FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Rewriting History: Fair label noise correction

Inês Oliveira e Silva

Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Carlos Soares

Co-Supervisor: Inês Sousa

July 27, 2023

Rewriting History: Fair label noise correction

Inês Oliveira e Silva

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. João Moreira
Referee: Pedro Saleiro

July 27, 2023

Resumo

A tomada de decisões arbitrárias, inconsistentes ou defeituosas levanta sérias preocupações [4], e prevenir modelos injustos é um desafio cada vez mais importante em *Machine Learning* [53]. Em muitos casos, os dados refletem comportamentos discriminatórios passados, e os modelos treinados com tais dados podem refletir enviesamento relativamente aos atributos sensíveis, como o género, raça ou idade [72]. Desenvolver modelos justos apresenta muitos desafios, principalmente relacionados com a avaliação da equidade, deteção de discriminação e mitigação de injustiças [53]. Uma abordagem para desenvolver modelos justos é pré-processar os dados de treino para simultaneamente remover a discriminação subjacente mas preservar as informações relevantes, por exemplo, através da correção de *labels* enviesadas. Embora existam vários métodos disponíveis para a correção de ruído nas *labels*, a investigação sobre o seu comportamento na identificação de discriminação é muito limitada. Neste trabalho, desenvolvemos uma metodologia empírica para avaliar sistematicamente a eficácia das técnicas de correção de ruído nas *labels* na garantia da equidade dos modelos treinados em datasets tendenciosos. A nossa metodologia envolve manipular a quantidade de ruído nas *labels* e pode ser usada com datasets benchmark de fairness, assim como com datasets standard de ML. Na avaliação empírica, aplicamos a nossa metodologia na análise de seis métodos de correção de ruído nas *labels* de acordo com várias métricas de fairness em datasets disponibilizados no OpenML. Nas experiências realizadas, aplicamos a nossa metodologia tanto em datasets benchmark de fairness como em datasets standard de ML nos quais o ruído é induzido nas *labels* através da segregação de um atributo escolhido arbitrariamente. Os resultados obtidos sugerem que o método *Hybrid Label Noise Correction* [70] alcança o melhor equilíbrio entre desempenho preditivo e equidade. O método *Clustering-Based Correction* [50] mostrou a maior redução da discriminação, porém, em prol de ter obtido um desempenho preditivo mais baixo. Adicionalmente, propomos um método para correção justa de ruído nas *labels*, que introduz modificações num método de correção de ruído proposto anteriormente, denominado *Ordering-Based Noise Correction* [23], de modo a tomar a equidade em consideração durante o processo de correção das *labels*. Avaliamos os nosso método em comparação com o método original, comparando a equidade das previsões e o desempenho preditivo dos modelos resultantes, tanto em datasets benchmark de fairness como em datasets standard de ML nos quais o ruído é injetado nas *labels* em proporções crescentes. Os resultados obtidos sugerem que a modificação de métodos já existentes de correção de ruído nas *labels* de modo a tomar em conta a equidade das correções pode resultar em modelos menos discriminatórios sem perda de desempenho preditivo.

Abstract

Arbitrary, inconsistent, or faulty decision-making raises serious concerns [4], and preventing unfair models is an increasingly important challenge in Machine Learning [53]. In many cases, data reflects past discriminatory behavior, and models trained on such data may reflect bias on sensitive attributes, such as gender, race, or age [72]. Developing fair models poses many challenges, which are mainly related to the measurement of fairness, the detection of unfairness, and the mitigation of unfairness [53]. One approach to developing fair models is to preprocess the training data to remove the underlying biases while preserving the relevant information, for example, by correcting biased labels. While multiple label noise correction methods are available, the existing ones are focused on model accuracy rather than fairness. In this work, we develop an empirical methodology to systematically evaluate the effectiveness of label noise correction techniques in ensuring the fairness of models trained on biased datasets. Our methodology involves manipulating the amount of label noise and can be used with fairness benchmarks but also with standard ML datasets. We apply the methodology to analyze six label noise correction methods according to several fairness metrics on OpenML datasets. In the conducted experiments, we leverage both fairness benchmark datasets and standard datasets where label noise is induced by the segregation of an arbitrarily chosen feature. Our results suggest that the Hybrid Label Noise Correction [70] method achieves the best trade-off between predictive performance and fairness. Clustering-Based Correction [50] can reduce discrimination the most, however, at the cost of lower predictive performance. Furthermore, we propose a method for fair label noise correction that adapts the Ordering-Based Noise Correction method [23] to take fairness into account during the process of label correction. We evaluate our method against the original one, comparing the fairness and predictive performance of the resulting models in several fairness benchmarks, as well as standard ML datasets where label noise is injected with varying rates. Our results show that the proposed method learns models that are less discriminatory without loss of predictive performance.

Acknowledgments

I would like to express my sincere appreciation to my supervisors, Carlos Soares and Inês Sousa, and to Rayid Ghani, for their invaluable support and guidance throughout my journey in completing this master's thesis. Their continuous encouragement, insightful feedback, and commitment to excellence have been instrumental in shaping this work.

I am profoundly grateful to my family for their love, support, and relentless belief in my abilities. A special appreciation goes out to my parents, who have always pushed me to strive for greatness and have instilled in me the confidence to face any challenges that come my way.

To my sister Francisca, who knows me more deeply than anyone else, thank you for being my best friend through the good and the bad. I cannot imagine how boring life would be without you.

Lastly, to my friends, who fill my days with joy, laughter, and inspiration. I am humbled and honored to have such amazing people in my life who have shaped my personal and academic growth in so many ways. I am graduating with the best memories and will always cherish my academic experience because of you.

Inês Silva

"Hoping for better times mustn't be a feeling but a doing something in the present"

Vincent van Gogh

Contents

List of Figures

List of Tables

Abbreviations

AUC	Area Under the ROC Curve
BE	Bayesian Entropy Noise Correction
CC	Clustering-Based Correction
CE	Cross Entropy
CL	Confident Learning
COMPAS	Correctional Offender Management Profiling for Alternative Sanctions
DNN	Deep Neural Networks
FNR	False Negative Rate
FN	False Negative
FP	False Positive
FPR	False Positive Rate
HLNC	Hybrid Label Noise Correction
MAE	Mean Absolute Error
ML	Machine Learning
OBNC	Ordering-Based Label Noise Correction
PL	Polishing Labels
ROC	Receiver Operating Characteristic
SL	Symmetric Learning
SVM	Support Vector Machine
SSK-means	Semi-Supervised K-Means
STC	Self-Training Correction
TN	True Negative
TPR	True Positive Rates
TP	True Positive

Chapter 1

Introduction

Machine Learning (ML) is a field that intersects computer science and statistics, addressing the problem of developing computer systems that can learn how to improve their performance through experience. The ongoing growth in data availability and computational resources have enabled the explosion of research in this area, with new algorithms and theory constantly emerging and making it one of the most rapidly evolving technical fields of the present day [37]. The wide range of domains where data-intensive machine learning methods can be applied has led to the increasing use of these models in decision-making, from healthcare to education or policing.

However, this widespread use of ML systems in sensitive environments profoundly impacts people's lives when given the power to make life-changing decisions [47]. One well-known example is the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) software. This computer program assesses the recidivism risk of individuals, and the American courts use it to decide whether to release a person from prison with bail. In a 2016 investigation conducted by ProPublica,[1] it was discovered that the system was biased against African-Americans, incorrectly classifying Black offenders as "high-risk" twice as often as White defendants. Another example relates to a less impactful yet more widely present tool in people's lives: Google's targeted ads. A group of researchers proposed AdFisher [20], a tool to gather insights on how user behaviors, Google's transparency tool "Ad Settings", and the presented advertisements interact. Their study revealed that male web users were more likely to be presented with ads for high-paying jobs than their female counterparts.

These are only a few cases amongst the growing number of examples of sensitive environments where ML is employed for decision-making. It is thus necessary that these decisions that risk limiting people's access to opportunities are based on the factors that are relevant to the desired outcome [4]. Learning such mapping from the factors to the outcome involves generalizing from historical examples, which can make algorithms vulnerable to the same biases people projected in their past decisions. For example, Amazon's ML experts tried to build a recruiting engine to automate the review of job applicants' resumes. However, they realized that their tool was discriminatory towards women. This behavior was suspected to be the consequence of using

[1] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

training data from the previous ten years, during which most applicants to technical positions were male, leading the system to discard most female applicants[2].

Intending to identify and mitigate these harmful and unacceptable inequalities, the field of *fair machine learning* has emerged [4]. In this context, we can classify an algorithm as unfair if its decisions reflect some kind of prejudice or favoritism towards certain groups of people based on their inherent or acquired characteristics [47].

As previously discussed, one of the critical aspects of creating reliable ML models is providing the system with high-quality data and having a sufficiently large, diverse, and well-annotated set of examples to learn from. Nevertheless, real-world datasets often contain noise, which can be described as non-systematic errors that blur the relationship between an instance's features and its class [26].

The ubiquity of noise and the high cost of acquiring high-quality data give rise to the extreme importance of implementing techniques that reduce noise and its consequences. However, despite the vast amount of literature on methods for dealing with noisy data, only a few of these studies focus on identifying and correcting noisy labels [50]. Our intuition is that this approach of correcting wrongly attributed labels can be leveraged in the context of fair machine learning if we consider discrimination present in the data as noise that can be removed. As such, one can apply noise correction techniques to obtain a feasibly unbiased dataset that can be used to train fair models. Thus, the motivation for this work comes from the need for more research on noise correction techniques and, to the best of our knowledge, the lack of work exploring the use of these methods in training fair models from biased data.

We develop an empirical methodology to systematically evaluate the usefulness of applying label noise correction techniques to guarantee the fairness of predictions made by models trained on biased data. Having an assumedly clean dataset, we first manipulate the labels to simulate the desired amount and type of label noise. The injected noise is group-dependent, meaning that it depends on the value of the specified sensitive attribute. We can parameterize the noise injection process to model various types of discrimination. The desired label noise correction technique is applied to the noisy data to generate a corrected version of the dataset. We train ML classifiers using the *original*, *noisy*, and *corrected* training sets. The obtained models are then evaluated under different assumptions, measuring the fairness and predictive performance of their predictions on the three test sets (*original*, *noisy*, and *corrected*). In this empirical study, we test and compare the effectiveness of six label noise correction techniques in improving the generated models' performance. We apply our methodology using multiple standard ML datasets available on OpenML and inject different types of label noise at varying rates. The models are evaluated using four well-known fairness metrics.

We further propose a variation of the Ordering-Based Noise Correction method [23] that takes fairness into account in the process of detecting and correcting noisy labels. Since the selection of labels to be corrected by OBNC takes into account all attributes, including the protected ones,

[2] https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

it may strengthen the prejudice. Thus, our first modification is to simply remove the sensitive attribute from the data used to identify misclassified labels. The second modification changes the criterion for deciding which labels should be corrected to balance the distribution of class labels across the sensitive groups. We test our method both in fairness benchmark datasets and also on standard ML datasets manipulated with our label noise manipulation methodology.

The rest of this document is organized as follows. In Chapter 2, we define the background concepts necessary for understanding this work and present an overview of the existing literature and state-of-the-art methods related to ML fairness and noise label correction. In Chapter 3, we detail the proposed methodology for a systematic analysis of label noise correction methods to improve ML fairness and describe the conducted empirical evaluation, presenting and analyzing the obtained results. In Chapter 4, we introduce our proposed algorithm for fair label noise correction, explain the conducted experiments, and present and discuss the obtained results. Finally, in Chapter 5, we analyze the conclusions derived from the developed work and propose future work directions.

Chapter 2

Literature Review

This chapter presents the main notions surrounding machine learning, fairness, and label noise. We first introduce the background knowledge related to fair machine learning and establish important concepts regarding the presence of label noise in datasets, reviewing the existing literature on these topics.

2.1 Machine Learning

Machine Learning concerns the problem of improving a system's performance according to some measure when executing a certain task by learning from experience through computational methods [37, 74]. In this scenario, experience takes the form of data, which, fed to a learning algorithm, results in a model that makes predictions for new samples [74]. Formally, a computer program is said to learn from experience E with respect to some classes of task T and performance measure P if its performance on the considered task can improve with experience [49].

There is a wide variety of ML algorithms, contrasting in how the candidate programs are represented and in how this space of programs is searched through [37]. An extensively used type of ML method is supervised learning, which is the learning task we focus on in this work. In this type of problem, the aim (i.e., the task T) is to make a prediction $\hat{y}$ for a given query x. The training data (i.e., the experience E) takes form in a set Tr of (x,y) pairs. Predictions are obtained via a learned mapping function, whose mapping can exist in many forms, ranging from decision trees to neural networks [32]. The target function $y = f(x)$ is the true function $f(\cdot)$ that we want to model. A hypothesis is the learned mapping function, $\hat{f}(\cdot)$, and its similarity to the target function is assessed using a loss function $L(\cdot)$ (i.e., the performance measure P) [56]. In machine learning, the terms *hypothesis* and *model* are used interchangeably. The learning algorithm is the set of instructions used to attempt to model the target function using training data.

The most simple case of supervised learning is binary classification, where y takes one of only two possible values, but there is an abundant variety of problems, from multiclass classification to ranking problems or multilabel classification [37].

As this work concerns classification, with a larger focus on the evaluation procedure, we will not go into as much detail regarding the remaining types of ML methods. These include unsupervised learning, where there is no need for labeled training data in order to successfully learn valuable representations of the input, and reinforcement learning, in which the task is to learn a policy for an agent to choose actions for a given state with the objective of maximizing the expected reward over time [37].

Regarding the binary classification problem, some classifiers output discrete values, indicating the predicted class label for the corresponding instance. Other classifiers assume the existence of a negative and a positive label (0 and 1, respectively) and produce continuous predictions corresponding to an estimate of the class membership probability of the instance. To determine the final predicted label, a threshold may be applied to this predicted probability: the predicted label is positive if the outputted value is above the threshold and negative otherwise.

2.1.1 Evaluation

In classification tasks, ML models output discrete values. Therefore the metrics for evaluating model performance must measure how well these perform in generalizing for unseen data and predicting the correct labels. In order to provide an overview of existing ML evaluation techniques, we present metrics for measuring the models' predictive performance in Section 2.1.1.1 and explain some relevant methods of performing such evaluation in Section 2.1.1.2.

2.1.1.1 Evaluation Metrics

Despite the existence of various performance metrics that are better suited for different problems and domains, most of them are defined based on a confusion matrix [74]. When in presence of a binary target, i.e., $y \in \{0,1\}$, there are four combinations of the ground truth and the predicted class:

- *True positive*, when both ground truth and predicted classes are positive, $y = \hat{y} = 1$;
- *False positive*, when the predicted class is positive but the ground truth one is negative, $y = 0, \hat{y} = 1$;
- *True negative*, when both predicted and ground-truth classes are negative, $y = \hat{y} = 0$;
- *False negative*, when the predicted class is negative but the ground-truth one is positive, $y = 1, \hat{y} = 0$.

When comparing the ground-truth labels with the predicted ones on a set of observations, we can compute the counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These counts constitute the confusion matrix, which is shown in Table 2.1.

Using the values in the confusion matrix, we can define a number of metrics, namely:

Table 2.1: The confusion matrix.

Ground-truth class	Predicted class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

- *Precision* [65], which represents the fraction of true positives from all predicted positive cases, $\frac{TP}{TP+FP}$;
- *Recall* [65], which is the fraction of true positives out of all the ground-truth positives, $\frac{TP}{TP+FN}$;
- *Accuracy* [24], which conveys the fraction of correctly labeled cases, $TP+TN$, from all samples, $\frac{TP+TN}{TP+FP+TN+FN}$;
- *F-measure* [24], which also measures the model's accuracy, but leveraging both *precision* and *recall* in a single value, defined as $F_1 = 2 * \frac{precision*recall}{precision+recall}$.

Finally, it is worth mentioning the area under the receiver operating characteristic (ROC) curve (AUC) [24] metric. ROC graphs are two-dimensional graphs where the y-axis represents the TP rate, and the FP rate is depicted on the x-axis, depicting relative tradeoffs between benefits (true positives) and costs (false positives) [22]. An example of a ROC graph is shown in Fig. 2.1, depicting the performance of a Logistic Regression classifier trained on the COMPAS dataset to predict whether an offender will commit a crime again.

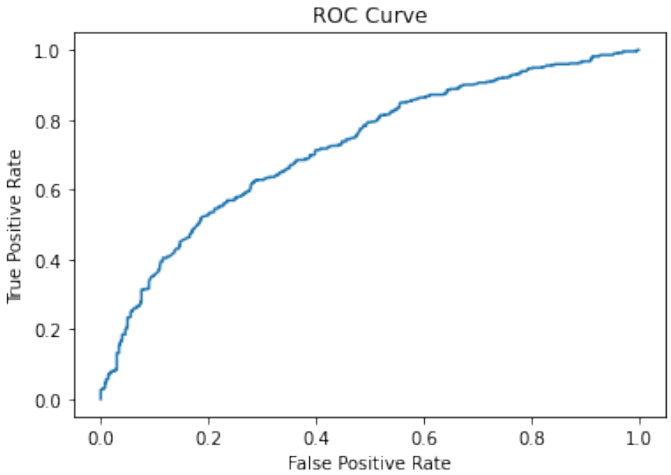


Figure 2.1: ROC graph obtained from the predictions of a Logistic Regression classifier trained on the COMPAS dataset.

If the used classifier outputs the probability of a sample belonging to each class and a threshold is applied to produce the discrete (binary) output, a ROC graph can be built from the predictions.

Each threshold value produces a different point in ROC space, so we can obtain a curve through ROC space by connecting the points corresponding to the different thresholds. By calculating the area under this curve, we obtain a single scalar value, the AUC, representing the model's performance.

2.1.1.2 Evaluation Methods

The model is obtained by applying the algorithm to a training set, but in order for the evaluation measures introduced earlier to represent the generalization ability of the model (i.e., its ability to make accurate predictions on new cases), a separate set of data is necessary. In most cases, only one set of data is available. As such, we must split it into training and test data.

In supervised learning, the simplest way to do so is through the holdout method. Using this technique, the data is split in a training and test set, fitting the model to the training data and using it to make predictions on the test data [56]. Model performance can then be estimated by comparing the true labels y of the test set to the predicted ones, $\hat{y}$, and calculating the desired metrics.

Splitting the dataset can easily be done by random subsampling, which consists of randomly choosing a part of the samples for the training set and using the rest as test data. However, applying random subsampling to naturally imbalanced datasets may result in minority classes being completely absent from the test set. To overcome this problem, a largely used technique is stratified subsampling, where the dataset is randomly split in such a way that maintains the original class proportion in both the training and test sets.

2.2 Fairness

The process of achieving fairness requires first defining it, which is not an easy task. While the vast abundance of different cultural backgrounds makes it harder to come up with a single widely accepted definition of fairness [47], it can be depicted as the absence of biases towards an individual or a group based on their innate or acquired traits that are irrelevant in a certain decision-making context [57]. Despite the numerous definitions of algorithmic fairness that have been proposed, which definition to apply in each particular case is still a subject of debate [65].

In this section, we review the state-of-the-art of fairness in ML, focusing mostly on research related to classification tasks. Section 2.2.1 highlights the main causes that potentially harm ML fairness. We then introduce several different measures to quantify fairness in Section 2.2.2, discussing relevant trade-offs. Finally, in Section 2.2.3, we present an overview of existing methods for enhancing ML fairness, comparing their characteristics, advantages, and disadvantages.

2.2.1 Unfairness Causes

An important first step when dealing with ML fairness is to understand the causes of the existing biases. In the work of [47], the authors propose a categorization of bias sources according to the

feedback loop phenomenon between biases present in data, algorithms, and user interaction, as shown in Fig. 2.2.

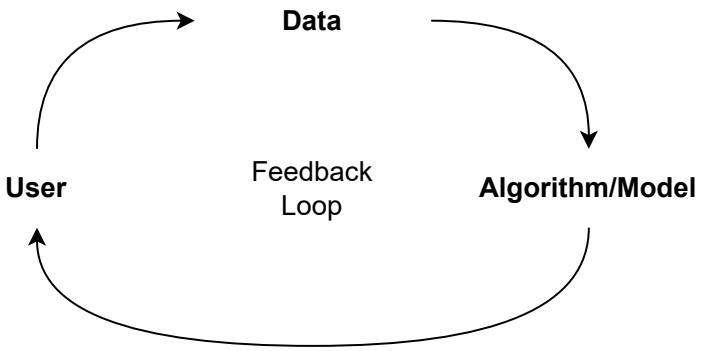


Figure 2.2: The feedback loop phenomenon between bias sources [47].

If the training data contains biases, the algorithms that leverage it will produce biased outcomes, likely intensifying and perpetuating the biases that were already present in the data. When feeding these biased outcomes to real-world systems, the users' decisions will be influenced by them, generating even more prejudiced data for prospective algorithm training. Based on this interaction loop, three categories of unfairness sources emerge.

Firstly, bias in the data can cause unfair algorithmic outcomes. For example, if there are significantly fewer instances of a certain group with positive labels than negative ones for cultural or historical reasons, then a classifier trained on such data will likely learn to classify most members of that group as negative [12]. Some specific biases in this category include:

- *measurement bias*, which arises during data collection when measuring the features [61];
- *representation bias*, which is related to the process of defining and sampling from a population [61];
- *aggregation bias*, which stems from wrongly made assumptions about the considered population that affect the model definition [61];
- *sampling bias*, which happens when the non-randomized sampling of subgroups hinders the generalization of estimated trends to new populations [47].

