Université de Montréal

Privacy evaluation of fairness-enhancing pre-processing techniques

par

Jean-Christophe Taillandier

Département d'informatique et de recherche opérationelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

31 Décembre 2020

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Privacy evaluation of fairness-enhancing pre-processing techniques

présenté par

Jean-Christophe Taillandier

a été évalué par un jury composé des personnes suivantes :

Louis Salvail
(président-rapporteur)

Alain Tapp
(directeur de recherche)

Sébastien Gambs
(codirecteur)

Gilles Brassard
(membre du jury)

Résumé

La prédominance d'algorithmes de prise de décision, qui sont souvent basés sur des modèles issus de l'apprentissage machine, soulève des enjeux importants en termes de la discrimination et du manque d'équité par ceux-ci ainsi que leur impact sur le traitement de groupes minoritaires ou sous-représentés. Cela a toutefois conduit au développement de techniques dont l'objectif est de mitiger ces problèmes ainsi que les les difficultés qui y sont reliées.

Dans ce mémoire, nous analysons certaines de ces méthodes d'amélioration de l'équité de type «pré-traitement» parmi les plus récentes, et mesurons leur impact sur le compromis équité-utilité des données transformées. Plus précisément, notre focus se fera sur trois techniques qui ont pour objectif de cacher un attribut sensible dans un ensemble de données, dont deux basées sur les modèles générateurs adversériaux (LAFTR [**67**] et GANSan [**6**]) et une basée sur une transformation déterministe et les fonctions de densités (Disparate Impact Remover [**33**]). Nous allons premièrement vérifier le niveau de contrôle que ces techniques nous offrent quant au compromis équité-utilité des données. Par la suite, nous allons investiguer s'il est possible d'inverser la transformation faite aux données par chacun de ces algorithmes en construisant un auto-encodeur sur mesure qui tentera de reconstruire les données originales depuis les données transformées. Finalement, nous verrons qu'un acteur malveillant pourrait, avec les données transformées par ces trois techniques, retrouver l'attribut sensible qui est censé être protégé avec des algorithmes d'apprentissage machine de base. Une des conclusions de notre recherche est que même si ces techniques offrent des garanties pratiques quant à l'équité des données produites, il reste souvent possible de prédire l'attribut sensible en question par des techniques d'apprentissage, ce qui annule potentiellement toute protection que la technique voulait accorder, créant ainsi de sérieux dangers au niveau de la vie privée.

Mots clés: Equité, respect de la vie privée, apprentissage machine, réseaux génératifs antagonistes.

Abstract

The prevalence of decision-making algorithms, based on increasingly powerful pattern recognition machine learning algorithms, has brought a growing wave of concern about discrimination and fairness of those algorithm predictions as well as their impacts on equity and treatment of minority or under-represented groups. This in turn has fuelled the development of new techniques to mitigate those issues and helped outline challenges related to such issues.

In this work, we analyse recent advances in fairness enhancing pre-processing techniques, evaluate how they control the fairness-utility trade-off and the dataset's ability to be used successfully in downstream tasks. We focus on three techniques that attempt to hide a sensitive attribute in a dataset, two based on *Generative Adversarial Networks* architectures (LAFTR [**67**] and GANSan [**6**]), and one deterministic transformation of dataset relying on density functions (Disparate Impact Remover [**33**]). First we analyse the control over the fairness-utility trade-off each of these techniques offer. We then attempt to revert the transformation on the data each of these techniques applied using a variation of an auto-encoder built specifically for this purpose, which we called *reconstructor*. Lastly we see that even though these techniques offer practical guarantees of specific fairness metrics, basic machine learning classifiers are often able to successfully predict the sensitive attribute from the transformed data, effectively enabling discrimination. This creates what we believe is a major issue in fairness-enhancing technique research that is in large part due to intricate relationship between fairness and privacy.

Keywords: Fairness, Privacy, Machine Learning, Generative Adversarial Network

Contents

List of tables

List of figures

Liste des sigles et des abréviations

GAN	Modèle génératif contradictoire, *Generative Adversarial Network*
MLP	Perceptron à plusieurs couches, *Multi-Layer Perceptron*
SVM	Machine à vecteurs de support, *Support Vector Machine*
DNN	Réseau de neurones profond, *Deep Neural Network*
NLP	Traitement de la langue naturelle, *Natural Language Processing*
DP	Confidentialité différentielle, *Differential Privacy*
BER	Taux d'erreur balancé, *Balanced Error Rate*
MLaaS	Service d'apprentissage machine, *Machine Learning as a Service*
NMI	Information mutuelle normalisée, *Normalized Mutual Information*
CART	Arbre de classification et regression, *Classification and Regression Tree*

Bag	Algorithme d'ensachage, *Bagging*
RF	Forêts aléatoires, *Random Forests*
GBM	Méthode de boosting du gradient, *Gradient boosting method*
MSE	Erreur moyenne carré, *Mean Square Error*

Remerciements

J'aimerais premièrement remercier mon directeur de recherche Alain Tapp ainsi que mon co-directeur Sébastien Gambs. Sébastien m'a accueilli dans son laboratoire, m'a intégré dans son équipe et m'a appuyé et offert un retour hebdomadaire tout au long de mes recherches. Je voudrais donc également remercier son équipe au Latece, notamment Rosin Claude Ngueveu, Ulrich Aïvodji et Antoine Laurent pour leur aide tant au niveau des connaissances techniques que leur aide durant la revue de littérature, mes recherches et expérimentations.

Finalement, j'aimerais remercier Céline Bégin pour son support tout au long de ma maîtrise et l'aide qu'elle m'a offerte a maintes reprises pour mieux naviguer l'administration du programme au DIRO. Après plusieurs visites inopinées à son bureau, je suis extrêmement reconnaissant de son pragmatisme.

Introduction

To some extent, excitement towards machine learning and artificial intelligence might seemed to be cooling down given the growth of private sector investment in AI-related projects decreasing in 2019 for the first time in 5 years [**31**]. However, this is at most a reckoning of the limitations of the underlying algorithms, not a reduction of their overall use or deployment. One area in which these algorithms are known to perform reasonably well is in support of decision-making systems, using anything from a dozen data points to a many thousands in the case of images as input and offer a probability as output, which will then be used by algorithm designers to make a decision. Use cases include bank loan applications [**32**], sentencing decisions [**84**], school admissions [**109**] and job applications [**92**]. For example the GRADE system [**109**] used by the computer science department of the University of Texas at Austin uses historical decisions made by admission committee to train a machine learning algorithm that will output a likelihood of PhD applicants being admitted to the program. Similarly, the unmanageable number of job applicants have encouraged research in systems using the candidate's job history in order to infer future behavior [**92**].

It is therefore becoming of increasing importance to understand how the use of machine learning models influence decision-making systems, amongst other things to prevent biases in the data to be reflected in the output of the algorithms. The risk is to perpetuate and reinforce those biases given the unprecedented scale at which those models are deployed in the real world. Another possible objective, which will not be discussed further in this thesis, is to better understand the inner working of algorithms by providing explanations for a decision, thus also increasing algorithmic accountability. A common example for the explain ability of algorithms is the decision tree, in which each branch characterizes the path that has led of a binary prediction. Here, the hope is to be able to provide a similarly simple explanation for other algorithms as to how they reach their predictions.

A second approach, which motivated this research, is to design methods to better control or bound the underlying decision-making process by compensating for a specific bias, which

has made headway mostly in computer vision [**82**]. This can be done in a variety of ways and according to different fairness metrics, a subset of which will be our focus in this thesis. Three main categories of fairness-enhancing algorithms exist: pre-processing, in-processing and post-processing techniques [**37**]. In a nutshell, post-processing consists in modifying the model output (*i.e.*, the decision) according to certain principles or metrics to ensure fairness. A straightforward example of this is offered in the same research by Kamiran et al. [**57**] consisting of modifying the leaves of their decision tree algorithm after its training in order to achieve better fairness. In-processing is about the model itself, and adjusting its inner working to prevent bias contained in the training data from being overly important in the decision-making process. A simple method by Kamiran et al. [**57**] consisting of the introduction of a regularising term for a decision tree. The regularising term will work against the algorithm training and attempt to compensate or prevent the use of biases contained in the training set. The last approach, which is investigated in this thesis, is to attempt to prevent the algorithm from relying on the data biases by transforming the data *a priori*, with the objective that the data should be usable by any downstream process without worries of discrimination and biases.

Furthermore, with the fast pace of new algorithms and architecture development in the field of machine learning and deep learning, some interest in the past few years has been on leveraging the adversarial learning approach that is the basis of Generative Adversarial Networks (GANs) to address the trade-off between utility and fairness discussed above.

In this thesis, we analyse three recent pre-processing methods that aim at creating a general transformation to be applied to input data, such as a loan applicant's personal information, to reduce bias measured through various metrics while preserving utility of the data, also known as a fairness-utility trade-off. The challenge is that besides directly removing the sensitive attribute, the correlations that this attribute has with the other features has to be mitigated as well, especially given the ease at which deep learning algorithms can recognize those patterns and correlations. Our contribution is to show that although those techniques allow their transformations to create (practical) guarantees that transformed data scores high on fairness metrics, it is often easy for an external classifier to predict the hidden sensitive attribute, which in effect nullifies their whole process. Two of the three algorithms studied leverage the GAN architecture (GANsan [**6**] and LAFTR [**67**]), while the third relies on deterministic transformations to change the training data before the algorithm training [**34**].

The main dataset used in this research to evaluate the success of attacks is the UCI data repository Adult dataset (presented in Section 1.2) and the Compas dataset (presented in

Section 2.2).

The main contributions of this thesis in the field of fairness enhancing pre-processing techniques can be summarized as follows:

(1) Optimizations based on a fairness metric do not offer sufficient protection against basic machine learning based inference attacks.

(2) This inference means that a properly tuned deep-learning model trained with the "fair" or "transformed" user profile might reconstruct this feature unbeknownst to its developer.

(3) The inter-dependency of fairness and privacy in fairness-related machine learning use cases means both need to be taken into account when developing fairness-enhancing techniques.

(4) The use of an additional external classifier exogenous to the base fairness-enhancing method during training seems to be able to mitigate some of those issues.

The next three chapters cover the relevant work necessary to understand this thesis' motivations. First, we review the concepts and previous work in machine learning and recent developments in generative models (Chapter 1). Second, we cover fairness concepts, popular metrics and recent techniques to improve fairness in decision-making algorithms (Chapter 2). Third, we look at privacy concepts, privacy-preserving methods and popular attack frameworks (Chapter 3) that motivate the development of fairness and privacy-enhancing techniques. We then go over our research objectives (Chapter 4), the experiments we ran on data generated by three different fairness optimising pre-processing techniques, as well as their resistance to simple machine learning algorithms (Chapter 5). Our evaluation of the fairness-enhancing methods is be three-fold: first, we look whether the method indeed produces data that is *fair* according to each method-optimized fairness metric. We then evaluate the control on the level of protection the method offers. Third, we evaluate whether it is possible to predict the original sensitive attribute that was hidden, using only the data that each method outputs. Finally, in Chapter 6, we conclude with recommendations for future development of fairness-enhancing methods.

Chapter 1

Machine learning basics

1.1. Preliminary notions

In this chapter, we first look at a brief history of machine learning, leading to the more recent subfield of deep learning [**42**]. We then explain and define fundamental machine-learning concepts before jumping to more specific deep learning models and architectures relevant to our experiments.

Finally, we have a look at current state-of-the-art architectures for various tasks as well as commonly used datasets for research and comparative analysis.

1.1.1. Machine learning categories and main concepts

Alexander Smola, in his 2002 book on machine learning states that *"[...] much of the art of machine learning is to reduce a range of fairly disparate problems to a set of fairly narrow prototypes. Much of the science of machine learning is then to solve these problems and provide good guarantees for solutions."* [**98**]. More concretely, it can be understood intuitively by looking at the linear regression algorithm that belongs to both statistics and was more recently borrowed by machine learning [**10**]. Linear regression takes as input historical data and learns to recognise patterns. Then, these patterns can be used to predict where future occurrences of data coming from a similar distribution will fall. This knowledge can then be used when looking at previously unseen occurrences of the data. As we will see, research in machine learning uses these principles to design powerful algorithms that are increasingly good at recognising such patterns in a variety of circumstances and tasks.

Machine Learning can be segmented into three broad main subfields: supervised, unsupervised and reinforcement learning. Supervised learning implies that the algorithm designer has a *labelled* dataset of points (x, y) in which x is a vector referred to as the

data input, which is of size d and $y \in \mathbb{R}$ as the label or target. The objective of the system is to be able to learn how to predict the y value having x as input by maximising the value of the probability $p(\hat{y}|x)$ in which $\hat{y}$ is the label the system predicts for its corresponding x. For example, x could consist of various information about a bank customer, and y the likelihood of this customer paying back a loan. Supervised learning, can be represented in a abstract manner agnostic to the underlying algorithm, model or architecture as a function with an input and output $\hat{y} = f(x)$. This function in practice sometimes represents a neural network, which is a collection or system with multiple units of calculation arranged one after another and in parallel, each taking a number as input and giving a transformed number as output, finally combining to offer what is often probabilities as final output. These neural networks are presented in depth in Section 1.1.

The system will slowly adapt its decision by updating the value of its parameters θ to improve the quality of its prediction by optimising an objective function. The objective function is optimised through a *loss function*. The loss function quantifies how good or bad the prediction of the system is. For example, the commonly used Mean Square Error (MSE) loss given in Equation 1.1.1 shows that for each of the n predictions of $\hat{y}$ made by the system, the loss will be calculated as the average of the square of difference between the true label y and $\hat{y}$.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1.1.1}$$

The loss function has to be chosen carefully, as it dictates how the system will be penalised for being wrong in its prediction, and therefore how it will adjust its weights θ internally, for instance through a back propagation algorithm for neural networks (see Section 1.1.2). Some loss function is specific to different types of tasks, for example the *Damage* loss is appropriate to compare the difference between the input and the output of a model.

Supervised learning can be, furthermore, split in two subdomains, depending on the type of the target value $\hat{y}$ it tries to predict. If y is a continuous variable, then the problem is a regression task, while when the system needs to decide in which of n classes the input belongs for $y \in [1, n]$, it corresponds to a classification task. In classification problems, the algorithm usually tries to identify the decision boundary, in which an input would be classified as one class instead of another. For example, for a binary classification task and for an input vector of size 2, a decision boundary could look like Figure 1.1.

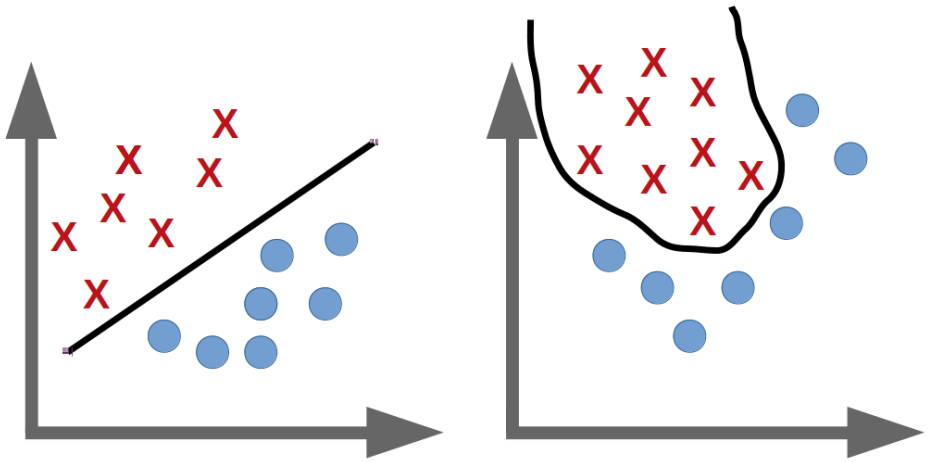

Fig. 1.1. Decision boundaries for a supervised learning algorithm. On the left, a linear regression is an algorithm able to find the decision boundary and correctly classify the data. On the right side, a much more expressive function (see Section 1.1.2) is needed to do so.

The second category, unsupervised learning does not rely on labelled data (*i.e.*, it has no knowledge of label y) but instead tries to recognise patterns within the data. For instance, one of the common unsupervised learning tasks is clustering, which attempts to group data points according to their similarities. A typical example of a clustering method is the k-means algorithm (see Figure 1.2).

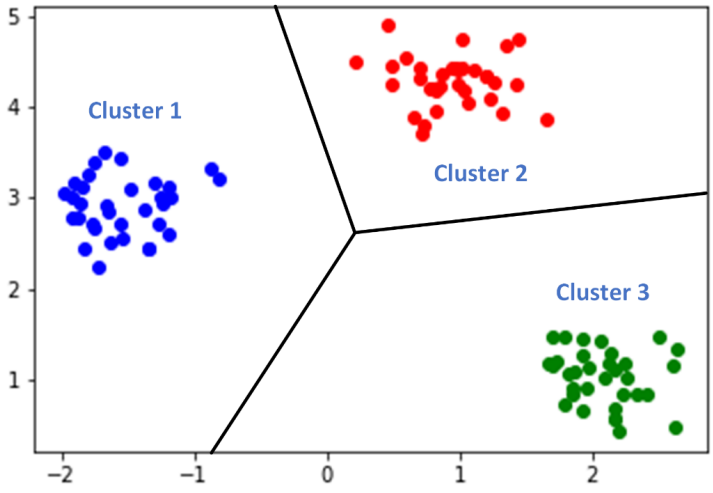

Fig. 1.2. Unsupervised Learning: Clustering of 2-dimensional dataset with k-means in which $k = 3$ [**69**].

Another unsupervised learning architecture is the Auto-Encoder (AE) [**42**], and more recently the Variational Auto Encoder (VAE) [**42**], which takes a data point as input and

passes it through a multi-layer neural network (a function $f(x)$ as described in Section 1.1) whose middle layer (referred to as the latent layer) is usually smaller than the input, and the output layer is the same size as the input. More on neural networks will be discussed in the following section. These auto-encoders are often used in dimensionality reduction tasks, where we want to find a representation of some data in lower dimensions, while retaining as much information as possible on the data. Typical use cases include data compression and visualising multidimensional data in a 2d or 3d graph. As shown in Figure 1.3, the portion before the latent layer is known as the *encoder* since it encodes data in smaller dimensional representations while the following portion is the *decoder* as it decodes the representation back to its original representation. Here, the objective function aims at producing an output that is as similar as possible to the input. When such an objective is attained, the middle (or latent) layer concentrates the useful information contained in the data to lower dimensions, hence removing *noise*. Because of this, it is also sometimes considered a dimensionality reduction technique. The Variational Auto Encoder is a version of the AE in which the latent layer is, instead of actual data, the parameters required to generate each piece of data. Taking the normal distribution as an example, the latent layer of a VAE will learn a means μ and standard deviation θ for each neuron of the latent layer, allowing to generate a value to be sent to the decoder. It is important to note that many other tasks, architectures and algorithms exist in unsupervised learning, but they are outside the scope of this thesis.

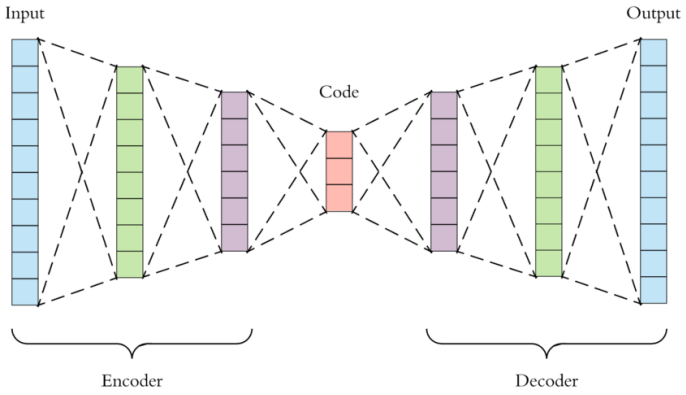

Fig. 1.3. Unsupervised Learning:
Auto-encoder architecture with 3 neuron
wide latent layer [**110**].

