthetic datasets to train several classifiers and check whether the classifiers can achieve fairness. We evaluate the classifiers with three settings: 1) the classifiers are trained and tested on the real dataset, called **REAL2REAL**; 2) the classifiers are trained and tested on the synthetic datasets, called **SYN2SYN**; 3) the classifiers are trained on the synthetic datasets and tested on the real dataset, called **SYN2REAL**. The ratio of the training set to testing set in these three settings is 1:1. We emphasize that only SYN2REAL is meaningful in practice as the classifiers are trained from the generated data and are adopted for decision making on the real data.

We adopt the following classifiers to evaluate the fair classification: 1) **SVM (linear)** which is a linear support vector machine with $C = 1$; 2) **SVM (RBF)** which is a support vector machine with the radial basis kernel function; 3) **Decision Tree** with maximum tree depth as 5; Note that we do not adopt the protected attribute and only use the unprotected attributes to train classifiers, which ensures no disparate treatment in classifiers.

**Fairness.** We adopt the *risk difference in a classifier* ($disc(\eta) = P(\eta(\mathbf{x}) = 1|s = 1) - P(\eta(\mathbf{x}) = 1|s = 0)$) to evaluate the performance of classifier on fair prediction. Table III shows the risk differences in classifiers on various training and testing settings. We can observe that when the classifiers are trained and tested on real datasets (i.e., REAL2REAL), the risk differences in classifiers are high. It indicates that if there is disparate impact in the training dataset, the classifiers also incur discrimination for prediction. Since SYN1-GAN is close to the real dataset, classifiers trained on SYN1-GAN also have discrimination in both SYN2SYN and SYN2REAL settings.

Although SYN2-NFGAN has similar distribution as the real dataset on unprotected attributes and decision, i.e., $P_G(\mathbf{x}, y) = P_{\text{data}}(\mathbf{x}, y)$, classifiers which are trained and tested in SYN2SYN settings achieve low risk differences. This is because the values of the protected attribute in SYN2-NFGAN are independently generated. Since both $\hat{\mathbf{x}}$ and $\hat{y}$ have no correlations with the generated $\hat{s}$ in SYN2-NFGAN, the statistical parity in classifiers can be achieved when trained and tested on synthetic datasets.

However, when classifiers are trained on SYN2-NFGAN and tested on the real dataset (i.e., SYN2REAL), the classi-

Table III: Risk differences in classifiers and classification accuracies on various training and testing settings

| | Classifier | Real2Real | SYN2SYN | | | SYN2REAL | | |
|---|---|---|---|---|---|---|---|---|
| | | | SYN1-GAN | SYN2-NFGAN | SYN3-FairGAN | SYN1-GAN | SYN2-NFGAN | SYN3-FairGAN |
| Risk Difference | SVM (Linear) | 0.1784 | 0.1341±0.0023 | 0.0018±0.0021 | 0.0371±0.0189 | 0.1712±0.0062 | 0.1580±0.0076 | 0.0461±0.0424 |
| | SVM (RBF) | 0.1788 | 0.1292±0.0049 | 0.0018±0.0025 | 0.0354±0.0206 | 0.1623±0.0050 | 0.1602±0.0053 | 0.0526±0.0353 |
| | Decision Tree | 0.1547 | 0.1396±0.0089 | 0.0015±0.0035 | 0.0535±0.0209 | 0.1640±0.0077 | 0.1506±0.0070 | 0.0754±0.0641 |
| Accuracy | SVM (Linear) | 0.8649 | 0.8281±0.0103 | 0.8162±0.0133 | 0.8247±0.0115 | 0.8363±0.0108 | 0.8340±0.0091 | 0.8217±0.0093 |
| | SVM (RBF) | 0.8433 | 0.8278±0.0099 | 0.8160±0.0100 | 0.8233±0.0103 | 0.8342±0.0036 | 0.8337±0.0060 | 0.8178±0.0128 |
| | Decision Tree | 0.8240 | 0.8091±0.0059 | 0.7926±0.0083 | 0.8077±0.0144 | 0.8190±0.0051 | 0.8199±0.0041 | 0.8044±0.0140 |