Furthermore, biases that stem from the algorithms can modulate biased user behavior. For example, algorithmic objectives can originate biased predictions by benefiting large groups to minimize the overall aggregated prediction error. Even if sensitive attributes are not being considered for decision-making, other non-sensitive features can be used to derive the sensitive ones, which are called proxy attributes. These can thus lead the ML algorithm to produce unfair decisions [53]. A few particular types of bias that are included in this category are:

- *algorithmic bias*, which is introduced by certain algorithmic design choices or by the use of statistically biased estimators, producing biased outputs [47];

- *popularity bias*, which may be seen in recommendation systems, for example, and is caused by the manipulation of popularity metrics by fake reviews or bots, giving more exposure to items that are popular but not necessarily the best option [17];
- *evaluation bias*, which happens during model evaluation, for example when using inappropriate benchmarking data that is not representative of the desired population [61].

Finally, user-generated data sources reflect intrinsic biases in the users that create them, namely:

- *historical bias*, which is observed when the collection of data accurately represents the world, also including discriminatory historical factors that have caused harm to certain populations [61];
- *social bias*, which occurs when a user's decision does not stem entirely from their own judgment but is rather influenced by others [47];
- *behavioral bias*, which arises from users adopting different behaviors depending on the context they are interacting in [47].

2.2.2 Fairness Measures

To empirically evaluate whether ML models are discriminatory, the previously discussed definition of fairness must be quantifiable. As such, in this section, we examine relevant fairness measures and discuss their trade-offs.

Fairness measures can be categorized according to how the underlying calculation is performed. In this way, we can consider statistical, similarity-based, and causal reasoning-based measures [65]. The statistical measures are mostly based on metrics that can be obtained from the confusion matrix. On the other hand, similarity-based measures focus on similar instances obtaining similar predictions. Finally, causal reasoning-based measures assess fairness by analyzing the relationships between attributes in a causal graph and how these attributes impact the predictions.

Statistical measures are easier to calculate than similarity and causal reasoning-based ones [65]. However, statistical measures focus only on the sensitive attributes, possibly hiding unfairness. To illustrate this, consider a classifier that predicts the same number of positive labels for protected and unprotected group instances. Statistical parity [21] would be guaranteed even if the classifier was determining the labels differently according to the group the instance belongs to [27] (e.g., by randomly classifying members of the protected group as positive but only classifying the ones in the unprotected group as positive based on some other feature). This problem of dismissing insensitive attributes is addressed by the remaining types of measures, however, at the cost of requiring expert knowledge to calculate.

Given a sample x, let $g \in \{0,1\}$ be the sensitive attribute. This attribute may not be binary and there may be multiple such attributes but we assume a single binary attribute, without loss of generality. Let $y \in \{0,1\}$ be the ground-truth label, and $\hat{y} \in \{0,1\}$ the predicted label for the

sample, where 1 is the positive class. Relevant statistical notions of fairness commonly used in literature are listed as follows.

Demographic Parity parity [21] (also known as Statistical Parity) is a statistical group fairness notion that is achieved when individuals from both protected and unprotected groups are equally probable to be assigned to the positive predicted class:

$$P(\hat{y} = 1|g = 0) = P(\hat{y} = 1|g = 1) \tag{2.1}$$

Disparate impact [53] mathematically represents the legal notion of disparate impact, which implies a more negative impact on the members of a protected class, even if the policy appears neutral. Like statistical parity, this measure is also based on both groups having similar positive prediction rates, but the ratio is observed instead of the difference. A classifier satisfies this definition if there is a similar proportion of positive predictions across groups:

$$\frac{P(\hat{y} = 1|g = 0)}{P(\hat{y} = 1|g = 1}) \geq 1 - \varepsilon \tag{2.2}$$

Predictive parity [16] is satisfied when both protected and unprotected groups have equal positive predictive value (PPV), which represents the probability of an individual whose predicted value is positive to truly belong to the positive class:

$$P(y = 1|\hat{y} = 1, g = 0) = P(y = 1|\hat{y} = 1, g = 1) \tag{2.3}$$

Predictive equality [16] requires both protected and unprotected groups to have the same false positive rate (FPR), which is related to the fraction of subjects in the negative class that were incorrectly predicted to have a positive value:

$$P(\hat{y} = 1|y = 0, g = 0) = P(\hat{y} = 1|y = 0, g = 1) \tag{2.4}$$

Equal opportunity [16] is obtained if both protected and unprotected groups have an equal false negative rate (FNR), the probability of an individual from the positive class to have a negative predictive value:

$$P(\hat{y} = 0|y = 1, g = 0) = P(\hat{y} = 0|y = 1, g = 1) \tag{2.5}$$

Equalized odds [31] is satisfied when protected and unprotected groups have equal true positive rates (TPR) and equal false positive rates (FPR):

$$P(\hat{y} = 1|y = c, g = 0) = P(\hat{y} = 1|y = c, g = 1),\ \forall c \in \{0, 1\} \tag{2.6}$$

Calibration [16, 65] requires that for any predicted probability score S, individuals in both protected and unprotected groups have an equal probability of correctly belonging to the positive class. For any given predicted probability score s in $[0, 1]$:

$$P(y=1|S=s,g=0) = P(y=1|S=s,g=1) \tag{2.7}$$

Research has shown that different measures have different pros and cons for each particular situation, and, more importantly, that these fairness definitions are not simultaneously achievable. For example, equalized odds and statistical parity cannot be both fulfilled when the proportions of actual positive samples are different [53], and statistical parity and individual fairness can only be simultaneously satisfied in trivial degenerate solutions [21].

2.2.3 Fairness-enhacing Mechanisms

In recent years many researchers have proposed new methods for tackling the ML fairness problem. These mechanisms are typically divided into three categories, according to which part of the ML pipeline they focus on: pre-processing, in-processing, and post-processing mechanisms [19, 47, 53]. The diagram in Fig. 2.3 depicts this categorization.

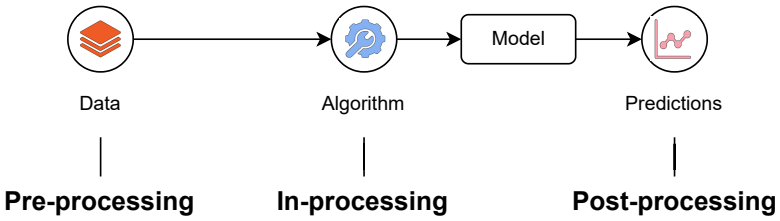


Figure 2.3: Categorization of fairness-enhancing mechanisms as pre-processing, in-processing, and post-processing, according to the part of the ML pipeline they focus on.

2.2.3.1 Pre-processing Mechanisms

Bias mitigation approaches in this category focus on eliminating bias by transforming the data before training [19, 53]. Pre-processing techniques can mostly be divided into two categories, according to the underlying approach: some change or reweight the labels, and others focus on altering feature representations.

Techniques that fall in the first category include massaging the dataset, a technique based on changing the labels of certain instances that are closer to the decision boundary in order to reduce discrimination [38]. Fairness can also be achieved without alteration of the labels by changing the data distribution through re-weighting the dataset, thus learning a fair classifier from the weighted data [38, 35]. Considering that some classifiers do not work directly with weights, sampling has further been proposed as an alternative to re-weighting by instead resampling the dataset to achieve the same non-discriminatory label distribution [38].

Recent work in developing pre-processing mechanisms suggests the use of increasing-to-balancing techniques [44]. These consist of inserting label noise to balance noise rates across classes. Such an approach is easy to implement and has been shown to improve both accuracy and fairness.

Confident learning (CL) [51] is an approach that aims at improving label quality by depicting existing label errors. This method combines the principles of pruning noisy data, counting with probabilistic thresholds to estimate noise, and ranking examples to train with confidence, to estimate the joint distribution between noisy and uncorrupted labels.

A different line of research suggests altering feature representations. One possible way of doing so is by using a convex optimization method to transform the biased data to a new mapping that enables the training of less discriminatory models. The goal is that the transformed data balances the trade-offs between achieving fairness, restricting distortion of individual samples, and preserving utility [13]. In this context, utility means that a model learned from the transformed dataset will have a similar predictive performance to the one learned from the original dataset.

A fair representation of data can also be obtained through Pairwise Fair Representation learning [40]. This approach takes advantage of human judgments on the similarity between instances as additional information to build a fairness graph. A Pairwise Fair Representation is then learned, combining the data-driven similarity between individuals with the pairwise side information from the graph.

A different approach is to perform recursive feature selection to train decoupled classifiers in such a way that satisfies specific preference guarantees [64]. Decoupling consists of training a separate classifier and recovering the most accurate model for each group using data from that group. The considered preference guarantees are *rationality* and *envy-freeness*. Rationality implies that each group prefers the classifier they were attributed to a pooled model that does not take group membership into account. Envy-freeness is related to each group preferring their assigned classifier to the model designated to any other group.

2.2.3.2 In-processing Mechanisms

This category of mechanisms may be applied when modifications to the learning procedures of the models can be made [8]. In-processing mechanisms ensure fairness during the training process by adapting algorithms to account for discrimination [19, 47, 53].

One typical approach is to treat the classification problem as a constrained optimization problem and enforce fairness constraints [7, 12, 41, 66, 69]. This has been achieved by rewriting the objective function to account for fairness [7, 69] or by exploiting the internal causal structure of data to model the label noise and counterfactual fairness simultaneously for causality-based fairness notions [69]. Zafar et al. [71] further describe how to design classifiers that satisfy *preference*-based fairness measures. This fairness definition, which they propose, assesses whether, in presence of the possible outcomes, a given group would collectively prefer the outcome they were attributed among the various options, irrespective of disparity with other groups.

This approach has also been applied to the problem of fair classification in the presence of noise. By considering standard fairness measures and assuming sensitive features are subject to the mutually contaminated learning model [58], Lamy et al. [41] have shown that it is possible to learn fair classifiers in the presence of noisy sensitive features. Their work demonstrated that fairness constraints on a clean distribution of the sensitive attribute are equivalent to a scaled

constraint on the noisy distribution. As such, fairness may be achieved by using any method for fair classification that takes a fairness-tolerance parameter and any existing noise estimation technique to determine how to adjust that parameter. This problem has also been addressed by performing empirical risk minimization with a fairness constraint [12] or by using specifically designed surrogate loss functions and surrogate constraints [66].

On the other hand, many approaches focus on applying reweighing techniques. Representation bias has been addressed in such a way by using an adaptative reweighing method that learns adaptive weights for each sample, achieving group-level balance among different demographic groups [14].

Another strategy for achieving fairness that has been recently proposed is the use of adversarial learning to learn a fair latent representation of the data that does not contain information about the sensitive attribute [10].

In some cases, sensitive features are available at training time but are not accessible for future data at prediction time. These situations are addressed by *privileged learning*. The privileged learning framework has been leveraged in the context of fair classification by treating protected features as privileged information, thus delivering fairness through unawareness [55]. In the conducted research, this was done by adding fairness constraints and regularizers to a privileged learning support vector machine (SVM) model.

Finally, fairness-aware hyperparameter optimization methods have also been proposed, enabling the integration of fairness objectives into ML pipelines [18].

2.2.3.3 Post-processing Mechanisms

When modifying the training data is not possible, and the learning process cannot be changed, post-processing mechanisms must be used to enhance fairness [8]. These methods rely on leveraging a holdout set taken from the training data that was not used in model training [19]. Some form of post-processing is then performed on the labels predicted for the test set to ensure fair results [47, 53].

The first post-processing approach to be proposed consists of taking an arbitrary learned predictor and constructing a classifier that fulfills the notion of equalized odds (Equation 2.6) through a post-processing step [31]. Subsequent work aimed to extend this method by applying a similar post-processing step to find the unique feasible solution to optimize both equalized odds and calibration (Equation 2.7) [54]. The authors observed that calibration and error-rate constraints are mostly incompatible. They found that even when substantially relaxing the equalized odds conditions (only requiring the weighted sums of the group error rates to match), calibration was still challenging to ensure. Despite achieving the intended goal of finding the unique feasible solution to optimize both fairness measures, they concluded that their method was inherently flawed. The predictions were technically fair, but the fact that a non-trivial portion of the individual predictions was randomly altered by the algorithm would have problematic implications in practice, especially in sensitive settings such as health care.

As a matter of fact, in succeeding work [68], it is concluded that post-processing methods may perform poorly in ensuring non-discrimination. To overcome the identified limitations, the authors propose a two-step framework. First, a possibly non-discriminatory predictor is estimated. In the second stage, discrimination is further reduced through post-processing correction.

More recently proposed post-processing methods include using plugin approaches that apply group-dependant thresholds to the predicted probability of each instance belonging to a class, such that the best trade-off between accuracy and fairness is achieved [48].

Unlike these methods that focus only on group fairness, further work has also been devised aiming to improve individual fairness through post-processing bias mitigation. Lohia et al. [45] propose deciding which samples to be altered to achieve group fairness based on the likelihood of individual fairness issues arising regarding each instance. By including such an individual bias detector in the used bias mitigation algorithm, both group and individual fairness can be achieved together.

2.3 Label Noise

In the previous section, we presented several methods that aim to ensure the fairness of machine learning classifiers, namely by imposing fairness constraints on the obtained predictions. Label noise is particularly important in the case of bias mitigation techniques, as data bias and label corruption are often closely related. This happens because the accuracy of certain labels is often affected by the subject belonging to a protected group [66]. However, bias mitigation techniques typically assume the existence of clean labels.

Noise can be defined as non-systematic errors that might complicate an algorithm's ability to uncover the relationship between the features and the class label of a sample [26]. When noise is related to wrongly assigned labels, we are in the presence of label noise. Label noise is a common phenomenon in real-world datasets, and the cost of acquiring non-polluted data is usually high. This makes it of great importance to develop methods that deal with this type of noise [1].

In this section, we define the most important concepts related to label noise, distinguishing the different types of noise in Section 2.3.1 and identifying some common causes of noisy labels in Section 2.3.2. Finally, in Section 2.3.3, we provide an overview of the currently existing methods for dealing with label noise.

2.3.1 Noise Types

Two different types of noise can occur in data: feature (or attribute) noise and label (or class) noise. Feature noise is related to the corruption of the data features, while label noise corresponds to wrongly assigned class labels. It has been shown that this second type of noise is usually more detrimental to the performance of classifiers [75], which can be explained by two factors.

On the one hand, each sample is comprised of numerous features but only a single label. Hence, if only a few features contain noise, the remaining ones counterbalance the impact of the

noise on learning. In fact, research shows that feature noise is more harmful the more correlated the feature under consideration is with the class label [75].

On the other hand, the substantial impact of the labels on learning contrasts with the features all having varying importance. Naturally, different learning algorithms attribute different significance to each feature, but the class label remains the most important [75]. Some attributes may not even provide any contribution to the resulting classifier. As such, feature noise usually has a far lesser impact than label noise.

Let y be the clean label and $\tilde{y}$ its noisy equivalent. Label noise can be further classified into one of three categories:

- **Random noise**, which corresponds to noise that is randomly distributed and does not depend on the instance's features or label [1], i.e., $P(\tilde{y}) = P(y)$;
- **Y-dependant noise** happens when instances belonging to a particular class are more likely to be mislabeled [1]. This type of label noise assumes that given y, the noisy label $\tilde{y}$ is conditionally independent of the instance x, i.e., $P(\tilde{y}|y,x) = P(\tilde{y}|y)$ [69];
- **XY-dependant noise** depends on both features and target values, meaning that the probability of a sample being mislabeled changes not only according to its particular class but also to the values of its features [1]. This is the type of noise commonly referred to as group-dependant [66] or instance-dependant [69] in the fairness literature. This type of label noise is often related to discrimination. Considering the COMPAS case, for example, the model unfairly predicts African-Americans as having a "high risk" of recidivism more often than Caucasians due to discrimination in past trials, which leads to models that reproduce the same kind of discrimination. In this situation, the probability of an offender being misclassified as "high risk", i.e., the label noise, depends on the *race* feature, so it is group-dependant.

2.3.2 Sources of Label Noise

Label noise is an intrinsic phenomenon to data collection processes, and it can be rooted in many causes [26]. We discuss some common causes of label noise that can occur both in human labeling and automatic data collection processes.

Firstly considering human labeling, the information provided to labelers might be of poor quality or insufficient to guarantee trustworthy labeling. The labeling task can be subjective or simply too complicated even for the field experts and thus be more prone to human errors in the process. In the medical field, for example, underdiagnosis is a severe systematic mislabeling problem where specific subgroups may be wrongly diagnosed by physicians more often than others. Training classifiers on such biased data causes models to not perform well across these subgroups [9]. Furthermore, having several labelers with different expertise levels on the subject might induce some conflict in the labeling. Despite the existence of techniques to measure and deal with the

lack of agreement between labelers [39], these disparities in the opinions of labelers may still induce noise in the labels. Finally, noise can be intentionally injected into datasets for privacy or data-poisoning reasons.

On the other hand, the high cost of having expert human labelers has led to the widespread adoption of automated classification frameworks. For example, the CheXpert [34] is a dataset of chest radiographs that were annotated using natural language processing methods on radiology reports. This dataset was revealed to contain a substantial amount of label noise, which negatively affects its reliability when benchmarked against the annotations of radiologists. This raises relevant concerns related to the automated labeling of data that will further be used to train models whose goal is to predict those same labels. Automatic data acquisition can also naturally have data encoding or dataset collection problems. Acquiring datasets from the web and social media might be beneficial because of the large amount of data available. However, it must be acknowledged that these datasets are often characterized by messy and unreliable information, which might introduce noise that complicates their utilization [1].

2.3.3 Dealing with Noisy Labels

Label noise can affect learning in many ways, from decreasing the performance of classifiers to increasing the complexity of the learned models [26]. Therefore, it is crucial to develop methods to deal with this type of noise to devise efficient and effective machine learning systems.

Several ways to categorize noise-dealing approaches have been proposed in the literature. One example is to divide the strategies into three main groups: methods that are naturally robust to label noise, filter methods to improve data quality, and models that directly model label noise [26]. In this work, we follow a different categorization, classifying the existing methods according to whether they model the noise structure or not [1]. The rest of this section explains these two categories and presents the most relevant noise-combating methods of each.

2.3.3.1 Noise Model-Based Methods

The goal of noise model-based methods is to extract information about the noise structure in the data to leverage it during training. This can be achieved, for example, by locating noise-free information or performing label noise correction. Despite the performance of such methods being tightly coupled with the correct noise estimation, they have the significant advantage of employing prior knowledge about the noise structure, also de-coupling the stages of noise estimation and classification [1].

Some of the proposed methods leverage a *noise transition matrix* that specifies the probability of a label being flipped to a different value. These techniques rely on estimating the noise transition matrix and modifying the loss based on its values [52, 43]. However, some noise transition matrix estimators have been devised under the *anchor-point* assumption: they presume the presence of instances that belong to a certain class with a probability of one, the *anchor points* [43]. Further work aimed at overcoming the need for this assumption by optimizing the cross-entropy loss

between the noisy label and the predicted probability, and the volume of the simplex formed by the columns of the transition matrix simultaneously [42].

Other approaches consist of correcting the labels that are most likely to be corrupted. It has been shown that when a classifier that was trained on noisy data has low confidence in the predicted label of an instance, it is likely to be corrupted [73]. Therefore its predictions can be used for deciding which labels to change based on a specified threshold value of confidence. A similar method uses the loss value to determine the probability of a sample being noisy, which can be achieved by fitting a two-component beta mixture model to each sample's loss, creating an unsupervised noise model to be used in loss correction [2]. A joint optimization framework to learn deep neural network parameters and estimate true labels has also been proposed [62]. The authors assume that by increasing the learning rate, the network will not fit noisy labels as easily. As such, the higher the loss, the higher the probability of a label being corrupted. This enables the correction of noisy labels to be performed while improving the performance of the classifier.

2.3.3.2 Noise Model Free Methods

These methodologies focus on algorithms that are inherently less sensitive to label noise and thus do not require the explicit modeling of the noise structure [1].

One approach to developing such algorithms is to focus on inherently noise-tolerant loss functions. It has been shown that 0-1 loss is robust to various types of label noise, while squared error loss is only tolerant to uniform noise, and most convex loss functions are not robust at all [46]. Further work demonstrated the noise-tolerance of the loss function based on the mean absolute value of error (MAE) [28].

Some methods assume a symmetry condition on the loss function [29]. A noise-tolerant convex barrier hinge loss has been proposed, which, while not symmetric everywhere, benefits particularly from the symmetric condition [15]. Regarding symmetric losses, a Symmetric Learning (SL) approach was suggested to address the problem of learning with Cross Entropy (CE) under noisy labels [67].

When training Deep Neural Networks (DNN), noise tolerance can be achieved by preventing the overfitting of noisy labels through regularization. Proposed methods include adding an additional softmax layer where dropout regularization is applied to prevent noise memorization and guarantee robustness [36], or leveraging the regularization effect of pre-training to enhance the model's tolerance to noise [33].

Ensemble learning approaches have also been used to address the problem of noise correction. Bagging has been proven to be robust to label noise while Boosting is far more sensitive [3]. To overcome its problem of overfitting noisy instances, noise-tolerant variations of AdaBoost have been proposed, such as a robust multi-class AdaBoost algorithm where a new weight updating scheme and a noise detection-based loss function have been included [59].

2.4 Dealing with Fairness in the presence of Label Noise

As discussed in the previous sections, while many methods have been proposed to guarantee the fairness of ML classifiers, these usually assume that the training data is not corrupted [66]. However, as previously examined, label noise is a common phenomenon in real-world data that may have negative consequences on model performance when not properly dealt with [26]. Taking this into consideration, some methods have been recently proposed that aim at fair ML in settings where data is corrupted by label noise.

One approach to achieve fair classification is to focus on re-weighting the training data to alter its distribution in a way that corrects for the noise process that causes the bias [35]. The authors have shown that training on the re-weighted dataset is equivalent to training on the unobserved unbiased labels. To evaluate how their method performed in comparison to previous approaches, they tested the various methods on a number of benchmark fairness datasets, measuring multiple fairness metrics.