The last type of machine learning subfield is reinforcement learning. Reinforcement learning is based on the idea of not telling the system directly what is considered a right or wrong decision, but instead give general principles of how to achieve success [**42**]. Reinforcement learning therefore also does not use explicit labels nor an objective function

to tell it when it is wrong. This technique aims at finding an equilibrium between the exploration of the data space and the exploitation of previously learned features. The system poses actions that are generated by a policy following an input vector. It then receives a positive feedback if the action was a good one, and negative feedback if it was not the case. The system learns by trying to optimize the reward it receives. In addition, the output of the policy is expressed as a distribution over possible actions, therefore adding some level of randomness to the exploration. This is needed to avoid repeating the same actions and hopefully reach faster convergence towards better outcomes (*i.e.*, higher rewards) [**74**]. Reinforcement learning will not be discussed further in this thesis.

The remainder of this chapter will delve into an important subfield of machine learning: deep learning.

1.1.2. A brief history of neural networks

The concept of artificial neural networks dates back from Rosenblatt's initial paper in 1958 introducing the Perceptron, which is basically a 1-layer neural network without activation function [**90**].

A neuron, a neural-networks' basic computation unit, takes as input a value, transforms it and outputs a new value (or possibly the same). The name comes from the fact that its output is a signal whose degree can change during training, analogous to how the brain's neurons communicate between themselves and evolve. Ideally during training, each neuron learns different features of the data distribution it is trained on, which is more easily visualised in computer vision as shown for various layers of the network in Figure 1.5. The concept of neuron exists in all neural networks models [**78, 82**].

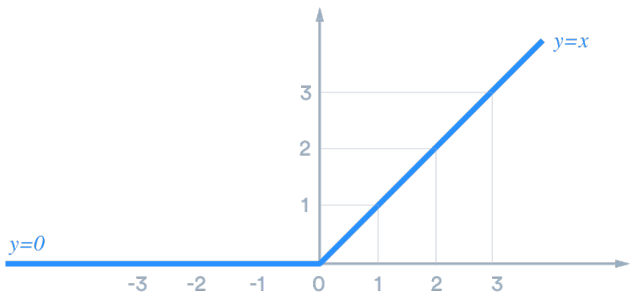

Fig. 1.4. ReLU Activation Function
$f(x) = max[0, x]$.

Activation functions will be explained in depth in later paragraphs, but simply put they are a mathematical operation made onto each neuron's output. A layer consists of one

or more $d \in \mathbb{N}$ neurons. Activation functions were introduced by Minsky and Papert in 1969 [**73**] but among the major breakthroughs was the discovery of the rectified linear unit (ReLU) activation function (Figure 1.4) [**76**], taking over from less popular sinus and tanh activation functions. ReLU has since gained the title of most widely used and successful activation function mostly due to its consistency, although extensive research is still being conducted to find better alternatives [**86**]. The simplicity of the ReLU allows for easy computation, good performance of algorithms and fast training of models.

Fig. 1.5. Use of *multifaceted feature visualization* to show what each layer of a Convolutional Neural Network learns [**78**]. Each pixel of each picture corresponds to a single neuron in the network.

A first key addition to the training of modern neural networks is back-propagation with gradient descent as a learning mechanism, solving the issue of how to adjust in an iterative manner the parameters of the neurons of the hidden layers (*i.e.*, the layer located between the input and output layers) [**91**]. It created a relatively inexpensive method of converging towards local (and sometimes global) minima. The alternatives used before the discovery of the back-propagation algorithm were extremely expensive, which for some times has limited the applicability and widespread use of neural networks (Newton [**75**], Quasi-Newton [**104**] [**41**], Levenberg-Marquardt Method [**62**], all of which were based on expensive *Hessian* matrices).

Both back-propagation and ReLU have greatly contributed to increase the accuracy of the predictions of neural networks. Nonetheless, the training of neural networks remains costly and the lack of understanding of its inner working due to its black-box aspect makes it a less popular choice against simpler machine learning algorithms, such as Support Vector Machines (SVMs) and decision trees that consistently offered similar performances throughout the 90s [**70**] in a much shorter training time [**14**]. More recently, an even wider choice of activation functions as well as different architectures, such as convolutional Neural Networks for image recognition and Recurrent Neural Networks for text processing, have emerged (more in Section 1.2). The neural networks have also gained in popularity due to the combination of factors such as Moore's Law, parallelisation, increase of spending (including development of GPU acceleration) as well as improvement of the algorithm's efficiency [**51**].

Among the important factors to consider when building a neural network is the bias-variance trade-off. Using our example from Figure 1.1, the linear regression on the left side is in a family of functions that could qualify as low expressivity functions because of their inability to discover the decision boundary for a dataset such as the one on the right, which is would lead to a high bias error. The bias error refers to how well a learning function (here the linear regression) is able to model the true function (represented by the training set) [**77**]. On the other hand, we could say that the function used to represent the decision boundary to the right is relatively more expressive and would have a lower bias when working on the right-hand dataset. Given various datasets, and even within the same dataset, there are many ways (*i.e.*, functions) to define the decision boundary. In particular, some functions allow us to draw more complicated ones, with the caveat that more expressive functions usually need more data to be successfully trained. It would therefore be tempting to always use a highly expressive function to be safe (as long as we have enough data). However this may lead to another issue called *overfitting* [**48**] and high variance, which refers to the fluctuation in performance of the trained model when facing

new data [**77**].

The bias-variance trade-off hence appears, when we want to use a function that is expressive enough to be able to find a good decision boundary (*i.e.*, low bias) while trying to avoid it causing too much overfitting and resulting in high variance. It is usually recommended to follow the Occam's Razor principle [**87**], which states to use the simplest (*i.e.*, low expressivity) function that works sufficiently well. High variance, usually undesirable, means that the model is not able to generalise to the general population, outside of the data it was trained on and that its performance varies significantly depending on the data it is used on. The decision criteria for identifying the expressivity level that we need include the size of data available for training, their properties, how confident we are that the data are representative of real-world data. Deep learning simply is a new, highly expressive function, with the advantages and disadvantages that were described above and controlled through the number of layers and their respective sizes. It is worth noting that some recent research indicates that the variance might not increase with the model's complexity when we also increase the width of its layers [**77**].

Another key concept already mentioned is generalization, defined by Google's Machine Learning Crash Course as *Generalization refers to your model's ability to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model* [**66**]. Generalization is a key quality for any machine learning algorithms and closely related to overfitting [**48**]. Given the higher-than-ever expressivity of models that deep-learning permits, generalization has to be kept in mind when training in order to ensure the model will be making accurate predictions when used on new data that has not been seen during training. One way to mitigate this is through a training-validation-testing separation of the dataset available for training. Following this methodology, an algorithm is trained by looking only at a subset of the data. The proportion of each dataset depends on many factors such as the task considered, the size of the full dataset as well as the architecture, depth (*i.e.*, amount of layers) of the neural network and acceptable prediction accuracy error rate [**45**]. The validation set is a relatively smaller subset compared to the training one and used during training to verify the quality of the prediction at regular intervals, leading to adjustment of the parameters of the model. Finally, the test set is used at the end of training to evaluate the generalization ability of the model.

The concepts of *generalization* and *train-valid-test split* are often dependent on a last key feature of the data, which is the extent to which the training data is representative of the underlying true data distribution. We humans work the same way; as a toddler if you are almost always presented with green apples and yellow pears (your training set), when

you see a green pear for the first time, there is a good probability you will classify it wrongly as an apple. The reason is that you never (or rarely) encountered such thing as a green pear. Supervised algorithms work the same: they are able to generalise, but have a limited capacity to extrapolate to data that is drawn from a different data distribution or domain.

1.1.3. Generative Adversarial Networks (GANs)

Generative Adversarial Networks were introduced at NIPS 2014 with the objective of applying deep learning to bypass the mathematical limitations of existing generative models (including auto-encoders), with an initial focus on image generation [**43**]. GANs are composed of two neural networks competing against each other, each with their own objective, architecture, loss function and training procedure but trained as a whole system. The first model called the *Generator* (red box of Figure 1.6), takes as input a vector of random (or coming from a particular distribution) noise and outputs what we would hope to be something similar to data coming from the real distribution of the training data. The second model named the *Discriminator* (blue box of Figure 1.6), is a simple binary (two classes) classifier that is randomly shown either an input from the real distribution or one generated by the *Generator* and tries to determine which of the two it observes.

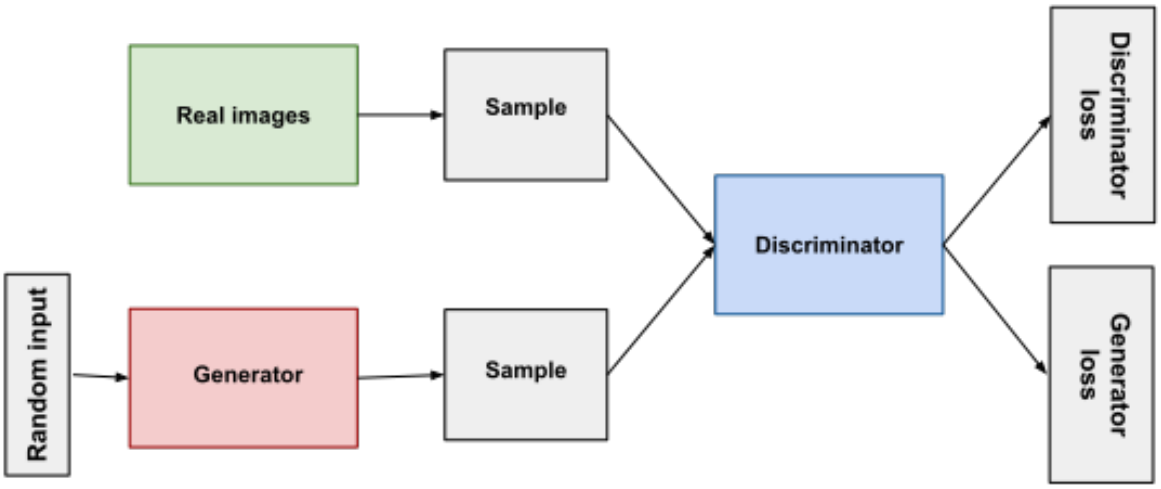

Fig. 1.6. Generative Adversarial Network general structure [**44**].

In its first versions, following the discriminator's prediction, the loss is compiled for both models and back-propagated on each through gradient descent. However, in practice the difficulty of successfully training such networks has often resulted in different rates of training and loss computation. This creates an adversarial game between the two networks where the generator G tries to minimize the accuracy of the discriminator accuracy while

the discriminator D tries to maximize its accuracy [**93**]. The objective function of this game can be formalized as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{1.1.2}$$

This non-exhaustive list of architectures has been developed over the years and is designed for a specific task such as predicting the probability of a customer to pay back his loan. Nonetheless, we have witnessed the emergence of a handful of benchmark tasks and datasets allowing to compare those different architectures. These are presented in the following section.

1.2. State-of-the-art in machine learning

In this section, we go over the common datasets used for benchmark in machine learning tasks and the current top performing architectures. This includes computer vision, Natural Language Processing (NLP) and tabular data (structured into rows, columns and cells often represented in an Excel-like table). We will go over the first two quickly and the last one in more detail, as tabular data is the focus of this research for reasons explained in Chapter 4.

Starting with computer vision, one of the most commonly used dataset is MNIST. This dataset consists of 60 000 training images and 10 000 test images, both 28x28 pixels, giving an input vector of size 784. The images represent handwritten digits from 0 to 9 (thus resulting in 10 classes in total) in black and white, and is nowadays considered a relatively easy task. Basic machine learning algorithms such as the k-nearest-neighbour algorithm consistently achieving less than 2% error on the test set, Support Vector Machines (SVM) less than 1%, and more advanced convolutional neural networks are able to reach less than 0.5% error rate [**116**]. Other popular image datasets include CIFAR10 (10 classes) and CIFAR100 (100 classes), which consist in slightly larger (32x32) colour images representing many things from aeroplanes to trucks, as well as cats and dogs. Regular Deep Neural Networks (DNN) are able to achieve above 90% on the 10-class version, while CNN reach closer to 95% since 2015 [**89**]. CIFAR100 high class numbers are still proving a challenge with state-of-the-art architectures unable to reach more than 76% accuracy, with unsurprisingly many top performers being the same architectures as CIFAR10.

Natural Language Processing is a field with a much wider range of tasks that can hardly be compared between themselves. These tasks range from translation, text classification, language modelling, speech recognition, sentiment analysis, to text summarising. However,

a key concept that has lead to better performance and faster training in language tasks are word embeddings and the use of context (other words before and after a target word). In particular, the research has gathered speed following Mikolov's 2013 Word2Vec embedding [**72**] and many more that followed. In addition, the performance of models varies across different languages. Google is trying to solve this issue and has recently released a multilingual benchmark consisting of 7 tasks, trying to benchmark complete NLP systems [**52**], although it is too recent to confirm widespread adoption. Nonetheless, various language models architectures for different tasks have shown consistent improvement in top performance scores achieved over the past 10 years [**106, 94**] with a majority of advances focused on English-based tasks or other languages to English translation tasks.

To summarise, given their obvious real-world use and attractiveness to the private sector, a lot of efforts have been put into developing increasingly complex deep-learning-based models for computer vision and language processing tasks. The development of CNN's and different language processing models has enabled neural networks to surpass the traditional methods used for vision and language tasks [**116, 94, 89**], but not for tabular data. That being said, Haldar et al. researching for AirBnB have shown in 2018 that for larger datasets (1.7 billion structured data points with 32 dimensions), DNN's outperform gradient boosting [**46**], and others have shown seemingly generalisable attempts to beat decision trees and gradient boosting techniques [**83**]. Nonetheless a handful of datasets stand out for often being used to benchmark new neural network architectures aimed at tabular data. The most common according to UCI Machine Learning Repository is the Adult Census Dataset [**24**]. This dataset is composed of 32 561 profiles extracted from the 1994 US Census Bureau Database, and the binary classification task is to determine whether each individual's (entry) revenue is above (>) or below (<=) \$50 000. Each entry contains 12 features plus the target (label), often used as 6 numerical and 8 categorical features. Historically, categorical features such as *native-country* had been embedded in a two dimensional one-hot (Figure 1.7) and min-max encoding (Equation 1.2.1). Min-max is a normalization method to improve a model's learning capabilities by bringing closer values that have large variance between each other.

$$X = \frac{X - X_{min}}{X_{max} - X_{min}}. \tag{1.2.1}$$

Deep neural networks achieve around 86% accuracy on the Adult dataset, which is about the same as other machine learning algorithms shown in Figure 1.8.

native-country	native-country=United-States	native-country=Canada	native-country=Cuba	native-country=Jamaica
United-States	1	0	0	0
United-States	1	0	0	0
United-States	1	0	0	0
Canada	0	1	0	0
Cuba	0	0	1	0
United-States	1	0	0	0
Jamaica	0	0	0	1
United-States	1	0	0	0
United-States	1	0	0	0

Fig. 1.7. Illustration of one-hot encoding.

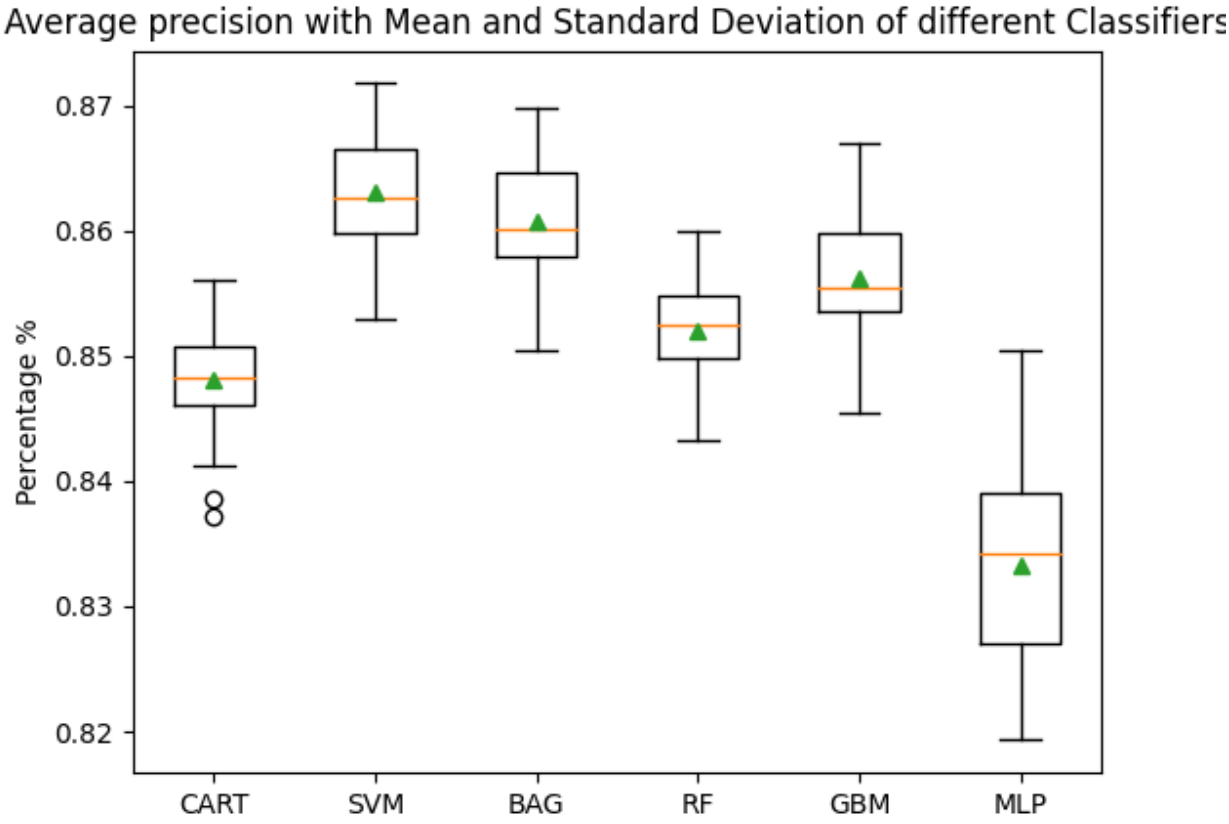

Fig. 1.8. Classification of Income on Adult Dataset. 10k-fold cross validation repeated 3 times for each algorithm. One-Hot and Min-Max Scalar are used for encoding [**11**].

A new trend to deal with tabular data has emerged in recent years. At its core it is a new way to encode the categorical features, borrowing techniques developed in natural language processing, more specifically 1-dimensional and 2-dimensional word embeddings [**100**]. This technique effectively creates a 2D image from textual or tabular data, in order to use the relatively higher performance of CNN's to solve this kind of task. Results have

only been marginally better, but they might show a path towards specified neural network architectures for structured data, the same way text and images have theirs.

Finally, the access to pre-trained or deep-learning based classifier services have been rolled out by all major cloud providers [**17, 95, 5**] allowing various levels of abstraction of machine learning fundamentals in order to ease the process of developing deep-learning based classifiers. This allows anyone with a training dataset to create its own state-of-the-art model and deploy it to for real-world applications, or to use one of the pre-trained models offered as a service. The fast pace advancement of new architectures and techniques, paired with the availability of an increasing number of non-technical users has created some moral hazards and potential issues. Those often have to do with the lack of explain ability of how algorithms achieve a certain outcome. For example, it is not trivial to know the extent of the impact of one feature, age for example, on the output of the model. In addition, an issue has emerged with libraries such as python's Scikit-Learn and pytorch, allowing users to create and train machine learning algorithms in a few lines of code. It is hence becoming increasingly difficult to keep track of where and how these algorithms are being used for accountability purposes. These issues will be discussed in the following chapter.

