Fall 12-13-2022

Fairness and Privacy in Machine Learning Algorithms

Neha Bhargava

Fairness and Privacy in Machine Learning Algorithms

A Thesis presented to The Faculty of the
Computer Science Department

By

Neha Bhargava

In Partial Fulfillment
of Requirements for the Degree
Master of Science, Computer Science

Kennesaw State University
Fall 2022

Fairness and Privacy in Machine Learning Algorithms

Approved:

DocuSigned by:
Ramazan Aygun
6A0A9AE64C45477...
December 13, 2022

Dr. Ramazan Aygun - Advisor

DocuSigned by:
Yong Pei
E5B40411E13C445...
December 13, 2022

Dr. Yong Pei – Computer Science Chair

DocuSigned by:
Sumanth Yenduri
B04458D098CE4E8...
December 13, 2022

Dr. Sumanth Yenduri - Dean

In presenting this thesis as a partial fulfillment of the requirements for an advanced degree from Kennesaw State University, I agree that the university library shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the dean of the appropriate school when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from or publication of this thesis which involves potential financial gain will not be allowed without written permission.

Neha Bhargava

Notice to Borrowers

Unpublished thesis deposited in the Library of Kennesaw State University must be used only in accordance with the stipulations prescribed by the author in the preceding statement

The author of this thesis is:

Neha Bhargava

The director of this thesis is:

Dr. Ramazan Aygun

Users of this thesis not regularly enrolled as students at Kennesaw State University are required to attest acceptance of the preceding stipulations by signing below. Libraries borrowing this thesis for the use of their patrons are required to see that each user records here the information requested.

Fairness and Privacy in Machine Learning Algorithms

An Abstract of
a Thesis Presented to
The Faculty of the Computer Science Department

By

Neha Bhargava
Bachelor of Science in Electronics and Instrumentation Engineering, DAVV Indore
India, 2006

In Partial Fulfillment
of Requirements for the Degree
Master of Science, Computer Science

Kennesaw State University
Fall 2022

Abstract

In this age of big data, one of the key concerns in the recent days has been bias present in the data and hence the need to ensure data fairness. According to dictionary definition, fairness refers to impartial and just treatment without any favoritism or discrimination among various groups of individuals. There is a need to ensure that bias in the data does not reflect in the models decision which in turn treats people from certain race, gender, sexual or political orientation unfairly and differently. The goal of fair data generation is to remove any prejudice which might be present in the data towards any specific demographic group. This is particularly of interest in decision making scenarios like financial lending, hiring, pretrial and immigration detention, health care, social services, and education where the system might favor one race and is biased towards the other. In this thesis, we propose ImpartialGAN to generate fair synthetic data from real data. The generated data is not only fair and free from bias but also ensures a good data utility while preserving data privacy. Hence this generated data can be used in place of real data for predictive analytics. We performed experiments on three datasets UCI Adult dataset, UCI German Credit Dataset and COMPAS dataset from ProPublica.

Fairness and Privacy in Machine Learning Algorithms

A Thesis Presented to The Faculty of the
Computer Science Department

By

Neha Bhargava

In Partial Fulfillment
of Requirements for the Degree
Master of Science, Computer Science

Advisor: Dr. Ramazan Aygun

Kennesaw State University
Fall 2022

Acknowledgment

Foremost, I am grateful and would like to express my sincere gratitude to my advisor Dr. Ramazan Aygun for his patience, motivation, enthusiasm, and immense knowledge. His course Machine Learning inspired me to undertake this research. He was very supportive and was always available to help me out throughout my research and thesis writing every time I had a problem. His guidance helped me in completing my research and in writing this thesis.

I would also like to give special thanks to my husband Nishank Jain and my sister Deepali Bhargava for their continuous support and understanding of the time and commitment required for my research and writing my thesis. I would like to also thank my father Nirmal Bhargava for his unconditional love and support throughout my studies.

I would like to thank my committee members Dr. Noh and Dr. Lee for their brilliant comments and suggestions. I would also like to thank the Computer Science department at KSU for providing the facilities and technical support to help me through this research.

Contents

Contents i

Chapter 1 Introduction 5

1.1. Motivation 5

1.2. Approach 6

1.3. Thesis Organization 7

Chapter 2 Related Work 8

2.1. Causal Graph based Approach 8

2.2. Achieving Fairness through Latent space de-biasing 10

2.3. Removing Bias through Adversarial Learning 12

2.4. Removing Algorithmic Bias Using Learned Latent Structure 13

2.5. Fairness using Flexibly Fair Representations 14

2.6. Fairness using Generative Adversarial Networks 16

2.7. Research Challenges 17

Chapter 3 Approach 19

3.1. GAN and Autoencoder setup for generating continuous and discrete data 19

3.2. ImpartialGAN Model 21

3.3. Fairness and discrimination metric 25

3.4. Algorithm 26

Chapter 4 Experiments 28

4.1. Experimental Setup 28

4.2. Classification Models and Settings 28

4.3. Evaluation on Adult Dataset. 29

4.3.1 Performance on Adult Dataset 31

4.3.2 Discussion about Adult Dataset 35

4.4 Evaluation on German Credit Dataset 36

4.4.1 Performance on German Credit Dataset 37

4.4.2 Discussion about German Credit Dataset 42

4.5 Evaluation on COMPAS Dataset 43

4.5.1 Performance on COMPAS dataset 44

4.5.2 Discussion on COMPAS dataset 49

5 Conclusion and Future Work 50

5.1 Future Work 50

References 52

List Of Figures

Figure 2.1-1: Example causal graph (Diagram based on [19]) 8
Figure 2.2-1: Picture taken from [20]: Training data augmentation 11
Figure 2.3-1: Model architecture (Diagram based on [21]) 13
Figure 2.4-1: Data debiasing (Picture from [22]) 14
Figure 2.5-1: Architecture of the proposed model (Picture from [23]) 15
Figure 2.6-1: Diagram taken from [7] : FairGAN architecture 16
Figure 3.1-1: Regular GAN architecture 19
Figure 3.1-2: Autoencoder architecture (Picture taken from [25]) 20
Figure 3.2-1: ImpartialGAN architecture 22

List Of Tables

Table 2.1-1: Example statistics (Table from [19]) 9
Table 4.3-1: Adult dataset: Risk difference in real and synthetic datasets 30
Table 4.3.1-1: Adult Dataset: Classifier risk difference and accuracy for RealtoReal setting 31
Table 4.3.1-2: Adult dataset: Classifier risk difference and accuracy for SyntoReal Dataset....... 33
Table 4.3.1-3: Adult dataset: Classifier risk difference and accuracy for SyntoSyn setting 34
Table 4.3.2-1: Adult Dataset: Pearson coefficient 35
Table 4.4-1: German Credit dataset: Risk difference in real and synthetic datasets 36
Table 4.4.1-1: German Credit dataset: Classifier risk difference and accuracy for RealtoReal setting 38
Table 4.4.1-2: German Credit dataset: Classifier risk difference and accuracy for SyntoReal setting 40
Table 4.4.1-3: German Credit dataset: Classifier risk difference and accuracy for SyntoSyn setting 41
Table 4.4.2-1: German Credit dataset: Pearson coefficient 42
Table 4.5-1: COMPAS dataset: Risk difference in real and synthetic datasets............................ 43
Table 4.5.1-1: COMPAS dataset: Classifier risk difference and accuracy for RealtoReal Dataset 45
Table 4.5.1-2: COMPAS dataset: Classifier risk difference and accuracy for SyntoReal setting .. 47
Table 4.5.1-3: COMPAS dataset: Classifier risk difference and accuracy for SyntoSyn setting.... 48
Table 4.5.2-1: COMPAS dataset: Pearson coefficient.. 49

Chapter 1 Introduction

Roughly 2.5 quintillion bytes of data is generated daily in this digital era. Manual processing of such huge amounts of data to extract useful information is nearly impossible but with the widespread use of machine learning algorithms and their ability to process enormous data in a fast, cost-effective, and scalable way has proven to be a preferred choice to glean useful insights and solve business problems in many domains. With this widespread use of machine learning algorithms there has always been concerns about the ethical issues that may arise from the use of this modern technology. While achieving high accuracies, accomplishing trustable [1], [2] and fair machine learning has been challenging. Maintaining data fairness and privacy is one of the top challenges faced by the industry as organizations employ various machine learning algorithms to automatically make decisions based on trends from previously collected data [3].

Protected group or attribute [4] refers to the group of individuals towards whom the system has some preconceived reservations and hence is discriminatory. Discrimination is the unjustified treatment towards a particular category of people based on their race, age, gender, religion, sexual orientation, or disability. If we use the data with preconceived reservation or inbuilt discrimination towards certain group, then the model trained on such data will also be discriminatory towards these specific individuals [5].

1.1. Motivation

While one approach can be to train the classifier without the protected attribute and use only the unprotected attribute for training the classifier but in many cases the protected attributes information is encapsulated in other unprotected attributes. For example, even if we remove protected attributes such as race and ethnicity from training data, the information related to these attributes might be present in other unprotected attributes such as postal zip code, county of residence, and country of origin. So, the model will implicitly learn the protected attribute information from these attributes and will be biased[6]. Hence, we also need a way to ensure that protected information is not stored in other attributes which are not protected and ultimately becomes a deciding factor in the model's outcome.

Another approach is to generate fair synthetic data from historical datasets. This approach was used by [7], [8]. Here in this thesis, we modified and improved FairGAN [7], which could still have implicit correlation between protected and unprotected attributes. As in [7], we similarly used real data which includes the protected attribute and used it to generate synthetic data which is free from bias. We used Generative Adversarial Networks (GANs) to generate synthetic data as GANs are able to closely replicate real data distributions and generate good quality synthetic data [9]. Once we have the synthetic data, it can be used for predictive modeling instead of using the real data that might be biased.

1.2. Approach

ImpartialGAN consist of four components: one generator and three discriminators. The first discriminator makes sure that the generated data is as close to real data as possible. The second discriminator ensures that the generated unprotected attributes along with the associated generated decision attribute taken together are jointly independent of the protected attribute. These components are very similar to the components in FairGAN [7]. To remove any residual correlation between unprotected attributes and the protected attribute, we introduce the third discriminator to enforce that, unprotected attributes do not encapsulate any information about the protected attribute. In this work we make sure that the generated data is similar to the real data and does not contain any information about the protected attribute while still maintaining a good correlation between the unprotected attributes and the output decision. Throughout this thesis I will be using the name synthetic and fake data interchangeably.

1.3. Thesis Organization

The thesis is organized as follows. Chapter 2 discusses related work about fairness. Chapter 3 explains the general mechanics of GAN followed by the in-depth components, architecture, and pseudocode of ImpartialGAN. The experimental setup along with the results obtained on the various datasets are explained in Chapter 4 followed by the conclusion and future work in Chapter 5.