A different line of work focuses on enforcing fairness constraints on the learning process to achieve fair predictions. Research has also been conducted in adapting this approach for learning fair classifiers in the presence of label noise [66, 69]. Some authors rewrite the loss function and fairness constraints to deal with label noise [69]. They further propose to model label noise and fairness simultaneously by uncovering the internal causal structure of the data. Surrogate loss functions and surrogate constraints have also been devised to ensure fairness in the presence of label noise [66].

2.4.1 Evaluation of Robustness

The performance of label noise correction methods depends on the level of noise in the data. They are expected to improve fairness by correcting possible biases. For practitioners to apply those methods safely in the real world, it is important to understand their behavior under different noise conditions. However, there is currently a lack of research in understanding how those techniques affect the fairness of models.

To address this limitation, a sensitivity analysis framework for fairness has been developed [25]. It assesses whether the conclusions about the fairness of a model derived from biased data are reliable. This is done by estimating bounds on the fairness metrics under assumptions about the magnitude of label noise. However, this approach still relies on a limited set of fairness benchmarks, limiting the scope of the conclusions since the existing datasets are not representative of many different types and levels of label noise.

In this work, we address the limitations of the empirical evaluation procedures that are usually conducted in the existing work. Instead of making assumptions about the level of label noise, we explicitly manipulate it.

2.5 Summary

In this chapter, we provided a short explanation of background knowledge regarding machine learning and analyzed the existing literature dealing with the problems of fairness and label noise in its context.

ML has enabled the automation of many essential decision-making processes, but these systems are subject to producing biased predictions that can have a significantly negative impact on disadvantaged people's lives. Recent research has been focused on identifying the causes of such bias, measuring the level of discrimination being generated by ML models, and devising methods that ensure the fairness of ML outputs.

A different problem that has been extensively studied in ML is label noise. We analyzed its possible sources, characterized the several types of label noise, and provided an overview of recent methods that have been proposed to train noise robust classifiers.

Finally, we presented some examples of relevant research in achieving algorithmic fairness in the presence of noisy labels. Few methods focus on simultaneously tackling both these problems, and there is a lack of research in developing procedures for the empirical evaluation of label noise correction methods when one of the goals is to achieve ML fairness.

Chapter 3

Systematic evaluation of the impact of label noise correction on ML Fairness

In this chapter, we propose a methodology for the systematic evaluation of label noise correction methods for ML fairness. We introduce the problem that motivates this work in Section 3.1 and detail the proposed methodology in Section 3.2. The experimental setup is described in Section 3.3 and the obtained results are analyzed and debated in Section 3.4

3.1 Problem Statement

As discussed in the previous chapter, label bias is a permeating problem that is becoming increasingly relevant due to the growing tendency to leverage ML classifiers in sensitive decision-making tasks. When the fairness of the decisions is not accounted for in the process of developing such models, the societal injustices present in the training data are reflected in the predictions of the models, perpetuating and possibly intensifying existing prejudice.

Many methods have been developed to deal with discrimination in ML classifiers, but these often assume that the data contains clean labels, which in many cases does not happen. Considering this scenario, some research has been conducted aiming to develop methods to ensure fairness in the presence of noisy labels [66, 69, 41].

In this work, we look at the fair classification problem in such a way that the bias present in the data is considered to be label noise, with the corruption rates being group-dependent. By assuming that there exist underlying, unknown, and unbiased labels that are overwritten by the observable biased ones, the natural approach is to apply label noise correction techniques to the data in order to obtain a clean dataset to be used in model training. The goal of this approach is to pre-process biased data to remove underlying discrimination, thus enabling classifiers trained on the corrected datasets to deliver predictions that are both accurate and fair.

Currently, there is a lack of research in understanding how the existing label noise correction techniques perform in achieving this goal of ensuring the fairness of models by correcting possible

biases present in the datasets used for training. As such, we address this problem by proposing a methodology for the systematic evaluation of label noise correction methods for ML fairness.

3.2 Methodology

With the objective of understanding the effect of existing label correction methods on improving the fairness of machine learning classifiers trained on the corresponding corrected data, we propose a methodology for empirically evaluating the efficacy of such techniques in achieving this goal. Our methodology can be applied to standard ML datasets, as well as to benchmark fairness datasets, and we describe the details of each case in the following subsections.

3.2.1 Using standard ML datasets

Having the *original* dataset, D_o, in which we assume the instances to have correctly assigned labels, the first step is to manipulate the labels. As we are applying this methodology to a standard classification dataset, we arbitrarily choose the positive class and a binary attribute to be considered as the sensitive one. Given noise rate τ, noise injection is performed by altering the label of instances with a certain probability depending on the noise rate and whether it belongs to the protected group. By parameterizing this process, we can simulate different types of discrimination. We thus obtain a *noisy* dataset, D_n, that is corrupted by the induced bias.

To simulate different types of biases, we inject group-dependant label noise in the clean datasets in two ways:

- **Asymmetrical Bias Noise**. This type of label noise is intended to simulate the cases where the instances belonging to the protected group are more likely to be given a positive label (or the ones not belonging to the protected group are systematically assigned to the negative class). For example, this would be equivalent to classifying African-American offenders as having a high risk of re-offending at a higher rate than their Caucasian counterparts. To simulate such bias, we set the label of each instance belonging to the protected group to the positive class with a probability equal to the desired noise rate. Naturally, as the noise rate gets higher, the data gets progressively more imbalanced.

- **Symmetrical Bias Noise**. In different situations, both the members of the protected group are benefited and the non-members are harmed. An example of this type of scenario is the automated selection of job applicants. If the selection process is biased towards preferring male applicants, there will be simultaneously more men being selected and more women being rejected. We simulate such bias by setting the label of each instance to the positive class if it belongs in the protected group or to the negative class otherwise, with a probability equal to the desired noise rate. We assume that the positive class is a good outcome, which is not always the case. Nevertheless, this is not expected to affect the conclusions.

The next step is to perform label noise correction by applying the method being analyzed on the dataset, obtaining a *corrected* dataset, D_c. This enables two types of evaluations of label correction methods: assessing whether the method is able to reconstruct the correct labels accurately and analyzing the impact of different levels of noise on the ability of the method to create data that leads to better models.

We first examine the similarity between the original labels and the ones obtained after applying label noise correction to the noisy data. Given a dataset with N instances, the ability to reconstruct the original labels is measured as the similarity between the *original* labels and the *corrected* ones, as shown in Eq. 3.1. Essentially, this is a measure of the accuracy of the label correction method in obtaining the original labels. However, to avoid confusion, we will refer to it as *reconstruction score*, r.

$$r = \frac{\sum_{i=1}^{N} \hat{y}_i = y_i}{N} \tag{3.1}$$

For each training set (D_o^{train}, D_n^{train}, and D_c^{train}), we then apply the chosen ML algorithm to it, obtaining the classifiers M_o, M_n, and M_c, respectively. These models are then evaluated under different scenarios.

Firstly, we want to consider the testing scenario where we only have access to corrupted data both for training and testing. The aim is to understand the effect of correcting training data in the case where the discrimination that was present when collecting the training data still exists at testing time. To achieve this, the *corrected* (M_c) and *noisy* (M_n) models are evaluated on the *noisy* test set, D_n^{test}. In this case, the intent is to observe if the noise correction methods are able to produce less discriminatory predictions without significant loss in predictive performance.

Our next objective is to understand the effect of correcting biased training data when the discrimination has been eliminated in the meantime and the testing data is unbiased. To achieve this, the models (M_o, M_n, and M_c) are evaluated on the *original* test set D_o^{test}.

Finally, we extend the previous scenario to remove the assumption that the original data is unbiased. In other words, we analyze the effect of correcting training data when the discrimination has been eliminated in the meantime but the original data was already biased and, thus, its labels are noisy. To achieve this, the *corrected* model, M_c, is evaluated on a test set with labels without noise. However, since we do not have access to the clean labels, we use a label noise correction method to correct the test data as well. We employ the same method that is being analyzed, but a more extensive empirical validation could use different methods or a combination of them. In any case, the results should be interpreted carefully, as the unbiased labels cannot be determined.

The diagram presented in Fig. 3.1 illustrates the explained methodology.

3.2.2 Using benchmark fairness datasets

When in the presence of an *originally biased* dataset, D_{ob}, the noise injection step is no longer needed as we assume the labels are already noisy. Since we are considering fairness benchmark

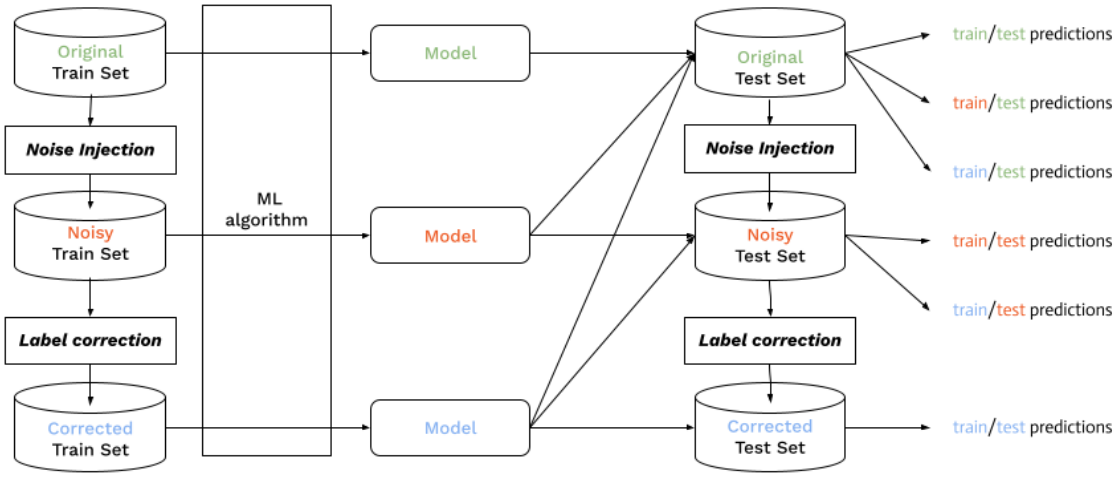


Figure 3.1: Diagram of the proposed methodology for empirically evaluating the efficacy of label noise correction methods in ensuring the fairness of classifiers using standard ML datasets.

datasets, we may use the data as expected, meaning that the sensitive attributes and positive class are the original ones.

The first step is to apply the considered label noise correction method on the training set, obtaining a *corrected* training set, D_c. Similarly to the methodology described for standard ML datasets, we then apply the selected ML algorithm to each training set (D_{ob}^{train} and D_c^{train}) to train the classifiers M_{ob} and M_c, respectively. These models are once again evaluated under different scenarios.

We begin by considering the testing scenario where only corrupted data is accessible both for training and testing. The *corrected* (M_c) and *originally biased* (M_{ob}) models are evaluated on the *originally biased* test set, D_{ob}^{test}. This way, we analyze whether the predictions of the models trained with corrected data are able to improve fairness and maintain predictive performance.

We then analyze the effect of correcting training data when the discrimination has been eliminated in the meantime. Since the original data is already biased, the *corrected* model, M_c, is evaluated on the *corrected* test set, D_c^{test}. The same label noise correction method that was applied to the training set is applied to the test set as well. As in the previous case, we could extend this methodology to use a combination of different methods.

This methodology is depicted in the diagram presented in Fig. 3.2.

3.3 Experimental Setup

To illustrate the use of the proposed methodology, we perform an empirical evaluation of six label noise correction methods to ensure the fairness of ML models. In this section, we describe its key aspects, explaining the considered label noise correction methods, characterizing the used datasets, and detailing how we evaluated the methods. The implementation of the proposed methodology is available at https://github.com/reluzita/fair-lnc-evaluation.

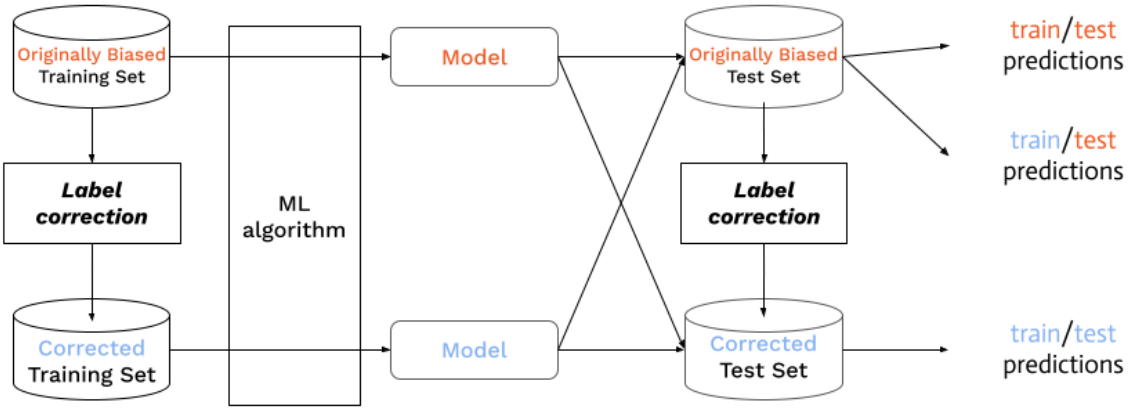


Figure 3.2: Diagram of the proposed methodology for empirically evaluating the efficacy of label noise correction methods in ensuring the fairness of classifiers using benchmark fairness datasets.

3.3.1 Label noise correction methods

We focus on problems where fairness issues are essentially caused by label noise, with the corruption rates being group-dependent. By assuming that there exist underlying, unknown, and unbiased labels that are overwritten by the observable biased ones, the natural approach is to apply label noise correction techniques to the data in order to obtain a clean dataset to be used in model training. The goal of this approach is to pre-process biased data to remove underlying discrimination, thus enabling classifiers trained on the corrected datasets to deliver predictions that are both accurate and fair. In the conducted experiments, we compared the following label noise correction methods:

- **Bayesian Entropy Noise Correction** (BE) [60]. In this method, multiple Bayesian classifiers are obtained with different training samples. These classifiers are used to obtain a probability distribution for each sample of it belonging to each considered class, which is applied in calculating the instance's information entropy. If the entropy of a sample is below the calculated threshold and its label is different from the predicted one, its value is corrected. These steps are repeated until a stopping criterion has been met;

- **Polishing Labels** (PL) [50]. This method replaces the label of each instance with the most frequent label predicted by a set of models obtained with different training samples;

- **Self-Training Correction** (STC) [50]. This algorithm works by first dividing the data into a noisy and a clean set using a noise-filtering algorithm. These methods identify and remove noisy instances from data, and in this case, the Classification Filter [63] is used. A model is obtained from the clean set and is used to estimate the confidence that each instance in the noisy set is correctly labeled. The most likely mislabeled instance is relabeled to the class determined by the classifier and added to the clean set. These steps are repeated until the desired proportion of labels is corrected;

- **Clustering-Based Correction** (CC) [50]. Firstly, a clustering algorithm is executed on the data multiple times, varying the number of clusters. A set of weights is calculated for each cluster based on its distribution of labels and size and is attributed to all the instances that belong to it. These weights are meant to benefit the most frequent class in the cluster. The weights obtained from each clustering are added up for each instance, and the label with the maximum weight is chosen;

- **Ordering-Based Label Noise Correction** (OBNC) [23]. The first step in this algorithm is to learn an ensemble classifier from the data. For each instance, the ensemble decides the label by voting, and the difference between the votes can be used to calculate an ensemble margin. The misclassified samples are ordered in a descending manner based on the absolute value of their margin. The most likely mislabeled instances are relabeled to their predicted classes;

- **Hybrid Label Noise Correction** (HLNC) [70]. In this approach, the first step is to separate the data into high-confidence and low-confidence samples. This is achieved by applying the k-means algorithm to divide the data into clusters and determining each cluster's label. The instances are classified as high-confidence if their label matches the cluster's label and low-confidence otherwise. The high-confidence samples are used to simultaneously train two very different models, using the semi-supervised k-means (SSK-means) [6] and Co-training [11] algorithms. These are applied to each low-confidence sample, and if both algorithms give it the same label, then the sample is relabeled and set as high-confidence. This process is repeated until all labels are high-confidence or after a specified number of times.

We implemented these label noise correction methods as described in their original papers, and the code is available at https://github.com/reluzita/label-noise-correction.

3.3.2 Algorithm and Parameters

In the conducted experiments, we used the Logistic Regression algorithm to obtain the classifiers. We applied stratified sampling to split the datasets into train and test sets, maintaining the original class ratio and using 20% of the instances for testing.

We implemented the BE method considering a value of 0.25 for the $alpha$ parameter, which is used to calculate the threshold T according to which a label is corrected or not. We used 10 folds to partition the dataset. For the PL method, we divided the dataset into 10 folds and applied the Logistic Regression algorithm to obtain the classifiers. Similarly, for the STC method, we also used 10 folds and the Logistic Regression algorithm. Additionally, the correction rate, i.e., the proportion of noisy instances to correct, was 0.8. Regarding the CC method, we ran it for 10 iterations, creating 100 clusters. Similarly, we used 100 clusters when clustering was performed in the HLNC method. Finally, for the OBNC method, we specified the proportion of labels to correct

as 20%. All these parameters were chosen to be consistent with the values used in the experiments described in the original papers of the considered label noise correction methods.

3.3.3 Datasets

The datasets used in the experiments are available on OpenML [1]. The standard ML datasets used in the noise injection experiments are summarized in Table 3.1 and the fairness benchmark datasets in Table 3.2. We selected such datasets to have a considerable range of examples with varying dimensions (in terms of both instances and features) and varying degrees of class imbalance.

Table 3.1: Characterization of the standard ML datasets used in the conducted experiments. Abbreviations:
$(+,\cdot)$ - instances in positive class
$(\cdot,p)$ - instances in protected group
$(+,p)$ - instances in positive class and protected group
$(+,u)$ - instances in positive class and unprotected group

dataset	OpenML id	# instances	# features	$(+,\cdot)$	$(\cdot,p)$	$(+,p)$	$(+,u)$
ads	40978	1377	1558	33 %	76 %	34 %	33 %
bank	1461	15111	30	33 %	51 %	24 %	43 %
biodeg	1494	1055	41	34 %	15 %	5 %	39 %
churn	40701	2121	22	33 %	23 %	21 %	37 %
credit	29	653	43	45 %	31 %	47 %	45 %
monks1	333	556	6	50 %	49 %	49 %	51 %
phishing	4534	11055	30	56 %	66 %	59 %	49 %
sick	38	636	26	33 %	39 %	12 %	47 %
vote	56	312	14	58 %	52 %	54 %	63 %

Table 3.2: Characterization of the fairness benchmark datasets used in the conducted experiments, according to the considered sensitive attribute. Abbreviations:
$(+,\cdot)$ - instances in positive class
$(\cdot,p)$ - instances in protected group
$(+,p)$ - instances in positive class and protected group
$(+,u)$ - instances in positive class and unprotected group

dataset	OpenML id	# instances	# features	$(+,\cdot)$	sensitive attribute	$(\cdot,p)$	$(+,p)$	$(+,u)$
adult	43898	45175	104	25 %	sex	68 %	31 %	11 %
					race	86 %	26 %	16 %
german	31	1000	58	70 %	sex	69 %	72 %	65 %
compas	45039	4966	11	50 %	sex	81 %	53 %	39 %
					race	40 %	42 %	55 %
ricci	42665	118	7	47 %	race	58 %	60 %	30 %
diabetes	43903	34071	48	33 %	race	75 %	34 %	32 %
titanic	40945	1309	11	38 %	sex	64 %	19 %	73 %

[1] https://www.openml.org/

3.3.4 Evaluation Measures

To evaluate the obtained models, we tested the predictive performance of the predictions by calculating the Area Under the ROC Curve (AUC) metric [30]. In terms of fairness, the following metrics were analyzed:

- **Demographic Parity** (also known as statistical parity) is a statistical group fairness notion that is achieved when individuals from both protected and unprotected groups are equally likely to be predicted as positive by the model [21]. We analyze the Demographic Parity difference between the two groups:

$$DP_{dif} = |P(\hat{y} = 1|g = 0) - P(\hat{y} = 1|g = 1)| \tag{3.2}$$

- **Equalized Odds** [31] is satisfied when protected and unprotected groups have equal true positive rates (TPR) and equal false positive rates (FPR). To calculate the Equalized Odds difference, EOD_{dif} between groups, we first obtain the TPR difference:

$$TPR_{dif} = |P(\hat{y} = 1|y = 1, g = 0) - P(\hat{y} = 1|y = 1, g = 1)| \tag{3.3}$$

  And the FPR difference:

$$FPR_{dif} = |P(\hat{y} = 1|y = 0, g = 0) - P(\hat{y} = 1|y = 0, g = 1)| \tag{3.4}$$

  Returning the largest of both values:

$$EOD_{dif} = max(TPR_{dif}, FPR_{dif}) \tag{3.5}$$

- **Predictive Equality** [16] requires both protected and unprotected groups to have the same false positive rate (FPR), which is related to the fraction of subjects in the negative class that were incorrectly predicted to have a positive value. We obtain the Predictive Equality difference:

$$PE_{dif} = |P(\hat{y} = 1|y = 0, g = 0) - P(\hat{y} = 1|y = 0, g = 1)| \tag{3.6}$$

- **Equal Opportunity** [16] is obtained if both protected and unprotected groups have an equal false negative rate (FNR), the probability of an individual from the positive class to have a negative predictive value. We use the Equal Opportunity difference:

$$EOP_{dif} = |P(\hat{y} = 0|y = 1, g = 0) - P(\hat{y} = 0|y = 1, g = 1)| \tag{3.7}$$

3.4 Results

In this section, we examine the results obtained from the conducted experiments. Firstly, we evaluate the label noise correction methods on standard ML datasets, injecting label noise at increasing noise rates, and we further test their performance on fairness benchmark datasets. Finally, we provide some insights into the observed outcomes and discuss possible limitations of the developed work.