Chapter 2

Fairness in machine learning

In the early days of deep learning developments, designers of decision-making algorithms were sometimes tempted to claim the absence of bias due to the objective nature of their algorithms as sufficient fairness criteria, considering that the bias within the dataset was not their fault or problem. Statistical bias occurs when the decision-making algorithm's expected value $\mathbb{E}[\hat{y}]$ is consistently different from the real value y [**3**].

For a classic machine learning designer, the main objective is to create a classifier with 100% accuracy on the test set (although rarely the case in practice), which would represent a perfect generalisation. One shortcoming of this approach is that statistical bias says nothing about the distribution of data. In particular, an algorithm could assign the average probability all the time, keeping statistical bias low, while having a very high error rate.

Another major issue is that these types of algorithms are increasingly deployed in the real world, having significant impact on people's lives through hiring decisions [**92**], bank loan applications [**32**], recidivism and sentencing [**84**], etc. Those are part of the reasons why the lack of statistical-bias argument was quickly rejected by the FAT (*Fairness, Accountability and Transparency*) research community, composed of computer scientists, ethicists and lawyers to name a few, who claimed that the real objective should be *"making algorithms systems to support human values"* [**3**]. In this chapter, we will first provide an overview of fundamental concepts and debates in the algorithmic fairness field. Then, we will present three state-of-the-art techniques that have been developed with the objective of mitigating the fairness issues of classical algorithms, each focusing on specific fairness metrics. All will then be analysed and compared in Chapter 5.

2.1. Fundamental notions of fairness

Finding the root causes of unfairness is a challenging task in itself. Previous research has identified three main possible causes: prejudice, underestimation and negative legacy [**58**]. To help understand, we can adjust the model's prediction formula used in the previous chapter: $\hat{y} = argmax(p(y|x))$ where y is the desired output and $x, \hat{y}$ the actual input and output respectively. We then extract from x the sensitive attribute s we want to protect (often race or gender), which would give us:

$$\hat{y} = argmax(p(y|x, s)). \tag{2.1.1}$$

With prejudice, the decision is made directly by looking at a certain feature, resulting in a *direct discrimination*. Prejudice is difficult to pinpoint when decisions are made by humans given that those prejudices are not straightforward to detect, and even sometimes unknown to the holder. A similar issue arises for black-box learning algorithms whose predictions are used by decision-making processes, and even more in deep learning, due to the lack of explainability (see Section 1.1.1), which makes it difficult to assess whether the decision-making process is prone to prejudice. Moreover, removing the feature that leads to discrimination is often not sufficient since a machine learning model can easily exploit the correlation between this attribute and use other features that will act as proxies, resulting in *indirect discrimination* [**81, 58**]. We will go over this phenomenon in more detail further on.

The second category, underestimation, refers to a model that cannot reach convergence or its full capacity, due to the availability of a small dataset. This phenomenon is related to our earlier concept of variance and capacity to generalise with high-capacity models needing more training data to reach convergence (see Section 1.1.2). This issue could possibly be mitigated by using a model with a lower capacity.

Finally, the third category is negative legacy, which reflects societal bias through either historical undersampling of a specific group or a history of humans consistently wrongly labelling subgroups, whether consciously or not. An example would be an implicit bias of a bank clerk that consistently and unreasonably classifies customers with blue eyes as less likely to pay back loans. This societal bias in the training dataset will likely be picked up by the learning algorithm and lead to a biased decision-making process. This issue has been reported on widely [**88, 97**]. For example, in computer vision a study has shown that facial recognition systems, including commercial systems [**19, 84**], are not able to recognise non-Caucasian people's pictures with the same accuracy as Caucasian ones [**80**]. This point demonstrates a known limitation of machine learning

algorithms, which is that they can never be better than the underlying data they are fed with.

Other challenges in the fairness community have not only to do with the difficulty of agreeing on terms, definitions and metrics, but also with which ones are the right ones to focus on. This is key to the development and comparison of fairness enhancing and/or privacy-preserving techniques. An often used high level segmentation distinguishes between *Individual Fairness* and *Group Fairness*. In a nutshell, individual fairness requires that similar individuals have similar expected value from a given predictive model's output [**27**]. Basically, this means that *similar individuals should be treated similarly* [**119, 28**]. For binary sensitive value, for any small value of ϵ, individual fairness can be defined as:

$$p(y|x) \approx p(y|x') \text{where } dist(x,x') \leq \epsilon \tag{2.1.2}$$

Group Fairness, also referred to as demographic parity, depends on different demographic groups having similar treatments [**27**]. For two distinct subgroups of the population, for instance male and female, we should have:

$$p(y|x, s = Male) \approx p(y|x, s = Female) \tag{2.1.3}$$

where s is the sensitive attribute (either male or female in this case), x represents all other attributes used as input to the model and y the desired output for each corresponding set of inputs x,s.

As a subset of Group Fairness, Feldman and co-authors [**34**] have proposed to apply the concept of *disparate impact* to machine learning. The concept dates back from a Supreme Court of the United States judgement in 1971, which has defined the term [**79**]. According to this judgement (and the interpretation of Feldman and co-authors [**34**]), Disparate Impact occurs "when a selection process has widely different outcomes for different groups, even as it appears to be neutral" [**34**], emphasising the effect that discrimination can have over specific attribute(s) at the group level. The acceptable rate set by the Supreme Court is less than 20% difference between both groups' decisions, which is why it is sometimes referred to as the 80% rule.

Take for instance, the scenario of an automatic bank loan application system that could lead to discrimination, which has gender as the sensitive attribute. In this setting, individual fairness would aim at ensuring that regardless of gender, similar profiles have a

similar chance of being approved, whereas group fairness would put emphasis on one (or possibly multiple) attributes [**34**], hereby requiring that the probability of women being approved is similar to that of men. Given the common use of Disparate Impact and general acceptance as a valuable fairness-enhancing method in the literature, we decided to include it as one of the three methods studied in Section 2.2.3 and compare it with other methods in our experiments.

Many other metrics of fairness can be considered to belong to one of those two categories. For example, Equalised Odds and Predictive Rate Parity [**39**] are instances of Group Fairness while Balanced Error Rate and Counter Factual Fairness [**61**] can be consider to belong to Individual Fairness. The relevance of using one type of fairness metric over the other is an ongoing debate in the field, which needs to take the context into account and is beyond the scope of this research. Notably, Dwork [**26**] argued that group fairness is problematic as it often leaves some individual with an unfair outcome. In addition, Chouldechova (2017) has shown that even when aiming to achieve group fairness, it is not possible to simultaneously optimize more than two metrics at the same time [**16**]. Nonetheless for the purpose of this research, we will focus on various definitions and methods of group fairness-enhancing methods as they have received extensive interest in recent years. We should keep in mind that the definition of what constitutes a *group* and its granularity has a huge implication when quantifying fairness. For example, we might split a group between male and female, and see very little discrimination, while further splitting the group of female using another attribute such as race, see more fairness issues arising.

A fundamental trade-off when improving an algorithm's fairness, regardless of how it is defined, is between utility and fairness. In particular, it has been demonstrated that optimising an algorithm with respect to a particular sensitive attribute and a single fairness metric usually works in opposition to the level of utility, as measured by the accuracy of the prediction [**20**]. This occurs for instance when the model's prediction relies on the very same sensitive attribute [**47**]. In other words, increasing fairness often means sacrificing some utility. This specific trade-off has been at the centre of the evaluation of new techniques and algorithms, with the objective of increasing the former as much as possible, while minimising the effect on the latter. Some of those techniques will be examined in depth in Section 2.2 and assessed in Chapter 5.

Recall that the removal of the sensitive attribute (*i.e.*, the race of a loan applicant), also referred to as blindness, from the data used to train the learning model or from the classifier's input is not sufficient to prevent the risk of discrimination, as it can lead to very little change in the results of the prediction [**47, 3**]. The problem lies within other features that

might contain hidden information about the sensitive attribute through correlations [**99**] effectively acting as a proxy or information leak of the sensitive attribute [**26**]. In addition, machine learning algorithms are very good at detecting even more complex patterns in the data, such as linear combinations of features, to improve its prediction accuracy. A typical example of proxy is how closely ZIP codes (not always considered as personal identifiable information) are related to race in some parts of the United States [**13**]. Proxies may give rise to indirect discrimination, especially given the complexity of deep learning models and the challenge to explain how information is conceptualised in each layer and neuron. As long as we lack the capacity to explain how deep-learning algorithms come up with their predictions, this issue will likely remain. In the meantime, the US Consumer Financial Protection Bureau recognises that to avoid discrimination it is better to include sensitive information in the algorithm to actively mitigate against potential bias than not collecting this information [**13**]. This can be considered as a final blow to the lack-of-statistical-bias argument.

Knowing that *blindness* of the sensitive attribute is not sufficient, and that after deciding which fairness metric to optimise, the next step is the choice of the method to find an ideal trade-off between the data utility (*e.g.*, with respect to a particular task) and this fairness criterion. Offering these types of guarantees has also been shown to be easily quantifiable but non-trivial to find the optimal solution. Note that the process of enhancing the fairness of machine learning algorithms has many names in the literature some calling it adversarial and fair representation [**67**], sanitised data [**49**], de-correlated data, censored and fair representation [**55**] [**55**], disentangled representation [**21**], just to name a few. A comprehensive analysis of various ways of tackling this trade-off by Friedler and co-authors has identified [**38**] three different types of interventions, namely *pre-processing*, algorithm modification, also called *in-processing*, and *post-processing* techniques modifying the model output.

We will delve deeper in the *pre-processing*, but first briefly explain the other two approaches. The *in-processing* or algorithm modification approach consists of adjusting the training mechanism of the algorithm to learn a model respecting a chosen fairness metric. Some of these so-called *fair classifiers* are fine-tuned for each algorithm we want to implement (*e.g.*, SVM or logistic regression) [**118**] while others offer a more generalised approach such as adding a regularising term to the loss function used during the training phase [**58**]. More precisely, this term will adjust the loss function to guide the training in the direction of increased fairness, which as mentioned earlier will often work in opposition to the main term of the loss that is optimising the accuracy of predictions. An example of a fairness-adjusted MSE loss is given in Equation 2.1.4 in which $R(\cdot)$ is the regularising term,

η a factor allowing to control the influence of R in the loss and s the sensitive feature.

$$Loss(x,s,y,\hat{y}) = MSE(x,y,\hat{y}) + \eta R(x,s,\hat{y}) \tag{2.1.4}$$

The *post-processing* approach adapts the output of a classifier that was previously trained, thus attempting to compensate for a negative bias embedded in it. One of the first papers to successfully introduce such an approach for decision trees in 2010 [**56**] also incorporates some algorithm modification techniques. However, this work is specifically tailored for decision trees only, which are known to have a decision-making process easy to explain, hence to control [**108**]. Other research has proved that post-processing techniques with binary sensitive attribute and prediction are computationally intractable and require strong relaxation of the equalised odds criteria when using some type of loss functions that otherwise are known to work well [**112**]. Post-processing can also be considered risky since the sensitive data would still need to be used as input to the model, thus causing potential privacy harms.

Finally, the *pre-processing* approach is the design of new representation of data that prevents or at least minimises biases and unfair treatment with respect to some demographic group in a later task (such as classification). It is especially relevant for this research because it can be done prior to any data analysis task and machine learning algorithms, and is therefore much more generic. The pre-processing approach transforms the data to remove the negative bias *a priori*. This transformation could possibly be performed by a third party, who would then share the transformed data with anyone. Another possible use case for the pre-processing approach is the situation in which the owner of the data (here the individual) locally *pre-processes* his personal data (*e.g.*, using his/her personal phone) before sharing it with a third parties or advertisers. This would contribute for a user to maintain the sovereignty on his own data while also minimising the trust assumption on external entities.

A lot of the research on structured data, such as tabular data, mentioned in previous paragraphs were developed for standard machine learning algorithms (*e.g.*, decision-Trees, SVM's, etc), given their high accuracy to computational cost ratio (see Section 1.1.1). Deep learning based techniques have also recently been investigated to address the utility-fairness trade-off. In the following section, we analyse and compare state-of-the-art techniques in

both deep-learning and the deterministic transformation-based method of Disparate Impact Remover (Section 2.2.3).

2.2. State-of-the-art in fairness-enhancing methods

This section presents common datasets that have been historically associated with studies of fairness-enhancing methods and techniques. We will see some of these have history that has wide societal implication. Others are datasets we already saw in Section 1.2, and where initially popular datasets for the general field of machine learning. Researchers later found fairness discrepancies with respect to specific attributes. Similarly to machine learning, many datasets that are currently used by researchers will not be discussed, the datasets discussed here are ones that standout by their relatively widespread use to compare fairness-enhancing techniques between themselves.

The Compas recidivism dataset is a sparsely populated dataset containing between 5 000 - 10 000 rows (depending on algorithms' designers data pre-processing) and around 50 attributes that were created by ProPublica [**84**] using a recidivism prediction tool developed by NorthPoint Inc. and used at least in one US state's legal system to compute the risk of recidivism [**54**]. The story of this dataset begins when ProPublica published a newspaper article outlining how the algorithm was biased towards white defendants, with among other things a much higher false positive (*i.e.*, predicting likelihood to recidivate) recidivism rate for black defendants and much higher false negative rate for white defendants (*cf.* Table 2.1). NorthPoint Inc. (now Equivant) rejection of the study results [**111**] sets off the debate about which fairness metric is the most appropriate in which situation. Not only the discussions were beneficial to raise the awareness of fairness issues in machine learning, but it also resulted in the Compas dataset becoming a benchmark for new fairness-enhancing techniques.

A second popular dataset in the fairness literature is the Adult Census dataset [**24**] discussed in Section 1.2. Although not initially collected for this purpose, this dataset is especially relevant for fairness-related tasks because it is biased, to different extents, both with respect to race in which whites account for 87% of profiles and gender as males represent 2/3 of the data (our research focus primarily on gender bias). Although the difference of false negative rate for the gender variable using most machine learning algorithms is not as large as with Compas, it has already been demonstrated that for individuals in executive or managerial occupations, women are more than twice as likely to fall in the false negative category than men [**15**]. This is a good example of how the concept of group in *group fairness* can be defined in different ways, which can significantly impact

the appraisal with respect to (the lack of) fairness (see Section 2.1). For this dataset, it was also demonstrated that the difference in false positive rate tends to decrease as the available training data increases [**15**]. This phenomenon relates to the *variance* (with respect to better generalisation or accuracy on test set) that tends to diminish with more data to train on.

Among other popular datasets used in fairness is the Ricci dataset coming from the US legal systems, the case was Ricci v. DeStefano Supreme Court (2009) discussing a firefighter promotion exam that is biased over race. This dataset has only 118 entries, which is a very small dataset for most learning algorithms. Another dataset commonly used is German Credit, which is composed of 1000 entries with gender and age being the potential sensitive features, although the bias is less pronounced than in the other datasets mentioned previously. These datasets are often used as secondary datasets to help support results, but rarely as the main ones.

2.2.1. Commonalities across the field

Before delving into the different frameworks and algorithms for tackling the fairness-utility trade-off, we will outline the commonalities that are shared by most of them. First, the utility-fairness trade-off is usually controlled by a parameter, which we will refer to as α. However, as it is often the case in machine learning, the naming convention for this parameter varies widely from one paper to another. It has become increasingly important, especially given the debates we mentioned in Section 2.1 about which fairness metric to choose, and which level of unfairness is acceptable. This parameter α offers the possibility to tune the level of fairness we want to achieve. Ideally, it would display a linear relationship with the fairness metric the algorithm optimises, but as we will see (Chapter 5) this is not always true.

Group fairness metrics have been studied extensively and among them, demographic parity, equalised odds and equal opportunity (at the core of the disparate impact method in Section 2.2.3) are relatively common (Equations for binary tasks in Section 2.2.1). Balanced Error Rate (BER) is also a well-established metric for measuring the accuracy in binary classification tasks. In particular, we will use it later to quantify the difficult of predicting the sensitive attribute in Section 2.2.2. The BER can be seen as the average rate of false prediction across each class. An optimal BER value varies between 0 and 1 and an ideal value sits as close as possible to 0.5.

Demographic Parity: $p(\hat{y}|s=0) = p(\hat{y}|s=1)$ (2.2.1)

Equal Opportunity: $p(\hat{y}=1 \mid s=0, Y=1) = p(\hat{Y}=1 \mid s=1, Y=1)$ (2.2.2)

Equalised Odds: $p(\hat{y}=1 \mid s=0, Y=y) = p(Y=1 \mid s=1, Y=y)$ (2.2.3)

Balanced Error Rate: $(\textit{False Negative rate} + \textit{False Positive Rate})/2$ (2.2.4)

One of the challenges is that not all datasets are biased towards the same attributes and/or fairness metrics, making it difficult to benchmark techniques across them. Tables 2.1 and 2.2 outline the metrics that are known to be specifically problematic for both the Compas and Adult datasets, in which statistics for Compas are computed with respect to race, while for Adult gender is considered for the bias.

Metric	**All Defendants**	**White Defendants**	**Black Defendants**
Total Data	7214	3696	2454
False Positive rate	32.35	23.45	44.85
False Negative rate	37.40	47.72	27.99
BER	0.1431	NA	NA
Disparate Impact [**34**]	0.7244	NA	NA

Table 2.1. Various Fairness Metrics of Compas dataset (White vs Others) [**84**].

Metric	**Adult dataset**
Disparate Impact	0.3482
Demographic Parity	0.3709
Equalised odds	0.16
Balanced Error Rate	0.1431

Table 2.2. Various baseline Fairness Metrics of Adult dataset (Male vs Female) [**67**].

These metrics were chosen because they are the ones implemented in the algorithms we will see in later sections, will be fundamental concepts used in the next chapters and are usually considered good proxy metrics for fairness.

2.2.2. Adversarial game for fair data generation

An idea that has gathered a lot of interest in recent years, is to apply a GAN-like approach to fairness-enhancing algorithms [**21, 6, 67, 115, 93**].

As illustrated in Figure 2.1, there are several components in the architecture. In our setting, two players (*i.e.*, the *generator* and the *discriminator*), which are usually

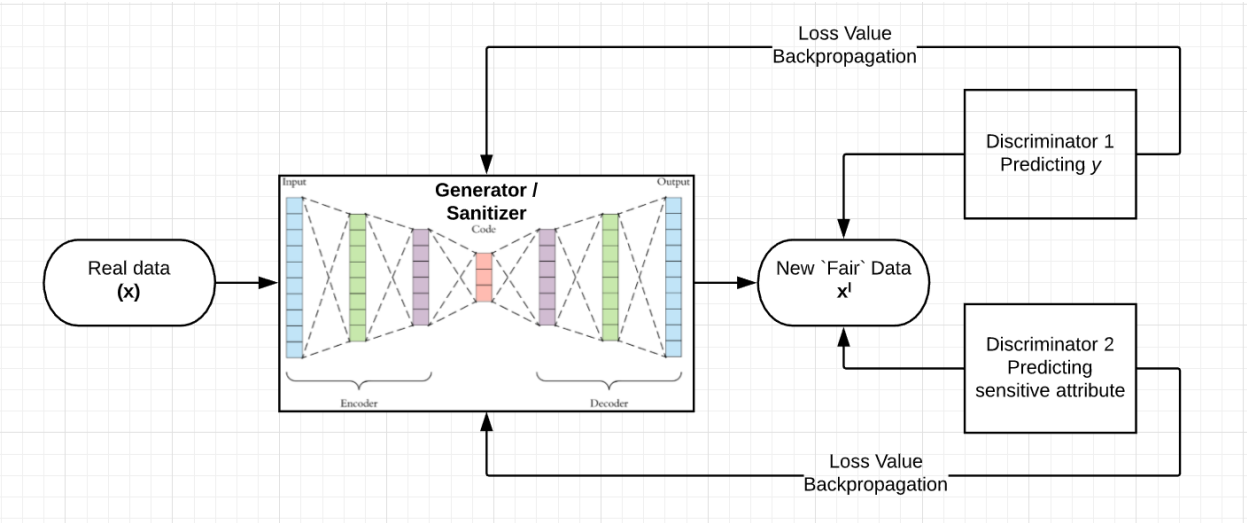

Fig. 2.1. High-level architecture of GAN-based fairness-enhancing techniques.

neural networks, are competing against each other to optimise the information leaked about the sensitive attribute, thus tuning the fairness-utility trade-off. While the details of the implementation and the architecture used may vary, the generator, usually an auto-encoder, takes a real profile as input and produces a new profile as output. More precisely, the generator's objective is to produce a modified version of the profile from which discovering the sensitive attribute is impossible by reducing correlations between the sensitive attribute and other attributes, which is why it is sometimes referred to as a sanitiser [**6**]. One of the main differences in implementations is the representation space in which the newly generated data lives. For instance, one possibility is that the profile produced remains in the same space (ie. the output profile share the same dimensions) as the input profile. One advantage of this situation is that the attributes are then of same size and can be easily compared with the original attributes [**6, 58**], allowing to easily analyse the data generated as well as the remaining correlations. Another possibility is that the profile generated is not in the same space as the original one [**67**], which offers more freedom for exploring alternative representations, but lacks the interpretability of the first method since human readable feature names such as "age" or "education" will be lost. Note that both methods implement the generator as an auto-encoder whose hyper-parameters (*i.e.*, number of layers, size of latent layer, ...) depend on the implementation, but in which the sensitive attribute that needs protection is removed from the output.