Chapter 2 Related Work

Fairness and bias mitigation research have taken three routes: a) remove bias from the real data, b) generate synthetic bias free data, and c) build classifiers sans discrimination for predictive modeling. Zhang et al. [10] categorize methods for constructing discrimination free classifiers as pre-process methods, in-process methods, and post-process methods. Pre- process methods [6], [11]–[14] use techniques like massaging, reweighing, or resampling that modify the training data to remove bias and then this modified data is used for predictive modeling. For in-process methods [15], [16], a fairness constraint or regularization term is applied to the classifier to achieve fair classification. Lastly the post-process methods [17], [18] change the predicted label to remove discrimination.

2.1. Causal Graph based Approach

To achieve fairness, we should be able to identify whether the discrimination is towards a specific group. Zhang et al. [19] proposed using causal graphs to find meaningful partitions in the data to identify that the discrimination in the decision is caused due to the individual's protected attribute.

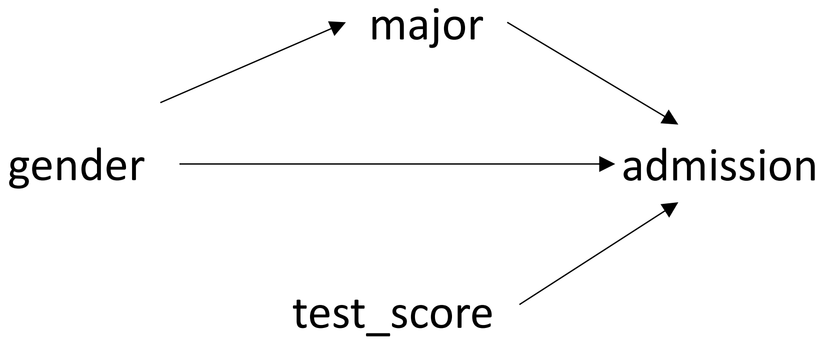

Figure 2.1-1: Example causal graph (Diagram based on [19])

To understand causal graphs and meaningful partitions the authors used the example of the university admission system with only four attributes namely gender, major, test_score and the result attribute admission. Figure 2.1-1 shows the causal graph for this university admission system. Here an arc between the attributes shows a causation. The cause of each node is its parent node. Next to explain a meaningful partition in a dataset they used the example statistics shown in Table 2.1-1 . An example of a meaningful partition with respect to the university admission system can be one where the data is partitioned based on the combination {major, test_score} and there is a substantial difference in the admission rates between male and females when a particular group of test score is considered. So, when we consider a test score of 'L' Table 2.1-1 shows discrimination against females based on the number of applicants. Similarly, there is discrimination against males when considering a test score of 'H'.

Major	**CS**				**EE**			
Gender	Female		Male		Female		Male	
Test Score	L	H	L	H	L	H	L	H
No. applicants	450	300	150	100	600	300	200	100
Admission Rate	30%	50%	36%	40%	40%	60%	45%	50%
	38%		38%		47%		47%	

Table 2.1-1: Example statistics (Table from [19])

Zhang et al. [19] then proposed two approaches to remove discrimination. In the first approach they modified the causal graph and used it to generate a dataset free from discrimination which can be used for predictive modeling. More specifically they generated data by modifying the conditional probability table of the decision attribute. This was done to remove bias from the relevant subgroups of the meaningful partitions. Hence the complexity of this approach is dependent on two factors: complexity of finding causal graphs and time required to solve quadratic programming.

The authors also suggested a second approach to remove discrimination by modifying the dataset. In this approach, random data points are selected with either the positive protected attribute and a negative decision or positive attribute and a positive decision. After this the decision attributes value is flipped for the selected data points. The complexity of this approach is dependent on finding the relevant sub population and the size of the dataset. By using the second approach the efficiency of the algorithm is compromised.

2.2. Achieving Fairness through Latent space de-biasing

Ramaswamy et al. [20] identified the need to remove the correlation between the decision, and the protected attributes in machine vision space. They proposed using GANs for data augmentation as they are able to produce realistic images. Their approach involved making perturbations in the GAN latent space which removes the correlation between the protected attribute and decision in the generated data set.

To understand the issue in machine vision space, they gave the example of a visual classifier where the classifier was trained to recognize whether a person is wearing a hat or not. In general, wearing a hat can be correlated to wearing sunglasses when it’s sunny outside, but this hypothesis is not always true.

In the case of imbalanced training datasets, the classifier will learn this correlation between hats and glasses. Hence it will fail to recognize the presence of hat in the absence of sunglasses on the other hand in the presence of sunglasses it might falsely predict that the person is wearing a hat even when they are not wearing one. To remove this correlation, they proposed data augmentation by adding GAN generated images to the training dataset. The generated images consisted of both types of images one without hat, but the person is wearing sunglasses and the second where the person is wearing a hat but has no sunglasses. This helps in removing the correlation between the two attributes. Figure 2.2-1 depicts this approach.

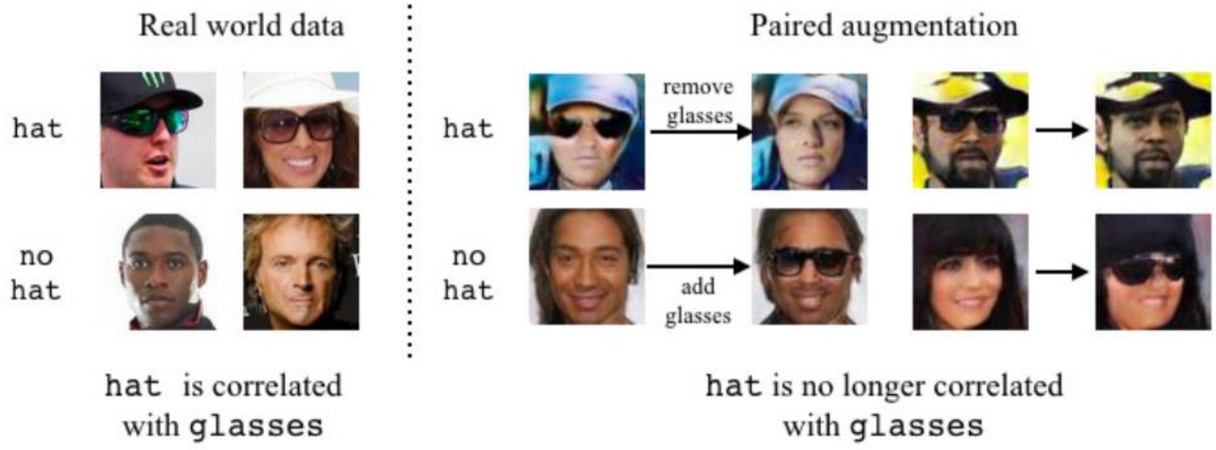

Figure 2.2-1: Picture taken from [20]: Training data augmentation

Its shortcoming are they do not consider the correlation between the unprotected attributes and the protected attributes which can influence the decision for the protected group.

2.3. Removing Bias through Adversarial Learning

Research in the literature has shown well trained models reflect biases that are present in the dataset. To mitigate such bias, Zhang et al. in [21] proposed an architecture comprising of two models namely a predictor model and an adversarial model. The predictor model is used to predict the target variable from the data. Next this prediction is fed as input to the adversary network which tries to predict the correct value of the protected attribute. Depending on the type of fairness that needs to be achieved whether it is demographic parity, equality of odds or equality of opportunity, there may be additional inputs to the adversarial network. The gradient of the adversarial model is incorporated in the predictor model via weight update to avoid leakage of information about the protected attribute. Figure 2.3-1 shows the architecture of their proposed model.

The aim over here is to maximize the Predictor model's ability to successfully predict the value of the decision/target variable. At the same time the adversarial network ensures the decision attribute does not encapsulate any information regarding the protected attribute. In the experiments the authors used the UCI Adult dataset for the classification task and used two logistic regression models - one for the predictor model and another for the adversarial model. However, in general, any gradient based learning models can be used. One of the drawbacks of this approach is that if the hyperparameters are not set correctly then the algorithm diverges, and the adversarial training becomes hard.

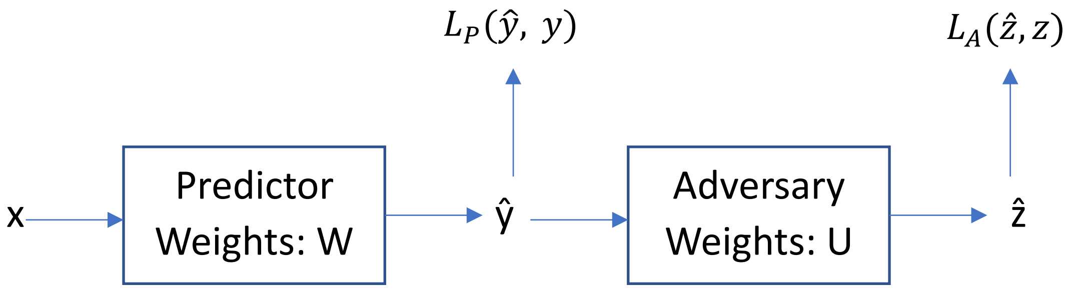

Figure 2.3-1: Model architecture (Diagram based on [21])

2.4. Removing Algorithmic Bias Using Learned Latent Structure

Amini et al. [22] used an extension of the variational autoencoder also known as debiasing variational autoencoder to mitigate bias and to increase classification accuracy. The purpose was to remove gender and racial bias in facial detection systems. In general, the system first learns all the latent/sensitive variables of the class in an unsupervised manner. Next these variables are used to resample the dataset while training so that the classifiers are unbiased.

Their approach can be better understood from Figure 2.4-1. Their algorithm uses Variational Autoencoders to identify the underrepresented attributes in the dataset. Next it increases the sampling probability of these attributes. In the Figure 2.4-1 the group of images on the left are sampled without debiasing whereas images on the right are with debiasing and hence have more diverse attributes like skin color, illumination etc. In their experiments they used images from CelebA and ImageNet datasets.

In facial detection systems the latent attribute can be skin color, age, or gender. In order to implement fairness in such classifiers the distribution of these latent attributes should be uniform. This is different from class imbalances. When there is a class imbalance in a training set, we try to have roughly the same number of samples of all the classes in a particular batch. Here in this proposed algorithm, it means these latent attributes are uniform within a particular class.

Simply put all the latent variables in a particular class should be balanced. For example, for a particular sample if we change the value of a latent variable (example skin tone from dark to light) then the classifier should still be able to predict the output label correctly.