3.4.1 Using standard ML datasets

Our goal is to analyze the robustness of label correction methods in terms of predictive accuracy as well as fairness, considering models trained in 3 different ways – M_o, M_n, M_c –, first analyzing the similarity between the original labels and the ones obtained after applying each label noise correction method to the noisy data.

3.4.1.1 Similarity to original labels after correction

Fig. 3.3 shows, on average, how similar each method's correction was to the original labels, considering both types of bias. Regardless of the type of bias, OBNC was the method that was able to achieve higher similarity to the original labels.

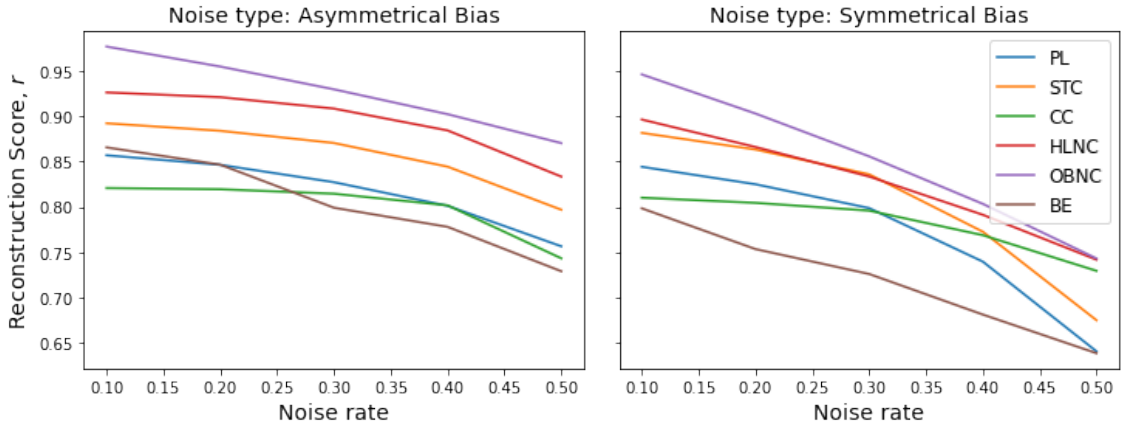
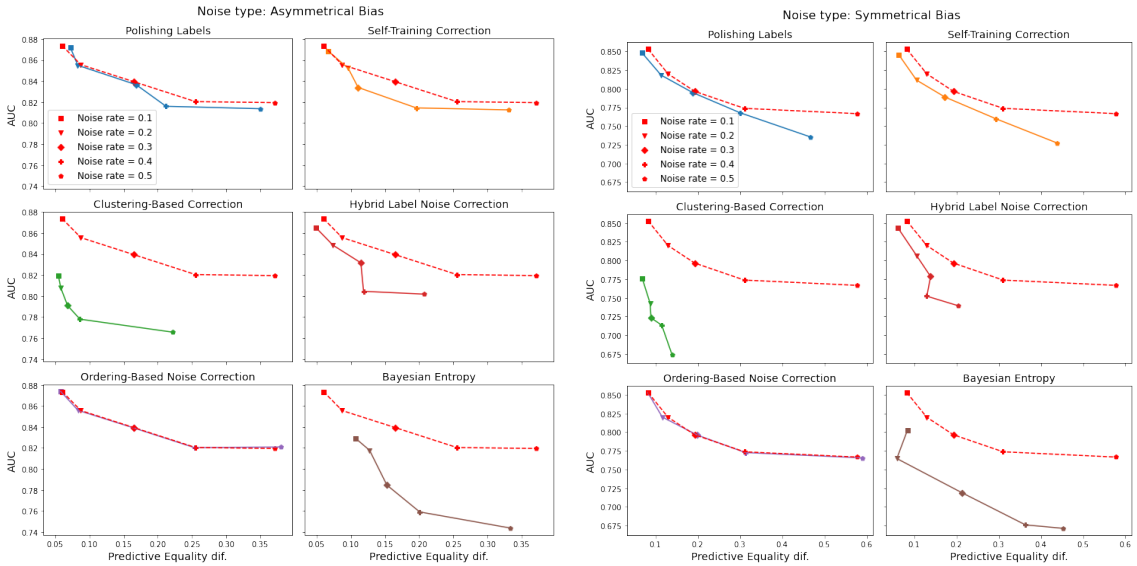


Figure 3.3: Reconstruction score (r), representing the similarity between original labels and the ones obtained after applying each label noise correction method for different noise rates.

3.4.1.2 Evaluation on the noisy test set

In some cases, we may only have access to biased data both for training and testing the models. As such, we evaluate the predictive performance and fairness of the predictions of the models on the *noisy* test set. The trade-off between the AUC metric and the Predictive Equality difference metric for different noise rates is shown in Fig. 3.4a, for the Asymmetrical Bias noise, and in Fig. 3.4b, for the Symmetrical Bias noise. In the remainder of this section, we only present the

results in terms of Predictive Equality difference since the same conclusions can be derived from the results that were obtained using any of the aforementioned fairness metrics (which can be found in Appendix A).



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

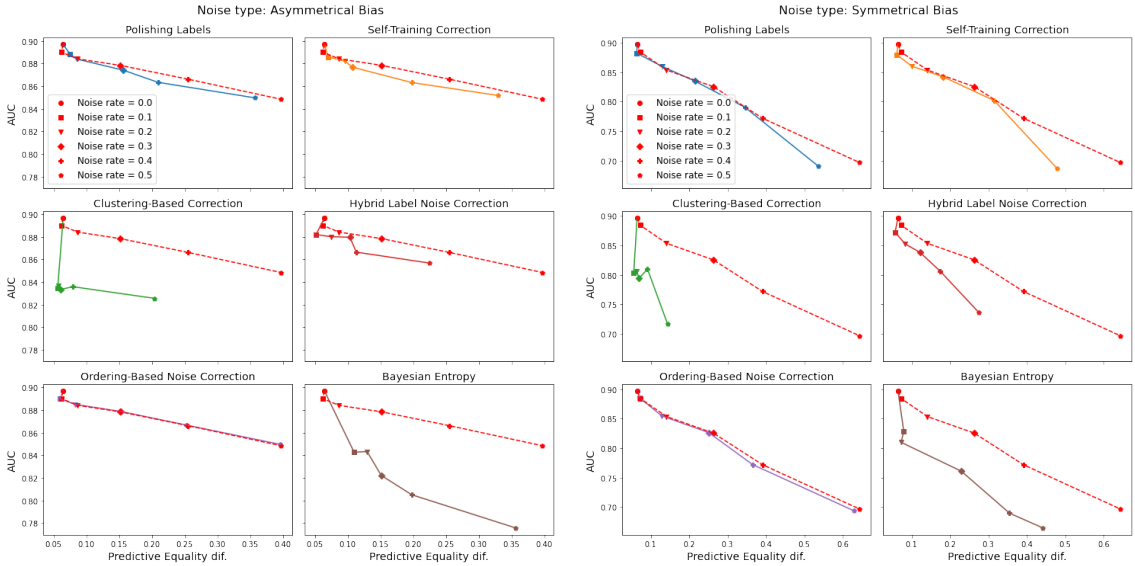
Figure 3.4: Trade-Off between AUC and Predictive Equality difference obtained on the *noisy* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

The OBNC method achieved performance similar to using the *noisy* data, while PL and STC show small improvements, mainly in terms of fairness. The CC method performs the best in achieving fairness, being able to keep discrimination at a minimum even at higher noise rates, as shown in Figure 3.4b, but losing significant predictive performance to do so. The HLNC method maintained its ability to improve fairness at minimum expense to the predictive performance of the resulting models.

3.4.1.3 Evaluation on the original test set

To understand how the label noise correction methods fare on producing accurate and fair predictions from biased data in an environment where these biases are no longer present, we evaluate the performance of the three obtained models on the original test set. The trade-off between the AUC metric and the Predictive Equality difference for each method at different noise rates is shown in Fig. 3.5a, for the Asymmetrical Bias noise, and in Fig. 3.5b, for the Symmetrical Bias noise.

In this testing scenario, the methods still behave in a similar way to the previous one in relation to each other. The OBNC method was shown to correct the labels in a way that is the most similar to the *original* train set. Still, the performance of the resulting model is comparable to using the *noisy* train set. The PL and STC methods achieve a slightly better trade-off between predictive



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

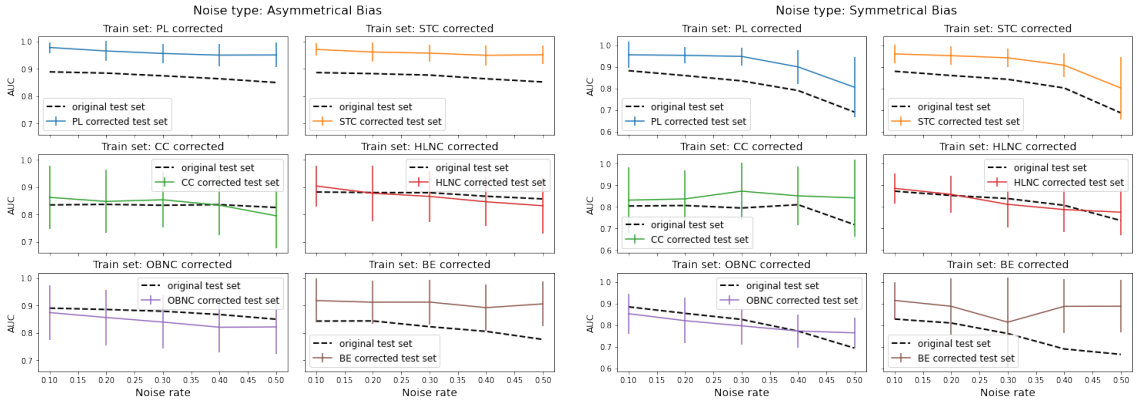
Figure 3.5: Trade-Off between AUC and Predictive Equality difference obtained on the *original* test set when correcting the data injected with each noise type at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

performance and fairness. On the other hand, the CC method shows significant improvements in terms of fairness, but at the expense of a lower AUC score. The BE method achieves a low score in both metrics. Finally, the HLNC method was found to be the best at simultaneously improving both predictive performance and fairness.

3.4.1.4 Evaluation on the corrected test set

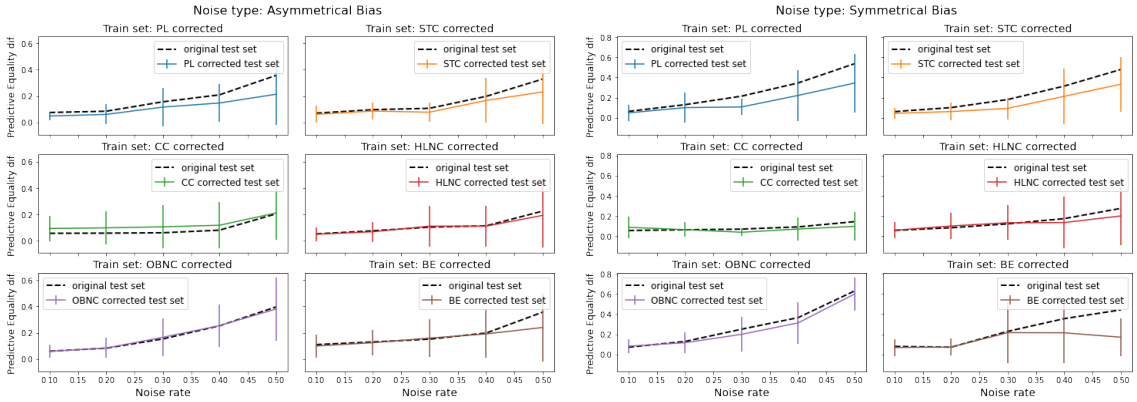
Finally, we investigate the possibility of applying label noise correction methods on the corrupted test set to simulate having an unbiased testing environment when only corrupted data is available for testing. To do so, we evaluate the performance of the models obtained using corrected train data on the test set corrected using the same method. We then assess whether that performance is similar to the one obtained when testing the same models on the original test set. The results for the AUC metric are presented in Fig. 3.6a, for the Asymmetrical Bias noise type, and in Fig. 3.6b, for the Symmetrical Bias noise type. Considering the Predictive Equality difference metric, the results are shown in Fig. 3.7a for the Asymmetrical Bias noise type, and in Fig. 3.7b, for the Symmetrical Bias noise type.

In terms of AUC, the PL, STC, and BE methods tend to result in an overestimation of the predictive performance of the resulting model. At the same time, the OBNC method appears to slightly underestimate it. The HLNC method shows similar performance to testing on the *original* test set in the presence of both types of noise, while the CC method only achieves this when dealing with Asymmetrical Bias noise. Regarding the Predictive Equality difference metric, all



(a) Injecting Asymmetrical Bias noise. (b) Injecting Symmetrical Bias noise.

Figure 3.6: Comparison in AUC between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.



(a) Injecting Asymmetrical Bias noise. (b) Injecting Symmetrical Bias noise.

Figure 3.7: Comparison in Predictive Equality difference between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.

methods show a performance very similar to using the *original* test set. A slight underestimation of discrimination can be seen for the PL, STC, and BE methods for the higher noise rates.

3.4.2 Using fairness benchmark datasets

We now analyze the robustness of label correction methods in terms of predictive accuracy and fairness by considering models trained in 2 different ways – M_{ob} and M_c.

3.4.2.1 Evaluation on the originally biased test set

We first consider the cases where we only have access to biased data both for training and testing the models, evaluating the predictive performance and fairness of the predictions of the models on

the *originally biased* test set. The trade-off between the AUC metric and the Predictive Equality difference is shown in Fig. 3.8.

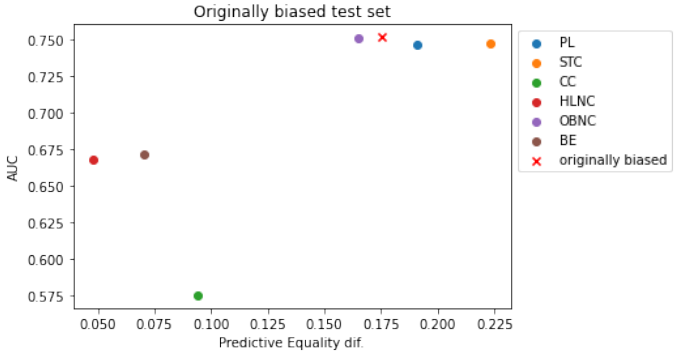


Figure 3.8: Trade-Off between AUC and Predictive Equality difference obtained on the *originally biased* test set.

In this scenario, we observe that the PL, STC, and OBNC methods are able to achieve the highest AUC scores, similar to the score obtained when using the *originally biased* training set. However, only the OBNC method, out of the three, is able to reduce discrimination (still very slightly). The CC method further improves fairness but at the cost of having the worst predictive performance out of all the methods. The HLNC and BE methods achieve similar predictive performance, but the HLNC method is able to reduce discrimination the most compared to the other five methods.

3.4.2.2 Evaluation on the corrected test set

We further investigate the possibility of applying label noise correction methods on the *originally biased* test set to simulate having an unbiased testing environment when only corrupted data is available for testing. To do so, we evaluate the performance of the models obtained using corrected train data on the test set corrected using the same method. The trade-off between the AUC and Predictive Equality difference metrics is presented in Fig. 3.9.

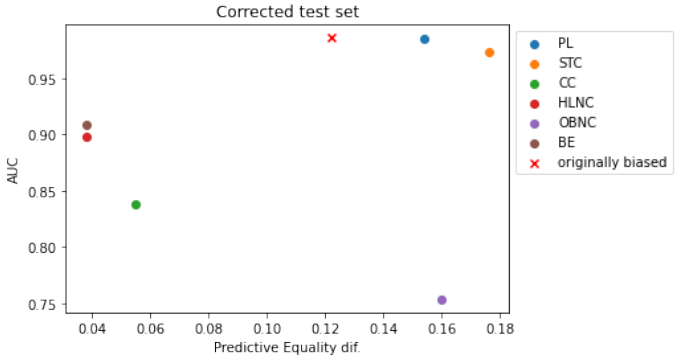


Figure 3.9: Trade-Off between AUC and Predictive Equality difference obtained on the *corrected* test set.

Firstly, we observe how the PL and STC methods appear to maintain a similar behavior to the previous scenario, achieving a predictive performance similar and higher discrimination compared to using the *originally biased* training set. However, considering that these methods were shown to overestimate the AUC score and underestimate the Predictive Equality difference when correcting the test set (Figs. 3.6 and 3.7), this trade-off would likely be worst if the methods were tested on noise-free data.

On the other hand, unlike in the previous experiment, the OBNC method achieved the lowest AUC score of the six methods (which might be an underestimation, according to the results of Fig. 3.6) and increased discrimination. Similarly, the BE method obtained a better trade-off between predictive performance and fairness compared to the previous case, but this might be due to its tendency to overestimate the AUC score when correcting the test set.

Finally, both the HLNC and CC methods were observed to correct labels in a way that is mainly similar to having a test set without noise. The HLNC method once again achieves the best fairness score out of all the methods without losing as much predictive performance as the CC method.

3.4.3 Discussion

The ability to correct the labels does not necessarily guarantee a good compromise between accuracy and fairness. For instance, the OBNC method obtained the highest similarity with the original labels. However, when assessing the compromise between predictive performance and fairness, the OBNC method had a much less satisfactory performance, showing barely any difference from training with the noisy training set.

On the other hand, the CC method, which did not show a high reconstruction score, kept discrimination at minimum values, even at the highest noise rates. However, this was achieved at the cost of lower predictive performance. The nature of the fairness metrics can explain this: e.g., the Predictive Equality metric calculates the difference between the FPR of each group, meaning that if both groups have a high but similar FPR, the predictions are technically fair but not accurate.

We must acknowledge some limitations of this study, as they can impact the generalizability of our findings. The first one is related to an important advantage of the proposed methodology: it may use standard benchmark datasets to assess the robustness of label correction methods. This means that analysis can be based on a much larger set of datasets than typical fairness studies use. However, the choice of both the sensitive attribute and the positive class is arbitrary. This means that these datasets do not necessarily have similar distributions to real problems with label noise caused by discrimination. However, the methodology can also be applied to benchmark fairness datasets to assess the generality of the results obtained. Additionally, the predicted classes were based on a threshold of 0.5, which is not realistic in many problems where discrimination might be an issue. As the choice of threshold impacts the fairness metrics, it is important to obtain results with other thresholds. In the case of benchmark fairness datasets, problem-specific thresholds can also be used.

Chapter 4

Fair Label Noise Correction

In this chapter, we explain our motivation and proposed method for accounting for fairness in label noise correction in Section 4.1, detailing the conducted experiments in Section 4.2 and analyzing the obtained results in Section 4.3.

4.1 Fair Ordering-Based Noise Correction

In the conducted empirical evaluation of label noise correction methods described in Chapter 3, the Ordering-Based Noise Correction (OBNC) method [23] performed in a peculiar way. Despite being the method that was able to achieve the highest reconstruction score out of all the considered label noise correction methods, it didn't show to be particularly good at ensuring the fairness of the resulting models.

The OBNC method takes advantage of ensemble margins as a measure of how likely an instance is to be mislabeled [23]. This method works by first training an ensemble classifier with the noisy data. The ensemble predicts a label for each instance, and the ensemble margin is calculated for all the mislabelled ones. The authors discuss four ways of calculating the ensemble margins. In this work, we only consider the supervised one, which was introduced in [5]. Since we are considering the particular case of binary classification, this ensemble margin can be defined by Equation 4.1, where v is the total number of votes and v_y is the number of votes for the true class y.

$$margin(x) = \frac{2v_y - v}{v} \tag{4.1}$$

These margin values are in the range $[-1, 1]$, being that misclassified instances have negative margin values. These samples are ranked in descending order according to a class noise evaluation function that relies on the ensemble's margin, as defined in Equation 4.2, where Tr is the training set of (x, y) pairs, and C is the ensemble classifier.

$$N(x_i) = |margin(x_i)|, \forall (x_i, y_i) \in Tr | C(x_i) \neq y_i \tag{4.2}$$

Higher values of $N(x_i)$ indicate a higher probability of y_i being noisy. Finally, the M samples with the highest values of $N(x_i)$ are corrected, flipping their labels to the ones predicted by the ensemble. A diagram depicting this described methodology is shown in Fig. 4.1.

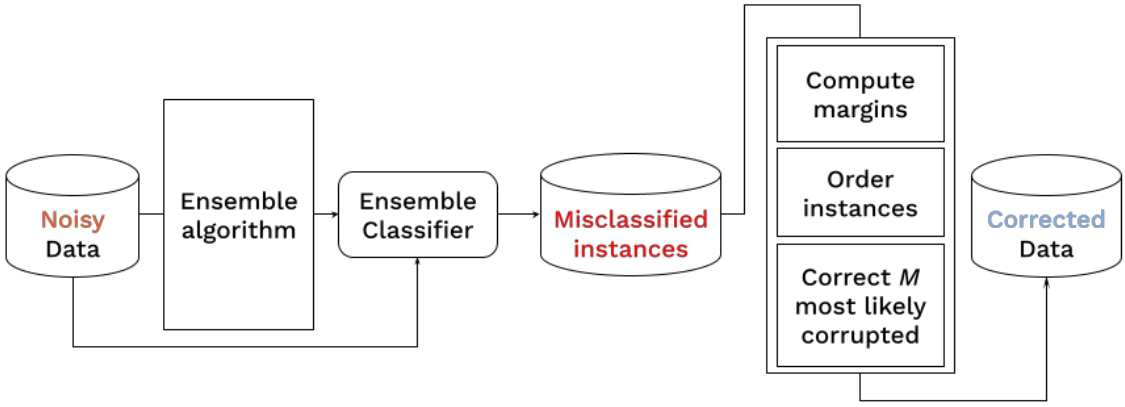


Figure 4.1: The Ordering-Based Noise Correction method.