The second player, the *discriminator*, can also be implemented in different ways but usually consists of two different classifiers, which each optimise different tasks. The first classifier that makes up the discriminator aims at predicting the original decision attribute y, thus addressing the utility part of the trade-off. Thus, its predictions are taken into

account in the computation of loss value of the generator. The second classifier tries to predict the sensitive attribute from the produced profile, thus having an opposite objective of the generator objective.

As mentioned in Section 1.1.3, successfully training regular GANs and achieving convergence for the underlying min max game is a notoriously difficult task. The same difficulties have been mentioned by the techniques for improving fairness based on GANs. More specifically, learning a generator that can produce profiles with a good level of variance has proven to be challenging [**6**]. Generators have a tendency to output a *median* or *average* profile, meaning it will consistently output a profile that is somewhere in the centre of the distribution (with *average* values for each feature) and therefore have a reasonably minimised loss value. This problem was identified from the early days of GAN's under the term *Mode Collapse* [**7**]. During our research, we replicated results from [**6**] using a vector of the loss value during gradient descent instead of an average value. This change makes it possible to achieve much higher diversity of the profiles that are created by the generator.

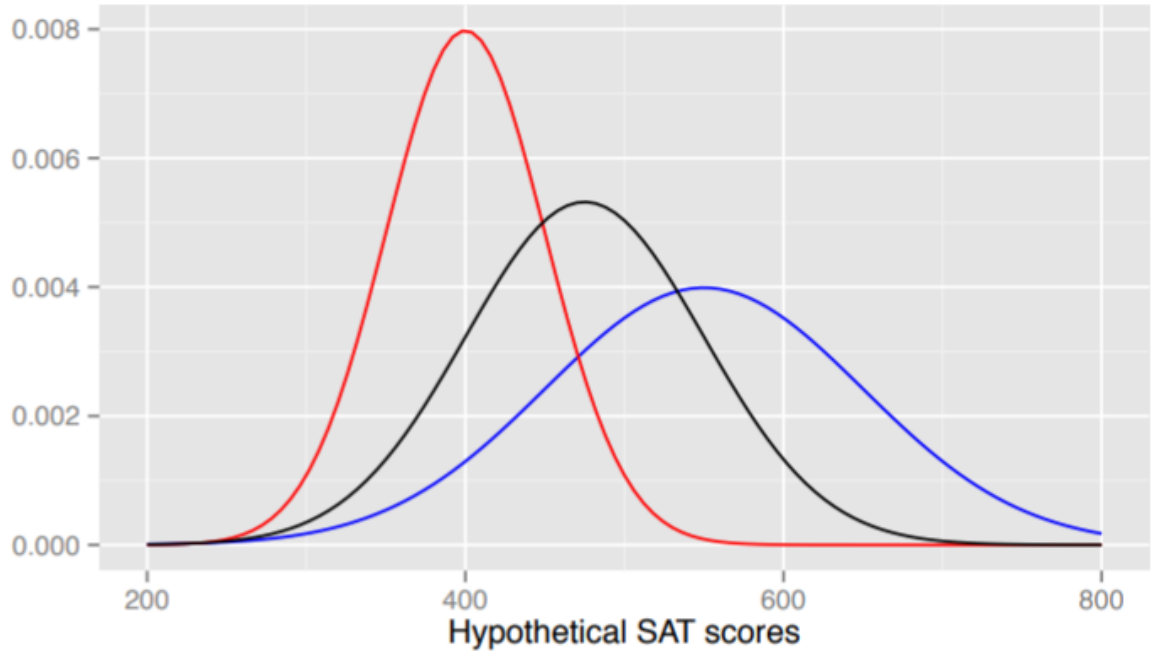

Fig. 2.2. Example of Disparate Impact Remover when applied on synthetic SAT scores (standardized test score used for admission in USA universities). The red curve is the protected group while the blue one is the non-protected one. The black curve is the *Repaired Data* that the method creates [**33**].

2.2.3. Disparate impact remover

Disparate Impact Remover has been developed in 2015 by Feldman and co-authors [**34**]. This method addresses the utility-fairness trade-off by building a deterministic mapping, which essentially transforms the value of all attributes of given profiles. It also offers theoretical guarantees of strongly preserving each attribute ranking with respect to other profiles by making a translation of same size on each attributes' cumulative density functions (CDF). With the CDF and the median of the original distribution, it then applies the translation to a new CDF while keeping the ordering intact. Finally, it uses a logistic regression algorithm. An example with synthetic data taken from the original paper is shown in Figure 2.2. The effectiveness of the disparate impact remover diminishes when it is applied to protect more than one attribute, although this is not something other methods discussed later seem to offer. Although the initial implementation by Feldman worked only with numerical attributes, a more advanced implementation enables them to integrate categorical attributes as well. This first implementation was also included in the IBM's AIF360 Fairness library and it is the one we have used in our experiments (see Chapter 5).

In the original paper, the method was evaluated mostly with the Adult dataset as well as synthetic data created for the purpose of the experimentations. Figure 2.2 illustrates the resulting distribution obtained when applying the method on synthetic data (red and blue lines). In addition, the results for Adult are also encouraging, demonstrating that it is possibly to comply with the 80% rule areas with a BER score of about 0.42 while maintaining an accuracy on the task (*i.e.*, income prediction) of 75%. Furthermore, changing the value of α, seems indeed correlated with the change of disparate impact score, while it does not seem to have a significant impact on the utility as computed by the accuracy of the initial task of income prediction.

2.2.4. Learning Adversarially Fair Representation (LAFTR)

Following the idea of learning a fair representation striking the right bargain between fairness and utility, LAFTR was among the first to realise this using an adversarial learning approach [**67**]. LAFTR builds on McNamara's work outlining that a fairness guarantee can be created by separating data users, regulators and producers, who each work to optimize their respective metric [**68**]. For example, we can think of one entity producing the data, individuals for example, another independent entity regulating the data and making all privacy and fairness enhancement needed (a non-profit organization or a government board), before the last entity (a company for example) uses the data. Similarly, the adversarial network model is trying to replicate this within one learning system. The

closest work to LAFTR, which is also cited as an explicit source of inspiration, is a method proposed by Louizos and collaborators that use variational auto-encoder to learn a new data representation increasing fairness while also maintaining a high level of utility [**64**]. Madras and co-authors have pushed the concept further by pairing this auto-encoder to a discriminator and create one of the first adversarial networks to achieve fair data representation simulating a GAN network.

As mentioned earlier, it might be desirable to optimise different fairness metrics, and in this sense LAFTR is quite flexible. In particular, the discriminator of the model can be designed to optimise one of three following metrics : demographic parity, equalised odds or equal opportunity. In Figure 2.3, x refers to the original input, with an encoder and decoder, just as in Figure 2.1. In LAFTR, the latent layer of the auto-encoder is directly used as the generated data. As the size of the latent layer is smaller than the original profile, the latent representation produced resides in a new data space, which is not directly interpretable by a human. Afterwards, the latent layer is passed through $g(Z)$, which tries to predict the label y and through the discriminator $h(Z)$, which aims at predicting the sensitive attribute A. In this method, the equivalent of α is called γ. Note that only the loss function of $h(Z)$ is included in the auto-encoder to improve the latent representation Z while the classifier $g(Z)$ is trained in parallel. The learning method requires to train the auto-encoder first, before training the classifier $g(Z)$ using the generated data Z with the original labels y. The loss values for the encoder, decoder and discriminator individual terms are then summed up, with γ having a direct impact on the discriminator loss value. The search for an optimal γ proceeds through "sweeping", by training several models using various values of γ to identify the best fairness-utility trade-off [**67**]. Contrary to other methods analysed in this paper, it is not clear that the process of tweaking the model to find the best trade-off follows a structured pattern. We will also see in Chapter 5 that there is not a linear correlation between γ and the amount of protection offered for the sensitive attribute.

The performance was evaluated on Adult dataset and varies slightly depending on the fairness metric optimised. Nonetheless all three implementations demonstrate a clear trend in which LAFTR achieves convergence with respect to the fairness metric optimised with a reasonable impact on the accuracy of prediction for the decision attribute. The biggest change is obtained when optimising the demographic parity. Indeed, starting with a regular deep-learning classifier $g(Z)$ with an accuracy of 85% for a demographic parity score of close to 0.2, the best values from LAFTR gives us a 92% decrease in fairness score, down to around 0.01, with less than 4% drop in accuracy (full results are displayed in Figure 2.4).

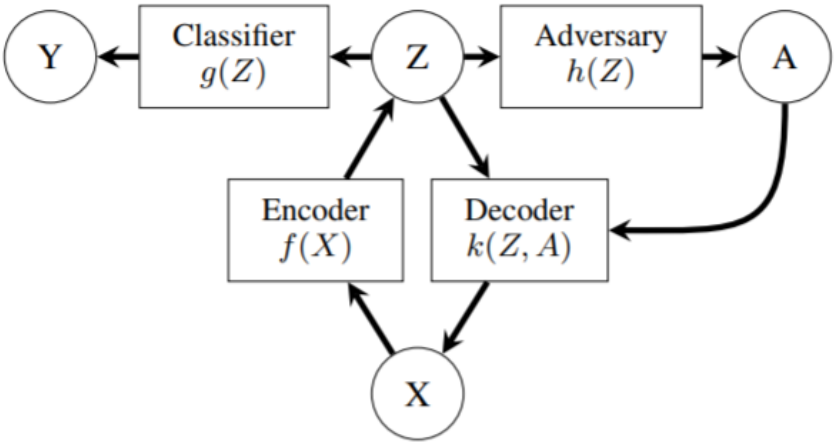

Fig. 2.3. LAFTR architecture as shown in original paper [**67**].
Starting with the real profile X, we train the auto-encoder represented by $f(X)$ and $k(Z,A)$. We then take the new representation Z and use the discriminator $h(Z)$ to ensure fairness towards attribute A. Finally, another classifier tries to predict the label Y to ensure Z still has relevant information about the initial profile X.

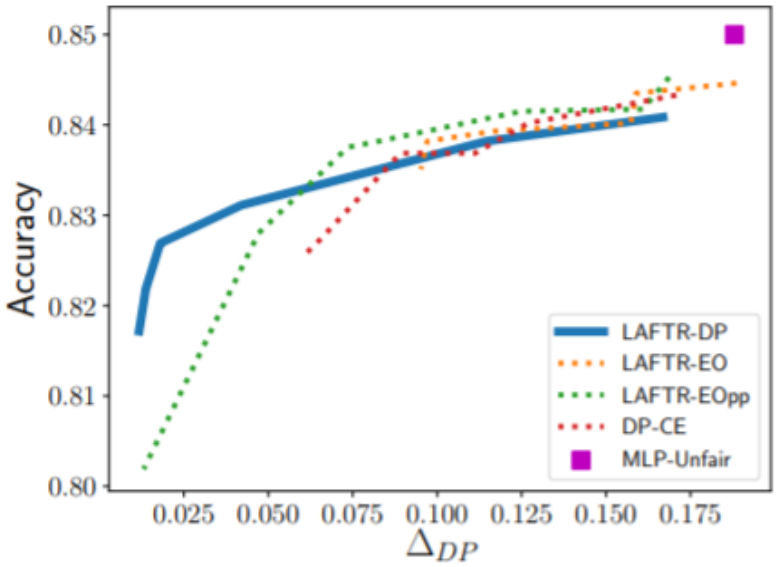

Fig. 2.4. LAFTR Score on demographic parity (ΔDP) optimization [**67**].

2.2.5. Local data debiasing through GAN-based local sanitiser (GANSan)

The final technique analysed as part of this research is named GANSan [**6**]. Referring to the figure accompanying the paper (Figure 2.5), in which the starting point is clearly

indicated, the input profile goes through the *sanitiser*, which outputs a sanitised version of this profile. The sanitiser is also implemented in the form of an auto-encoder but the vector produced by the final output layer output is used as the sanitised profile. As a result, the output resides in the same space, which means that it produces the same attributes and after decoding the values, can easily be compared to the original profile. This output profile is then handed over to the discriminator, which tries to predict the sensitive attribute from it.

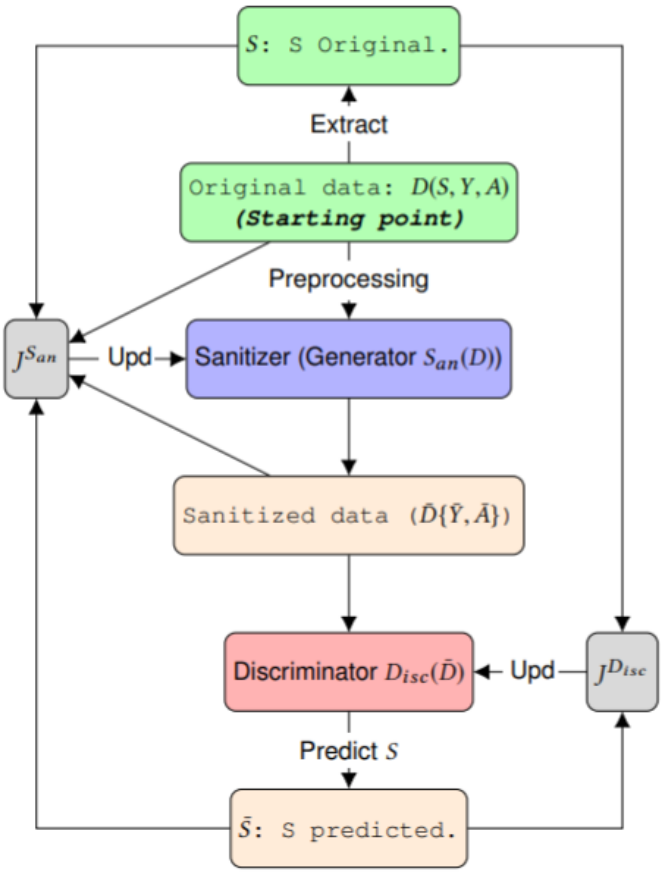

Fig. 2.5. GANSan architecture [**6**].

The sanitiser optimisation measures the distance (in the original paper, the L_2 distance is implemented) between the original profile and the sanitised one, to quantify the utility loss. Similarly to LAFTR, the optimisation also integrates the quality of the prediction of the discriminator with respect to the sensitive attribute. More precisely, the Balanced Error Rate (BER) is used to measure the error rate of each of the output classes. As for the back propagation, a variant of a standard implementation is used. The usual implementation average output loss value for all dimensions, resulting in one value to use in our gradient descent algorithm. GANSan uses the *un-reduced* vector of loss values, meaning it does not

compute the average, but keep one loss value for each attribute, and applies the gradient descent iteratively over each one. In contrast to LAFTR, the discriminator is trained jointly with the sanitiser, albeit at different rates. This is more in line with the original GAN architectures and results in the discriminator becoming increasingly better at identifying the sensitive attribute from the sanitised profile. The training of the discriminator relies on the Mean Square Error (see Section 1.1.1) between its prediction and the corresponding original sensitive attribute. In addition, the *fidelity* between the original and sanitise profile is used to quantify the utility of the sanitised data (how similar the two profiles are). The α parameter enables to control the fairness-utility trade-off and is shown in the original paper to be linearly correlated to a higher fairness and lower utility (*i.e.*, fidelity).

A key feature of this method, as describe in more details in Chapter 5, is the use of additional external classifiers (not shown in Figure 2.5) at each of the algorithm's iteration (also referred to as *epochs*), and for each value of α to identify the ideal fairness-utility trade-off, which would be a BER of 0.5 and a Fidelity value of 1. The results obtained on the Adult dataset demonstrate that GANSan can achieve an optimal BER of 0.5 for a Fidelity rate above 93% but with a relatively high cost in terms of the prediction of the decision y of the initial task at hand (determining whether a profile's income is above or below \$50 000). Reducing slightly the value of α incrementally decreases the BER with an inversely proportional increase in the prediction of decision.

A wealth of other adversarial training algorithms and methods have been developed in recent years that we do not mention here. We picked these three for a few reasons including a trade-off between diversity of methods (one deterministic, two based on deep learning based), comparability (they are all pre-processing methods), common dataset implementation (Adult dataset) as well as availability and reproducible code (for all three we used code available as it is).

Chapter 3

Privacy models and attacks

This chapter presents notions and protection mechanisms regarding privacy that have been introduced in machine learning. First, we introduce the two main privacy models that have been developed and are used by some of the biggest technology companies to mitigate privacy issues in real-world applications [**105**]. Afterwards, we review the most common privacy attacks against machine learning models. Each of these attacks targets a specific type of setting, in which various amounts of information about the model is released by its designers. Finally, we discuss how privacy and fairness are interrelated and have to be considered as a whole when designing nondiscriminatory algorithmic systems.

3.1. Preliminary notions in privacy

The renewed interest in machine learning has also brought new worries with respect to data privacy. Encryption, which can be used to ensure the confidentiality of communications, is fundamental to protect the privacy of data at rest or in transit [**8**]. However, it does not answer all privacy problems, in particular when considering the situations in which data is shared or published (see Section 2.1). Given that a profile might be used to predict other characteristics (*e.g.*, the ZIP code is a good predictor of race) it would be tempting to encrypt all the data. This would prevent anyone from using the data to build predictive algorithms, thus ensuring privacy while sacrificing utility. The concept of privacy in machine learning is intrinsically linked to fairness, particularly when a fairness-enhancing technique attempts to prevent knowledge discovery or usage of a specific attribute to overly influence the prediction. For example, if we want to prevent a banker from using the race as a basis for decision-making (thus achieving fairness), we need to ensure that the race of the customer is well hidden (thus ensuring privacy). Dwork and co-authors have clearly outlined the relationship between those two and the need to create methods to ensure privacy when working on fairness [**26**]. The first step of such process is to make sure that

the attribute in question remains hidden and cannot be retrieved by an adversary. As we will see the increase in sophistication of attacks (Section 3.2) has accelerated the pace at which previous privacy-preserving techniques come to offer insufficient levels of protection, which also brings potential fairness issues.