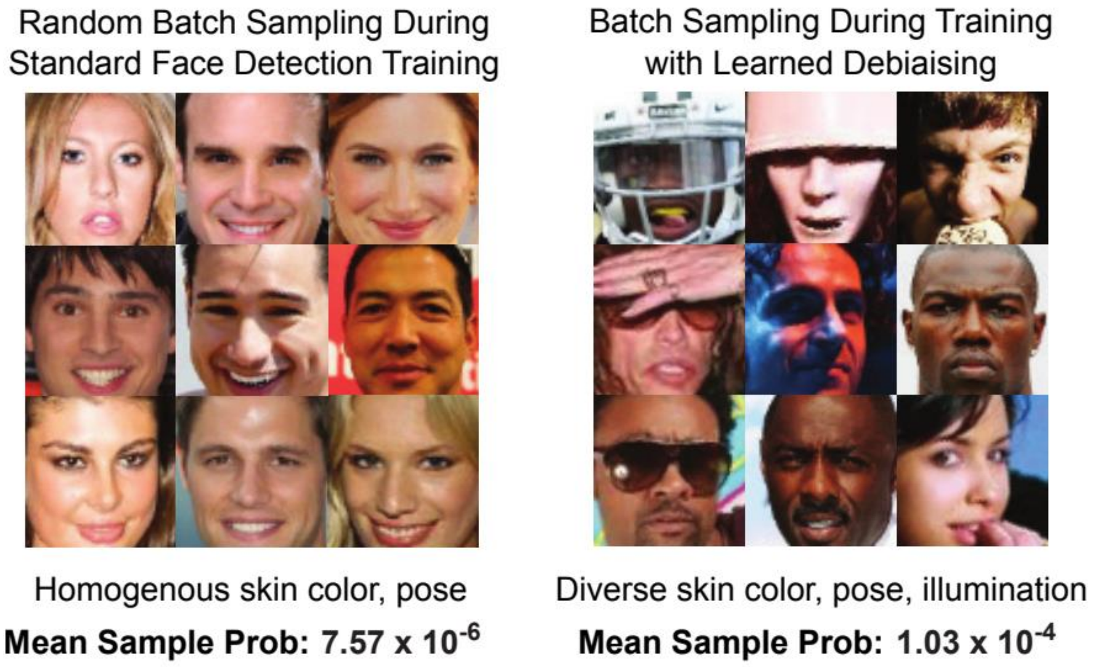
Random Batch Sampling During Standard Face Detection Training
Batch Sampling During Training with Learned Debiaising
Homogenous skin color, pose
Mean Sample Prob: 7.57 x 10^{-6}
Diverse skin color, pose, illumination
Mean Sample Prob: 1.03 x 10^{-4}

Figure 2.4-1: Data debiasing (Picture from [22])

2.5. Fairness using Flexibly Fair Representations

Creager et al. [23] used flexibly fair representations to build a fair model for a variety of protected groups. Their method can be used for a variety of downstream tasks as learned representations are disentangled from multiple sensitive attributes during training. In their experiments they satisfied demographic parity so that the prediction label was independent of the set of sensitive attributes. Figure 2.5-1 helps in understanding their approach. Here protected attributes are referred as sensitive attributes and the unprotected attributes are referred to as non-sensitive.

Given a dataset D all the unprotected attributes can be represented by x. The set of all the sensitive attributes is represented by a and y is the label to predicted. The Variational

Autoencoder learns the latent representation of the unprotected attributes which is represented by z. Latent representation of the protected attributes are represented by b. The author's approach was to have a latent subspace for each protected attribute in a way that a subspace for a particular protected attribute is independent of the subspace of the other protected attributes.

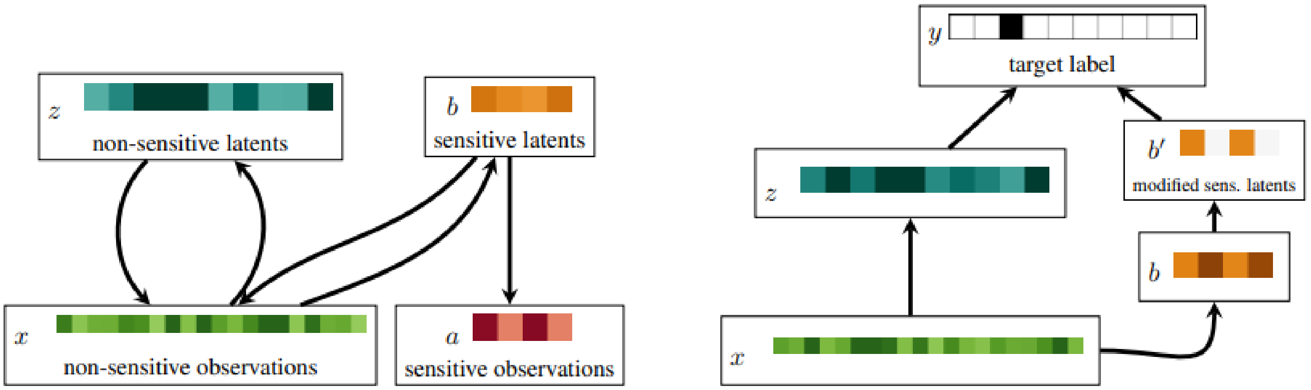

Figure 2.5-1: Architecture of the proposed model (Picture from [23])

They performed three tasks namely fair classification, predictiveness, and disentanglement to ensure the performance of their method. For fair classification the model was trained to predict y given the vectors z and b. Here they removed the concerned protected attributes dimensions from b and evaluated the model's performance on the test set. This task was repetitively performed for each protected attribute one by one.

For the second task on predictiveness a classifier was trained to correctly predict the value of a protected attribute from the latent representations b.

Lastly for disentanglement a separate classifier was trained to predict the value of a specific protected attribute say a_i from the latent space of the unprotected attributes and the latent space of the remaining protected attributes. If the classifier loss is low, then this shows predictiveness and if the loss is high, it shows disentanglement.

The authors applied their method on two datasets the Communities and Crime Dataset and the Celeb-A dataset. One of the drawbacks of their approach is that much of the research uses synthetic data which has uniform distribution of the various factors to check for disentanglement, which may not be the case in the real world.

2.6. Fairness using Generative Adversarial Networks

Xu et al. [7] used a GAN to generate fair synthetic data along with the decision from noise conditioned on the protected attribute gender by using an additional discriminator to enforce fairness by removing the correlation between the protected attribute gender and the other unprotected attributes along with the decision.

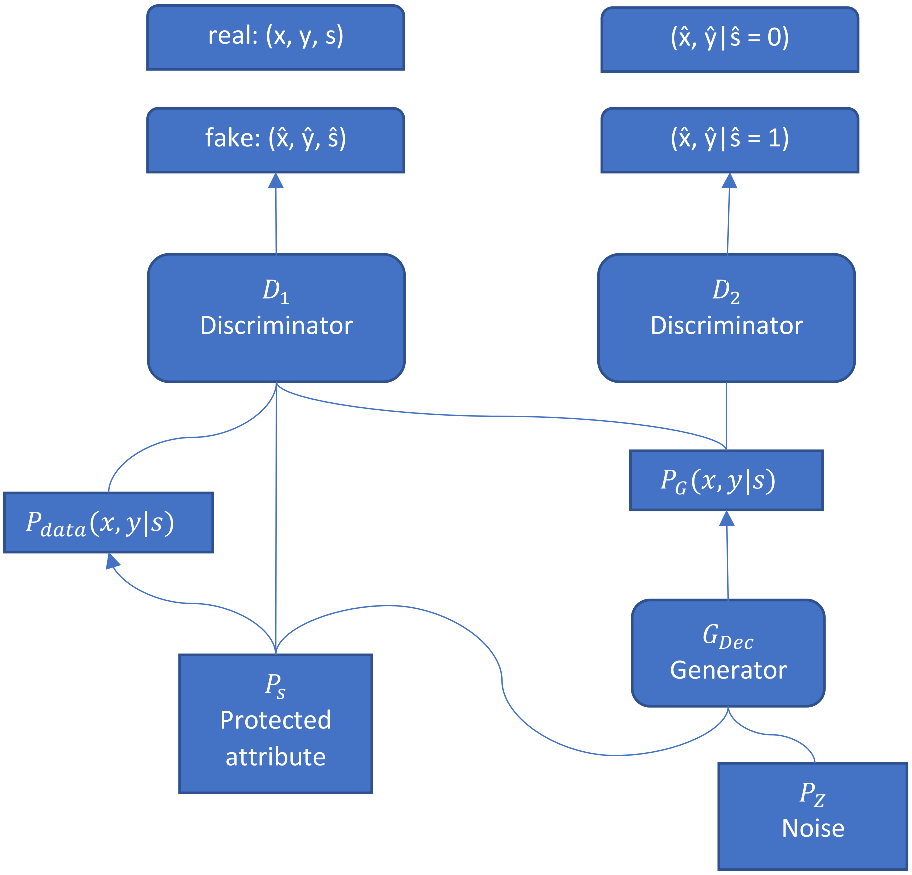

Figure 2.6-1: Diagram taken from [7] : FairGAN architecture

Figure 2.6-1 shows the architecture of FairGAN. Since GANs are able to generate good quality data the authors used GANs in their architecture. However instead of using one discriminator, their framework used two to achieve fairness. They used a modified GAN generator which was conditioned on noise and the protected attribute to generate synthetic data. The first discriminator ensures that synthetic data is like real data. The second discriminator ensures data fairness in the remaining attributes including the decision attribute by ensuring that no information regarding the protected attribute is stored.

While reproducing their experiments we found that the unprotected attributes still had information encapsulated about the protected attributes which might be affecting the output decision of the model. Our algorithm, ImpartialGAN, removes this correlation between the protected and the unprotected attributes.

2.7. Research Challenges

As discussed earlier we can leave the protected attribute and use only the unprotected attributes for training the classifier but it’s highly likely that the protected attributes information is encapsulated in other unprotected attributes. For example, there was an initiative at Amazon to automate the hiring process. The algorithm was designed to shortlist the resume of the people that Amazon should hire. Later on it was discovered that the algorithm was biased towards females as majority of the software engineers hired by the company were males [24]. This happened because the historical data on which the algorithm was trained was biased. Now we can argue that removing the gender or say the names of the applicants from the resume will make the system bias free but in reality, the algorithm can learn about a person’s gender through the words mentioned in their hobbies like women’s rugby team and the college they attended.

The system can infer that a person is a female if she attended an all-women's college. So, the model will implicitly learn the protected attribute's information from these attributes and will be biased. Hence, we need a way to ensure that protected information is not stored in other attributes which are not protected and ultimately becomes a deciding factor in the model's outcome. This issue has been addressed by our proposed algorithm ImpartialGAN.

Chapter 3 Approach

3.1. GAN and Autoencoder setup for generating continuous and discrete data

GANs consist of two parts: a generator G and a discriminator D. The generator model produces synthetic data from random noise z following the noise distribution P_z. The data from the generator along with real data x from a training data set are given as inputs to the discriminator, which attempts to distinguish between the inputs x and the G(z) data generated by the generator. Over the course of the training the generator gets better at creating samples that look more and more like the real data by following the real data distribution P_{data} while the discriminator is unable to distinguish between the real data and the synthetic data.