With the objective of reducing discrimination present in the labels through label noise correction, we propose Fair-OBNC, a variation of the OBNC method that takes fairness into consideration.

An intuitive and straightforward approach to prevent classifiers from learning associations between the sensitive attribute and the class labels is to simply remove the sensitive attribute from the training data. This is done so that sensitive attributes do not influence the decisions of the resulting models, avoiding the perpetuation of existing biases related to that attribute. As such, we first make a simple adjustment to the original OBNC algorithm, simply ignoring the sensitive attribute when training the ensemble of classifiers and calculating the margins. We name the version of our method that only removes the sensitive attribute Fair-OBNC-rs.

Moreover, let us consider the notion of Demographic Parity [21], which implies that instances of the unprotected group are as probable to be assigned to the positive predicted class as the members of the protected group. In this following modification, which we name Fair-OBNC-dp when performed by itself, we apply additional criteria for correcting each label such that the distribution of positive labels across the groups is as balanced as possible. The pseudo-code depicting how this criterion is applied is shown in Algorithm 1.

After ordering the misclassified labels by decreasing margins, we assess whether performing the correction will be beneficial in achieving a balanced distribution of labels. If the probability of an instance that belongs to the protected group having a positive label is higher than that of an instance of the unprotected group, we will correct protected samples that have a positive label and unprotected ones that have a negative label. The opposite happens when the probability of an unprotected sample belonging to the positive class is higher than that of a negative one.

Our method, Fair-OBNC, combines these two modifications, ignoring the sensitive attribute when training the ensemble and then performing the corrections in a way that decreases the difference in the proportion of positive labels between both groups. The implementation of Fair-OBNC

Algorithm 1 Fair-OBNC-dp: criterion for correction

Require: ordered misclassified instances' index vector M, proportion of instances to correct m, sensitive attribute vector A, corresponding label vector Y

$corrected \leftarrow 0$
for $i \in M$ **do**
  $dp_1 \leftarrow P(y = 1|a = 1), y \in Y, a \in A$
  $dp_0 \leftarrow P(y = 1|a = 0), y \in Y, a \in A$
  if $dp_1 > dp_0$ and $Y[i] = A[i]$ or $dp_1 < dp_0$ and $Y[i] \neq A[i]$ **then**
    $Y[i] \leftarrow 1 - Y[i]$
    $corrected \leftarrow corrected + 1$
  end if
  if $corrected \geq m * |Y|$ **then**
    break
  end if
end for

can be found at https://github.com/reluzita/label-noise-correction.

4.2 Experimental Setup

We evaluate our Fair-OBNC method using our previously described evaluation methodology. As a baseline, we use the original OBNC method, implemented as described in the original paper. We further test the application of each of the proposed variations, Fair-OBNC-rs and Fair-OBNC-dp, to analyze the effects of applying each modification separately. We either only remove the sensitive attribute (Fair-OBNC-rs) or only apply the fairness criterion for correcting each label (Fair-OBNC-dp). We set the number of labels to correct as 20% of the total number of labels in the dataset for both the original OBNC and our Fair-OBNC method, as the same was done in the original OBNC paper.

The experiment with each dataset was carried out as follows. We split the data into training and test sets (80%/20%) with stratified sampling. After label correction, a Logistic Regression algorithm is used to learn a model from the training set.

For these experiments, we tested our method in the same datasets used in Chapter 3, using the ones summarized in Table 3.1 for the noise injection experiments, and the fairness benchmark ones described in Table 3.2.

The evaluation measure is the Area Under the ROC Curve (AUC) metric [30]. The fairness of the predictions was assessed in terms of Predictive Equality difference (Eq. 3.6), Demographic Parity difference (Eq. 3.2), Equalized Odds difference (Eq. 3.5) and Equal Opportunity difference (Eq. 3.7).

4.3 Results

In this section, the results obtained from the conducted experiments are examined. We first consider the experiments using standard ML datasets, where noise is injected at increasing noise rates. The methods are additionally evaluated on the fairness benchmark datasets. Finally, the observed effects are discussed and some limitations of the work are pointed out.

4.3.1 Using standard ML datasets

In these noise injection experiments, we begin by comparing the label noise correction methods in terms of achieved similarity between the original labels and the corrected ones, further analyzing their performance on the *noisy*, *original*, and *corrected* test sets.

4.3.1.1 Similarity to original labels after correction

Firstly, we compare the *corrected* dataset to the *original* one, being that the correction is applied to datasets with different levels of injected noise. Fig. 4.2 shows, on average, how similar each method's correction was to the original labels, considering both types of bias.

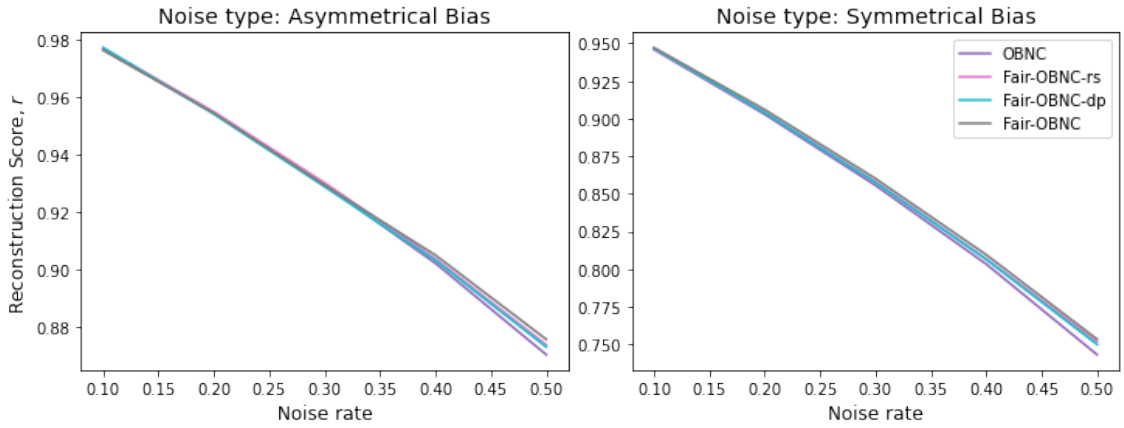


Figure 4.2: Reconstruction score (r), representing the similarity between original labels and the ones obtained after applying each label noise correction method for different noise rates.

From the aggregated results, both the original OBNC method and our proposed variation appear to achieve a very similar reconstruction score. At higher noise rates, a slight difference between these values can be observed: the original OBNC achieves the lowest reconstruction score, while the Fair-OBNC method achieves the highest value.

We further analyzed the results for each dataset separately. The results are shown in Figs. 4.3 and 4.4 for the Asymmetrical and Symmetrical Bias types, respectively, and we observe that in many of the datasets, both methods perform the same. In fact, if we analyze the characteristics of the datasets, the ones where some difference is observable between methods are the ones where both the proportion of positive instances and the proportion of instances in the protected group are above 50%.

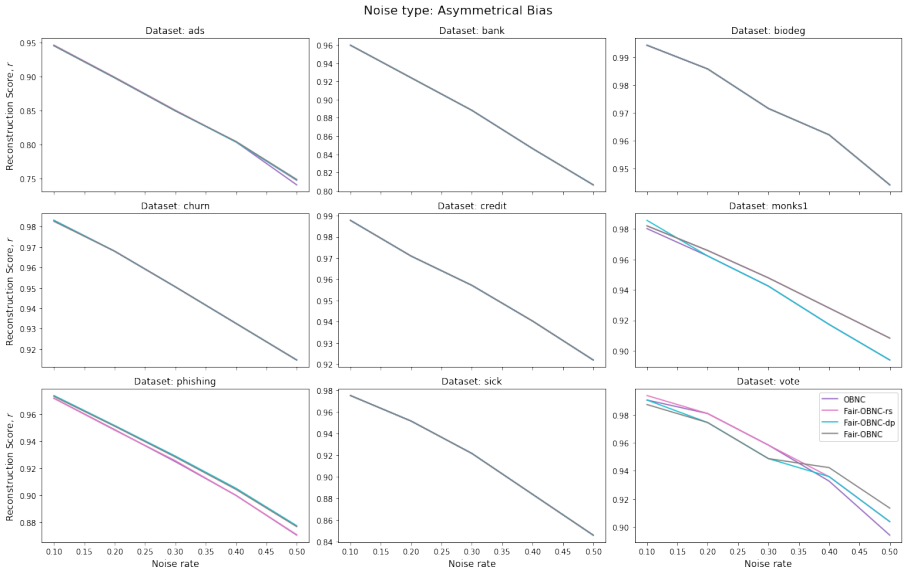


Figure 4.3: Reconstruction score (r), representing the similarity between original labels and the ones obtained after applying each label noise correction method on each dataset, with Asymmetrical Bias noise injected at increasing rates.

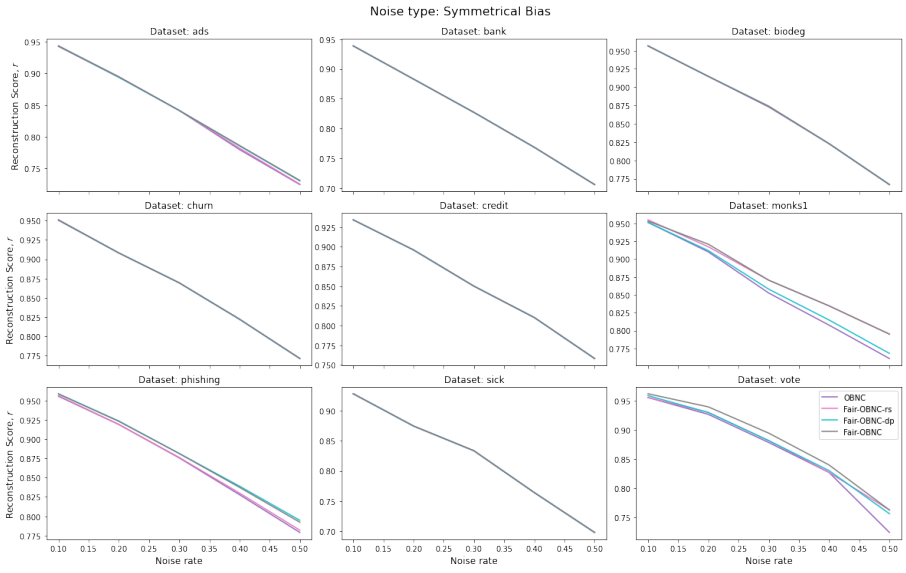


Figure 4.4: Reconstruction score (r), representing the similarity between original labels and the ones obtained after applying each label noise correction method on each dataset, with Symmetrical Bias noise injected at increasing rates.

4.3.1.2 Evaluation on the noisy test set

This scenario represents cases where only biased data is available for training and testing (i.e. we use the *noisy* training and test sets). The trade-off between the AUC metric and the Predictive Equality difference metric for different noise rates is shown in Fig. 4.5. The results presented in this section only consider the Predictive Equality difference metric as a fairness measure, as the

same conclusions can be derived from the results that were obtained using any of the remaining fairness metrics (which can be found in Appendix B).

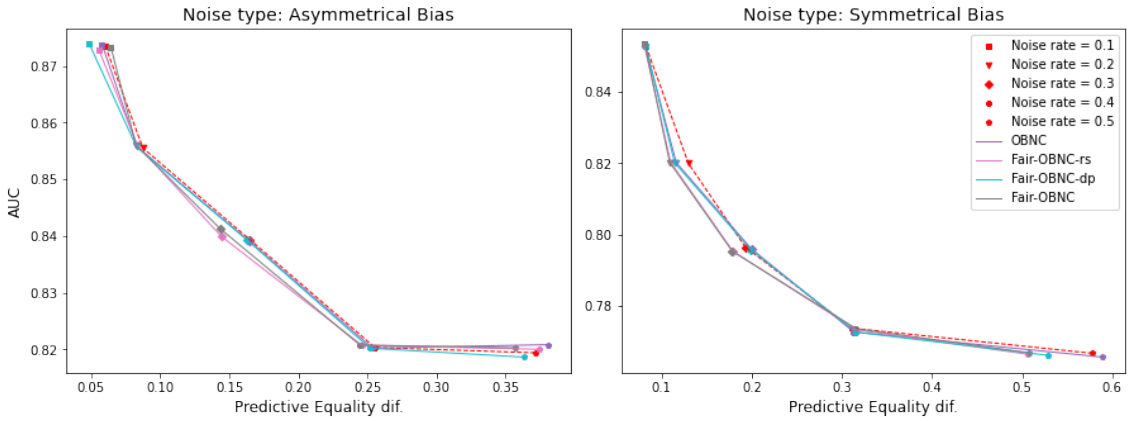


Figure 4.5: Trade-Off between AUC and Predictive Equality difference obtained on the *noisy* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

The results are, in general, very similar, and the differences are unlikely to be statistically significant. Nevertheless, we observe that Fair-OBNC improves fairness when compared to the original OBNC without loss in predictive performance. On the other hand, the comparison between the two alternatives of Fair-OBNC, Fair-OBNC-rs and Fair-OBNC-dp, doesn't show any strong pattern. These observations are confirmed by analyzing the results with a noise rate of 0.5, which are shown in Fig. 4.6.

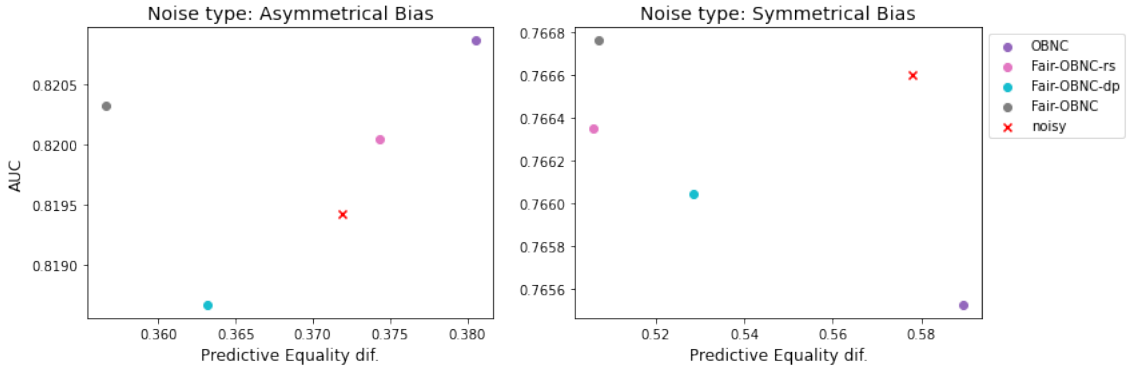


Figure 4.6: Trade-Off between AUC and Predictive Equality difference obtained on the *noisy* test set when correcting the data injected with each type of noise at a noise rate of 0.5 using each of the label correction methods.

Given the previous observations that the results are only different for datasets with a higher proportion of positive examples, we also show the trade-off between fairness and predictive performance on the *phishing* dataset. This dataset has a proportion of positive instances and of protected instances above 50%. By analyzing Fig. 4.7, we can observe a more evident distinction between

the performance of the methods. This further confirms the previous remarks on the positive impact of our method, which improves fairness without loss of predictive performance.

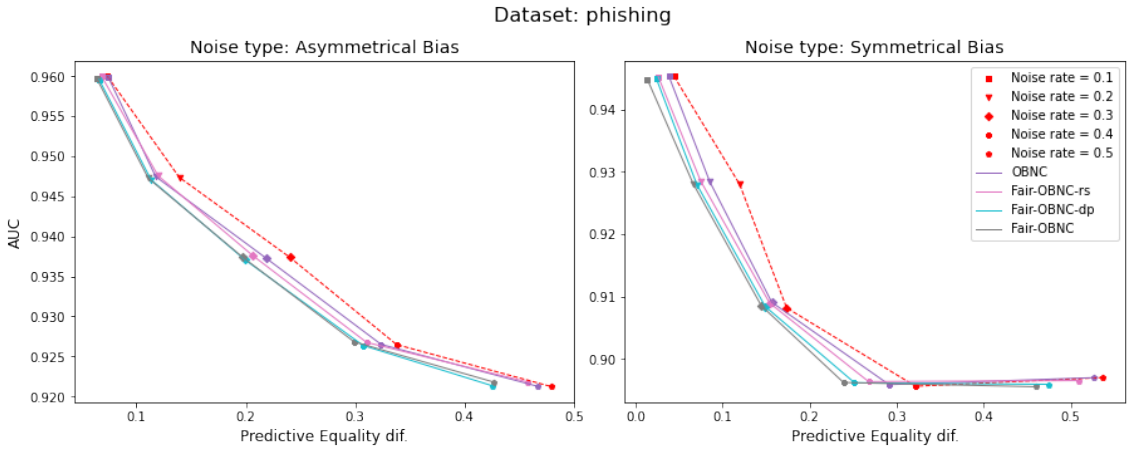


Figure 4.7: Trade-Off between AUC and Predictive Equality difference obtained on the *noisy* test set when correcting the *phishing* data injected with each type of noise at different rates using each of the label correction methods.

4.3.1.3 Evaluation on the original test set

We further evaluate the obtained models on the *original* test set to investigate how the methods would perform in a testing scenario where the biases that were present in the training data have been corrected in more recent data (i.e. we use the *noisy* training set but the *original* test set, in which we assume that there is no noise). The trade-off between the AUC metric and the Predictive Equality difference metric for different noise rates is shown in Fig. 4.8.

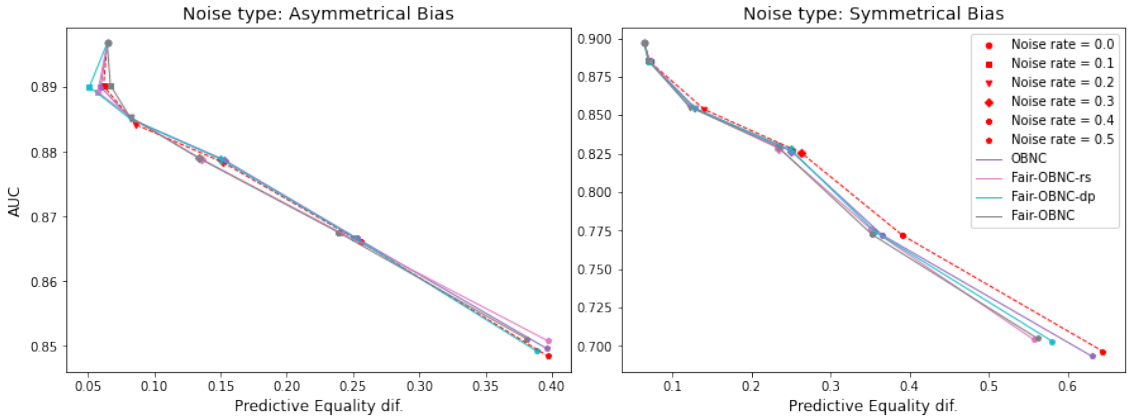


Figure 4.8: Trade-Off between AUC and Predictive Equality difference obtained on the *original* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

As in the previous scenario, our method shows improvements in terms of fairness, without loss of predictive performance, when compared to the original one. As before, no conclusive observations can be made concerning the two variations of Fair-OBNC. For better visualization and interpretation of the results, we illustrate this trade-off at a noise rate of 0.5 in Fig. 4.9. Similarly to the previous testing scenario, these results provide further evidence concerning the previous observations. The Fair-OBNC method consistently performs better in terms of fairness without loss in model accuracy. The Fair-OBNC-rs modification is slightly better than Fair-OBNC when dealing with Symmetrical Bias noise, but the difference is quite small.

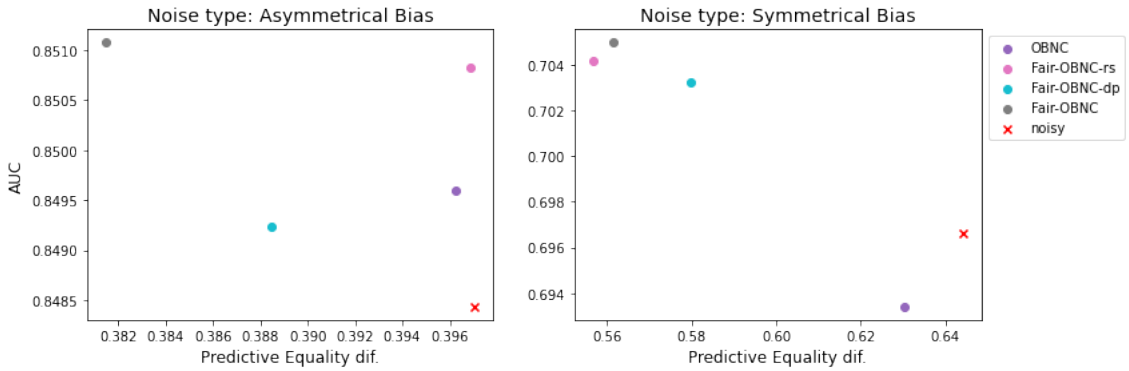


Figure 4.9: Trade-Off between AUC and Predictive Equality difference obtained on the *original* test set when correcting the data injected with each type of noise at a noise rate of 0.5 using each of the label correction methods.

Furthermore, the results obtained on the *phishing* dataset, which are shown in Fig. 4.10, also corroborate these statements.