The public release of medical history data in the United States for research purposes is a great illustration of those concerns. For example, the re-identification of the then governor of Massachusetts' [**102**] demonstrates that simply protecting datasets is not enough. The medical history is often anonymized in the United States by removing names, street address and other personal information. It was believed that the privacy of individuals was guaranteed following this procedure. However, Sweeney has shown in 2000 that she was able to identify 87% of USA residents with only the combination of three attributes (ZIP code, gender and date of birth), all publicly available [**101**]. More precisely, she has built on that research by cross-referencing publicly released health data with another dataset she bought (legally) for $20 (the voter registration list). From this, she was able to demonstrate that it is possible to infer the governor's address, party affiliation and other information (see Figure 3.1).

We can easily see how this could translate in fairness concerns, in which people with certain medical conditions exposed could be excluded from insurance schemes or knowing the ZIP code would allow for discrimination based on race (see Section 2.1). In another medical data-related research, Malin and co-authors have performed a systematic review of re-identification, which is discussed in Section 3.2.2, in which they found a potential gap between the anonymisation mechanisms mandatory for medical data release without the patient's consent and the re-identification methods available where the latter, under some circumstances, tend to outperform the former [**50**]. Those are the type of issues and topics that this chapter covers. In particular, different frameworks for privacy attacks on machine learning models are discussed in Section 3.2.1. However, first we review two of the most important privacy models.

3.1.1. *k*-anonymity

Sweeney did not stop with the re-identification of Massachusetts' governor in her 2002 paper but also proposed a potential solution that has been for some time considered one of the best approaches to anonymize data. To address the increasing concerns of privacy attacks such as linkage, membership inference and information leakage that can occur following multiple queries (see Section 3.2 for more details), she introduced the k-anonymity

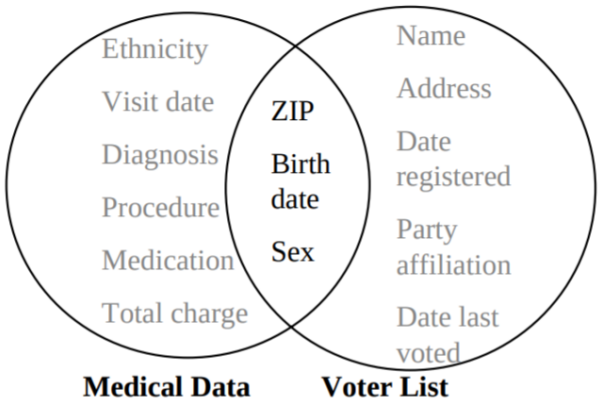

Fig. 3.1. Example of a linkage attack between two datasets, which relies on quasi-identifiers to associate an identity to an "anonymized profile" as shown by Latanya Sweeney [**102**].

privacy model. The first step when applying her approach is to identify the attributes that can lead to a privacy breach. Building on her previous work [**101**], she defined these attributes quite broadly to refer to any attribute that could be used in a linkage attack. Linkage attacks can be considered to be part of a greater category called Re-Identification (or de-anonymisation) attacks. She refers to such attributes as *quasi-identifiers* (QID), meaning they can be used to identify anonymized attributes by cross-referencing datasets.

As we can already see one limit of this method is that the QID selection depends on the knowledge of public data available to create successful linkage attack. In practice, malicious actors rely on data that is not always known to be available for such purposes. Sweeney has also defined the notion of sensitive attributes (different from the sensitive attribute we saw in the context of fairness) as attributes that will not be considered to be QID and will be released to the public (see Figure 3.2) resulting in fairness issues (see Section 2.1). Once the QIDs are identified, the anonymisation process modifies the entries of those attributes through generalisation and suppression to achieve k-anonymity. The main objective of k-anonymity is to guarantee that any record in a sanitised database will be indistinguishable from at least $k-1$ other entries. Note that they are usually multiple ways to reach k-anonymity (see Figure 3.2), but that in order to preserve utility the records should be changed as little as possible. With respect to the computational costs of this method, finding the optimal procedure that would change the minimal number of records has been proven to be an NP-hard problem [**71**]. Although most algorithms for k-anonymity can be considered as heuristics that efficiently find an acceptable solution, it is important to keep in mind the trade-off with utility discussed in Section 2.1. In this setting, the trade-off reappears albeit being bounded by computing costs; finding the method (through

heuristics) to achieve the desired level of privacy while minimize the decrease in utility.

Race	BirthDate	Gender	ZIP	Problem
black	1965	male	02141	short of breath
black	1965	male	02141	chest pain
person	1965	female	0213*	painful eye
person	1965	female	0213*	wheezing
black	1964	female	02138	obesity
black	1964	female	02138	chest pain
white	1964	male	0213*	short of breath
person	1965	female	0213*	hypertension
white	1964	male	0213*	obesity
white	1964	male	0213*	fever
white	1967	male	02138	vomiting
white	1967	male	02138	back pain

Race	BirthDate	Gender	ZIP	Problem
black	1965	male	02141	short of breath
black	1965	male	02141	chest pain
black	1965	female	02138	painful eye
black	1965	female	02138	wheezing
black	1964	female	02138	obesity
black	1964	female	02138	chest pain
white	1960-69	male	02138	short of breath
white	1960-69	human	02139	hypertension
white	1960-69	human	02139	obesity
white	1960-69	human	02139	fever
white	1960-69	male	02138	vomiting
white	1960-69	male	02138	back pain

Fig. 3.2. Two versions of k-anonymized dataset in which $k = 2$.
An example of a complementary release attack as the two versions can be linked on "problem" attribute [**102**].

Sweeney herself has outlined the following limitations of her approach. First, if the data entries are not randomly shuffled (*e.g.*, if they are ordered alphabetically by name or chronologically by age), this could be a potential issue since an attacker could rely on the ordering to assume data that had been transformed. For example, a k-anonymized database release that heavily relied on changing the date of birth would be problematic if entries were ordered by age. Second, multiple releases of a same database with different k-anonymity schemes could allow privacy attacks using those different releases, known as Complementary Release Attack (see Figure 3.2). This is created by the possible identification of unique entries in k-anonymized databases released at different times. By identifying unique entries, an attacker can then have additional information by combining both database releases. Finally, Temporal Attacks occur when adding, deleting or removing entries to a database and rerunning the k-anonymity algorithm. Under such an attack, it is possible that some records will be re-identified. In this type of attack, an attacker will use multiple k-anonymized datasets that are released with different anonymisation. Then, by merging the different data that was released in each table, the attacker could gain additional insights leading to privacy issues.

A few alternatives have been proposed to improve on Sweeney's initial privacy model. For instance, ℓ-diversity aims at preventing the low occurrence of some attributes by requiring each k-anonymous group have at least ℓ occurrences of a sensitive value [**65**]. t-proximity extends this approach by ensuring that each k-anonymous group has a sensitive attribute distribution that is t-close to the overall attribute distribution [**63**]. The two main distance metrics used to calculate t-proximity are the variational distance (Equation 3.1.2)

and the Kullback-Leibler distance (Equation 3.1.1).

$$\mathrm{KL}(P \parallel Q) = -\sum_{x \in \mathcal{X}} p(x) \log \left(\frac{Q(x)}{p(x)} \right) \tag{3.1.1}$$

$$D[\mathbf{P}, \mathbf{Q}] = \sum_{i=1}^{m} \frac{1}{2} |p_i - q_i| \tag{3.1.2}$$

3.1.2. Differential privacy

Another privacy model designed to safely answer database queries that has gained wide interest since the last decade is Differential Privacy (DP) [**22, 25**]. This privacy model was originally developed to offer stronger privacy guarantees than k-anonymity. While it was only popular in the research community at its beginning, it has since been implemented by Apple [**114**] and Google [**105**] for preserving the privacy in various applications. The main idea behind the privacy model is to prevent any inference that could be done with respect to a particular record by ensuring that its contribution to a query or computation has only a limited impact. Without going in the details, DP is usually achieved by the addition of random noise or through randomisation of the underlying computation. Once again the objective is to limit the inference that an adversary could perform with respect to a specific individual, while being able to discover or exploit global statistical information about the dataset.

The level of protection is parametrised by the privacy parameter ϵ [**8**]. The smaller the value of ϵ, the more protected the data is. Note that DP can also be implemented at the local level, in which the choice of ϵ is made by an individual or globally in which the choice of ϵ is the same for all individuals. This last model was designed for use cases in which the database administrator would provide answers to queries following a DP algorithm, hence users could leverage the data as they pleased [**102**] given the data they receive is already anonymized. There are inherent limitations of differential privacy, some of them that could apply to any privacy model. For instance, the publication of differentially private results of a study finding a relationship between beer drinking and happiness inherently leaks information about happiness of beer drinkers, regardless of one's inclusion in the study dataset. In addition, developing DP algorithms for a particular task, finding an optimal value of ϵ and then implementing the algorithm have proven to be difficult [**40**]. In addition even if a DP algorithm was successfully implemented, for example at Apple [**114**] using

local-DP, the choice of ϵ, which sets the trade-off between privacy and utility of data, has shown to be a challenge. For instance, some research has demonstrated that Apple has used values of ϵ that are too high to provide useful privacy guarantees [**103**].

3.2. Families of privacy attacks

This section discusses common attack frameworks that adversaries could apply on released models or data to jeopardize users' privacy. Before that, we review the main dimensions influencing the attack framework.

The first dimension is whether the attack takes place in a white-box or black-box setting. A white-box setting implies that the adversary has some knowledge about the model, such as the number and width of layers as well as other hyper-parameters. A common use case is when a company sells a pre-trained deep-learning model, which the buyer either uses as it is or further trains with his own data. The black-box approach is, as the name indicates, more restrictive. It assumes no knowledge of the underlying model and the adversary can only query the model with an input of his choice and receive the corresponding output. Note that several types of outputs are possible. Indeed while the output is usually a vector composed of as many values as there are classes, each with a confidence score, the model could also only return the most probable class, with the latter usually making the task of the adversary more challenging. Such use cases include the MLaaS (Machine Learning as a Service) paradigm, in which a client usually interacts with the service provider through pay-per-request application programming interface (API) systems. Releasing these systems is known to potentially leak information about the training data (see Section 3.2.3), and commercial value is among the reasons why the model's inner workings are unknown to the public and users.

A second dimension in segmenting the different approaches to privacy attacks has been with respect to what is the target of the discriminator [**113**]. In particular, attacks optimising a specific privacy metric such as the ones we saw in Section 3.1.2 often led to improvement and development of those techniques by somehow raising the bar in terms of how well data is protected. On the other hand, attacks targeting specific attributes aim at finding the minimal amount of perturbation needed to offer a certain level of privacy for these particular attributes.

3.2.1. Linking attack

We already covered a linking attack in our example with the Governor of Massachusetts (Figure 3.1) without naming it. This attack generally uses an external database sharing a subset of attributes with the data attacked. Another example of such de-anonymization attacks was shown by Narayanan and Schmatikov [**4**] in which they were able to re-identify individuals from the Netflix's allegedly anonymized movie rating dataset by using Internet Movie Database as an auxiliary dataset. It has since been reproduced using publicly available Amazon reviews as auxiliary information for the linkage attack [**2**].

Genomics is another field in which there is a strong tension between sharing data for research purposes and privacy [**9**] as shown by linkage attacks. This is also true in healthcare in general in which data sharing is a core principle and existing privacy-preserving mechanisms are often insufficient [**30**]. A review of the literature has found that since 2009, 72% of successful re-identification attacks have been completed through linkage using global datasets such as social medias [**50**].

3.2.2. Model inversion attack

The second attack we discuss is the *Model Inversion Attack*. This attack usually occurs in a black-box setting and leverages the knowledge of the model to infer additional attributes on specific profiles. It was initially proposed in genomics by Fredrikson and co-authors [**36**], before being later formalized [**113**]. In model inversion attack, the adversary exploit the confidence intervals released with the prediction of a model to draw conclusions about relationships between inputs and outputs of the model. A possible way to implement a model inversion attack is by changing the value for the attacked attribute to look at which value maximises the posterior probability of the sensitive attribute belonging to a specific input $p(x_s|x_0,x_1..x_d,y)$. Repeating this process multiple times using inputs for which the output is known will allow an attacker to have an increasingly accurate idea of what inputs achieve what output and possibly predict sensitive attributes. Unless optimal DP (with the minimal amount of data transformation) is implemented (which we saw in Section 3.1.2 is computationally costly), the success of the attacks usually increases with the augmentation of ϵ [**36**].

A possible use case for such attack is the service offered by most major cloud providers, in which an API is available to send input to, and where the service returns a prediction with a confidence score, known as the MLaaS we covered in Section 3.2. Another use case developed with a new version of the attack also by Fredrikson and collaborators [**35**] involved a facial recognition task, in which the confidence score can be used to mount a

model inversion attack whose results are shown qualitatively in Figure 3.3. This new version also avoids the computationally expensive calculation involved with finding all possible combinations of the attacked attribute through posterior probability analysis. It has further been shown that the confidence score is in some case not even required for the attack to be successful [**107**]. Note that this attack does not provide any guarantee that the queried input was used in the training of the target model, something that we will discuss in the following section.

Fig. 3.3. The image on left was generated by a model inversion attack while the image on the right is the original one [**35**].

Recent research also proposes to rely on Generative Adversarial Networks (GANs) to conduct an attack in the black-box setting. For instance, GAMIN [**1**] consists of a generator network that maps noise z to an output x while a surrogate model takes x as input and tries to predict what this same x's output would be when passed through the target model. The generator and surrogate are trained jointly using a cross-entropy loss computed not only from the prediction of x from both the surrogate and true model, but also using the target model's prediction of the noise z. The result is a surrogate model that has learned the decision boundaries of the target model and can approximate it on any x. As the experiments were conducted on pictures, the evaluation of whether the attack is successful had to be done by qualitative human appraisal. Surveys were therefore passed in order to ask people as to whether they could recognise the image as the class it belongs to. In this case, the images generate were black and white digits from 0 to 9, trained on the MNIST dataset discussed in Section 1.2. The results varied with which class (*i.e.*, which digit) but people had more difficulty recognising images generated from deeper networks such as CNN's [**1**].

3.2.3. Membership inference attack

In a membership inference attack, the attacker attempts to determine whether a specific input has been previously used in the training of a black-box model. This can create privacy concerns for instance in the case of hospital discharge data, which was used in a major paper showing such attacks on Google's and Amazon's MLaaS services [**96**]. More precisely, the authors have first set up a black-box membership attack, in which the problem is transformed into a simple binary classification task, while the classifier is trained to distinguish between data used in training for the target model and data that was not [**96**]. In parallel, a *shadow model* is trained to replicate the predictions of the target model. More precisely, first multiple shadow models are trained to imitate the target model either synthetic or any real-world data in which the true labels are known. Afterwards, the main attack model is trained by using a combination of input-output to recognise whether they were used in the shadow models.

Once trained, this attack model was able to achieving 90% accuracy in membership inference attack on the Google's MLaaS model by using only synthetic data for shadow model training. It also reached over 70% accuracy on the hospital discharge data confirming privacy concerns and, as we now know, potential fairness issues. Note that to know whether a profile is in training data or not with an accuracy above 50% can prove to be problematic. A key information needed for this attack to perform well was the knowledge of the confidence score for each class that those MLaaS service returns through their API. The authors therefore outlined that removing the confidence score of the k least probable classes was the best mitigating strategy to prevent their attacks from being successful with the reduction in the overfitting of the model being another possible approach [**96**].

Membership attacks have since increased in sophistication, for example showing that differentially-private deep learning models do not protect against membership attacks unless the value of ϵ is so low that the data loses most of its utility [**85**]. Another work has attacked state-of-the-art generative models by training a shadow GAN with the data generated by the target model [**49**]. Once that network is trained, the problem is now in the white-box setting, and from there querying the discriminator of the GAN has shown to have learned the distribution of the training data, hence becoming a binary classifier that predicts whether a specific data point was used in training [**49**].

3.2.4. Model stealing attack

The next type of attack covered is the model stealing attack. As mentioned earlier, many firms are reluctant to release their models entirely for various reasons. For instance,

with respect to privacy as we just discussed, once we have a copy of the target model, we fall in a white-box setting in which it becomes easier to predict membership or to make multiple inferences about the sensitive attributes. These attacks have been shown to be extremely straightforward and simple, not only against basic learning algorithms such as decision trees and SVMs, but also against deep neural networks and state-of-the-art models offered by Google and Amazon through their API.

Tramer and collaborators have used the same output from these services as Fredrikson and co-authors [**113**] for their model inversion attack in which the output is a vector of confidence score for each class predicted by the classifier. More precisely, their attack compares the shadow model's prediction and the one given by the API. In their experiments, a shadow model as simple as a linear regression allowed to reach near-perfect score for binary classification tasks with fewer than 113 queries. For multi-classes, their attack [**107**] achieved 100% equivalent models against Google and Amazon within minutes and a few thousand API queries (see Figure 3.4). The results were similar even when the depth of the neural network increases which intuitively leads to more difficult learning from shadow models.

Service	Model Type	Data set	Queries	Time (s)
Amazon	Logistic Regression	Digits	650	70
	Logistic Regression	Adult	1,485	149
BigML	Decision Tree	German Credit	1,150	631
	Decision Tree	Steak Survey	4,013	2,088

Fig. 3.4. Metrics of successful attacks of various multi-class models that were able to extract a 99% equivalent model [**107**] computed by comparing similarity in each model's output.

As mentioned previously, a possible counter-measure is the removal of the confidence score from the prediction API output. In this case, attacks were equally successful with the only caveat that it required more queries to achieve the same performance. The results of this research highlight the simple, quick, and relatively cheap (in actual dollars) cost of this model stealing attack, which could be used as a first step to facilitate the other attack discussed previously.

3.2.5. Reconstruction attack

Lastly, the concept of reconstruction attack was introduced by Kasiviswanathan and co-authors [**59**] in which they applied *linear reconstruction attacks* developed in previous

work by Dwork and co-authors [**29, 23**] to non-linear contexts. Those attacks, as applied by [**59**] are meant to find a more robust lower bound of noise or disruption to original data to ensure privacy, adding formal guarantees to methods discussed in Sections 3.1.1 and 3.1.2. The objective of the attack is to reverse the noise or distortion that was introduced in databases, which would imply the protection has been removed. The results show that with simple machine learning algorithms such as logistic regressions and decision trees, such attacks were surprisingly successful [**59**], fuelling a development in the area.

3.3. Insufficient privacy leading to lack of fairness

A common theme from most attacks is that their efficiency is increasing with the amount of overfitting [**117**], which translates in higher confidence scores. As a consequence, those high confidence scores are providing more information to attackers. The silver lining that this creates is a rare common ground between privacy advocates and algorithms designers endless quest for better models, since as we saw, overfitting is undesirable as it increases variance and offers a less generalisable model to use in the real world. As discussed previously, removing or adding noise to these confidence scores is not always sufficient to prevent all attacks, especially model stealing attacks, which can facilitate many other attacks by leading to the white-box setting.

Another observation that becomes clear after looking at the state-of-the-art on privacy attacks and data leakage frameworks is how privacy and fairness are intricately linked. This relationship is an interdependence since offering guarantees on one almost automatically requires offering some level of protection on the other. However, neither fairness-enhancing nor privacy-enhancing techniques seem to directly offers protection on the other. As discussed a lot of research has focused on one or the other but very few on the interplay between them. We will see in the following chapters that leaving aside privacy concerns when developing fairness-enhancing techniques open the door for extremely simple privacy attacks. Such successful attacks could then endanger the fairness guarantees that the technique was originally designed to provide, which has been the main driver of our contribution in this research.