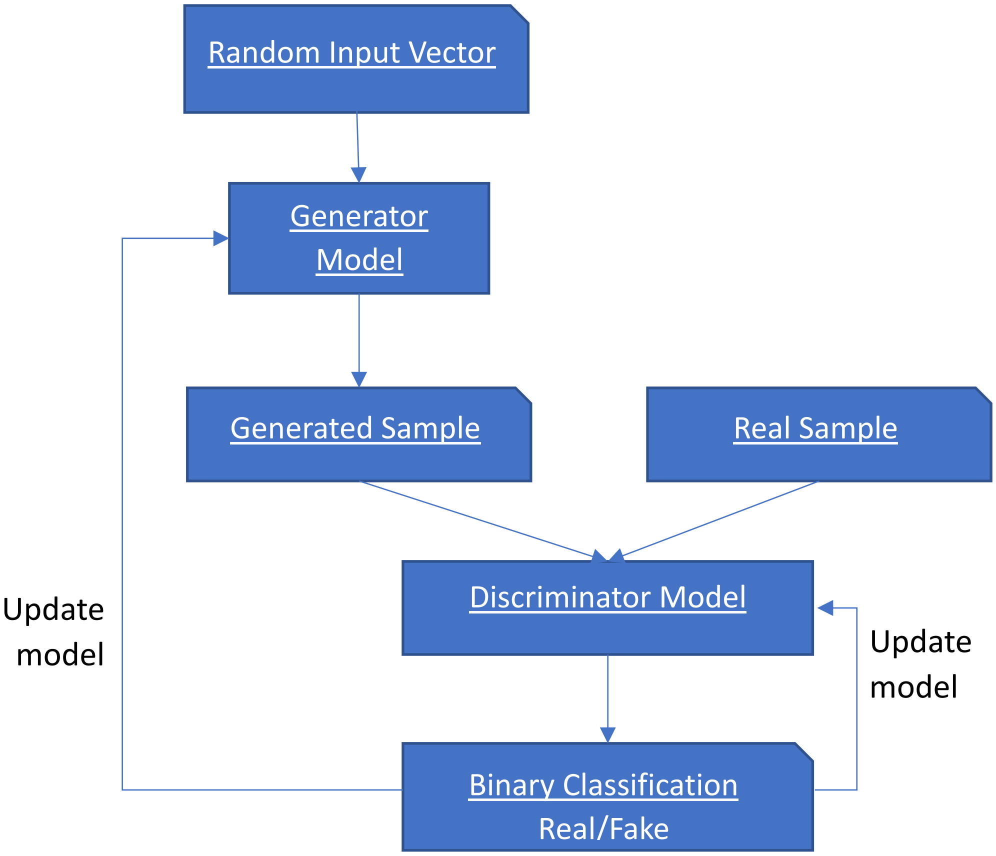

Figure 3.1-1: Regular GAN architecture

The architecture of a regular GAN is as shown in Figure 3.1-1. For simplicity and consistency, we have adopted the same notation convention as in FairGAN [7]. A GAN value function can be represented as in Equation 1.

$$V(G,D) = E_{x\sim P_{data}}[logD(x)] + E_{z\sim P_z}[\log(1 - D(G(z))]$$

Equation 1

Autoencoders are based on neural networks. Their objective is to first compress the input data to a latent space also known as the bottleneck which consists of the most important representations of the input data. Next the input is reconstructing from this compressed form. This process helps the Autoencoder in learning the most important hidden features in the input.

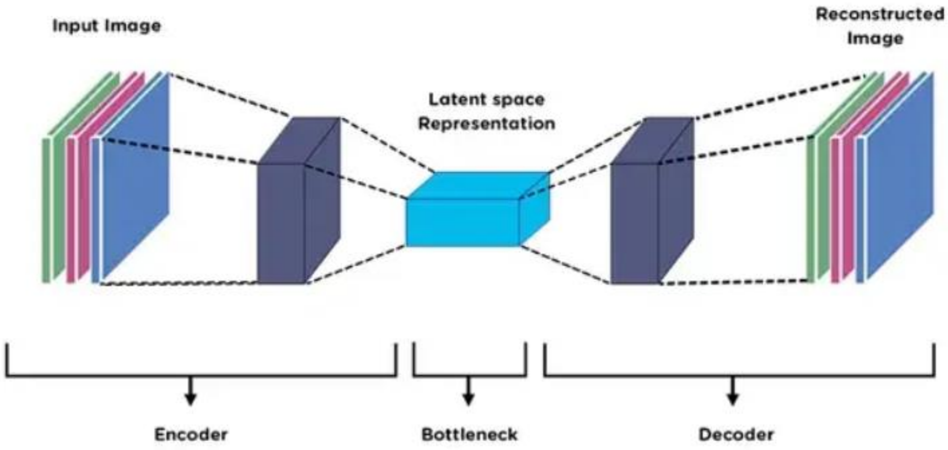

Figure 3.1-2: Autoencoder architecture (Picture taken from [25])

Figure 3.1-2 depicts a general Autoencoder architecture [25]. Autoencoders are only able to compress data on which they are trained on. An Autoencoder setup consists of an Encoder and a Decoder.

An Encoder compresses the data to a lower dimension known as latent representations. These representations are different from the original input. Decoder part of the Autoencoder architecture tries to reconstruct the original input from the compressed version generated by the encoder.

As GANs are unable to generate discrete data, FairGAN adopted the modified generator G_{Dec} from medGAN [26], and similarly we also used it instead of the generator model from GAN architecture. Here, the generator of the GAN generates the salient representations over a noise variable z and then the decoder from an autoencoder model tries to reconstruct the synthetic data from these representations. The modified generator G_{Dec} can be realized by the following function

$$G_{Dec}(z) = Dec(G(z))$$

Equation 2

3.2. ImpartialGAN Model

ImpartialGAN has four major components one generator and three discriminators. Figure 3.2-1 shows the architecture of ImpartialGAN. The modified generator G_{Dec} produces fake samples which consist of i) unprotected attributes, x̂ ii) the decision attribute, ŷ, and iii) the protected attribute ŝ. These are generated from noise variable z and the real protected attribute s following the joint distribution for (x,y) given the conditional probability of s, where x represents the unprotected attributes and y the decision label.

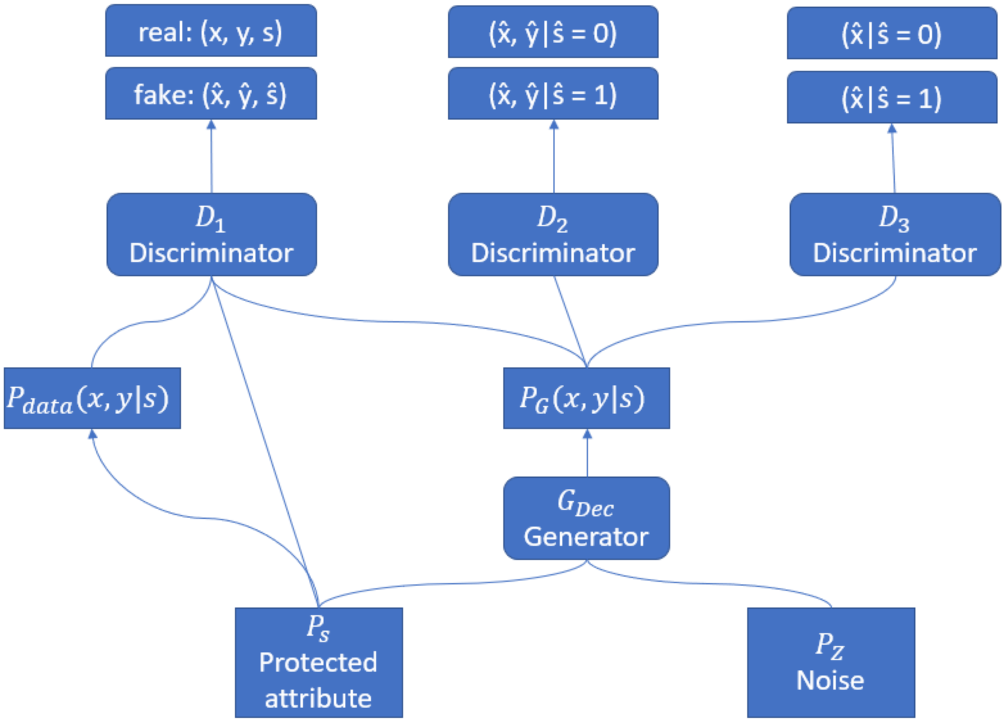
real: (x, y, s)
fake: $(\hat{x}, \hat{y}, \hat{s})$
$(\hat{x}, \hat{y} | \hat{s} = 0)$
$(\hat{x}, \hat{y} | \hat{s} = 1)$
$(\hat{x} | \hat{s} = 0)$
$(\hat{x} | \hat{s} = 1)$
D_1 Discriminator
D_2 Discriminator
D_3 Discriminator
$P_{data}(x, y|s)$
$P_G(x, y|s)$
G_{Dec} Generator
P_s Protected attribute
P_Z Noise

Figure 3.2-1: ImpartialGAN architecture

$$(\hat{x},\hat{y}) = G_{Dec}(z,s) = \mathrm{Dec}(G(z,s)),\ z\sim P_z$$

Equation 3

FairGAN Components of ImpartialGAN. In Equation 3, P_Z represents the noise distribution. Discriminator D_1 identifies fake samples $(\hat{x},\hat{y},\hat{s})$ from the real samples (x, y, s). This enforces the generator to align the fake samples more and more to the probability distribution of the real data given the protected attribute from random noise. Once the generated fake samples are marked as real, they are fed as input to the discriminator D_2 to enforce the fairness constraint.

Discriminator D_2 tries to find the value of the protected attribute given the unprotected attributes and the associated decision. It makes sure the unprotected attribute and the decision together does not encapsulate any information regarding the protected attribute value.

New Discriminator for ImpartialGAN. The third discriminator D_3 ensures that there is no correlation between the generated unprotected attributes and the protected attribute. In the following equations bold expressions indicate extensions provided by ImpartialGAN compared to FairGAN. The minmax game between the generator and the various discriminators can be described with the following equations:

$$\underset{G_{Dec}}{min}\ \underset{D_1 D_2 D_3}{max}\ V(G_{Dec}, D_1, D_2, D_3) = V_1(G_{Dec}, D_1) + \lambda_1 V_2(G_{Dec}, D_2) + \boldsymbol{\lambda_2 V_3(G_{Dec}, D_3)}$$

Equation 4

where,

$$V_1(G_{Dec}, D_1) = E_{s\sim P_{data}(s),(x,y)\sim P_{data}}(x, y|s)[logD_1(x, y, s)] +$$

$$E_{\hat{s}\sim P_G(s),(\hat{x},\hat{y})\sim P_G(x,y|s)}[\log{(1 - D_1(\hat{x}, \hat{y}, \hat{s}))}]$$

Equation 5

$$V_2(G_{Dec}, D_2) = E_{(\hat{x},\hat{y})\sim P_G}(x, y|s = 1)[logD_2(\hat{x}, \hat{y})] +$$

$$E_{(\hat{x},\hat{y})\sim P_G(x,y|s=0)}[\log{(1 - D_2(\hat{x}, \hat{y}))}]$$

Equation 6

$$\boldsymbol{V_3(G_{Dec}, D_3) = E_{(\hat{x})\sim P_G}(x|s = 1)[logD_3(\hat{x})] +}$$

$$\boldsymbol{E_{(\hat{x})\sim P_G(x|s=0)}[\log{(1 - D_3(\hat{x}))}]}$$

Equation 7

In Equation 4, λ_1 and λ_2 indicates the weightage whether more weight is given to fairness in joint combination of unprotected attribute along with the associated decision or to the fairness in the unprotected attributes.