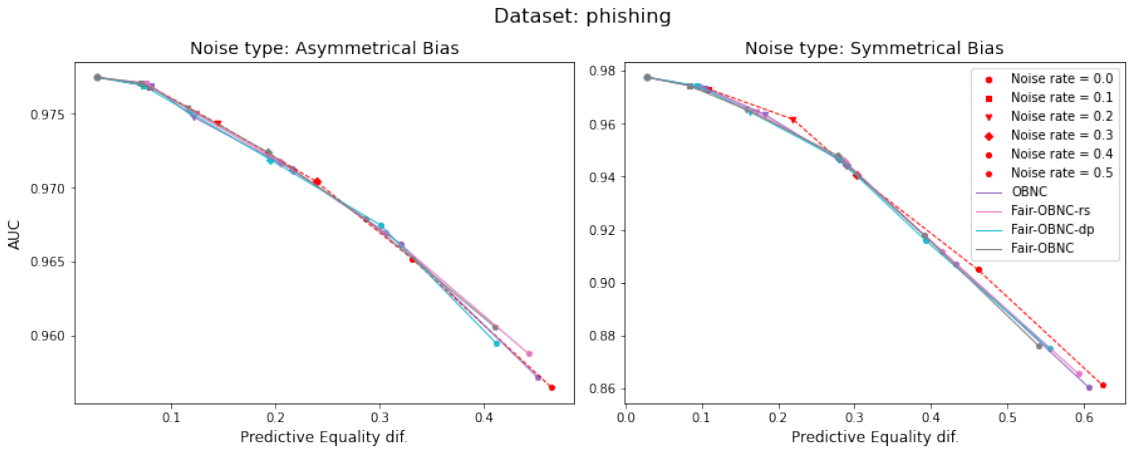
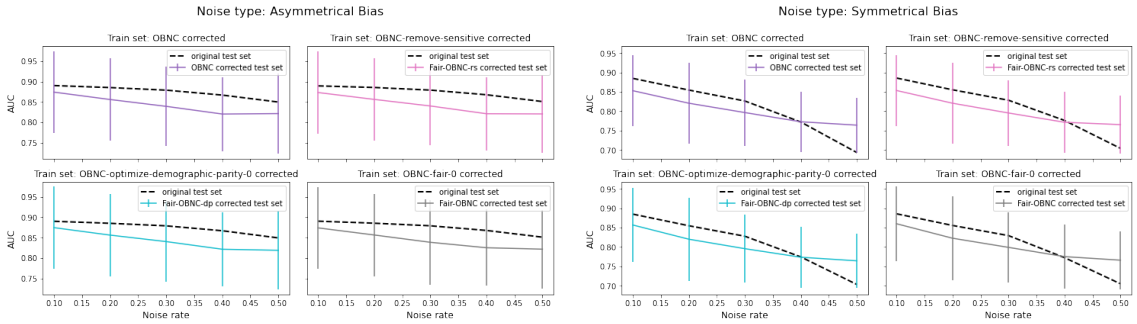


Figure 4.10: Trade-Off between AUC and Predictive Equality difference obtained on the *original* test set when correcting the *phishing* data injected with each type of noise at different rates using each of the label correction methods.

4.3.1.4 Evaluation on the corrected test set

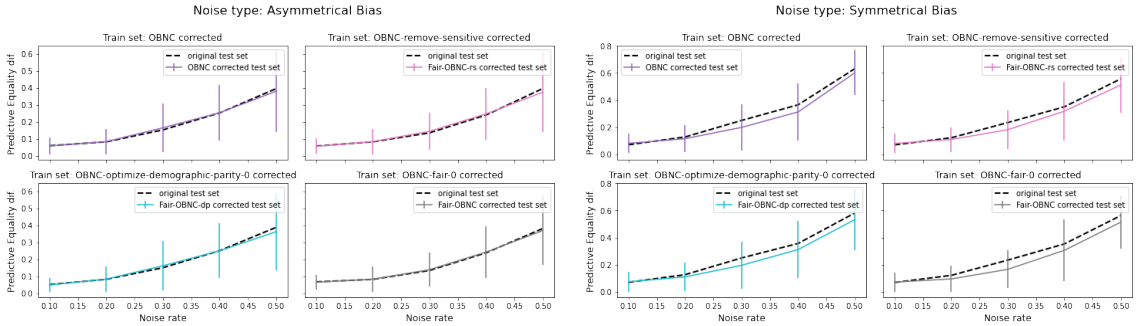
We now analyze the effect of correcting training data when the discrimination has been eliminated in the meantime, but the original data was already biased (i.e. we use the *corrected* training and test sets). In these experiments, we apply the same label noise correction method in both the training and the test set. As the available testing labels are noisy, we use a label noise correction method to correct the test data as well. The results for the AUC metric are presented in Fig. 4.11, and for Predictive Equality difference in Fig. 4.12.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure 4.11: Comparison in AUC between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure 4.12: Comparison in Predictive Equality difference between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.

Both our Fair-OBNC method and the application of the modifications separately show similar behavior to the original method, which is a tendency to precisely estimate fairness but underestimate the predictive performance. This is consistent with the results obtained by the OBNC method in the empirical evaluation conducted in Chapter 3. Since the test set is corrected in the same manner as the training set, we could expect an optimistic estimation of the predictive performance of the models. However, although the differences are not statistically significant, we observe the opposite: the models systematically obtain higher predictive performance on the original test set.

4.3.2 Using benchmark fairness datasets

Finally, the performance of the proposed method was further tested in the fairness benchmark datasets described in Table 3.2, measuring the predictive accuracy and fairness of the models trained in the *originally biased* and *corrected* training sets.

4.3.2.1 Evaluation on the originally biased test set

First, as we only have access to biased data both for training and testing, we evaluate the models on the *originally biased* test set. The trade-off between the AUC metric and the Predictive Equality difference is shown in Fig. 4.13.

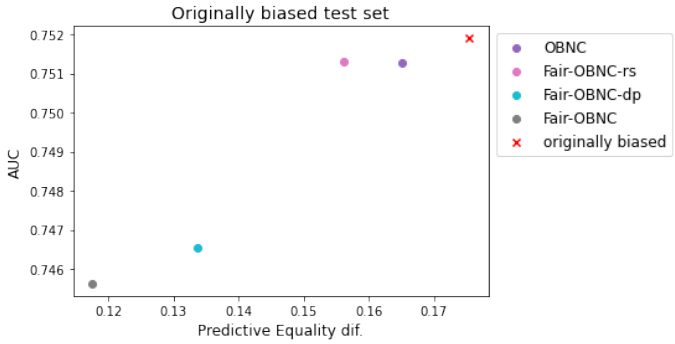


Figure 4.13: Trade-Off between AUC and Predictive Equality difference obtained on the *originally biased* test set.

Our method improves the fairness of the predictions in comparison to the original OBNC method. Although the difference is also unlikely to be statistically significant, as in the results with the injected noise on standard datasets, there is a small decrease in predictive performance: Fair-OBNC achieves the lowest value of discrimination but, at the same time, the lowest value of AUC. Additionally, the Fair-OBNC-rs modification reduces discrimination the least in comparison to the other modifications.

4.3.2.2 Evaluation on the corrected test set

Moreover, we apply the noise correction methods on the *originally biased* test set to simulate a presumed noise-free testing environment. The trade-off between the AUC and Predictive Equality difference metrics is presented in Fig. 4.14. These results are almost identical to the ones obtained when using the *originally biased* test set, and as such, the same conclusions can be derived.

4.3.3 Discussion

From the conducted experiments, we can conclude the positive impact of modifying existing label noise correction specifically for the task of improving the fairness of ML models. Our proposed

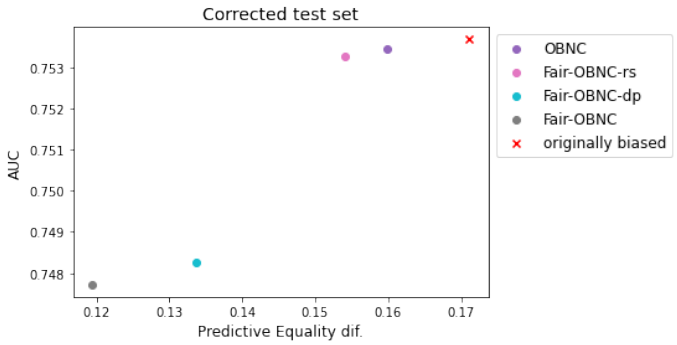


Figure 4.14: Trade-Off between AUC and Predictive Equality difference obtained on the *corrected* test set.

modifications to the original OBNC method resulted in improvements in fairness without a significant effect on the accuracy of the predictions of models trained on the corrected data when compared to the baseline.

The results should be interpreted carefully, as we have observed that for many datasets, the results of OBNC and Fair-OBNC are the same (Fig. 4.3). The larger differences were obtained when the proportion of positive instances and the proportion of protected instances is above 50% (Table 4.12b). We believe this is a limitation of the original OBNC. This method is sensitive to imbalanced class distributions, as the definition of the margin does not take into account the distribution of classes. This effect was not observed in the experiments conducted in the original OBNC paper since only datasets with balanced class distributions were used [23]. Furthermore, we also believe this limitation of the OBNC method explains the surprising results obtained in Section 4.3.1.4 when using the corrected test set. In this testing scenario, we compared the performance of the model obtained from the *corrected* training data on the *original* and *corrected* test sets. When correcting the test set, we observed that not only do the methods underestimate model performance but also no difference is observed between our proposed method and the original one.

We can also identify opportunities for improvement in each of the modifications that we introduce in our proposed algorithm. On the one hand, the method ignores the sensitive attribute. This is a simple strategy that works well in this case where we arbitrarily select a single sensitive attribute and are in control of the noise injection process. However, this strategy may be too simplistic to achieve fairness in more complicated scenarios where there might be multiple sensitive attributes and more complex noise models. It is also well known that, in many cases, there are proxy attributes that can be used to infer the sensitive information we are trying to hide, and in such cases, this adjustment would not be very effective.

On the other hand, the intuition behind the application of fairness criteria when deciding which labels to correct could be extended to apply different and more elaborate fairness definitions. This is particularly important due to our findings related to how class distributions impact the performance of our method. As such, we highlight the importance of future work in investigating how to modify the margin calculation to account for imbalanced class distributions. Moreover,

work could be developed to extend our fair criterion for correcting the labels.

Finally, in the conducted experiments, we set the number of instances to corrected to 20% of the total number of instances. The effect of changing this parameter could also be further investigated.

Chapter 5

Conclusions

In this work, we tackle the problem of learning fair ML classifiers from biased data. In such a scenario, we look at the inherent discrimination in datasets as label noise that can be eliminated using label noise correction techniques. This way, the corrected data could be used to train fair classifiers using standard ML algorithms without further application of fairness-enhancing techniques.

To provide a comprehensive overview of the background knowledge related to this topic, we reviewed the existing literature related to ensuring the fairness of ML models and dealing with noisy labels. We identified some limitations of the fairness empirical evaluation procedures that are usually conducted in the existing literature, as well as a lack of work in developing methods that aim at improving the fairness of ML models through label noise correction.

To fill this gap, we propose a methodology to empirically evaluate the effect of different label noise correction techniques in improving the fairness and predictive performance of models trained on previously biased data. Our framework involves manipulating the amount and type of label noise and can be used on both fairness benchmarks and standard ML datasets.

Having an *original* standard ML dataset, which we assume to have clean labels, the first step of our framework involves injecting the desired type and amount of label noise to obtain a *noisy* dataset. We then apply the considered label noise correction method to generate a *corrected* dataset. Classifiers are trained using each of the training sets, and the obtained models are tested under different assumptions. We simulate the cases where only biased data is available by testing the models on the *noisy* dataset. The scenario where the discrimination that exists in the training data is no longer present at testing time can be enacted by testing on the *original* test set. We can further test the usefulness of applying label noise correction to the test set in cases where the discrimination has been eliminated, but we still only have access to biased data by testing the models on the *corrected* test set.

In the conducted experiments, we analyzed six label noise correction methods. We observed that the Hybrid Label Noise Correction method [70] was able to achieve the best trade-off between fairness and predictive performance.

Furthermore, we propose Fair-OBNC, a method for fair label noise correction that applies two different modifications to the Ordering-Based Noise Correction method [23] to take fairness into

consideration during noise correction. These variations include ignoring the sensitive attribute and deciding which labels to correct in a way that balances the distribution of classes across groups. In the conducted experiments, we evaluated our method against the original one, also testing each modification by itself, and analyzed the fairness and predictive performance of the obtained models in both standard ML datasets and fairness benchmarks. We observed that the performed modifications to an existing label noise correction method resulted in models that were less discriminatory without loss of predictive performance.

References

[1] Görkem Algan and Ilkay Ulusoy. Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems*, 215:106771, 2021.

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pages 312–321. PMLR, 2019.

[3] Robert E Banfield, Lawrence O Hall, Kevin W Bowyer, and W Philip Kegelmeyer. A comparison of decision tree ensemble creation techniques. *IEEE transactions on pattern analysis and machine intelligence*, 29(1):173–180, 2006.

[4] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019. http://www.fairmlbook.org.

[5] Peter Bartlett, Yoav Freund, Wee Sun Lee, and Robert E Schapire. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[6] Sugato Basu. Semi-supervised clustering by seeding. In *Proc. ICML-2002*, 2002.

[7] Yahav Bechavod and Katrina Ligett. Penalizing unfairness in binary classification. *arXiv preprint arXiv:1707.00044*, 2017.

[8] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

[9] Mélanie Bernhardt, Charles Jones, and Ben Glocker. Potential sources of dataset bias complicate investigation of underdiagnosis by machine learning algorithms. *Nature Medicine*, 28(6):1157–1158, 2022.

[10] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

[11] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[12] Avrim Blum and Kevin Stangl. Recovering from biased data: Can fairness constraints improve accuracy? *arXiv preprint arXiv:1912.01094*, 2019.

[13] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30, 2017.

[14] Junyi Chai and Xiaoqian Wang. Fairness with adaptive weights. In *International Conference on Machine Learning*, pages 2853–2866. PMLR, 2022.

[15] Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. On symmetric losses for learning from corrupted labels. In *International Conference on Machine Learning*, pages 961–970. PMLR, 2019.

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

[17] Giovanni Luca Ciampaglia, Azadeh Nematzadeh, Filippo Menczer, and Alessandro Flammini. How algorithmic popularity bias hinders or promotes quality. *Scientific reports*, 8(1):15951, 2018.

[18] André F Cruz, Pedro Saleiro, Catarina Belém, Carlos Soares, and Pedro Bizarro. A bandit-based algorithm for fairness-aware hyperparameter optimization. *arXiv preprint arXiv:2010.03665*, 2020.

[19] Brian d'Alessandro, Cathy O'Neil, and Tom LaGatta. Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data*, 5(2):120–134, 2017.

[20] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *arXiv preprint arXiv:1408.6491*, 2014.

[21] 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*, pages 214–226, 2012.

[22] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[23] Wei Feng and Samia Boukir. Class noise removal and correction for image classification using ensemble margin. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4698–4702. IEEE, 2015.

[24] César Ferri, José Hernández-Orallo, and R Modroiu. An experimental comparison of performance measures for classification. *Pattern recognition letters*, 30(1):27–38, 2009.

[25] Riccardo Fogliato, Alexandra Chouldechova, and Max G'Sell. Fairness evaluation in presence of biased noisy labels. In *International Conference on Artificial Intelligence and Statistics*, pages 2325–2336. PMLR, 2020.

[26] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[27] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. Fairness testing: testing software for discrimination. In *Proceedings of the 2017 11th Joint meeting on foundations of software engineering*, pages 498–510, 2017.

[28] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[29] Aritra Ghosh, Naresh Manwani, and PS Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015.

[30] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[31] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[32] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[33] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019.

[34] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[35] Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 702–712. PMLR, 2020.

[36] Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 967–972. IEEE, 2016.

[37] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[38] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012.

[39] Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*, 2017.

[40] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. Operationalizing individual fairness with pairwise fair representations. *arXiv preprint arXiv:1907.01439*, 2019.

[41] Alex Lamy, Ziyuan Zhong, Aditya K Menon, and Nakul Verma. Noise-tolerant fair classification. *Advances in Neural Information Processing Systems*, 32, 2019.

[42] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *International Conference on Machine Learning*, pages 6403–6413. PMLR, 2021.

[43] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[44] Yang Liu and Jialu Wang. Can less be more? when increasing-to-balancing label noise rates considered beneficial. *Advances in Neural Information Processing Systems*, 34:17467–17479, 2021.

[45] Pranay K Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R Varshney, and Ruchir Puri. Bias mitigation post-processing for individual and group fairness. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 2847–2851. IEEE, 2019.

[46] Naresh Manwani and PS Sastry. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3):1146–1151, 2013.

[47] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[48] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, pages 107–118. PMLR, 2018.

[49] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[50] Bryce Nicholson, Jing Zhang, Victor S Sheng, and Zhiheng Wang. Label noise correction methods. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–9. IEEE, 2015.

[51] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

[52] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017.

[53] Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

[54] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

[55] Novi Quadrianto and Viktoriia Sharmanska. Recycling privileged learning and distribution matching for fairness. *Advances in Neural Information Processing Systems*, 30, 2017.

[56] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

[57] Nripsuta Ani Saxena, Karen Huang, Evan DeFilippis, Goran Radanovic, David C Parkes, and Yang Liu. How do fairness definitions fare? examining public attitudes towards algorithmic definitions of fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 99–106, 2019.

[58] Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference on learning theory*, pages 489–511. PMLR, 2013.

[59] Bo Sun, Songcan Chen, Jiandong Wang, and Haiyan Chen. A robust multi-class adaboost algorithm for mislabeled noisy data. *Knowledge-Based Systems*, 102:87–102, 2016.

[60] Jiang-wen Sun, Feng-ying Zhao, Chong-jun Wang, and Shi-fu Chen. Identifying and correcting mislabeled training instances. In *Future generation communication and networking (FGCN 2007)*, volume 1, pages 244–250. IEEE, 2007.

[61] Harini Suresh and John V Guttag. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002*, 2(8), 2019.

[62] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5552–5560, 2018.

[63] Isaac Triguero, José A Sáez, Julián Luengo, Salvador García, and Francisco Herrera. On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, 132:30–41, 2014.

[64] Berk Ustun, Yang Liu, and David Parkes. Fairness without harm: Decoupled classifiers with preference guarantees. In *International Conference on Machine Learning*, pages 6373–6382. PMLR, 2019.

[65] Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, pages 1–7. IEEE, 2018.

[66] Jialu Wang, Yang Liu, and Caleb Levy. Fair classification with group-dependent label noise. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 526–536, 2021.

[67] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[68] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *Conference on Learning Theory*, pages 1920–1953. PMLR, 2017.

[69] Songhua Wu, Mingming Gong, Bo Han, Yang Liu, and Tongliang Liu. Fair classification with instance-dependent label noise. In *Conference on Causal Learning and Reasoning*, pages 927–943. PMLR, 2022.

[70] Jiwei Xu, Yun Yang, and Po Yang. Hybrid label noise correction algorithm for medical auxiliary diagnosis. In *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 567–572. IEEE, 2020.

[71] Muhammad Bilal Zafar, Isabel Valera, Manuel Rodriguez, Krishna Gummadi, and Adrian Weller. From parity to preference-based notions of fairness in classification. *Advances in Neural Information Processing Systems*, 30, 2017.

[72] Tianxiang Zhao, Enyan Dai, Kai Shu, and Suhang Wang. Towards fair classifiers without sensitive attributes: Exploring biases in related features. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1433–1442, 2022.

[73] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pages 11447–11457. PMLR, 2020.

[74] Zhi-Hua Zhou. *Machine learning*. Springer Nature, 2021.

[75] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210, 2004.

Appendix A

Additional Empirical Evaluation Results

In this appendix, we present the additional results from the conducted empirical evaluation of label noise correction methods for improving ML fairness, considering the remaining fairness metrics.

A.1 Using standard ML datasets

A.1.1 Performance evaluation on the noisy test set

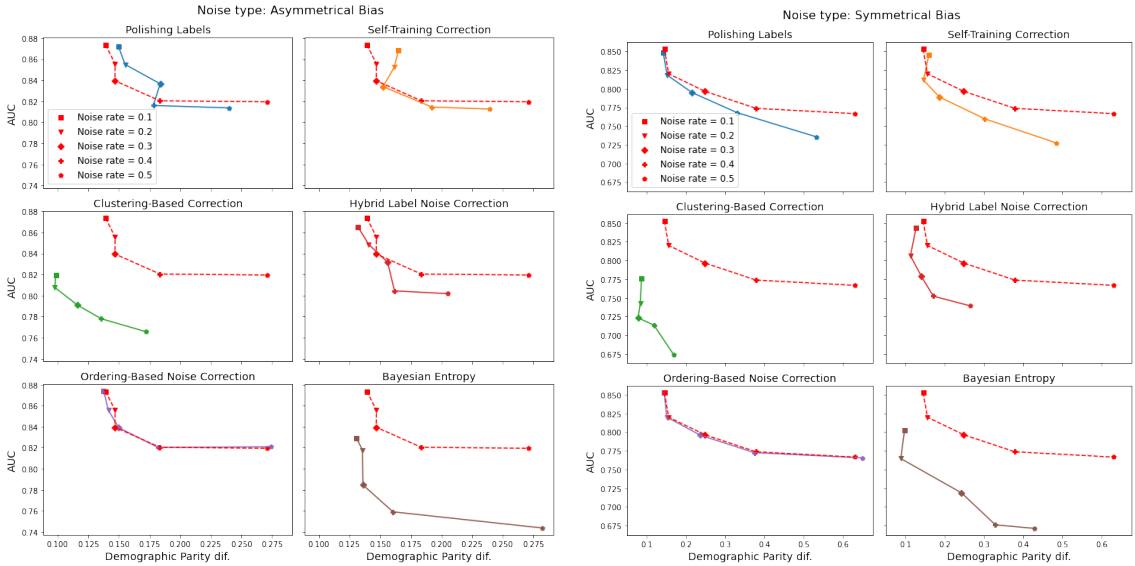
The trade-off between the AUC metric and the several fairness metrics for each type of noise and at different noise rates is shown in Fig. A.1 for the Demographic Parity difference metric, in Fig. A.2 for the Equalized Odds difference metric, and in Fig. A.3 for the Equal Opportunity difference metric.