Chapter 4

Design of privacy attacks against pre-processing methods

The emergence of fairness and privacy concerns in the wake of the current deep learning wave has shown that there can be a multitude of fronts from which to approach the development of fairness-enhancing techniques. In particular not only the choice of the fairness metric, but also the technique used vary widely, which makes the comparison between approaches difficult. In this research, we decided to focus on pre-processing algorithms because they seem to offer the most direct and generic way to avoid discrimination while being agnostic to the form of the input data or the future use of this data. In particular, we envision situations in which a pre-processing approach could occur directly on users' devices before the sharing of the data with a third party. In this scenario, the data producer (*i.e*, the user) would give his profile as input to the pre-processing algorithm, with a desired α value (*i.e.*, the level of protection) that could be chosen by him or decided in advance according to an external criterion. This would contribute to data sovereignty as it would give more control of their data back to their owner, while still enabling to benefit from personalised services.

4.1. Research objectives

While this research field has been burgeoning in recent years, as exemplified by the increased interest in conferences focused on fairness, accountability and transparency [**18**], one possible gap is a comparative analysis of the different existing methods to have a better understanding of the circumstances under which a particular method is more appropriate, as well as the extent to which those techniques are generic. Given the advent of adversarial methods for fairness enhancement described in Section 2.2.2, we chose to investigate two pre-processing methods : one in which the sanitised data remains in the space as the original data (GANSan [**6**]) and the other in which the representation produced lives in

a different space (LAFTR [**67**]). Finally, we chose the third method to be a non-machine learning based approach, in which we could potentially achieve better or at least similar performance with relatively simpler deterministic transformation to the data, instead of more complex and expensive machine learning-based techniques. For this, we picked the Disparate Impact Remover [**34**], a method that has been implemented in IBM's AIF360 library for AI fairness [**53**] and which has been cited many times in the fairness literature.

Our objective is to test the resistance and protection of the sensitive attribute of these methods when facing various attacks on the sanitised data they produced. For reasons outlined in Section 2.1, we do not expect *a priori* these techniques to perform well with respect to other fairness metrics than the one they explicitly optimised.

Hence, the first hypothesis that we were interested to explore is whether or not the parameter α (or its equivalent) does offer a direct control on the fairness-utility trade-off as explained in Section 2.2. Although this might not have been desired by the model creators, as we explained it is highly desirable to allow the user to control the fairness-utility trade-off.

The second part of our experiments is to reconstruct the original profile from the sanitised profile produced by these fairness-enhancing methods, which is undesirable both from a privacy and fairness point of view since it would allow attackers to return to the initial situation, before the protection. In particular, as two out of three methods that we have analysed the sanitised profile is generated through an auto-encoder, one of our hypothesis was that it may be possible with a properly tuned auto-encoder to reverse the process using similar learning methods than those that were used to transform it initially. The objective there would be to learn an inverse transformation that would take the sanitised profile as input and produced as output a version of the profile that is closer to the original, the ideal case being that this profile is exactly identical to the original profile including the sensitive attribute. For the third method, an even simpler approach should be able to reverse the transformation, although tested it with a model based on reconstruction attacks seen in Section 3.2.5. We called this part of our model architecture a *reconstructor*, for the sake of comparison.

Finally, we looked at how well the sensitive attribute was protected with respect to correlations with other attributes. As explained in Chapter 3, privacy with respect to the sensitive attribute is required, in our context to guarantee fairness. To assess this, we have built standard machine learning classifiers taking as input $\hat{x}$, the sanitised data without the sensitive attribute or label y, which attempts to predict the original sensitive attribute. We refer in particular to the Scenario 2 from Table 4.1 in which our classifiers (here $f(x)$) are

searching for a function that maximises the prediction accuracy of the original sensitive value by taking as input the sanitised data (see Equation 4.1.1). Throughout this section, when referring to the average of our set of machine learning classifiers, unless otherwise indicated, this will include a Multi-Layered Perceptron (MLP), SVM, Bagging, Gradient Boosting and decision tree (CART).

$$E[s] = f(\hat{x}) \tag{4.1.1}$$

Within the context of this attack, a successful classifier would remove a lot of guarantees in terms of fairness that the method should provide, since it would make it possible to reconstruct the original value of the sensitive attribute, thus potentially allowing for direct or indirect discrimination. This attack is similar in spirit to the linear reconstruction attacks developed against differential privacy [**59**], our objective being to set a lower bound on the amount of privacy the fairness-enhancing technique offers. This is a concrete example of the relationship between privacy and fairness, as a lack of the former leads to concerns about the latter.

4.2. Experimental setting

Hereafter, we explain in more details the setting in which our experiments take place, including the information available to a potential adversary. This will help us to understand under which circumstances such experiments could be replicated by a malicious actor. This analysis will mainly be performed using the Adult dataset while the Compas dataset will be used for Disparate Impact Remover and GANSan to validate the results obtained on Adult.

We have so far discussed different variants of the same data: original, sanitised and reconstructed. To be precise, the experiments mostly involve using data that has previously been sanitised, while keeping the original values for the sensitive and decision attributes. This can be seen as the most likely to occur in a future where fairness-enhancing methods are being used. Only the transformed profiled is known to the public, but we can assume that the associated label $\hat{y}$ is close enough to the real label y, since that would be the objective to the algorithm designer. Moreover, scenario 2 corresponds to a situation in which an adversary has used (possibly as a black-box) the publicly available pre-processing approach under attack to generate a training dataset. More precisely, the adversary receives as input the pre-processing method as well as the user profiles for which he knows the true label y as well as the sensitive attribute, obtaining values for x, s, y and $\hat{x}$.

Scenario	Train set composition		Test set composition	
	A	Y	A	Y
Baseline	Original	Original	Original	Original
Scenario 1	Sanitized	Sanitized	Sanitized	Sanitized
Scenario 2	Sanitized	Original	Sanitized	Original
Scenario 3	Sanitized	Sanitized	Original	Original
Scenario 4	Original	Original	Sanitized	Original

Table 4.1. Different possible use cases for using fairness-enhancing methods [**6**] in which A is the input profile and Y the label to predict. In this research, we have adopted scenario 2.

With this information in hand, the objective of the discriminator is to predict the sensitive attribute from the sanitised input by maximizing the conditional probability as defined in Equation 4.2.1.

$$\hat{s} = argmax(p(s|\hat{x})) \tag{4.2.1}$$

An important point to reiterate is that unlike LAFTR and GANSan, the data generated by Disparate Impact Remover contains only a subset of the attributes (columns) as the implementation currently available in IBM's AIF360 library is based on the initial version of the method [**34**]. An extended version was released later, which is able to deal both with categorical and numerical attributes. However the difficulty to reproduce previous reported results with this version made us more confident to work with the initial version. The attributes that the different methods are able to handle are summarised in Table 4.2. Hereafter, we summarise the fairness result that we were able to obtain with the three methods that we have analysed.

Finally, once we had generated the sanitised data with the three different methods, we had to make sure that the results obtained were in agreement with what was reported in their associated papers. As mentioned in the introduction, we only expect the data to perform similarly well to what was outlined by its authors on the specific metric it was designed to optimise. Table 4.3 displays the results for the optimal value of α value. We see that all three methods bring the values of the optimised metric within the acceptable range discussed in Section 2.1 which is above 0.8 for disparate impact, below 0.2 for demographic parity, and close to 0.5 for the balanced error rate.

	Disp. Impact	LAFTR	GANsan
age	x		x
fnlwgt	x		x
education-num	x		x
capital-gain	x		x
capital-loss	x		x
hours-per-week	x		x
Workclass			x
Marital-Status			x
Occupation			x
Relationship			x
Native Country			x
Total Dimension	6	6	11

Table 4.2. Attributes included in the sanitised data generated by each of the three methods. Note that LAFTR generates a sanitised profile that lives in a new space, and thus all 6 attributes can be considered "*new*" and are not interpretable like age, education and others.

Methods	Metric	Original Value	Value	α value
Disp. Impact Remover	Disparate Impact	0.6964	0.9331	1.0
LAFTR	Weighted Demo. Parity	0.3709	0.0931	0.1
GANsan	Balanced Error Rate	0.1431	0.4830	0.9875

Table 4.3. Metrics on data generated and used in this research and its corresponding α value.

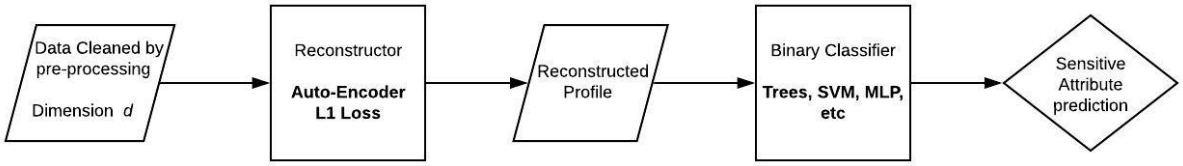

Fig. 4.1. High-level overview of the reconstructor attack.

4.3. Architecture of the reconstructor

Before proceeding to the experiments, in this section we detail the architecture implemented for the reconstruction attack. This attack takes place in the black-box setting, assuming no prior knowledge of the models behind the methods under attack. The architecture of the reconstructor model (*i.e.*, effectively an auto-encoder) is shown in Figure 4.1. The implemented loss used for the training is the absolute distance error (also called the L_1 error) shown in Equation 4.3.1.

$$\text{L1 Loss} = \sum_{i=1}^{n} \left| x_{\text{i}} - x^{\text{i}} \right| \tag{4.3.1}$$

The reconstructor takes a sanitised profile x as input and its objective is to learn a transformation outputting a profile $\hat{x}$, which ideally should be more similar to the original profile (our labels y) than the sanitised profile. For each attribute, the loss function computes the absolute difference between the predicted value $\hat{x}_i$ and the true value x_i (the definition of L1 Loss). As mentioned in Section 2.2.2 and illustrated in Figure 2.5, using the common approach of taking the average value of the loss over all attributes for the gradient descent results in a very low diversity of outputs and often converges to a median profile. In our case, every single value outputted was always the same one, irrespective of the input, which causes issues for the prediction task later on. If all profiles are the same, how can a bank decide to whom to lend. However, the solution taken from Section 2.2.2 has also been successful in increasing the diversity of output. As previously mentioned, this solution keeps the multiple values and applies the gradient descent iteratively for each value. Another architecture decision was to generate a profile that does not contain the sensitive attribute and to predict it afterwards with another classifier instead of having the auto-encoder outputting the attribute directly (although we experimented with both versions). Intuitively these two approaches should give similar results and the assumption taken was that one fewer attribute to predict would be easier for the reconstructor. In addition, it is possible that the mutual information contained in other attributes of the reconstructed profile might give additional information to the external classifier trying to predict the sensitive attribute. It is important to note that it could also be argued that during the reconstruction phase, the information that passes through the auto-encoder might facilitate the reconstruction of the sensitive attribute right away, but we did not explore this possibility further.

Once the reconstructor has completed its training and selected the epoch in which the average L_1 loss over all attributes was minimised, then the five type of classifiers were used with 10 k-fold cross-validation. This means the same training is run 10 times, by splitting the train and test set differently each time, in order to reduce statistical noise. Additionally, the attack does not require to encode the categorical attributes or other types of attributes, since the output of the three methods provides ready to use encoded data. Finally, we chose the Adam optimizer, a variant of the regular gradient descent that adjusts the rate of learning in the model's parameter instead of keeping the same one throughout [**12**], which is a common choice for many classification tasks on tabular data such as Adult dataset.

These choices were made to ensure the success of the reconstruction attack of the profiles. First, it needs to ensure the data coming out of our reconstructor should be on average more

similar to the original data than the data that came from the models attacked, meaning our model would take a transformed profile as input, and output a version that is more similar to its original, pre-transformed version. Second, it needs to increase the amount of information leaked about the sensitive attribute by making it easier to predict it. To summarise, increasing the risk of discrimination by being able to infer the hidden sensitive attribute is the end goal of the adversary.

Chapter 5

Experimental evaluation of privacy attacks

As mentioned, our experiments can be separated into three distinct, although related sub-experiments. First we look at whether or not, and if so to what extent, the parameter we referred to in Section 2.2 as α offers a direct control on the fairness-utility trade-off. Secondly, we attempt to inverse the transformation to the original profile that these 3 methods made with the help of our custom-built reconstructor. Lastly, we evaluate the level of protection these methods produce and whether they effectively protect the sensitive attribute (here the person's gender) from external attacks. We use various external classifiers and train them to try to infer this attribute using the transformed profile as input.

5.1. Control on level of protection with α

Before testing extensively how well the three fairness-enhancement methods are able to prevent the inference of the sensitive attribute, our first analysis consists in measuring how well the parameter α enables to control the fairness-utility trade-off. To realise this, we have observed the evolution of both the accuracy with respect to the initial task on the Adult dataset (*i.e.*, predicting the income) and the fairness metric optimised. Figure 5.1 summarises the results obtained for those two metrics on the y-axis, with the corresponding values of α on the x-axis. Note that the direction of the correlation is not as important in our context as observing the general trend resulting from the change in α. Indeed, the analysis of the whether the discrimination observed is beneficial or detrimental to the people is beyond the scope of our research. We simply want to establish the presence and extent of the discrimination. From the bottom row, we can see that both disparate impact remover (first column) and GANSan (third column) display a clear trend in which an increase in α leads to a better protection. However, such trend is not observed for LAFTR. The red horizontal bar represents for the leftmost and centre graph the level above which each fairness metric value is considered to offer an acceptable level of protection. For the rightmost graph,

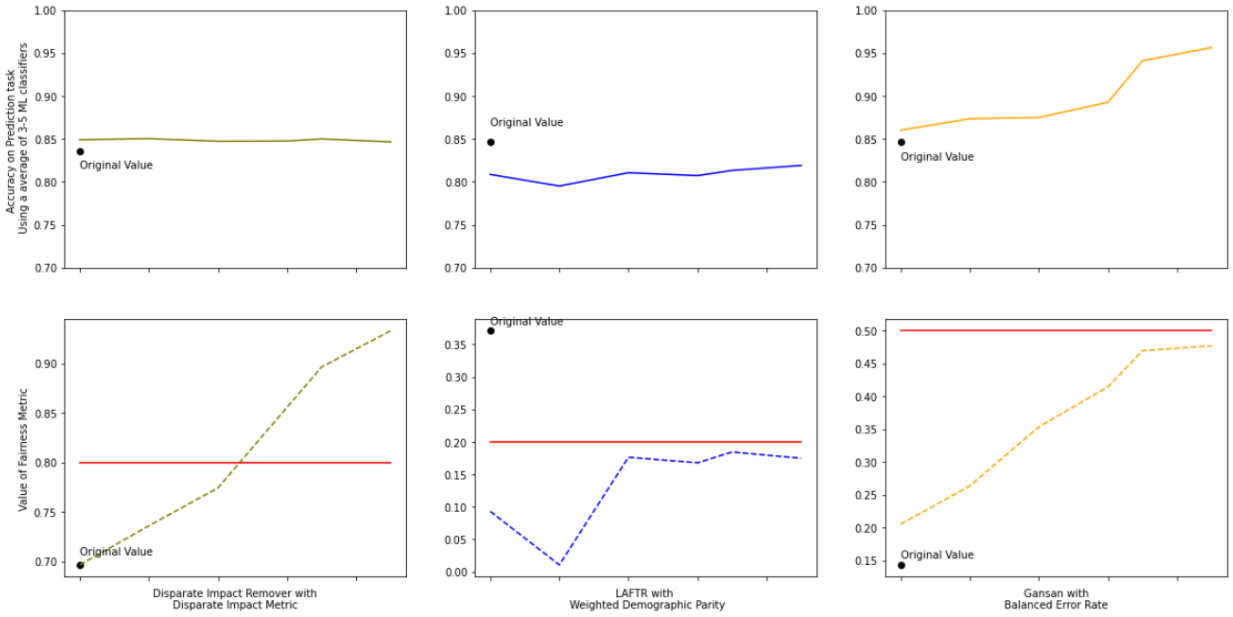

Fig. 5.1. For each method, the impact of the change of value of α is assessed.
Top Row: Accuracy on classification task (*i.e.*, predicting income)
Bottom Row: Fairness Metric
Note: GANSan results are the averaged results of Gradient Boosting, MLP and SVM.

we consider that the closest to the red bar (representing 0.50) results are, the fairer the data is.

The second step of the analysis is to look at the top row of Figure 5.1. An ideal scenario discussed in Section 2.1 would be that the accuracy of the classification task does not decline too much as the fairness metric improves. In our experiments, this occurs only for Disparate Impact Remover, which is not surprising given the minimal distortion it creates on the data as we saw earlier in Figure 5.4 from Section 5.2. Thus, we can conclude that Disparate Impact Remover is particularly efficient in changing the data as little as possible to improve the metric it is optimising. The results for LAFTR are less clear as the fairness evolution seems to be rather unstable. The results obtained for GANSan are the most surprising and have led to multiple reviews of our code implementation and results. In particular, the accuracy is constantly improving with the increase of the value of α, which also leads to an improvement of the fairness metric. This would signify that the data is at the same time fairer and better suited for the initial classification task, which is a win-win situation. While the in-depth study of this phenomenon observed with GANSan is beyond the scope of our research, a possible explanation could be that the training mechanism of GANSan has learned to distil the information about the label y in each attribute.

When looking at our second dataset, Compas, we can see the limits of using only one technique as well as the importance of choosing carefully the metric to optimise.

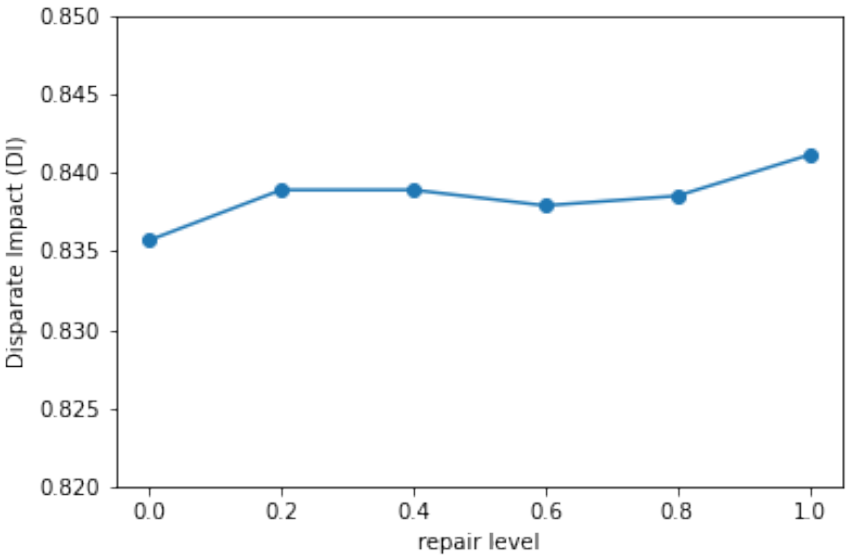

Fig. 5.2. Disparate Impact obtained for various values of the repair level α on the Compas data for Disparate Impact remover.

The original disparate impact value of the Compas dataset is 0.8371, which is within the acceptable range as discussed in Chapter 2. Using only numerical attributes, which are the only ones that the IBM-AIF360 implementation can handle, we see that Disparate Impact Remover barely changes the data with a 0.11% improvement of disparate impact metric with a maximised α value (see Figure 5.2). This is not surprising as the process quantifying the extent to which the data need to be transformed is directly based upon the metric's initial value. Given the known bias in the Compas dataset (see Section 2.2), it seems clear that relying only on one metric prevents from seeing the complete picture in some cases. Surprisingly, we still see the increase in prediction accuracy for the initial task, with values upwards of 84% accuracy to predict y, the two-year recidivism rate. This seems to indicate that the Disparate Impact Remover in this specific case, although minimally changing the data, helps to achieve a higher prediction accuracy, which can be interesting not for fairness research but more basic research techniques.