As in FairGAN, using Equation 5, the generator first follows the probability distribution of the protected attribute (s) from real data. After that the generator uses the joint distribution of the pair (x, y) to generate a tuple (x̂, ŷ, ŝ) from random noise given the conditional distribution of the protected attribute (s). Once the generated tuple is close to real data and the discriminator D_1 marks them as real then using Equation 6 the discriminator D_2 is trained to predict the value of (ŝ) from the pair (x̂, ŷ) whereas the generator is trained to ensure that the P (x̂, ŷ|ŝ = 0) = P (x̂, ŷ|ŝ = 1) so that the discriminator D_2 is unable to predict the correct value of (ŝ) given a pair (x̂, ŷ). This training of D_2 and G_{Dec} ensures the generated unprotected attributes and the associated decision are not correlated with the protected attribute.

After achieving this, D_3 and G_{Dec} are trained using Equation 7 in ImpartialGAN. The unprotected attributes x̂ are given as input to D_3 which is trained to predict the value of (ŝ) while the generator ensures that the P (x̂|ŝ = 0) = P (x̂|ŝ = 1). This joint training of D_3 and G_{Dec} ensures data fairness in the unprotected attributes.

3.3. Fairness and discrimination metric

Ideally, statistical parity or fairness in a dataset should be represented as

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

where y is the decision and s is the protected attribute. The metric risk difference yields the amount of discrimination in the dataset and is expressed as follows:

$$\textit{riskDiff(Dataset)} = P(y = 1|s = 1) - P(y = 1|s = 0)$$

Statistical parity or fairness in a classifier can be determined by replacing the true label y with the prediction of the classifier as

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

where a classifier uses η(x) function to output decision ŷ. Here, x represents the attributes. While the previous formula considers risk difference defined in FairGAN, it does not consider the actual true decision. Hence, a classifier with low accuracy can reduce the risk difference easily. To address this issue, in our ImpartialGAN, the discrimination of a classifier, η, can be measured by the risk difference considering the actual true label as follows

$$\textit{riskDiff}(\eta) = P((\eta(x) = 1 \text{ and } y = 1)|s = 1) - P((\eta(x) = 1 \text{ and } y = 1)|s = 0).$$

3.4. Algorithm

As previously mentioned, we espoused the modified generator from FairGAN [7] which they adopted from medGAN [26] to produce discrete data. In order for the decoder to be able to reconstruct the data, we first trained the Autoencoder using the loss function in Equation (8).

$$\text{Loss} = ||Dec\big(Enc(x)\big) - x\ ||_2^2$$

Equation 8

where x represents the input features, Enc is the encoder, and Dec is the decoder in an Autoencoder setup.

Then we used this trained decoder along with the generator of a regular GAN [9] to create the generator for ImpartialGAN. The trained decoder produces synthetic data from the representations produced by G(z,s). Algorithm 1 shows how the various sub-modules of ImpartialGAN are trained. The Autoencoder is trained in lines 6 through 13. The discriminator D_1 and G_{Dec} is trained in lines 14 through 22 so that the synthetic data is as similar to real as possible. Then the discriminator D_2 and G_{Dec} are trained as in lines 23 through 31 to apply fair constraint on (x̂, ŷ) jointly. Lastly, the discriminator D_3 and G_{Dec} are trained to apply the fair constraint on (x̂) as shown from lines 32 through 40.

Algorithm 1 Pseudocode for the implementation of ImpartialGAN

```
i: the number of iterations
b: the number of train batches
v: the number of validation batches
$d_T$ : training dataset
$d_V$ : validation dataset
for i iterations do
    for b training batches do
        Train Autoencoder, AE on $d_T$ using Loss = $||Dec(Enc(x)) - x||_2^2$
    end for
    for v validation batches do
        Validate AE's performance on $d_V$
    end for
end for
for i iterations do
    for b training batches do
        Train Discriminator $D_1$ on a batch of real and synthetic data using loss
        function in Equation 5
        Train Generator $G_{Dec(z,s)}$ using loss function in Equation 5
    end for
    for v validation batches do
        Validate $D_1$ and $G_{Dec(z,s)}$ on $d_V$
    end for
end for
for i iterations do
    for b training batches do
        Train Discriminator $D_2$ on a batch of real and synthetic data using loss
        function in Equation 6
        Train Generator $G_{Dec(z,s)}$ using loss function in Equation 6
    end for
    for v validation batches do
        Validate $D_2$ and $G_{Dec(z,s)}$ on $d_V$
    end for
end for
for i iterations do
    for b training batches do
        Train Discriminator $D_3$ on a batch of real and synthetic data using loss
        function in Equation 7
        Train Generator $G_{Dec(z,s)}$ using loss function in Equation 7
    end for
    for v validation batches do
        Validate $D_3$ and $G_{Dec(z,s)}$ on $d_V$
    end for
end for
```

Chapter 4 Experiments

4.1. Experimental Setup

In this chapter, we evaluate the performance of ImpartialGAN on three datasets and compare with FairGAN's [7] performance. All the experiments were conducted on a system with Intel Core i7-8550U CPU @1.80GHz and 16 GB RAM. We briefly explain the various datasets and the experimental setup used.

Implementation Details. We implemented and tested ImpartialGAN by varying the values of the coefficients λ_1 and λ_2 to determine the best coefficient values that maintain a balance between utility and fairness. We adopted the same architecture for D_1, D_2, G_{Dec} as FairGAN [7] and extended it to implement D_3. The autoencoder consists of encoder and decoder each having one hidden layer with 128 neurons. We trained the autoencoder for 200 epochs. The generator and all the discriminators are feed forward neural networks with two hidden layers in each. Generator's each hidden layer has 128 dimensions. The first layer of discriminators has 256 dimensions, and the second layer has 128 dimensions. First D_1 and G_{Dec} are trained for 2,000 epochs. Next, we trained $D_2$and G_{Dec} for 2,000 epochs. Lastly, we trained D_3 and G_{Dec} for 2,000 epochs.

Datasets. We conducted our experiments on three datasets, which are UCI Adult Income Dataset, German Credit Dataset and COMPAS Dataset.

4.2. Classification Models and Settings

After generating the synthetic data using different values of λ_1 and λ_2, we trained three different classifiers to check the utility of the generated data along with the risk difference of the various classifiers: i) linear Support Vector Machine(SVM), ii) Support Vector Machine with Radial Basis Function kernel(RBF), and iii) Decision trees and used grid search to find optimal hyperparameter values for SVM(RBF).

We used two different configurations to evaluate the performance of the classifiers. 1) (SyntoSyn): We trained the classifiers on the synthetic dataset and evaluated them on the synthetic dataset. 2) (SyntoReal): We trained the classifiers on the synthetic dataset and evaluated them on the real dataset.

4.3. Evaluation on Adult Dataset.

In our experiments, we used the preprocessed datafile obtained from [7]. Xu et al. used the UCI Adult Dataset [27] which contains 48,842 instances. After removing the instances with unknown values, the dataset size reduces to 45,222. The instances in the original dataset have 14 attributes and the binary decision attribute reflects if the income is less than 50,000 or greater than 50,000. Xu et al. preprocessed this dataset by converting each attribute to one hot encoded form and then combining the one hot encoded form of each attribute to create a dataset that resulted in a total of 58 attributes for each instance. As in [7], in our experiments, we have considered only one protected attribute which is the gender of the individual whose values were either male or female. The decision attribute income was also binary whose output was either a positive outcome or a negative outcome.

	Real Data	FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	0.1989	0.0562 ± 0.0190	0.0966 ± 0.0134	0.0957 ± 0.0036	0.0419 ± 0.0124	0.0265 ± 0.0118	0.0222 ± 0.0082

Table 4.3-1: Adult dataset: Risk difference in real and synthetic datasets

Risk Difference in Real and Generated Data. We compare the risk difference between FairGAN and ImpartialGAN while varying the parameters λ_1 and λ_2 using

riskDiff (Dataset) = P (y = 1|s = 1) − P (y = 1|s = 0) described in the previous chapter. The risk difference for the real and synthetic datasets are shown in Table 4.3-1. The risk difference in the real data is .1989 which shows that the protected attribute information is present in the output label, and there is discrimination against females. The risk difference for FairGAN is 0.0562 which shows fair data generation but there is still correlation between the unprotected attributes and the protected attribute. For (λ_1 = 0, λ_2 = 1) and (λ_1 = 0, λ_2 = 2) the risk difference is lower than real dataset but higher than FairGAN as there is still correlation between ŷ and s in the generated data. But as we increase the value of λ_2 keeping λ_1 value constant at 1, the risk difference drops as now the correlation between (x̂, ŷ) and s is minimized as well as the correlation between x̂ and s.

4.3.1 Performance on Adult Dataset

For the SVM classifier with linear kernel, the regularization parameter C value is set as 1.0. For SVM with RBF kernel C value is set as 1 along with the kernel coefficient ϒ as .001. Lastly, for decision trees we used the maximum depth of the tree as 5.

Table 4.3.1-1 presents the risk difference and accuracy for classifiers in RealtoReal setting when classifiers are trained and evaluated on the real dataset. We consider these results as baseline for comparison purposes. While the accuracy is high for the classifiers in RealtoReal setting so is the risk difference. This proves the real dataset is biased, and hence the classifiers trained on it are likely to be biased as well. We also believe the most important experimental setting is SyntoReal same as emphasized by Xu et al. [7]. For practical purposes we can only train the classifiers on synthetic data and then can use these trained classifiers for unbiased prediction on real datasets.