fiers still have significant discrimination against the protected group. Because the unprotected attributes of SYN2-NFGAN are close to the real dataset, the correlations between the generated $\hat{\mathbf{x}}$ and the real $s$ are still preserved. The disparate impact in terms of the real $s$ on SYN2-NFGAN is not removed. When classifiers are tested on the real dataset where the correlations between $\mathbf{x}$ and $s$ are preserved, the classification results indicate discrimination. On the contrary, when the classifiers are trained on SYN3-FairGAN and tested on the real dataset, we can observe that the risk differences in classifiers are small. Since the FairGAN prevents the discrimination by generating $\hat{\mathbf{x}}$ that don't have correlations with the real $s$, the classifier trained on SYN3-FairGAN can achieve fair classification on the real dataset. It demonstrates the advantage of FairGAN over the NaïveFairGAN on fair classification.

**Classification accuracy.** Table III further shows the classification accuracies of different classifiers on various training and testing settings. We can observe that the accuracies of classifiers on the SYN2REAL setting are close to the results on the REAL2REAL setting. It indicates synthetic datasets generated by different GAN models are similar to the real dataset, showing the good data generation utility of GAN models. Meanwhile, accuracies of classifiers which are trained on SYN3-FairGAN and tested on real dataset are only slightly lower than those trained on SYN1-GAN, which means the FairGAN model can achieve a good tradeoff between utility and fairness. The small utility loss is caused by modifying unprotected attributes to remove disparate impact in terms of the real $s$.

## VI. Conclusions and Future Work

In this paper, we have developed FairGAN to generate fair data, which is free from disparate treatment and disparate impact, while retaining high data utility. As a result, classifiers trained on the generated fair data are not subject to discrimination when making decisions on the real data. FairGAN consists of one generator and two discriminators. In particular, the generator generates fake samples conditioned on the protected attribute. One discriminator is trained to identify whether samples are real or fake, while the other discriminator is trained to distinguish whether the generated samples are from the protected group or unprotected group. The generator can generate fair data with high utility by playing the adversarial games with these two discriminators. The experimental results showed the effectiveness of FairGAN.

## Acknowledgment

This work was supported in part by NSF 1564250, 1646654 and 1841119.

## References

[1] M. Joseph, M. Kearns, J. H. Morgenstern, and A. Roth, "Fairness in learning: Classic and contextual bandits," in *NIPS*, 2016.

[2] A. Beutel, J. Chen, Z. Zhao, and E. H. Chi, "Data decisions and theoretical implications when adversarially learning fair representations," in *FAT/ML*, 2017.

[3] R. Binns, "Fairness in machine learning: Lessons from political philosophy," *arXiv:1712.03586 [cs]*, 2017.

[4] L. Zhang, Y. Wu, and X. Wu, "A causal framework for discovering and removing direct and indirect discrimination," ser. IJCAI, 2017.

[5] F. Kamiran and T. Calders, "Classifying without discriminating," in *Control and Communication 2009 2nd International Conference on Computer*, 2009.

[6] ——, "Data preprocessing techniques for classification without discrimination," *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, 2012.

[7] M. Feldman, S. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *KDD*, 2015.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *NIPS*, 2014.

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

[10] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," in *MLHC*, 2017.

[11] T. Calders, F. Kamiran, and M. Pechenizkiy, "Building classifiers with independency constraints," in *ICDM Workshops*, 2009.

[12] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," *arXiv:1104.3913 [cs]*, 2011.

[13] Y. Wu and X. Wu, "Using loglinear model for discrimination discovery and prevention," in *DSAA*, 2016.

[14] T. Kamishima, S. Akaho, and J. Sakuma, "Fairness-aware learning through regularization approach," in *ICDM Workshops*, 2011.

[15] M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *AISTATS*, 2017.

[16] F. Kamiran, T. Calders, and M. Pechenizkiy, "Discrimination aware decision tree learning," in *ICDM*, 2010.

[17] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *NIPS*, 2016.

[18] L. Zhang, Y. Wu, and X. Wu, "Achieving non-discrimination in data release," in *KDD*, 2017.

[19] H. Edwards and A. Storkey, "Censoring representations with an adversary," *arXiv:1511.05897 [cs, stat]*, 2015.

[20] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning adversarially fair and transferable representations," *arXiv:1802.06309 [cs, stat]*, 2018.

[21] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *AIES*, 2018.

[22] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784 [cs, stat]*, 2014.

[23] D. Dheeru and E. Karra Taniskidou, *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2017.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.