5.2. Reconstruction of the sanitised profile

The second step in our experiments is to attempt to reconstruct the original profile, as well as the sensitive attribute, using the sanitised profile outputted by each fairness-enhancement techniques, which is composed of the attributes mentioned in Table 4.2. The design choices in terms of architectures and hyper-parameters (*e.g.*, batch size, learning rate, and number of layers to the auto-encoder) were made by iteratively trying

out different combinations using a grid-search approach. In practice, the batch size has little impact on the quality of reconstruction. However, a faster convergence is observed with a bigger batch size, something generally expected in machine learning. The total number of layers leads to better results when set to only one hidden layer on each size of the latent layer, which can be explained by the small amount of training data available. As mentioned in Section 1.1.1 to work with models with higher capacity, more data needs to be available to be trained on. Finally, other things being equal, the learning rate has an impact similar to the batch size. More precisely, a small learning rate simply slowed down the convergence rate (*i.e.*, the same loss would be achieved but in more epochs).

With respect to the attacks on GANSan, the training results are shown in Table 5.1 and Figure 5.3, for reconstruction generated with different mix of hyper-parameters and data transformed initially by GANSan algorithm. The curves of reconstruction shows the loss hitting the same local minima at about the same loss values, under various combinations of hyper-parameters. We can also assume from the first column of Figure 5.3 that lower loss values passed this local minimum do not seem likely since the loss tends to shoot upwards. Therefore, although it seemed that we have successfully optimised our model by minimising the L_1 distance on GANSan, which has made the reconstructed profile "more similar" on average, the actual profiles created by our reconstructor were not consistently giving more information about the sensitive feature. In particular, we found out that the best performing model, although being able to reconstruct 61 out of 78 (encoded) attributes of the input, was unable to predict sensitive attributes with better accuracy compared to models with a hyper-parameter mix achieving less well the reconstruction of attributes. The model reconstructs much more easily the continuous attributes (*e.g.*, age, *fnlwgt*, education-num, capital-gain, capital-loss, hours-per-week) compared to categorical attributes (*e.g.*, native-country and marital-status). The lack of diversity was acute for these two attributes, as the model continuously chose the majority value (90% are from the USA in original data) and assigned it to all profiles. At the end, we can say that the original objective of the reconstruction attack, which was to increase the ability to predict the sensitive attribute from the reconstructed profile, has not been conclusive despite the fact that the reconstructed profile is closer to the original profile than the sanitised one.

The reconstruction attempts on data sanitised by Disparate Impact Remover [**33**] yields even less conclusive results, which may be due to the small percentage of values changed, as seen in Table 5.2. Figure 5.4 (top row) further shows that the number of data points modified for 4 of the 5 attributes used as input 4 remains lower than 1000 (out of 35 000 total) regardless of the value of α. Moreover, the average change for the modified data points remains below 6% (second row). In practice, the reconstruction of this data was

Batch Size	Learning Rate	Ave. Loss	Epoch
2048	1e-05	23.809	59
2048	1e-07	23.8078	936
2048	1e-08	23.811	1268

Table 5.1. Learning rate value and impact on lowest average loss (and its epoch) for the training of the reconstructor training. Tested on GANSan for a value of α=0.2 and BER=0.205.

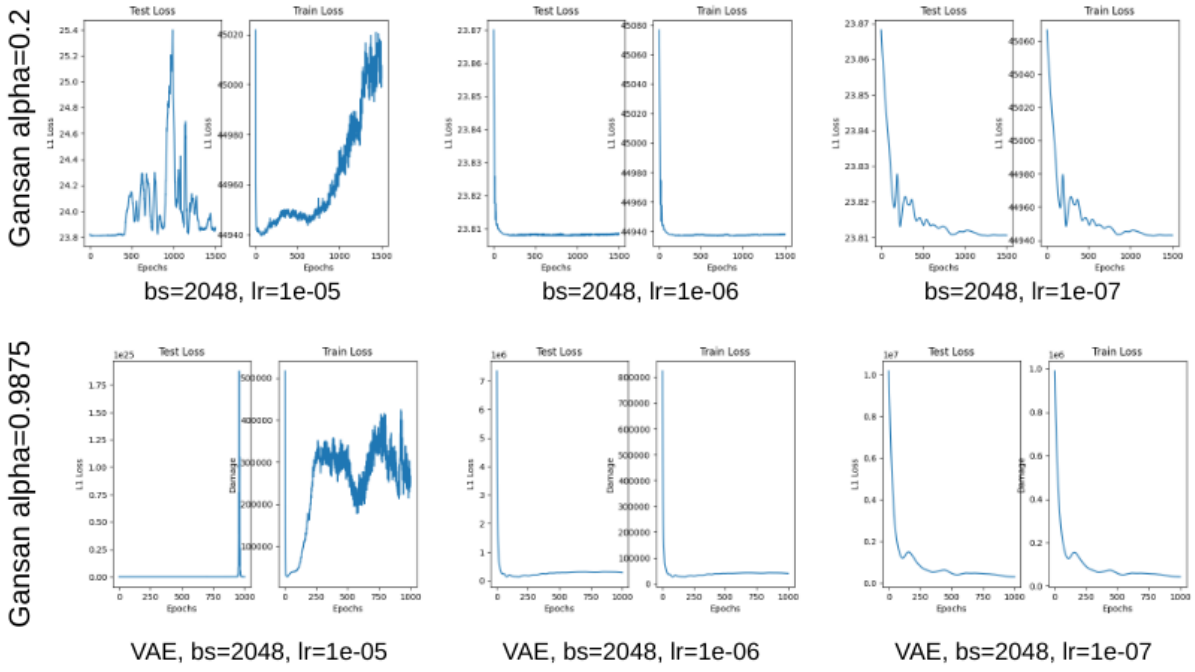

Fig. 5.3. Various plots showing the loss when reconstructing the GANSan data showing the minimal impact of learning rates. In general we are looking for downward slopes which would mean continuous learning and improvement of our machine-learning models.

even less successful than with GANSan. In particular, even though the loss could reach a minimum, none of the attributes were reconstructed closer to their original values while the prediction of the sensitive attribute was also unsuccessful. The easiest explanation for this phenomenon is that the scope for reducing the very small change on the data was so narrow that our model was not able to address it.

Similarly inconclusive results were achieved for LAFTR, although more difficult to interpret due to the new space in which the data resides. In addition, the training method did not allow us to match the original profiles with their sanitised versions, so for a given profile transformed by LAFTR we could not match it with the original profile (which is

Attribute	Number of values changed (32 562)	Ave. Change	Std. Dev. of change
Age	1	0.013699	0.0
Education-Num	0	X	X
Capital-gain	611	0.003798	0.00824
Capital-Loss	1058	0.02144	0.025818
Hours-Per-Week	21 671	0.067362	0.19506

Table 5.2. Statistics on the extend to which Disparate Impact Remover transforms data for α=1.0.

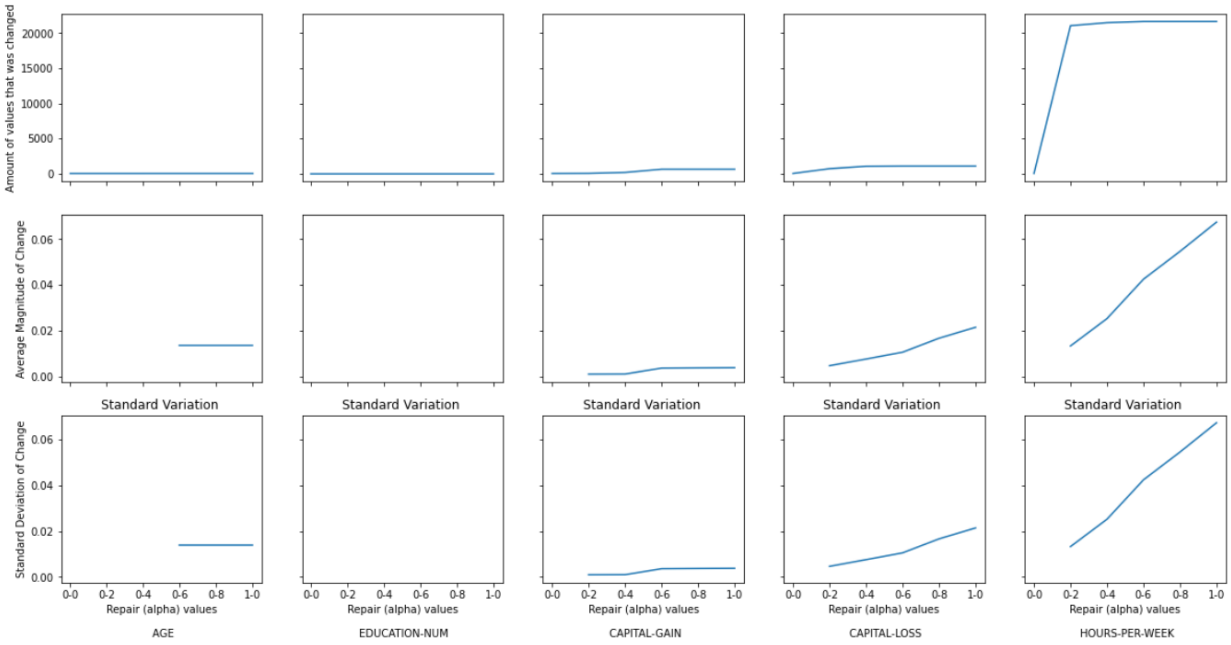

Fig. 5.4. Visual representation of Disparate Impact impact/change of original data for different α values.

used as the output $\hat{x}$ in our generator).

We have tried other methods to achieve better results such as a Variational Auto-Encoder version of the reconstructor (where the middle layer is composed of the mean and standard variation that generates the distribution of value, see Section 2.2.4), a version without classifiers in which the auto-encoder directly recreates the sensitive attribute, using Mean Square Error (L_2) or Damage (see Section 1.1.1) as loss functions. All these attempts did not result in any improvement with respect to the quality of reconstructed profiles. We believe that more research would need to be done since it should intuitively be possible to reverse more or less perfectly the protection provided by the adversarial methods. This intuition is due to the fact that the model fundamentally does a mathematical operation on the input data, which should be reversible. In addition, a discriminator that has a white-box knowledge of

the Disparate Impact Remover algorithm (meaning he was aware of the algorithm's transformation discussed in Section 2.2.3) could easily reverse it without even relying on learning methods as we did. This is true because Disparate Impact remover applies deterministic transformations whereas the other two algorithms do not. Nonetheless, as we tested the mutual information between the sensitive attribute (gender) and all others from data generated by the three different methods, it gave us the intuition that much simpler attack methods might be able to break the protection these methods offer, these external classifiers attacks will be the core of the analysis of Section 5.3.

5.3. External classifiers attacks

As mentioned in the previous section, the reconstruction attack has failed. The last step of the reconstructor system, referring back to Figure 4.1, involves a classifier that takes the reconstructed data, and tries to predict the original sensitive attribute. We noted that for GANSan, although the profile was successfully reconstructed (at least partially), the accuracy on the prediction of the sensitive attribute did not improve, or even worsen. Similarly for Disparate Impact, although the average profile was not successfully reconstructed, these external classifiers predicting the sensitive attribute surprisingly reached an average accuracy of around 84% depending on the value of α.

Further investigation (still relative to Scenario 2 from Table 4.1) summarised in Figure 5.5 shows the average classification accuracy of the sensitive attribute of our five machine learning classifiers for the sensitive attribute on disparate impact remover for different values of α. Despite the fact that the fairness metric is successfully optimised (above the 80% bar here), our classifier displays a very high prediction accuracy, even higher than the 84.88% on the original data. The results with the Compas dataset are inconclusive as the Disparate Impact Remover algorithm as implemented by IBM's AIF-360 does not allow to reach a disparate impact value above the desired 80% threshold even with an extreme value of $\alpha = 1.0$ (here called the repair-level value). When we ran our classifiers against the data sanitised with this method, we achieve a high 85.66% accuracy in predicting the sensitive attribute, which is much higher than the 69% baseline accuracy on the original data.

This surprising result creates potential issues, both in terms of privacy and fairness. This result made us wonder whether other methods also failed to prevent the prediction of sensitive attribute despite their successful fairness metric improvement. Figure 5.6 shows the same graph for LAFTR-generated data. Although the accuracy of the prediction of sensitive attribute does not reach the 90% range observed with Disparate Impact Remover, the method fails to reduce the accuracy towards values closer to the actual male-female

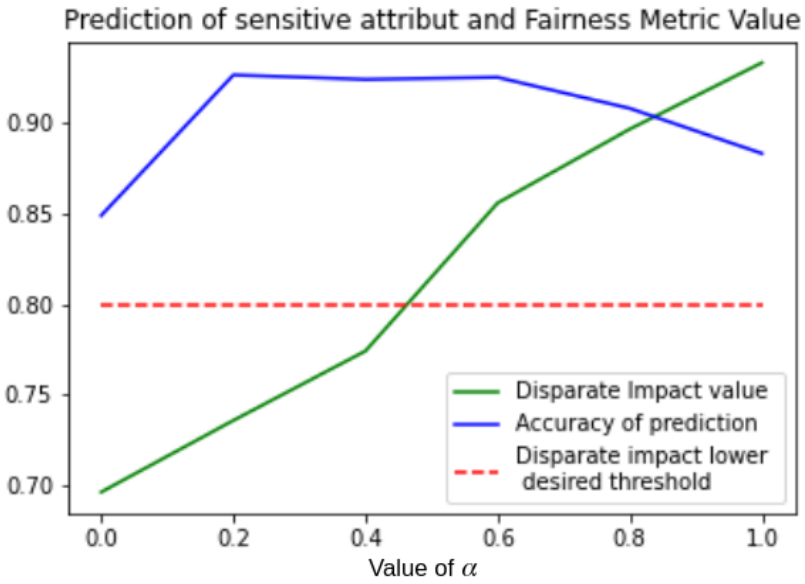

Fig. 5.5. Accuracy of the prediction of the sensitive attribute prediction for the data sanitised by disparate impact remover for various values of α.

distribution of 67%-33%, staying around 85% for most values of α. Thus, we observe here the same issue in which the method, by optimising a fairness metric only, does not necessarily protect against external actors from inferring the sensitive attribute, opening the door to issues of privacy which themselves lead to fairness issues, a relationship that was discussed in Section 3.3.

Finally, Figure 5.7 offers the same analysis for data sanitised by GANSan. This method does not lead to the same conclusions as the previous two methods. In particular, the prediction accuracy of the sensitive attribute decreases quite proportionately with an increase in the value of α (which as we previously saw also improves the fairness metric). This can be considered the most ideal results so far, in fact, the only one offering a significant level of privacy protection for the sensitive attribute. The lowest accuracy value around 69% is extremely close to what can be considered an optimal 67%-33%, which corresponds to a naïve classifier predicting the gender according to the true proportion of male-female in training data (this classifier could for instance discard the information available on the sanitised profile and always predict the majority class). It also seems that the α parameter offers relatively good control on the level of protection offered. One caveat is the accuracy peak with the highest value of α, which does not follow the trend that is otherwise fairly consistent for all other values.

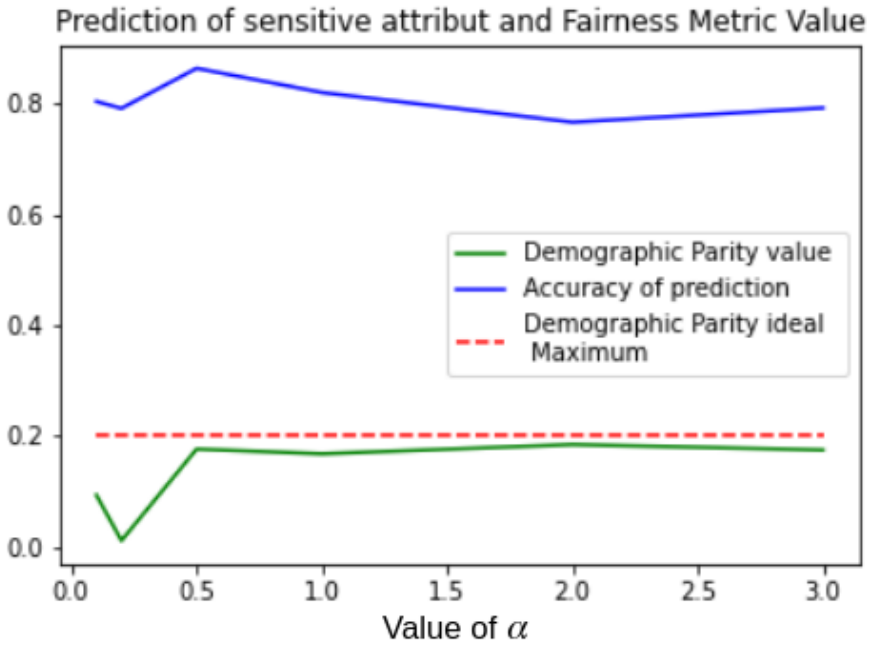

Fig. 5.6. Accuracy of the prediction of the sensitive attribute using the LAFTR-generated data for various values of α.

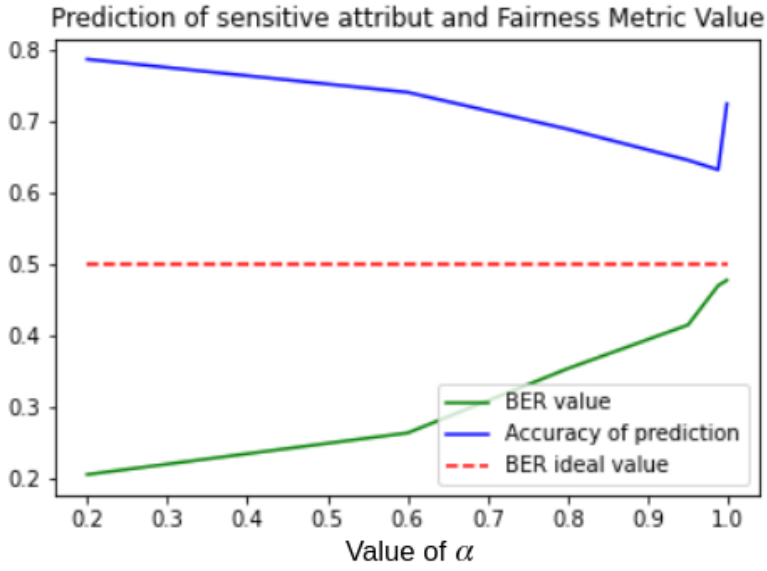

Fig. 5.7. Prediction accuracy of sensitive attribute on GANSan sanitised data for various values of α. Recall that an accuracy of 67% is considered optimal.

5.4. Analysis of the source of leaked information

As mentioned in earlier chapters, to prevent indirect discrimination a fairness-enhancing method should reduce the correlations of the sensitive attribute with other attributes,

in addition of removing the sensitive attribute. One of the possible reasons for the high prediction of the sensitive attribute is probably the high level of mutual information between the original sensitive attribute and the other attributes. The machine learning classifiers can rely on this mutual information to make accurate prediction. To verify this, we have computed the Normalised Mutual Information (NMI), for each attribute of each method with the sensitive attribute and compared it with the original Adult dataset and we obtained the results displayed in Figure 5.8. NMI is computed between two attributes and returns how much information from the first attribute can be extracted from the second attribute. For example if we can predict the race of a person with his or her postal code, NMI between race and postal code is likely to be high.

	age	work clas	fnlw gt	ed	ed-num	mari	occ	rela	gain	loss	Hour	country
Adult Baseline	**.004**	**.014**	**.114**	**.003**	**.003**	**.121**	**.070**	**.258**	**.018**	**.013**	**.026**	**.004**
Gansan 0.9875	**.107**	**.002**	**.110**	**.001**	**.111**	**.002**	**.001**	**.002**	**.040**	**.025**	**.110**	**.001**
Disp. Impact 1.0	**.004**				**0.004**				**0.019**	**0.018**	**.023**	
Laftr	**Latent representation of 7 features all with 0.128**											

Fig. 5.8. Normalized Mutual Information for each attribute of each method studied. Red boxes indicates that the sanitisation made the feature significantly more correlated with the sensitive attribute.