Classifier	Risk Difference	Accuracy
SVM(Linear)	0.1295	0.8425
SVM(RBF)	0.1022	0.8307
Decision Tree	0.1212	0.8234

Table 4.3.1-1: Adult Dataset: Classifier risk difference and accuracy for RealtoReal setting

For training and evaluating these classifiers, we only used the unprotected attributes without the protected attribute gender for predicting the income. We also used the classifier (η) risk difference, *riskDiff(η)* = P ((η(x) = 1 and y = 1)|s = 1) − P ((η(x) = 1 and y = 1)|s = 0) as explained in the previous chapter to see the fairness of classifiers in predicting the output label. Table 4.3.1-2 shows the accuracy and risk difference results for the classifiers in SyntoReal setting. For SVM(Linear) and SVM(RBF), the accuracies are slightly better whereas for decision trees the accuracy decreased and then increased while increasing λ_2. For all the three classifiers the risk difference increased when compared to FairGAN but was still lower than the difference obtained in RealtoReal setting. For all the classifiers the change in accuracy proves that ImpartialGAN maintains good data utility in SyntoReal setting. Table 4.3.1-3 shows the results we obtained for risk difference and accuracy in classifiers when the classifiers are both trained and tested on synthetic data. For the risk difference in classifiers for SyntoSyn setting we observed that if discriminator D_2 is not used (λ_1 = 0), the risk difference for the classifiers increased compared to FairGAN. But if all three discriminators of ImpartialGAN are used (λ_1 > 0, λ_2 > 0), the risk difference dropped significantly. The risk difference shows there is no correlation between the protected attribute (s) and (x̂, ŷ). Our classification accuracy increased in most of the cases compared to the FairGAN. The accuracy increased for SVM(Linear) compared to FairGAN. However, the accuracy slightly dropped for the SVM(RBF) and Decision trees. This can be attributed to the drop in the risk difference for both of these models. This proves ImpartialGAN is capable of generating fair data while maintaining a good data utility in SyntoSyn setting as well.

It is noteworthy to mention that in SyntoSyn setting the risk difference of the classifiers dropped drastically compared to the risk differences of the classifiers in RealtoReal setting. On the other hand, the accuracies for both SyntoSyn and SyntoReal reduced slightly when compared to RealtoReal setting. The slight difference in accuracy with significant reduction in risk difference emphasizes that the synthetic data has a good data utility.

	Classifier	SyntoReal					
		FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	SVM(Linear)	0.0949 ± 0.0172	0.1205 ± 0.0062	0.1183 ± 0.0071	0.1079 ± 0.0015	0.1066 ± 0.0176	0.1126 ± 0.0219
	SVM(RBF)	0.0667 ± 0.0258	0.0872 ± 0.0083	0.0881 ± 0.0148	0.0860 ± 0.0008	0.0855 ± 0.0081	0.0829 ± 0.0324
	Decision Trees	0.0453 ± 0.0694	0.1032 ± 0.0187	0.0830 ± 0.1253	0.0795 ± 0.0351	0.0628 ± 0.0514	0.1030 ± 0.0097
Accuracy	SVM(Linear)	0.8311 ± 0.0064	0.8372 ± 0.0021	0.8341 ± 0.0019	0.8331 ± 0.0021	0.8323 ± 0.0044	0.8343 ± 0.0036
	SVM(RBF)	0.8194 ± 0.0115	0.8233 ± 0.0038	0.8234 ± 0.0146	0.8265 ± 0.0032	0.8260 ± 0.0016	0.8217 ± 0.0102
	Decision Trees	0.7979 ± 0.0206	0.8094 ± 0.0126	0.7562 ± 0.0298	0.7986 ± 0.0107	0.7884 ± 0.0286	0.8074 ± 0.0059

Table 4.3.1-2: Adult dataset: Classifier risk difference and accuracy for SyntoReal Dataset

	Classifier	SyntoSyn					
		FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	SVM(Linear)	0.0177 ± 0.0288	0.0485 ± 0.0070	0.0337 ± 0.0021	0.0166 ± 0.0077	0.0079 ± 0.0143	0.0063 ± 0.0112
	SVM(RBF)	0.0051 ± 0.0258	0.0146 ± 0.0064	0.0158 ± 0.0086	0.0055 ± 0.0062	0.0037 ± 0.0115	0.0033 ± 0.0115
	Decision Trees	0.0390 ± 0.0205	0.0637 ± 0.0081	0.0637 ± 0.0058	0.0221 ± 0.0128	0.0259 ± 0.0208	0.0142 ± 0.0175
Accuracy	SVM(Linear)	0.8271 ± 0.0115	0.8319 ± 0.0130	0.8257 ± 0.0051	0.8306 ± 0.0094	0.8292 ± 0.0011	0.8309 ± 0.0179
	SVM(RBF)	0.8096 ± 0.0143	0.8022 ± 0.0246	0.7998 ± 0.0177	0.8080 ± 0.0146	0.8082 ± 0.0058	0.8069 ± 0.0287
	Decision Trees	0.8251 ± 0.0089	0.8296 ± 0.0116	0.8251 ± 0.0047	0.8166 ± 0.0123	0.8236 ± 0.0102	0.8197 ± 0.0201

Table 4.3.1-3: Adult dataset: Classifier risk difference and accuracy for SyntoSyn setting

4.3.2 Discussion about Adult Dataset

Measuring risk difference in a meaningful way is challenging. Xu et al. [7] define risk difference as , *riskDiff(η)* = P (η(x) = 1|s = 1) − P (η(x) = 1|s = 0). This formula focuses on the prediction of classifier ignoring the correct class, and hence, if a classifier mispredicts it can lower the risk difference. In risk difference assessment, rather than just using the prediction, the original label should also be used. Then the risk difference would be stated as

riskDiff(η) = P ((η(x) = 1 and y = 1)|s = 1) − P ((η(x) = 1 and y = 1)|s = 0). This would mean that it is more critical to decrease risk difference on correctly predicted data. However, in this case, the risk difference will be similar to the risk difference in the dataset.

The Pearson coefficient for the protected attribute is shown in Table 4.3.2-1 and shows some degree of correlation with the attributes relationship and hours worked per week.

Attribute Names	Pearson Coefficient for protected attribute
Age	0.0888
Work Class	0.0959
Education-num	0.0122
Marital status	-0.1293
Occupation	0.0803
Relationship	**-0.2734**
Race	-0.0678
Sex	1
Capital-gain	0.0484
Capital-loss	0.0455
Hours per week	**0.2293**
Native Country	-0.0081
Decision	0.2159

Table 4.3.2-1: Adult Dataset: Pearson coefficient

4.4 Evaluation on German Credit Dataset

For our experiments, we used all numeric datafile produced by Strathclyde University. The data file consists of 1000 instances and 25 attributes including the decision attribute. The binary decision attribute reflects the credit risk associated with the customer(whether a person is a good credit risk or a bad credit risk). We preprocessed this dataset by converting each attribute to one hot encoded form and then combining the one hot encoded form of each attribute to create a dataset that resulted in a total of 68 attributes. In this dataset we have one protected attribute which is the gender of the individual whose values were either male or female. The decision attribute credit risk is also binary whose output was either a positive outcome or a negative outcome.

	Real Data	FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	0.0748	0.0460 ± 0.0322	0.0366 ± 0.0453	0.0072 ± 0.0262	0.0050 ± 0.0378	0.0088 ± 0.0801	0.0342 ± 0.0327

Table 4.4-1: German Credit dataset: Risk difference in real and synthetic datasets

Risk Difference in Real and Generated Data. We compare the risk difference in real data with datasets generated by FairGAN and ImpartialGAN by varying the parameters λ_1 and λ_2. Risk difference was calculated using the formula *riskDiff (Dataset)* = P (y = 1|s = 1) − P (y = 1|s = 0) described previously. The risk difference for the real and synthetic datasets are shown in Table 4.4-1. The risk difference in the real data is 0.0748 and the risk difference for data generated from FairGAN is 0.0460 which shows fair data generation but there is still correlation between the unprotected attributes and the protected attribute. For (λ_1 = 0, λ_2 = 1) and (λ_1 = 0, λ_2 = 2) the risk difference further decreases as compared to real data and FairGAN both. But as we increase the value of λ_2 keeping λ_1 value constant at 1, the risk difference drops and then increases slightly (still the risk difference is lower than both real data and FairGAN).

4.4.1 Performance on German Credit Dataset

For the SVM classifier with linear kernel, the regularization parameter C value is set as 1.0. For SVM with RBF kernel C value is set as 10 along with the kernel coefficient ϒ as .01. Lastly, for decision trees we used the maximum depth of the tree as 5.

Table 4.4.1-1 presents the risk difference and accuracy for classifiers in RealtoReal setting where classifiers are trained and evaluated on the real dataset. These results are considered as baseline for comparison purposes. While the accuracy is not very high for the classifiers in RealtoReal setting but the risk difference is still high for this small dataset. The real dataset is biased, and hence the classifiers trained on it are likely to be biased as well.

Classifier	Risk Difference	Accuracy
SVM(Linear)	0.0950	0.726
SVM(RBF)	0.0743	0.734
Decision Tree	0.1206	0.702

Table 4.4.1-1: German Credit dataset: Classifier risk difference and accuracy for RealtoReal setting

For training and evaluating these classifiers, we only used the unprotected attributes without the protected attribute gender for predicting the credit risk. We also used the classifier (η) risk difference, *riskDiff(η)* = P ((η(x) = 1 and y = 1)|s = 1) − P ((η(x) = 1 and y = 1)|s = 0) as explained in the previous chapter to see the fairness of classifiers in predicting the output label.

Table 4.4.1-2 shows the risk difference and accuracy results for the classifiers in SyntoReal setting. For the risk difference in classifiers for SVM(Linear) and SVM(RBF) we observed that if discriminator D_2 is not used (λ_1 = 0), the risk difference for the classifiers first increased compared to FairGAN and then decreased for λ_2=2. But if all three discriminators of ImpartialGAN are used (λ_1 > 0, λ_2 > 0), the risk difference dropped significantly for λ_1=1 , λ_2=3. For Decision Trees the risk difference was lowest for FairGAN. The risk difference for ImpartialGAN was higher than FairGAN but was still lower than the risk difference in RealtoReal setting and kept on dropping with increasing values of λ_1 and λ_2. For SVM(Linear) and SVM(RBF), the accuracies are highest for λ_1=1 , λ_2=2 whereas for decision trees the accuracy decreased with increasing values of λ_1 and λ_2. However, the accuracies were still better than FairGAN. For all the classifier's the change in accuracy proves that ImpartialGAN maintains good data utility in SyntoReal setting.

Table 4.4.1-3 shows the results we obtained for risk difference and accuracy in classifiers when the classifiers are both trained and tested on synthetic data. For the risk difference in classifiers for SyntoSyn setting we observed that if discriminator D_2 is not used (λ_1 = 0), the risk difference for the classifiers decreased compared to FairGAN and was lowest for all the three classifiers. But if all three discriminators of ImpartialGAN are used (λ_1 > 0, λ_2 > 0), the risk difference started increasing but was still lower than FairGAN. The risk difference shows there is no correlation between the protected attribute (s) and (x̂, ŷ). Our classification accuracy increased for all the three classifiers compared to the FairGAN. This proves ImpartialGAN is capable of generating fair data while maintaining a good data utility in SyntoSyn setting.

It is noteworthy to mention that in both SyntoSyn and SyntoReal setting the risk difference of the classifiers dropped drastically compared to the risk differences of the classifiers in RealtoReal setting. On the other hand, the accuracies for classifiers increased significantly for SyntoSyn setting when compared to RealtoReal setting. The accuracies increased for SVM(Linear) and SVM(RBF) but dropped significantly for Decision Trees in SyntoReal setting when compared to RealtoReal setting. While the accuracy dropped for Decision Trees it can be attributed to the significant drop in risk difference. These results prove that ImpartialGAN maintains good data utility in both SyntoReal and SyntoSyn settings.