A.1.2 Performance evaluation on the original test set

The trade-off between the AUC metric and the several fairness metrics for each type of noise and at different noise rates is shown in Fig. A.4 for the Demographic Parity difference metric, in Fig. A.5 for the Equalized Odds difference metric, and in Fig. A.6 for the Equal Opportunity difference metric.

A.1.3 Performance evaluation on the corrected test set

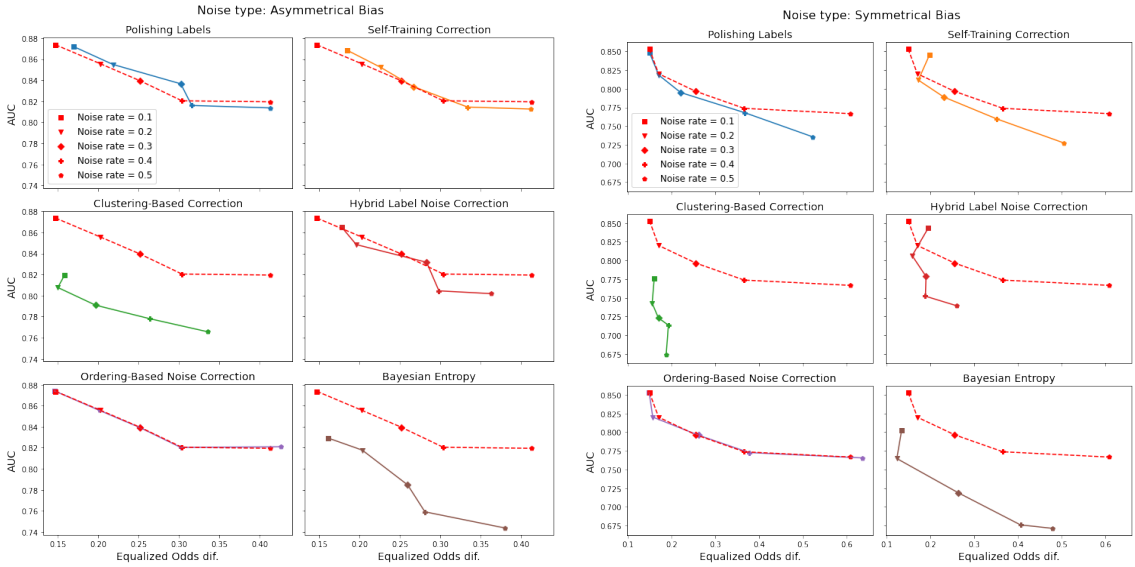
The results for the Demographic Parity difference metric are presented in Fig. A.7, for the Equalized Odds difference metric in Fig. A.8, and for the Equal Opportunity difference metric in Fig. A.9.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

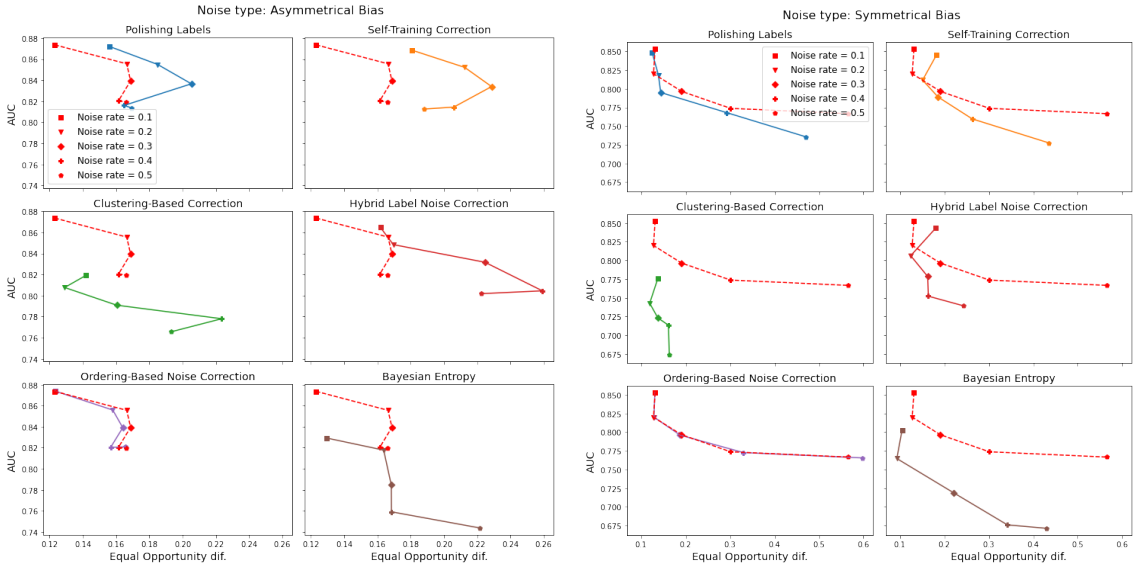
Figure A.1: Trade-Off between AUC and Demographic Parity difference obtained on the *noisy* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

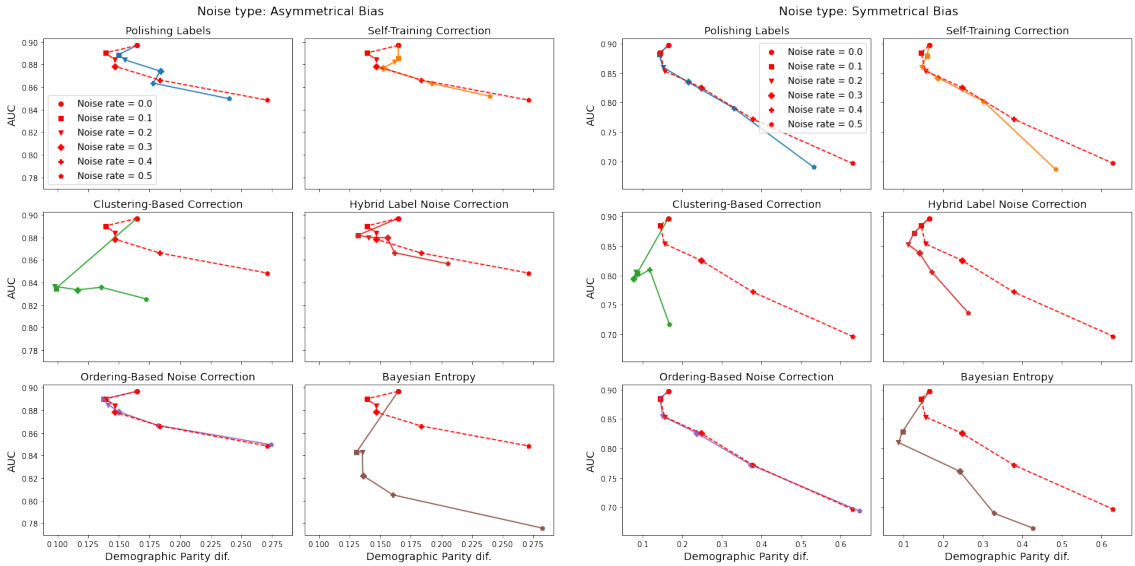
Figure A.2: Trade-Off between AUC and Equalized Odds difference obtained on the *noisy* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure A.3: Trade-Off between AUC and Equal Opportunity difference obtained on the *noisy* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.



(a) Injecting Asymmetrical Bias noise.

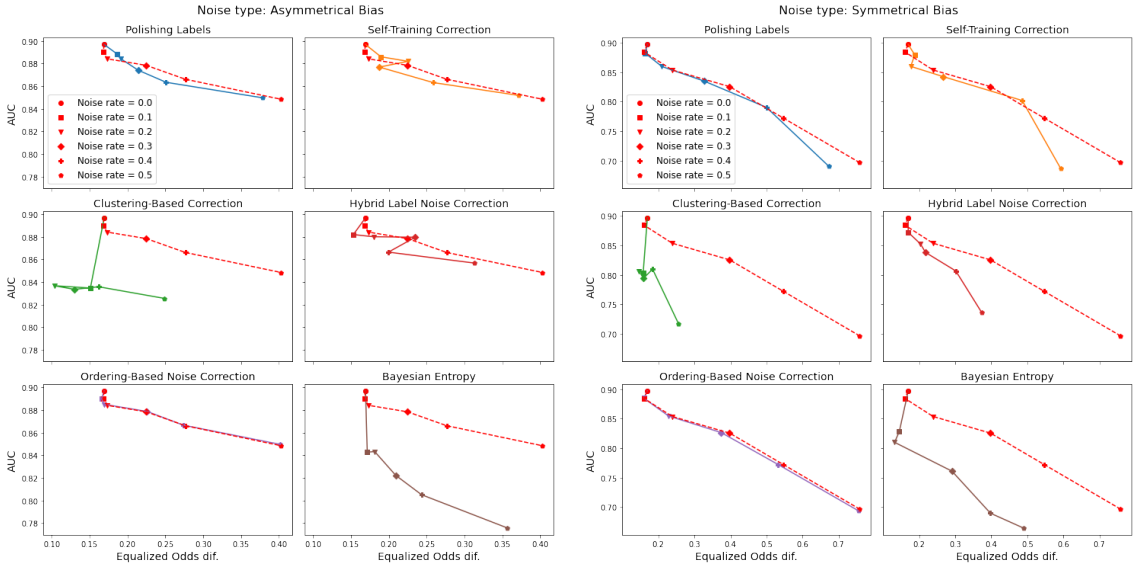
(b) Injecting Symmetrical Bias noise.

Figure A.4: Trade-Off between AUC and Demographic Parity difference obtained on the *original* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *original* train set at each noise rate.

A.2 Using benchmark fairness datasets

A.2.1 Performance evaluation on the noisy test set

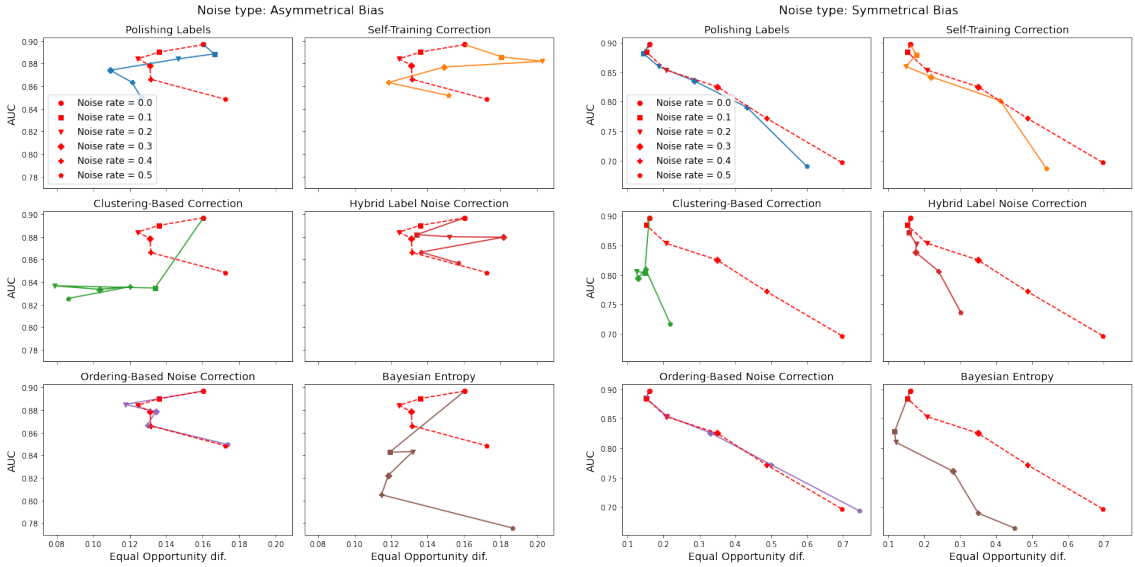
The trade-off between the AUC metric and the several fairness metrics is shown in Fig. A.10, for the Demographic Parity difference metric, in Fig. A.11, for the Equalized Odds difference metric,



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure A.5: Trade-Off between AUC and Equalized Odds difference obtained on the *original* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *original* train set at each noise rate.

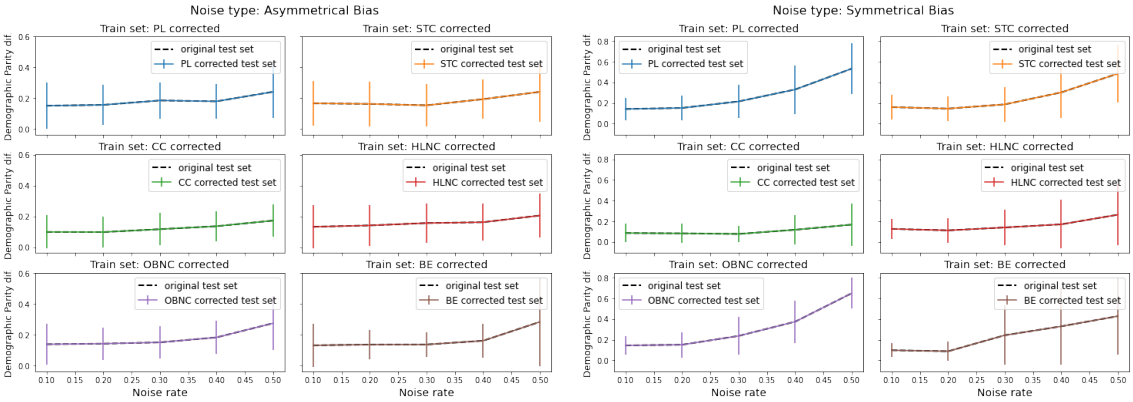


(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure A.6: Trade-Off between AUC and Equal Opportunity difference obtained on the *original* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *original* train set at each noise rate.

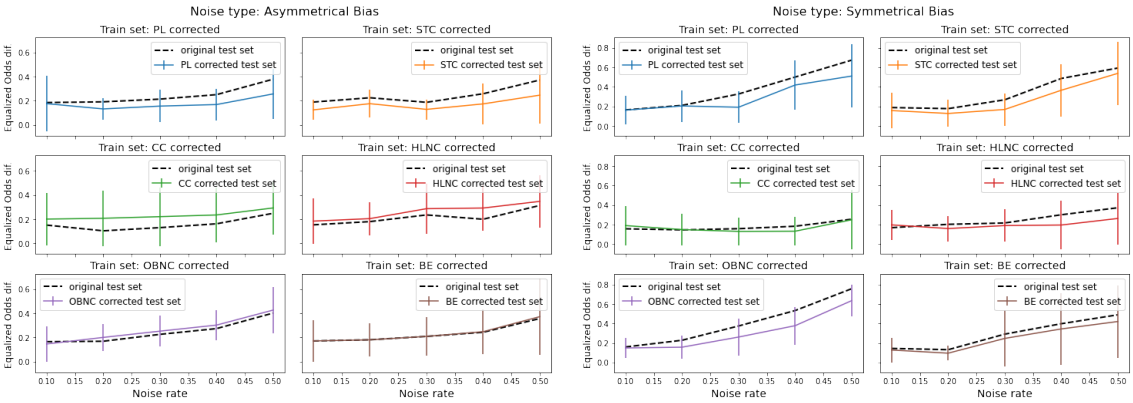
and in Fig. A.12, for the Equal Opportunity metric.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure A.7: Comparison in Demographic Parity difference between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.



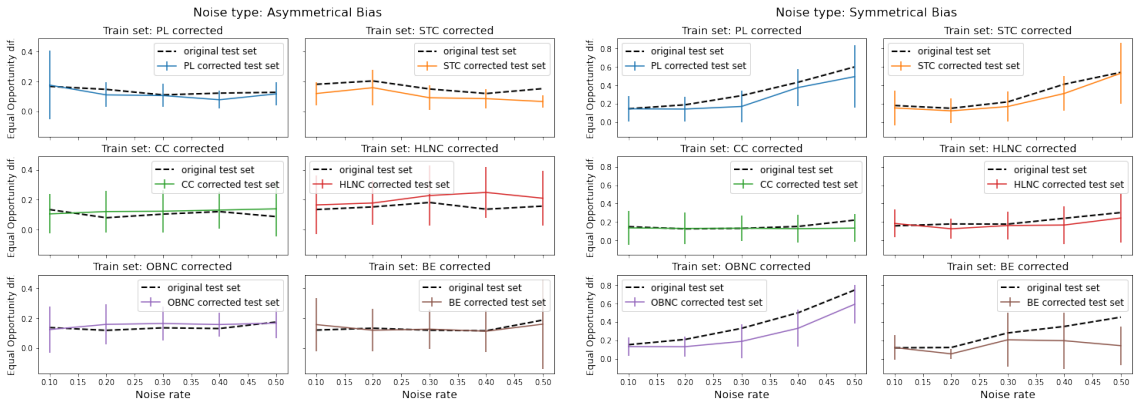
(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure A.8: Comparison in Equalized Odds difference between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.

A.2.2 Performance evaluation on the corrected test set

The trade-off between the AUC metric and the several fairness metrics is shown in Fig. A.13, for the Demographic Parity difference metric, in Fig. A.14, for the Equalized Odds difference metric, and in Fig. A.15, for the Equal Opportunity metric.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure A.9: Comparison in Equal Opportunity difference between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.

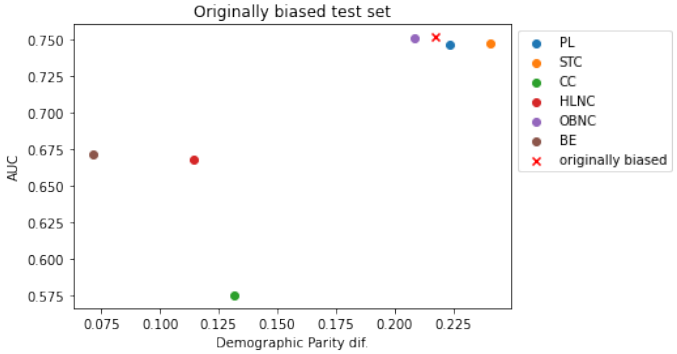


Figure A.10: Trade-Off between AUC and Demographic Parity difference obtained on the *originally biased* test set.

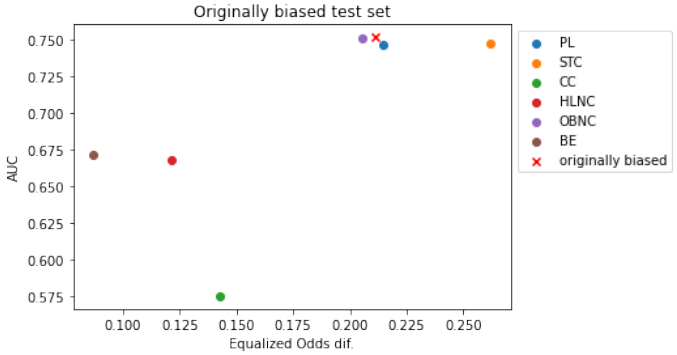


Figure A.11: Trade-Off between AUC and Equalized Odds difference obtained on the *originally biased* test set.

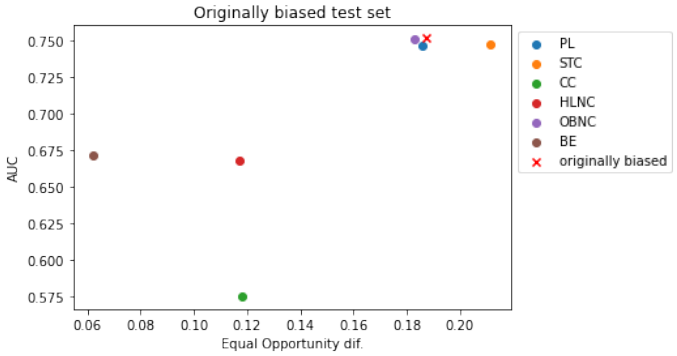


Figure A.12: Trade-Off between AUC and Equal Opportunity difference obtained on the *originally biased* test set.

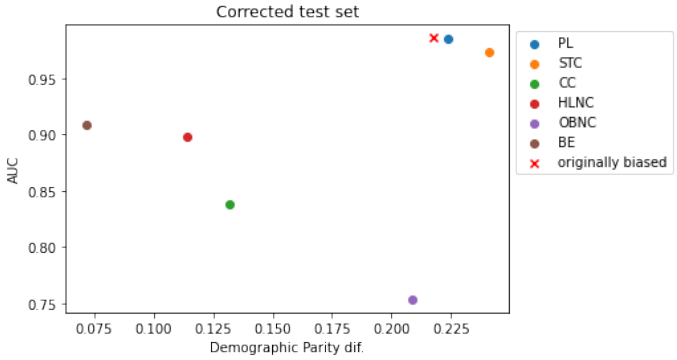


Figure A.13: Trade-Off between AUC and Demographic Parity difference obtained on the *originally biased* test set.

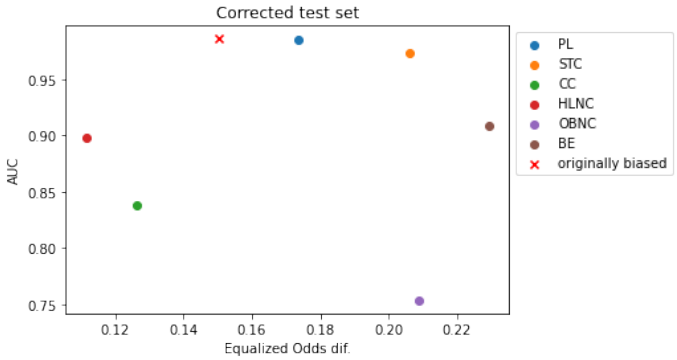


Figure A.14: Trade-Off between AUC and Equalized Odds difference obtained on the *originally biased* test set.