For Disparate Impact Remover as well as GANSan, we computed the NMI using the data sanitised with the value of α offering the lower prediction accuracy for the sensitive attribute, which are respectively 1.0 and 0.9875. In the case of LAFTR, all 7 attributes have an NMI of 0.128, regardless of the value of α. As 0.127 denotes a higher correlation than most of the original data, this could explain the high accuracy of classification of the sensitive attribute for LAFTR. In practice, we believe that it is very likely that a machine learning classifier would be able to leverage this information to help predict the sensitive attribute. Given the issue of output diversity (constant generation of a median profile) when working with adversarial models (discussed in Section 1.1.3 and seen in Section 5.2), it might be the case that LAFTR model ended up settling for this average NMI value across attributes, although more investigation would need to be done to confirm this.

A surprising finding from Figure 5.8 is that individual attributes sanitised by Disparate Impact Remover seem to be less correlated with the sensitive attributes compared to GANSan's sanitisation, although it was much easier to predict the sensitive feature with the former. This means that there is a possibility of a remaining more complex correlation resulting from combination of features, which is not encapsulated in the NMI. For example, a combination of age and education-num could potentially contain a lot of information to predict the gender of the profiles, while each of this attribute alone might be weakly correlated.

5.5. Potential avenues for explaining variations in performance of the attacks

GANsan has very little similarities with Disparate Impact Remover in terms of the architecture and training procedure with the exception of the external classifier that Disparate Impact Remover uses in its computation of the transformation to be applied to the data. The relatively unconvincing results of Disparate Impact Remover on hiding the sensitive attribute could be attributed to the algorithm designers' choice of a single linear, low-expressivity logistic regression as classifiers and no other classifiers external to the transformation procedure. A simple way to support this hypothesis is to use the same logistic regression classifier (python library 'sklearn' implementation with balanced class weights) to compare its performance on tests we made thus far. This will help us understand whether this specific logistic regression algorithm is able to perform well on the Adult dataset task of predicting profiles' income. More precisely, two results are specifically relevant to our research. First, the performance of such algorithms on the core Adult dataset task of predicting income from all other inputs achieves 80.124% accuracy, which is 4% lower than our group of five classifiers (MLP, SVM, Bagging, Gradient Boosting and CART). Second, using the Disparate Impact Remover with an extreme value for the sanitisation of $\alpha = 1.0$, this logistic regression only achieves 53.454% in predicting the sensitive attribute, much lower than the baseline of 84.88% for our five machine classifiers. This suggests that the logistic regression used in the implementation of the Disparate Impact Remover does not have a high-enough complexity to predict the sensitive attribute on the Adult dataset.

Comparing the training procedure of GANSan in Figure 2.5 and LAFTR (Figure 2.3), there are some similarities with respect to the optimisation procedure. In particular, both have to optimise two functions that take similar inputs, one for the data generator and one for the discriminator (albeit not trained at the same rate). Nonetheless, some differences were outlined in Section 2.2 and the training order of the generator and discriminator as well as the choice of loss function is unlikely to be able to account for and explain the whole story.

The biggest difference is GANSan's training with 10-k-fold for any values of α, is that at each epoch and for each fold, additional external classifiers are used to predict the sensitive attribute. In addition, the fairness metric for each of the 10 folds returns multiple different values of the fairness-utility trade-off (fairness-fidelity is used as proxy in the paper) from which we choose the best one. The training process then chooses the data generated by the sanitiser according to the best fairness/utility, defined in the paper as:

$$BestValue = \min \left\{ \left(BER_{\min} - \frac{1}{2} \right)^2 + fid_e \right\} \tag{5.5.1}$$

in which $\text{BER}_{\min}$ refers to the lowest BER value from the external classifiers. The inclusion of these external classifiers, which simulate an attack on the sensitive attribute, has shown to have strong implications in the privacy guarantees the sanitised data offers. Counter-intuitively, it even seems to have an even bigger impact on the mutual information about the sensitive attribute contained in the sanitised data. Although Figure 5.8 has shown that the mutual information was higher for most of the numerical attributes, it was lower for a majority of the categorical. It also outlines the issue that relying only on the discriminator classifier whose training is bound by the full training procedure of the system is therefore unlikely to provide a full account of the possible privacy risks in terms of inference. This is a known phenomenon of adversarial learning, in which the discriminator tends to converge (*i.e.*, improve) faster than the generator [**93**]. This issue is usually mitigated by constraining the generator in a number of ways, most commonly by manually slowing down the convergence of its training, which gives a desirable conclusion in typical data-synthetization use cases, but seemingly not for fairness-enhancing methods.

Chapter 6

Conclusion

In conclusion, our initial objectives of reconstructing the user profiles to retrieve the information supposed to be hidden by the three fairness-enhancing methods assumed the data was well protected. First, while optimizing using a fairness metric, most of these methods intrinsically offer little guarantees with respect to privacy. Second, relying only on the traditional GAN-learning procedure prevents the adversary classifier to reach its full potential since the discriminator is known to have trouble reaching its full capacity (see Section 5.5). The main effect is giving a perception that information about sensitive attributes is removed from the data, what we showed is not the case. Third, minimising the mutual information between the sensitive attribute and the other sanitised attributes is not sufficient in itself to prevent the prediction of the sensitive attribute.

Indeed in all cases we have investigated, the analysed methods succeeded in optimising fairness according to various well-established metrics. Even if finding some form of inverse transformation to rebuild the original data does not seem to be an easy task, the remaining mutual information in the released data allows in some cases to retrieve the value of the sensitive attribute with simple machine learning classifiers. In other cases, the use of more powerful classifiers confirms that it is increasingly difficult to infer the hidden sensitive attribute from the sanitised data. These classifiers are necessary during two distinct phases. They are embedded in the training, but also used externally to validate the protection with high-expressivity classifiers as is the case in GANSan.

Although those fairness-improving methods based on optimising a single fairness metric open the door to very big privacy concerns, research on approaches such as GANSan [**6**] showed that the inclusion of external classifiers throughout the training (as opposed to their inclusion at the end only) seem to mitigate some of those risks.

6.1. Future work

The reconstruction attack approach started in this research would clearly benefit from further investigation. In particular, as we mentioned, intuition suggests that it should be possible to learn a reverse transformation for the data that was transformed with all three methods under investigations. If not a deep-learning model like the auto-encoder we used, an adversarial model, more similar to the one used to transform the data, could prove more accurate. Additionally, seeing that relative convergence of the L_1 loss did not result in an accurate classification of the sensitive value, changing or tweaking of the loss used in training seems to be a potential avenue to investigate.

A second promising avenue to explore is the increase in the diversity of output that was achieved not by merely reducing the loss vector to its average value, and applying gradient descent for every value at every epoch, which is far from standard in most machine learning applications. A quick verification in Section 4.3 showed us that, summing up, the vector and applying a single gradient for each epoch did not have the same effect. This suggests the model gains from moving around more in the data space (given we apply gradient descent multiple times in one epoch as opposed to once), and further investigation should give us theoretical knowledge of how and why this is the case.

Some assumptions discussed in earlier sections could also be challenged for a better understanding of the fairness-utility trade-off. For example, some work suggests rather counter-intuitively that including the sensitive attribute in the algorithm's input might increase performance of both fairness and the downstream task [**60**]. Their algorithm's result for student's GPA in university admission showed that a classifier that is *aware* of the sensitive attribute (on which most of all this paper is based on) can achieve better results.

Finally, as we saw with Disparate Impact Remover applied on Compas data, a single metric may lead to false sense of comfort with respect to fairness in a specific dataset. Taking into account more than one fairness metric simultaneously during training could potentially mitigate these dangers, although we have yet to see such an implementation.

The increased research in the field of fairness-enhancing algorithms will lead to improve methods and techniques for improving fairness. We should nonetheless make sure to keep an eye on the root of the cause that makes this work necessary. In particular, finding out why there is such a disparate prevalence among certain groups in the first place and how can we build a society in which such fairness-enhancing algorithms are superfluous. Using the Adult dataset as an example, we saw that non-white profiles are disproportionately likely

to earn incomes lower than $50 000. It is great that technology could be used to make sure these imbalances are not amplified, but tackling the root of the problem here would involve figuring out why these imbalances are in the data in the first place as well as how can we, as a society mitigate those when we deem them undesirable.

Bibliography

[1] Ulrich Aïvodji, Sébastien Gambs, and Timon Ther. Gamin: An adversarial approach to black-box model inversion. *arXiv preprint arXiv:1909.11835*, 2019.

[2] Maryam Archie, Sophie Gershon, Abigail Katcoff, and Aaron Zeng. De-anonymization of netflix reviews using amazon reviews.

[3] Narayanan Arvind. 21 fairness definitions and their politics. tutorial presented at the conference on fairness, accountability, and transparency. 2018.

[4] Vitaly Shmatikov Arvind Narayanan. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, 2008.

[5] Microsoft Azure. Azure machine learning. `https://azure.microsoft.com/en-us/services/machine-learning/`. [Online; accessed 2020-08-06].

[6] Ulrich Aïvodji, François Bidet, Sébastien Gambs, Rosin Claude Ngueveu, and Alain Tapp. Agnostic data debiasing through a local sanitizer learnt from an adversarial network approach, 2019.

[7] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[8] Mihir Bellare and Phillip Rogaway. Introduction to modern cryptography. *Ucsd Cse*, 207:207, 2005.

[9] Bonnie Berger and Hyunghoon Cho. Emerging technologies towards enhancing privacy in genomic data sharing. *Genome Biology*, 20, 12 2019.

[10] Jason Brownlee. Gentle introduction to the adam optimization algorithm for deep learning, December 2020. [Online; posted 15-August-2020].

[11] Jason Brownlee. Imbalanced classification with the adult income dataset, 2020. [Online; accessed August 1st, 2020].

[12] Jason Brownlee. Linear regression for machine learning, October 2020. [Online; posted 20-August-2020].

[13] United States Consumer Financial Protection Bureau. Using publicly available information to proxy for unidentified race and ethnicity. Report, US CFPB, Washington DC, 2014.

[14] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. 1998.

[15] Irene Chen, Fredrik D. Johansson, and David Sontag. Why is my classifier discriminatory?, 2018.

[16] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, Jun 2017.

[17] Google Cloud. Cloud automl. `https://cloud.google.com/automl/`. [Online; accessed 2020-08-06].

[18] CM FAccT Conference. Acm conference on fairness, accountability, and transparency (acm facct), 2020.

[19] C. M. Cook, J. J. Howard, Y. B. Sirotin, J. L. Tipton, and A. R. Vemury. Demographic effects in facial recognition and their dependence on image acquisition: An evaluation of eleven commercial systems. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(1):32–41, 2019.

[20] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness, 2017.

[21] Elliot Creager, David Madras, Jörn-Henrik Jacobsen, Marissa A. Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement, 2019.

[22] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, page 202–210, New York, NY, USA, 2003. Association for Computing Machinery.

[23] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, page 202–210, New York, NY, USA, 2003. Association for Computing Machinery.

[24] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[25] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[26] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery.

[27] Cynthia Dwork and Christina Ilvento. Fairness under composition, 2018.

[28] Cynthia Dwork and Christina Ilvento. Individual fairness under composition. *Proceedings of Fairness, Accountability, Transparency in Machine Learning*, 2018.

[29] Cynthia Dwork, Frank McSherry, and Kunal Talwar. The price of privacy and the limits of lp decoding. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 85–94, June 2007.

[30] Khaled Emam, Elizabeth Jonker, Luk Arbuckle, and Bradley Malin. A systematic review of re-identification attacks on health data. *PloS one*, 6:e28071, 12 2011.

[31] Saurabh Mishra et al. Artificial intelligence index report 2019 steering committee. Technical report, Stanford University, 2019.

[32] Daniel Faggella. Artificial intelligence applications for lending and loan management, 2020. [Online; accessed September 31, 2020].

[33] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, 2015.

[34] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery.

[35] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery.

[36] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. *Proceedings of the ... USENIX Security Symposium. UNIX Security Symposium*, 2014:17—32, August 2014.

[37] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*, 2019.

[38] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 329–338, New York, NY, USA, 2019. Association for Computing Machinery.

[39] Pratik Gajane and Mykola Pechenizkiy. On formalizing fairness in prediction with machine learning, 2017.

[40] Simson L. Garfinkel, John M. Abowd, and Sarah Powazek. Issues encountered deploying differential privacy. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, WPES'18, page 133–137, New York, NY, USA, 2018. Association for Computing Machinery.

[41] P. E. GILL and W. MURRAY. Quasi-Newton Methods for Unconstrained Optimization. *IMA Journal of Applied Mathematics*, 9(1):91–108, 02 1972.

[42] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive Computation and Machine Learning series. MIT Press, 2016.

[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[44] Google Developers Documentation. Overview of gan structure, 2020. [Online; accessed July 29, 2020].

[45] I. Guyon, J. Makhoul, R. Schwartz, and V. Vapnik. What size test set gives good error rate estimates? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):52–64, 1998.

[46] Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C. Turnbull, Brendan M. Collins, and Thomas Legrand. Applying deep learning to airbnb search, 2018.

[47] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3315–3323. Curran Associates, Inc., 2016.

[48] Douglas M. Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12, 2004.

[49] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019:133–152, 01 2019.

[50] Jane Henriksen-Bulmer and Sheridan Jeary. Re-identification attacks—a systematic literature review. *International Journal of Information Management*, 36(6, Part B):1184 – 1192, 2016.

[51] Danny Hernandez and Tom B. Brown. Measuring the algorithmic efficiency of neural networks, 2020.

[52] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization, 2020.

[53] IBM. Ai fairness 360 open source toolkit. 2020.

[54] Surya Mattu Julia Angwin, Jeff Larson and ProPublica Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. May 2016.

[55] Peter Kairouz, Jiachun Liao, Chong Huang, and Lalitha Sankar. Censored and fair universal representations using generative adversarial models, 2019.

[56] F. Kamiran, T. Calders, and M. Pechenizkiy. Discrimination aware decision tree learning. In *2010 IEEE International Conference on Data Mining*, pages 869–874, 2010.

[57] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. Discrimination aware decision tree learning. In *In Proc. of the 10th IEEE Int'l Conf. on Data Mining*, pages 869–874, 2010.

[58] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 35–50, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[59] Shiva Prasad Kasiviswanathan, Mark Rudelson, and Adam Smith. The power of linear reconstruction attacks, 2012.

[60] Jon M. Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Advances in big data research in economics algorithmin fairness. 2018.

[61] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness, 2017.

[62] KENNETH LEVENBERG. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[63] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, 2007.

[64] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder, 2015.

[65] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. volume 1, page 24, 01 2006.

[66] Machine Learning Crash Course. Generalization, 2020. [Online; accessed December 1st, 2020].

[67] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations, 2018.

[68] Daniel McNamara, Cheng Soon Ong, and Robert C. Williamson. Provably fair representations, 2017.

[69] Medium. Applied deep learning - part 3: Autoencoders, 2017. [Online; accessed July 27, 2020].

[70] David Meyer, Friedrich Leisch, and Kurt Hornik. The support vector machine under test. *Neurocomputing*, 55(1):169 – 186, 2003. Support Vector Machines.

[71] Adam Meyerson and Ryan Williams. On the complexity of optimal k-anonymity. In *Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '04, page 223–228, New York, NY, USA, 2004. Association for Computing Machinery.

[72] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[73] Marvin Minsky and Seymour Papert. Perceptrons - an introduction to computational geometry. 1969.

[74] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[75] J. J. More and D. C. Sorensen. Newton's method.

[76] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[77] Brady Neal. On the bias-variance tradeoff: Textbooks need an update, 2019.

[78] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks, 2016.

[79] Supreme Court of the United States. Griggs v. duke power co.

[80] Kayee Hanaoka Patrick Grother, Mei Ngan. Face recognition vendor test (frvt), part 3: Demographic effects. Report, US Department of Commerce; National Institute of Standard in Technology, Washington DC.

[81] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 560–568, New York, NY, USA, 2008. Association for Computing Machinery.

[82] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[83] Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data, 2019.

[84] ProPublica. compas-analysis, 2016. [Online; accessed August 5th, 2020].

[85] Mahabur Rahman, Thohedur Rahman, R. Laganière, N. Mohammed, and Y. Wang. Membership inference attack against differentially private deep learning model. *Transactions on Data Privacy*, 11:61–79, 04 2018.

[86] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017.

[87] Carl Edward Rasmussen and Zoubin Ghahramani. Occams razor. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 294–300. MIT Press, 2001.

[88] Thomson Reuters. Black and asian faces misidentified more often by facial recognition software.

[89] Rodrigo Benenson. Cifar-10 who is the best in cifar-10 ?, 2016. [Online; accessed July 30, 2020].

[90] Frank F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[91] David Rumelhart, Geoffrey Hinton, and Ronald Williams. Learning representations by back propagating errors. *Nature*, 323:533–536, 10 1986.

[92] Sima Sajjadiani, Aaron J Sojourner, John D Kammeyer-Mueller, and Elton Mykerezi. Using machine learning to translate applicant work history into predictors of performance and turnover. *Journal of Applied Psychology*, 2019.

[93] Mathew Salvaris, Danielle Dean, and Wee Hyong Tok. Generative adversarial networks. *Deep Learning with Azure*, page 187–208, 2018.

[94] Sebastian Ruder. Nlp-progress repository to track the progress in natural language processing (nlp), including the datasets and the current state-of-the-art for the most common nlp tasks., 2020. [Online; accessed July 30, 2020].

[95] Amazon Web Services. Machine learning on aws. [Online; accessed 2020-08-06].

[96] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2016.

[97] Tom Simonite. The best algorithms struggle to recognize black faces equally.

[98] Alexander J. Smola and Shahar Mendelson. Machine learning, proceedings of the summer school 2002, 2002.

[99] Moritz Hardt Solon Barocas. Fairness in machine learning. 2017.

[100] Baohua Sun, Lin Yang, Wenhan Zhang, Michael Lin, Patrick Dong, Charles Young, and Jason Dong. Supertml: Two-dimensional word embedding for the precognition on structured tabular data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[101] Latanya Sweeney. Simple demographics often identify people uniquely. *Health*, 671, 01 2000.

[102] Latanya Sweeney. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[103] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12, 2017.

[104] Richard A Tapia. Diagonalized multiplier methods and quasi-newton methods for constrained optimization. *Journal of optimization theory and applications*, 22(2):135–194, 1977.

[105] Google Differential Privacy Team. Differential privacy. `https://github.com/google/differential-privacy`, 2020.

[106] The Stanford Natural Language Processing Group. The stanford natural language inference (snli) corpus. [Online; accessed July 30, 2020].

[107] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 601–618, Austin, TX, August 2016. USENIX Association.

[108] Ana Valdivia, Javier Sánchez-Monedero, and Jorge Casillas. How fair can we go in machine learning? assessing the boundaries of fairness in decision trees, 2020.

[109] Austin Waters and Risto Miikkulainen. Grade: Machine learning support for graduate admissions. *AI Magazine*, 35(1):64, Mar. 2014.

[110] Wikipedia, the free encyclopedia. Autoencoder, 2020. [Online; accessed July 27, 2020].

[111] Tim Brennan William Dieterich, Christina Mendoza. Compas risk scales:demonstrating accuracy equity and predictive parity. July 8 2016.

[112] Blake Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors, 2017.

[113] Xi Wu, Matt Fredrikson, Somesh Jha, and Jeffrey F. Naughton. A methodology for formalizing model-inversion attacks. *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pages 355–370, 2016.

[114] Apple WWDC 2016. Engineering privacy for your users. June 2016.

[115] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. *CoRR*, abs/1805.11202, 2018.

[116] Yan LeCun et al. The mnist database of handwritten digits. [Online; accessed July 30, 2020].

[117] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting, 2017.

[118] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification, 2015.

[119] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.