	Classifier	SyntoReal					
		FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	SVM(Linear)	0.0779 ± 0.0161	0.0787 ± 0.0214	0.0712 ± 0.034	0.0698 ± 0.0247	0.0717 ± 0.03	0.0557 ± 0.0093
	SVM(RBF)	0.0839 ± 0.0245	0.0862 ± 0.0305	0.0731 ± 0.0172	0.0754 ± 0.021	0.0711 ± 0.0533	0.0570 ± 0.0085
	Decision Trees	0.0295 ± 0.0237	0.0774 ± 0.0345	0.0593 ± 0.0502	0.0448 ± 0.0402	0.0468 ± 0.0253	0.0464 ± 0.0213
Accuracy	SVM(Linear)	0.7582 ± 0.0018	0.7518 ±0.0112	0.7424 ±0.0096	0.7520 ±0.0100	0.7610 ± 0.0070	0.7484 ±0.0156
	SVM(RBF)	0.7546 ± 0.0044	0.7520 ± 0.0090	0.7456 ± 0.0154	0.7564 ± 0.0206	0.7588 ± 0.0092	0.7504 ± 0.0236
	Decision Trees	0.6468 ± 0.0752	0.6980 ± 0.0320	0.6632 ± 0.0408	0.6720 ± 0.0400	0.6656 ± 0.0364	0.6640 ± 0.0320

Table 4.4.1-2: German Credit dataset: Classifier risk difference and accuracy for SyntoReal setting

	Classifier	SyntoSyn					
		FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	SVM(Linear)	0.0763 ± 0.0550	0.0154 ± 0.0535	-0.0027 ± 0.0836	0.0081 ± 0.0459	0.0282 ± 0.0615	0.0481 ± 0.0713
	SVM(RBF)	0.0777 ± 0.0594	0.0229 ± 0.0632	-0.0039 ± 0.0700	0.0080 ± 0.0405	0.0257 ± 0.0763	0.0378 ± 0.0695
	Decision Trees	0.0637 ± 0.0343	0.0398 ± 0.0612	0.0065 ± 0.0314	-0.0064 ± 0.0344	0.0285 ± 0.0799	0.0410 ± 0.0600
Accuracy	SVM(Linear)	0.7828 ± 0.0352	0.8092 ± 0.0328	0.8160 ± 0.0120	0.7748 ± 0.0412	0.8236 ± 0.0304	0.8208 ± 0.0252
	SVM(RBF)	0.7852 ± 0.0388	0.8088 ± 0.0352	0.8124 ± 0.0176	0.7884 ± 0.0276	0.8296 ± 0.0384	0.8292 ± 0.0308
	Decision Trees	0.7312 ± 0.0408	0.7512 ± 0.0408	0.7512 ± 0.0328	0.7040 ± 0.0300	0.7428 ± 0.0272	0.7620 ± 0.0420

Table 4.4.1-3: German Credit dataset: Classifier risk difference and accuracy for SyntoSyn setting

4.4.2 Discussion about German Credit Dataset

As previously mentioned in the discussion about UCI Adult dataset we used a modified formula for calculating the classifier risk difference when compared to FairGAN [7] which has an impact on the results. Also, this dataset is very small containing only 1000 instances which further impacts the results. Table 4.4.2-1 shows the Pearson Correlation coefficient for the attribute gender. The table shows for this dataset gender has high correlation with a person's status which can be single, widowed, married, divorced etc.

Attribute Names	Pearson Coefficient for protected attribute
Balance Checking account	0.0256
Loan Months	0.0745
Credit History	0.0718
Credit Amount	0.1082
Savings Balance	0.0350
Months Employed	0.1970
Person Status	**0.7380**
Person Residence	-0.0138
Property	0.0515
Age	0.2225
Other Installment Plans	-0.0330
Number of Existing credits at this Bank	0.0943
Number of people being liable to provide maintenance for	0.2034
Telephone	0.0760
Foreign Worker	0.0512
Purpose Car New	0.0130
Purpose Car Used	0.0564
Other debtors / guarantors – None	-0.0136
Other debtors / guarantors – co-applicant	0.0077
House rent vs Free	-0.2228
House owns vs Free	0.1196
Job unemployed vs Management	-0.0764
Job unskilled vs Management	-0.0108
Job skilled vs Management	-0.0076
Person Sex	1.0000

Table 4.4.2-1: German Credit dataset: Pearson coefficient

4.5 Evaluation on COMPAS Dataset

For our experiments, we used the COMPAS dataset published by ProPublica. The data file consists of 7214 instances and 53 attributes including the decision attribute. After dropping the attributes that did not have any impact on the decision attribute like the name, middle name etc., we were left with 13 attributes per instance. The binary decision attribute is recidivism which tells whether a person will reoffend or not. We preprocessed this dataset by converting each attribute to one hot encoded form and then combining the one hot encoded form of each attribute to create a dataset that resulted in a total of 31 attributes for each instance including the decision attribute. In this dataset we have one protected attribute which is the race of the individual whose values were either African American or others. The decision attribute recidivism is also binary whose output was either the person recidivated or not.

	Real Data	FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	0.1305	0.0034 ± 0.0207	0.0018 ± 0.0223	-0.0130 ±0.0161	0.0073 ± 0.0152	-0.0051 ± 0.0267	-0.0131 ± 0.02

Table 4.5-1: COMPAS dataset: Risk difference in real and synthetic datasets

Risk Difference in Real and Generated Data. Here we compare the risk difference in real data with datasets generated by FairGAN and ImpartialGAN again by varying the parameters λ_1 and λ_2. As mentioned in this chapter previously Risk difference was calculated using the formula *riskDiff (Dataset)* = P (y = 1|s = 1) − P (y = 1|s = 0). The risk difference for the real and synthetic datasets are shown in Table 4.5-1. The risk difference in the real data is 0.1305 which is high and the risk difference for data generated from FairGAN is 0.0034 which shows fair data generation but there is still correlation between the unprotected attributes and the protected attribute. For (λ_1 = 0, λ_2 = 1) and (λ_1 = 0, λ_2 = 2) the risk difference further decreases as compared to real data and FairGAN both. As we increase the value of λ_2 keeping λ_1 value constant at 1, the risk difference drops further and moves in the negative direction. Ideally risk difference should be closer to zero. Values further away from zero in either direction be it positive or negative are not ideal.

4.5.1 Performance on COMPAS dataset

For the SVM classifier with linear kernel, the regularization parameter C value is set as 1.0. For SVM with RBF kernel C value is set as 100 along with the kernel coefficient ϒ as .01. Lastly, for decision trees we used the maximum depth of the tree as 5.

Table 4.5.1-1 presents the risk difference and accuracy for classifiers in RealtoReal. We consider these results as baseline for comparison purposes in other settings. The accuracy is extremely high for the classifiers in RealtoReal setting and so is the risk difference. This proves the real dataset is biased, and hence the classifiers trained on it are likely to be biased as well.

Classifier	Risk Difference	Accuracy
SVM(Linear)	0.1306	0.9706
SVM(RBF)	0.1306	0.9706
Decision Tree	0.1323	0.9695

Table 4.5.1-1: COMPAS dataset: Classifier risk difference and accuracy for RealtoReal Dataset

For training and evaluating these classifiers, we only used the unprotected attributes without the protected attribute race for predicting the decision attribute recidivism. As previously mentioned, we used the classifier (η) risk difference,

riskDiff(η) = P ((η(x) = 1 and y = 1)|s = 1) − P ((η(x) = 1 and y = 1)|s = 0).

Table 4.5.1-2 shows the risk difference and accuracy results for the classifiers in SyntoReal setting. For the risk difference in classifiers for SVM(Linear) we observed that if discriminator D_2 is not used (λ_1 = 0), the risk difference for the classifiers deceased only slightly compared to FairGAN. But if all three discriminators of ImpartialGAN are used (λ_1 > 0, λ_2> 0), the risk difference dropped slightly for λ_1=1 , λ_2=3. In general for SVM(RBF) we got slightly better results than SVM(Linear). For Decision Trees the risk difference was lowest for the setting λ_1 =1, λ_2 = 1. For all the classifier's the accuracies were roughly the same for both FairGAN and ImpartialGAN.

Table 4.5.1-3 shows the results we obtained for risk difference and accuracy in classifiers when the classifiers are both trained and tested on synthetic data. For the risk difference in classifiers for SyntoSyn setting we observed that if discriminator D_2 is not used (λ_1 = 0), the risk difference for the classifier SVM(Linear) decreased significantly compared to FairGAN. The risk difference for all the classifiers was lowest when we only used discriminator D_3. But if all three discriminators of ImpartialGAN are used (λ_1 > 0, λ_2 > 0), the risk difference started decreasing and started moving in the negative direction (below zero). For all the classifier's the accuracy was highest when we only used discriminator D_3.

In SyntoReal setting the risk difference and accuracy of the classifiers dropped slightly compared to the risk differences of the classifiers in RealtoReal setting. On the other hand, the risk difference dropped significantly in SyntoSyn setting while the accuracies remained close to the accuracy achieved in RealtoReal setting.