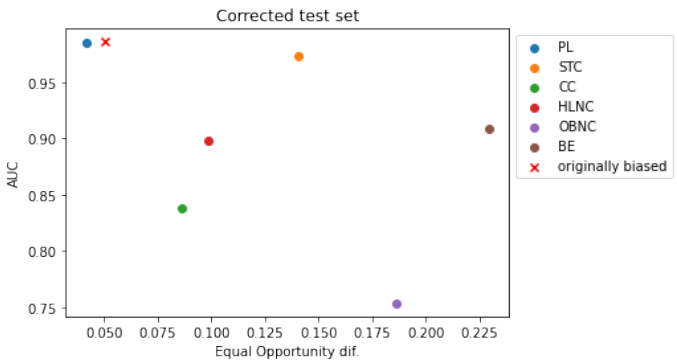


Figure A.15: Trade-Off between AUC and Equal Opportunity difference obtained on the *originally biased* test set.

Appendix B

Additional Fair Ordering-Based Noise Correction Results

In this appendix, we present the additional results from the conducted experiments to evaluate our proposed algorithm, Fair-OBNC, considering the remaining fairness metrics.

B.1 Using standard ML datasets

B.1.1 Performance evaluation on the noisy test set

The trade-off between the AUC metric and the several fairness metrics for each type of noise and at different noise rates is shown in Fig. B.1 for the Demographic Parity difference metric, in Fig. B.2 for the Equalized Odds difference metric, and in Fig. B.3 for the Equal Opportunity difference metric.

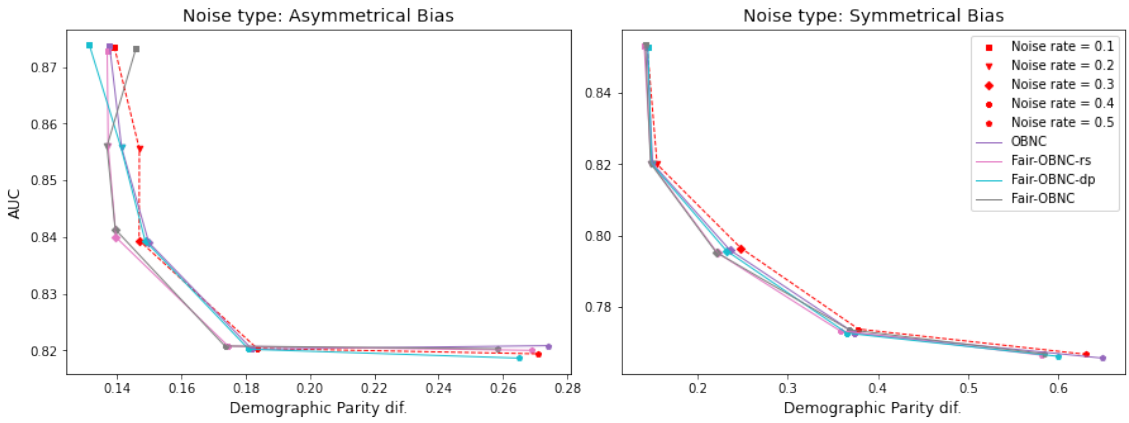


Figure B.1: Trade-Off between AUC and Demographic Parity difference obtained on the *noisy* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

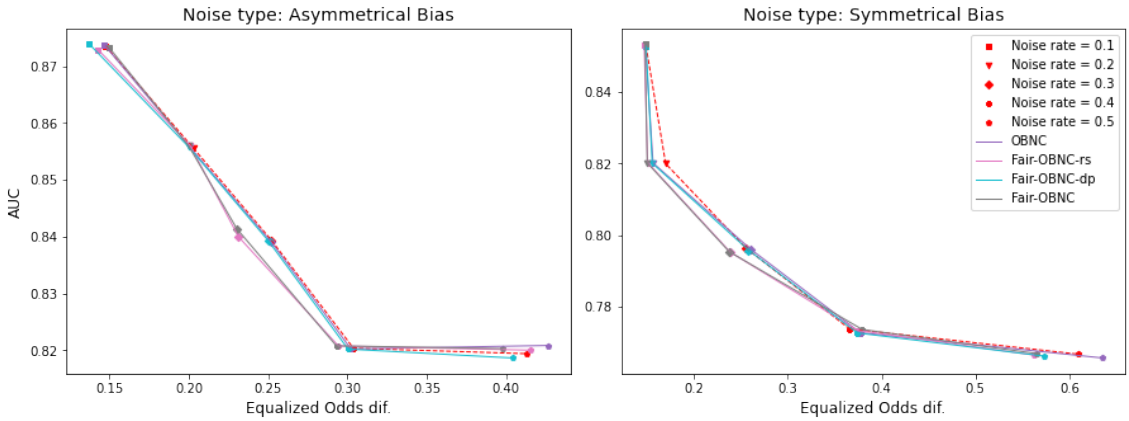


Figure B.2: Trade-Off between AUC and Equalized Odds difference obtained on the *noisy* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

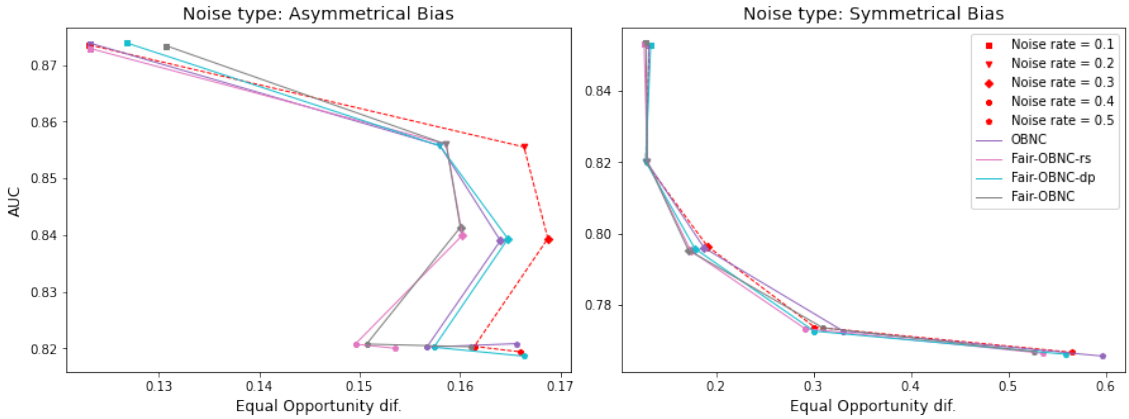


Figure B.3: Trade-Off between AUC and Equal Opportunity difference obtained on the *noisy* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

B.1.2 Performance evaluation on the original test set

The trade-off between the AUC metric and the several fairness metrics for each type of noise and at different noise rates is shown in Fig. B.4 for the Demographic Parity difference metric, in Fig. B.5 for the Equalized Odds difference metric, and in Fig. B.6 for the Equal Opportunity difference metric.

B.1.3 Performance evaluation on the corrected test set

The results for the Demographic Parity difference metric are presented in Fig. B.7, for the Equalized Odds difference metric in Fig. B.8, and for the Equal Opportunity difference metric in Fig. B.9.

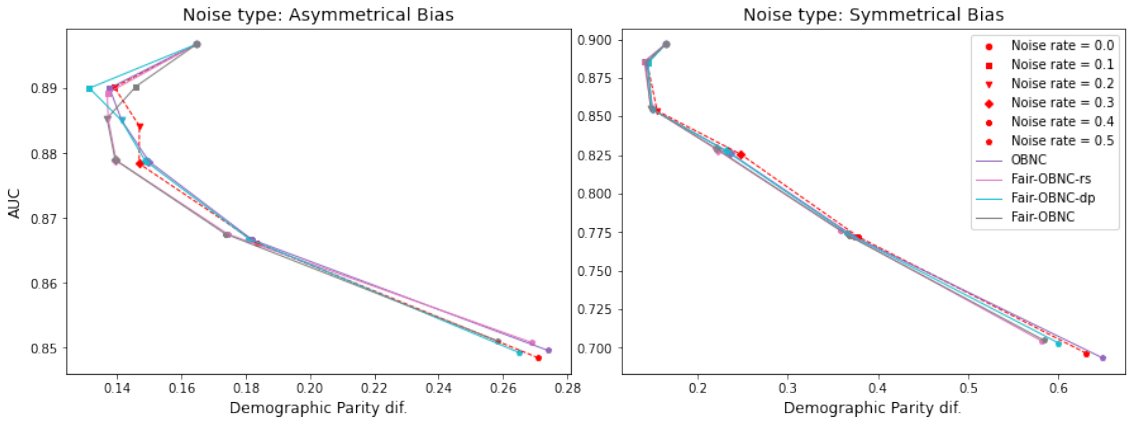


Figure B.4: Trade-Off between AUC and Demographic Parity difference obtained on the *original* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

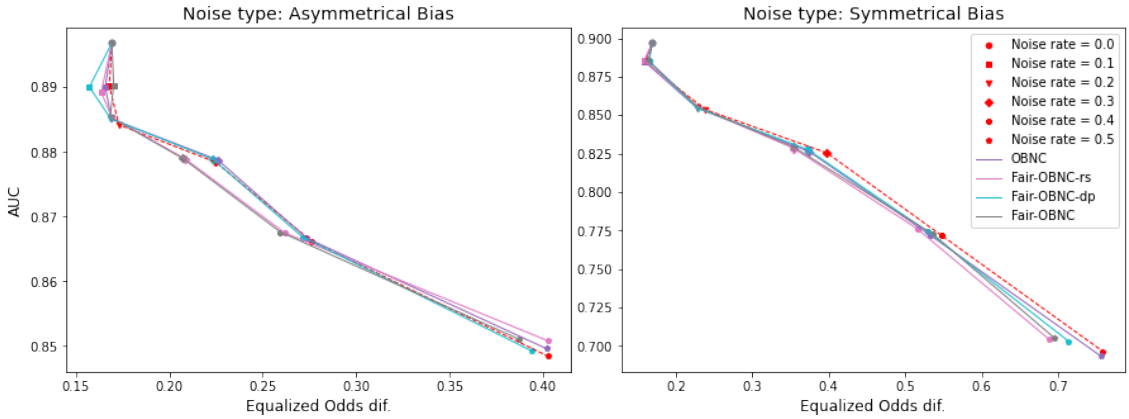


Figure B.5: Trade-Off between AUC and Equalized Odds difference obtained on the *original* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.

B.2 Using benchmark fairness datasets

B.2.1 Performance evaluation on the noisy test set

The trade-off between the AUC metric and the several fairness metrics is shown in Fig. B.10, for the Demographic Parity difference metric, in Fig. B.11, for the Equalized Odds difference metric, and in Fig. B.12, for the Equal Opportunity metric.

B.2.2 Performance evaluation on the corrected test set

The trade-off between the AUC metric and the several fairness metrics is shown in Fig. B.13, for the Demographic Parity difference metric, in Fig. B.14, for the Equalized Odds difference metric,

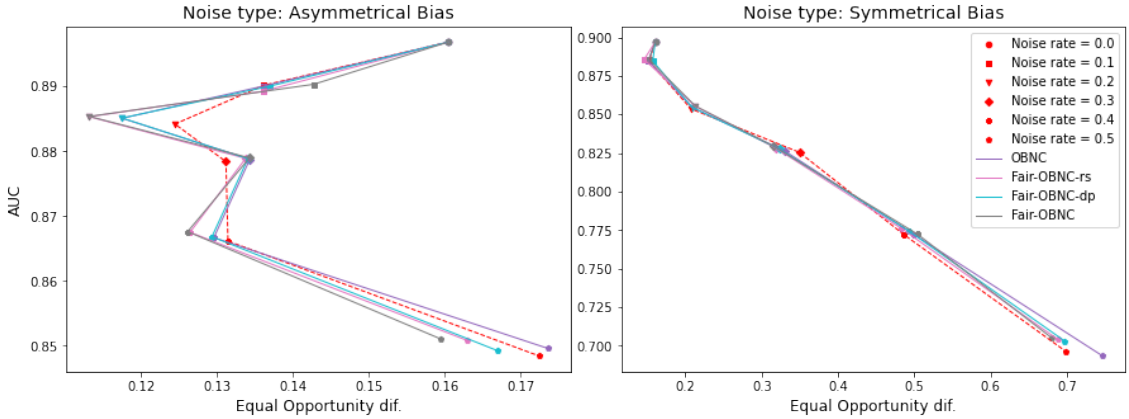
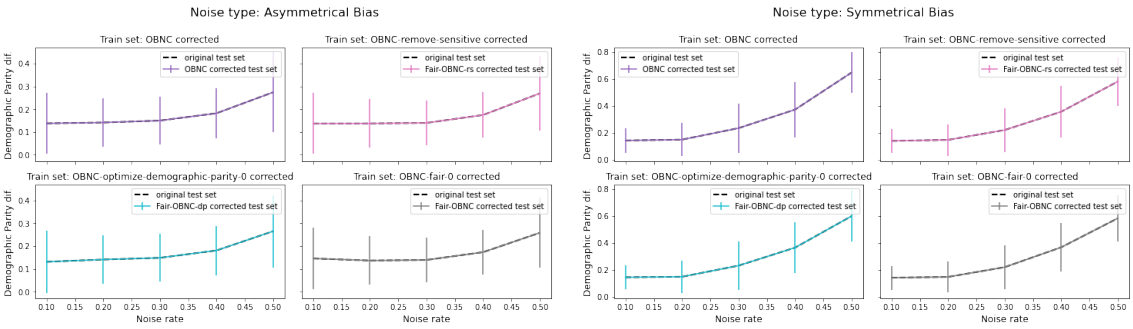


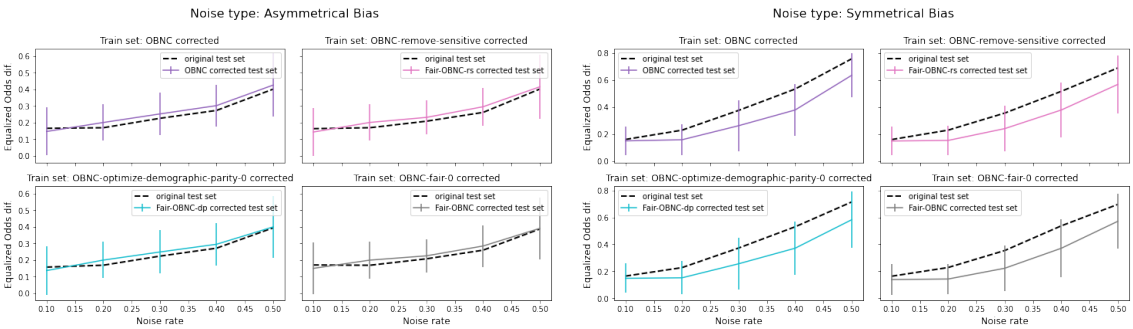
Figure B.6: Trade-Off between AUC and Equal Opportunity difference obtained on the *original* test set when correcting the data injected with each type of noise at different rates using each of the label correction methods. The red dashed line shows the performance of the model obtained from the *noisy* train set at each noise rate.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

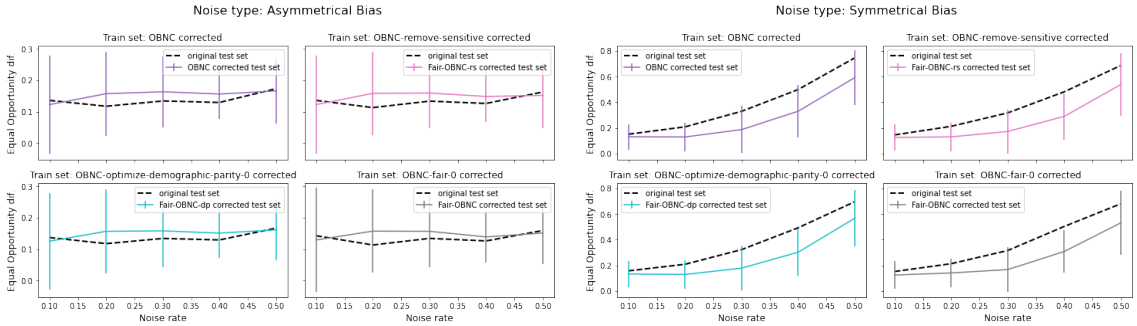
Figure B.7: Comparison in Demographic Parity difference between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure B.8: Comparison in Equalized Odds difference between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.



(a) Injecting Asymmetrical Bias noise.

(b) Injecting Symmetrical Bias noise.

Figure B.9: Comparison in Equal Opportunity difference between testing the model obtained from the data corrected by each method on the original test set and on the test set corrected by the same method.

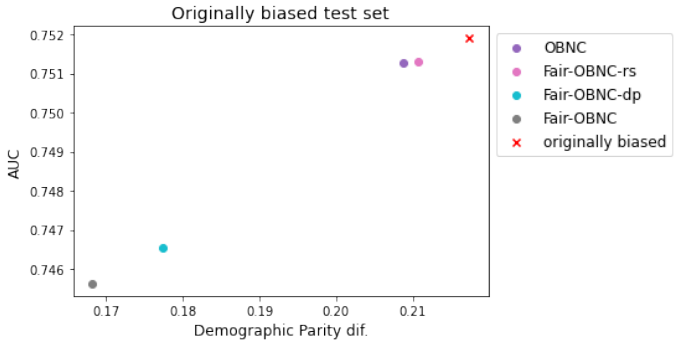


Figure B.10: Trade-Off between AUC and Demographic Parity difference obtained on the *originally biased* test set.

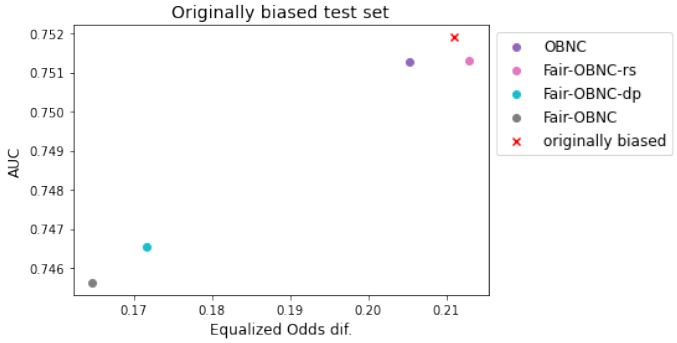


Figure B.11: Trade-Off between AUC and Equalized Odds difference obtained on the *originally biased* test set.

and in Fig. B.15, for the Equal Opportunity metric.

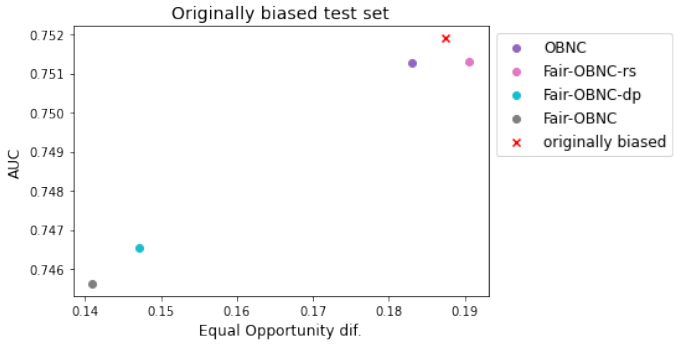


Figure B.12: Trade-Off between AUC and Equal Opportunity difference obtained on the *originally biased* test set.

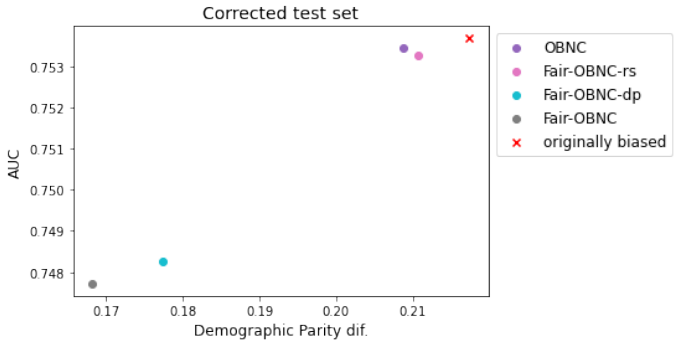


Figure B.13: Trade-Off between AUC and Demographic Parity difference obtained on the *originally biased* test set.

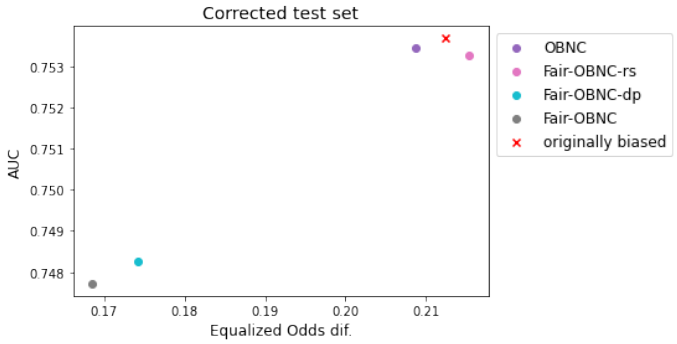


Figure B.14: Trade-Off between AUC and Equalized Odds difference obtained on the *originally biased* test set.

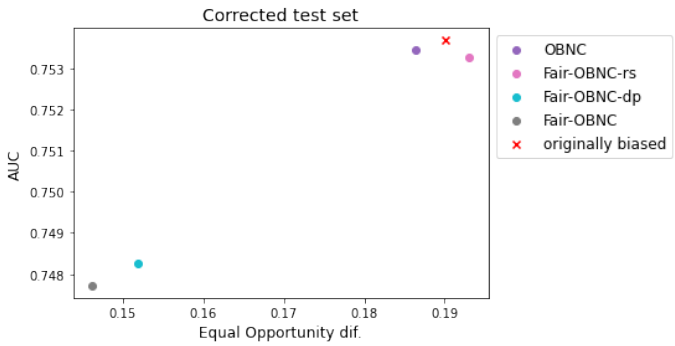


Figure B.15: Trade-Off between AUC and Equal Opportunity difference obtained on the *originally biased* test set.