	Classifier	SyntoReal					
		FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	SVM(Linear)	0.1306 ± 0.0000	0.1306 ± 0.0000	0.1296 ± 0.0010	0.1306 ± 0.0000	0.1306 ± 0.0000	0.1298 ± 0.0008
	SVM(RBF)	0.1306 ± 0.0000	0.1305 ± 0.0001	0.1289 ± 0.0017	0.1296 ± 0.0010	0.1306 ± 0.0000	0.1298 ± 0.0008
	Decision Trees	0.1294 ± 0.0012	0.1286 ± 0.0020	0.1280 ± 0.0026	0.1267 ± 0.0036	0.1318 ± 0.0097	0.1277 ± 0.0029
Accuracy	SVM(Linear)	0.9695 ± 0.0000	0.9695 ± 0.0000	0.9686 ± 0.0009	0.9695 ± 0.0000	0.9695 ± 0.0000	0.9683 ± 0.0012
	SVM(RBF)	0.9695 ± 0.0000	0.9695 ± 0.0000	0.9681 ± 0.0014	0.9686 ± 0.0009	0.9695 ± 0.0000	0.9683 ± 0.0012
	Decision Trees	0.9682 ± 0.0013	0.9680 ± 0.0015	0.9672 ± 0.0023	0.9661 ± 0.0033	0.9597 ± 0.0098	0.9663 ± 0.0032

Table 4.5.1-2: COMPAS dataset: Classifier risk difference and accuracy for SyntoReal setting

	Classifier	SyntoSyn					
		FairGAN	ImpartialGAN				
		λ_1=1	λ_1=0	λ_1=0	λ_1=1	λ_1=1	λ_1=1
		λ_2=0	λ_2=1	λ_2=2	λ_2=1	λ_2=2	λ_2=3
Risk Difference	SVM(Linear)	-0.0089 ± 0.0238	-0.0026 ± 0.0124	-0.0061 ± 0.0293	-0.0113 ± 0.0185	-0.0044 ± 0.0262	-0.0246 ± 0.0127
	SVM(RBF)	-0.0092 ± 0.0241	0.0025 ± 0.0128	-0.0064 ± 0.0296	-0.0114 ± 0.0186	-0.0039 ± 0.0257	-0.0245 ± 0.0121
	Decision Trees	-0.0096 ± 0.0229	-0.0026 ± 0.0119	-0.0081 ± 0.0323	-0.0109 ± 0.0180	-0.0033 ± 0.0246	-0.0252 ± 0.0136
Accuracy	SVM(Linear)	0.9676 ± 0.0046	0.9626 ± 0.0030	0.9690 ± 0.0038	0.9661 ± 0.0070	0.9687 ± 0.0069	0.9664 ± 0.0050
	SVM(RBF)	0.9678 ± 0.0050	0.9627 ± 0.0023	0.9693 ± 0.0032	0.9668 ± 0.0063	0.9686 ± 0.0075	0.9668 ± 0.0051
	Decision Trees	0.9727 ± 0.0054	0.9707 ± 0.0068	0.9739 ± 0.0047	0.9752 ± 0.0051	0.9737 ± 0.0044	0.9724 ± 0.0048

Table 4.5.1-3: COMPAS dataset: Classifier risk difference and accuracy for SyntoSyn setting

4.5.2 Discussion on COMPAS dataset

As previously mentioned in the discussion about UCI Adult dataset we used a modified formula for calculating the classifier risk difference when compared to FairGAN [7] which has an impact on the results. Also, this dataset is small containing only 7214 instances which further impacts the results. The results show that the discriminator D_2 and D_3 may be affecting each other due to the size of the dataset. Table 4.5.2-1 shows the Pearson Correlation coefficient for the attribute race. The table shows for this dataset race is correlated to maybe age, juvenile misdemeanor count, priors count, and the flag is recid. However, these correlations are very low, and this can be one of the reasons for the algorithm cannot improve the removal of the bias further. In other words, D_3 may not lower risk difference further considering low correlation between other attributes and the protected in addition to the high accuracy of classifiers.

Attribute Names	Pearson Coefficient for protected attribute
Sex	-0.0229
Age	**0.1339**
Race	1.0000
Juvenile Felony Count	-0.0914
Juvenile Misdemeanor Count	**-0.1010**
Juvenile Other Count	-0.0727
Priors Count	**-0.1889**
Charge Degree Type Count	0.0756
is_recid flag	**-0.1335**
is_violent_recid flag	-0.0548
Score Text Type Category	0.0413
Violent Score Text Type Category	-0.0289

Table 4.5.2-1: COMPAS dataset: Pearson coefficient

5 Conclusion and Future Work

In this thesis, we proposed ImpartialGAN that addresses the correlation between unprotected and protected attributes compared to FairGAN [7]. ImpartialGAN consists of one generator and three discriminators. The generator produces fake data from noise conditioned on the protected attribute given the joint distribution of (unprotected attributes, decision). While the first discriminator ensures the fake data is as similar to real data, the remaining two discriminators ensure the data is fair and free from bias towards the protected group. The experimental results on UCI Adult, German Credit and COMPAS datasets show the effectiveness of ImpartialGAN in generating fair data while maintaining the data utility in both SyntoSyn and SyntoReal settings. We have used gender as the protected attribute in the UCI Adult and German Credit datasets. For the COMPAS dataset, we have tested our approach on the race attribute.

5.1 Future Work

The model ImpartialGAN that we proposed in the thesis can be extended as follows:

1. Currently the model is trained to work on protected attribute whose value is binary. It can be extended to support protected attributes whose values are non-binary.
2. With the ever-changing nature of data, it is possible that there are multiple protected attributes present in the dataset. Currently the model takes into consideration only one protected attribute, but it can be extended to accept multiple protected attributes (e.g., gender and race together).
3. Currently the Pearson coefficient is calculated on the real data, but we can calculate the Pearson coefficient on the preprocessed dataset (one hot encoded form) to ascertain how the correlation between the protected attribute and the other attributes changes.

4. As of now the training of the discriminators is done in the order D_1 first followed by D_2 and then the last discriminator, D_3, is trained with the generator G_{Dec}. But we can change the order and train the discriminator D_3 before D_2 and see how this affects the risk difference and accuracy.
5. Lastly, we can run more experiments by giving more weightage to λ_1 keeping λ_2 constant at a value to see an in-depth comparison of the discriminators D_2 and D_3 .

References

[1] T. X. Tran, M. L. Pusey, and R. S. Aygun, “Else-Tree Classifier for Minimizing Misclassification of Biological Data,” in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec. 2018, pp. 2301–2308. doi: 10.1109/BIBM.2018.8621322.

[2] T. X. Tran and R. S. Aygun, “WisdomNet: trustable machine learning toward error-free classification,” *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2719–2734, Apr. 2021, doi: 10.1007/s00521-020-05147-4.

[3] K. Makhlouf, S. Zhioua, and C. Palamidessi, “On the Applicability of Machine Learning Fairness Notions,” *ACM SIGKDD Explor. Newsl.*, vol. 23, no. 1, pp. 14–23, May 2021, doi: 10.1145/3468507.3468511.

[4] S. Sayenju *et al.*, “Directional Pairwise Class Confusion Bias and Its Mitigation,” in *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, Jan. 2022, pp. 67–74. doi: 10.1109/ICSC52841.2022.00017.

[5] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A Survey on Bias and Fairness in Machine Learning,” *ACM Comput. Surv.*, vol. 54, no. 6, p. 115:1-115:35, Jul. 2021, doi: 10.1145/3457607.

[6] L. Zhang, Y. Wu, and X. Wu, “A Causal Framework for Discovering and Removing Direct and Indirect Discrimination,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, Aug. 2017, pp. 3929–3935. doi: 10.24963/ijcai.2017/549.

[7] D. Xu, S. Yuan, L. Zhang, and X. Wu, “FairGAN: Fairness-aware Generative Adversarial Networks,” in *2018 IEEE International Conference on Big Data (Big Data)*, Dec. 2018, pp. 570–575. doi: 10.1109/BigData.2018.8622525.

[8] P. Sattigeri, S. C. Hoffman, V. Chenthamarakshan, and K. R. Varshney, “Fairness GAN.” arXiv, May 24, 2018. doi: 10.48550/arXiv.1805.09910.

[9] I. Goodfellow *et al.*, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems*, 2014, vol. 27. Accessed: Oct. 23, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

[10] L. Zhang, Y. Wu, and X. Wu, “Achieving Non-Discrimination in Prediction,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, Jul. 2018, pp. 3097–3103. doi: 10.24963/ijcai.2018/430.

[11] F. Kamiran and T. Calders, “Classifying without discriminating,” in *Control and Communication 2009 2nd International Conference on Computer*, Feb. 2009, pp. 1–6. doi: 10.1109/IC4.2009.4909197.

[12] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, “Certifying and Removing Disparate Impact,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Aug. 2015, pp. 259–268. doi: 10.1145/2783258.2783311.

[13] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, and K. R. Varshney, "Optimized Pre-Processing for Discrimination Prevention," in *Advances in Neural Information Processing Systems*, 2017, vol. 30. Accessed: Oct. 23, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/9a49a25d845a483fae4be7e341368e36-Abstract.html

[14] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowl. Inf. Syst.*, vol. 33, no. 1, pp. 1–33, Oct. 2012, doi: 10.1007/s10115-011-0463-8.

[15] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi, "Fairness Constraints: Mechanisms for Fair Classification," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Apr. 2017, pp. 962–970. Accessed: Oct. 23, 2022. [Online]. Available: https://proceedings.mlr.press/v54/zafar17a.html

[16] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-Aware Classifier with Prejudice Remover Regularizer," in *Machine Learning and Knowledge Discovery in Databases*, Berlin, Heidelberg, 2012, pp. 35–50. doi: 10.1007/978-3-642-33486-3_3.

[17] F. Kamiran, T. Calders, and M. Pechenizkiy, "Discrimination Aware Decision Tree Learning," in *2010 IEEE International Conference on Data Mining*, Dec. 2010, pp. 869–874. doi: 10.1109/ICDM.2010.50.

[18] M. Hardt, E. Price, E. Price, and N. Srebro, "Equality of Opportunity in Supervised Learning," in *Advances in Neural Information Processing Systems*, 2016, vol. 29. Accessed: Oct. 23, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html

[19] L. Zhang, Y. Wu, and X. Wu, "Achieving Non-Discrimination in Data Release," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Aug. 2017, pp. 1335–1344. doi: 10.1145/3097983.3098167.

[20] V. V. Ramaswamy, S. S. Y. Kim, and O. Russakovsky, "Fair Attribute Classification Through Latent Space De-Biasing," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9301–9310. Accessed: Oct. 23, 2022. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Ramaswamy_Fair_Attribute_Classification_Through_Latent_Space_De-Biasing_CVPR_2021_paper.html

[21] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating Unwanted Biases with Adversarial Learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, New York, NY, USA, Dec. 2018, pp. 335–340. doi: 10.1145/3278721.3278779.

[22] A. Amini, A. P. Soleimany, W. Schwarting, S. N. Bhatia, and D. Rus, "Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, New York, NY, USA, Jan. 2019, pp. 289–295. doi: 10.1145/3306618.3314243.

[23] E. Creager *et al.*, "Flexibly Fair Representation Learning by Disentanglement," in *Proceedings of the 36th International Conference on Machine Learning*, May 2019, pp. 1436–1445. Accessed: Oct. 23, 2022. [Online]. Available: https://proceedings.mlr.press/v97/creager19a.html

[24] "Why Amazon's Automated Hiring Tool Discriminated Against Women | News & Commentary," *American Civil Liberties Union*, Oct. 12, 2018. https://www.aclu.org/news/womens-rights/why-amazons-automated-hiring-tool-discriminated-against (accessed Dec. 07, 2022).

[25] S. Serengil, "Convolutional Autoencoder: Clustering Images with Neural Networks," *Sefik Ilkin Serengil*, Mar. 23, 2018. https://sefiks.com/2018/03/23/convolutional-autoencoder-clustering-images-with-neural-networks/ (accessed Dec. 07, 2022).

[26] 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 *Proceedings of the 2nd Machine Learning for Healthcare Conference*, Nov. 2017, pp. 286–305. Accessed: Oct. 23, 2022. [Online]. Available: https://proceedings.mlr.press/v68/choi17a.html

[27] "UCI Machine Learning Repository: Adult Data Set." https://archive.ics.uci.edu/ml/datasets/adult (accessed Oct. 23, 2022).