Fairea: A Model Behaviour Mutation Approach to Benchmarking Bias Mitigation Methods

Max Hort
max.hort.19@ucl.ac.uk
University College London
UK

Jie M. Zhang
jie.zhang@ucl.ac.uk
University College London
UK

Federica Sarro
f.sarro@ucl.ac.uk
University College London
UK

Mark Harman
mark.harman@ucl.ac.uk
University College London
UK

ABSTRACT

The increasingly wide uptake of Machine Learning (ML) has raised the significance of the problem of tackling bias (i.e., unfairness), making it a primary software engineering concern. In this paper, we introduce *Fairea*, a model behaviour mutation approach to benchmarking ML bias mitigation methods. We also report on a large-scale empirical study to test the effectiveness of 12 widely-studied bias mitigation methods. Our results reveal that, surprisingly, bias mitigation methods have a poor effectiveness in 49% of the cases. In particular, 15% of the mitigation cases have worse fairness-accuracy trade-offs than the baseline established by *Fairea*; 34% of the cases have a decrease in accuracy *and* an increase in bias.

Fairea has been made publicly available for software engineers and researchers to evaluate their bias mitigation methods.

CCS CONCEPTS

• **Software and its engineering** → **Software creation and management**; **Extra-functional properties**.

KEYWORDS

Software fairness, bias mitigation, model mutation

ACM Reference Format:
Max Hort, Jie M. Zhang, Federica Sarro, and Mark Harman. 2021. Fairea: A Model Behaviour Mutation Approach to Benchmarking Bias Mitigation Methods. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Machine Learning (ML) software is widely used for critical decision making applications, such as loan applicant filtering [48], justice risk assessment [3, 6], and job recommendations [68]. Nevertheless, ML software can exhibit unwanted discriminatory and unfair behaviours [50]. The consequences of such bias[1] can be highly detrimental, for example affecting human rights [46], university admissions [7], profit and revenue [47]. Furthermore, many ML software system fall within legal or regulatory control, bringing to the software engineers who deploy them, additional legal risk [19, 50, 53].

ML fairness, an important non-functional testing property of ML software [66], has been widely studied in the past few years in both software engineering [10, 17, 31, 64, 66] and machine learning literature [5, 14, 38, 40]. Some approaches adapt the training data to reduce data bias (i.e., pre-processing) [15, 17, 24, 36], some create classification models that consider fairness during the training process (i.e., in-processing) [5, 16, 41, 60, 62], others apply changes to the model prediction outcomes to reduce bias (i.e., post-processing) [14, 28, 37, 39, 51].

While these bias mitigation methods are able to reduce bias in light of a given fairness metric, the improvement in fairness often comes at the cost of a lower prediction accuracy [5].[2] In other words, there is a *software engineering trade-off* between accuracy and fairness for ML software, as revealed by many previous theoretical and empirical studies [22, 24, 36].

The existence of such trade-offs brings challenges for judging the effectiveness of bias mitigation methods. Previous work presented the trade-offs in a qualitative manner. They either report and analyse the bias mitigation effectiveness by plotting the accuracy and fairness for a visual comparison [36, 39, 51], or display accuracy and fairness separately [17, 40, 61, 69] (in tables or bar charts). As far as we know, there is no trade-off **baseline**, nor is there any **quantitative** approach that can automatically evaluate and compare the fairness-accuracy trade-offs of software bias mitigation methods.

This paper introduces *Fairea*. *Fairea* is a novel **model behaviour mutation approach** to automatically **benchmarking and quantifying** the fairness-accuracy trade-off achieved by bias mitigation methods for ML software. With *Fairea*, we conduct a large-scale empirical study to benchmark and compare the effectiveness of 12 widely-studied bias mitigation methods that are publicly available in the popular IBM AI Fairness 360 library (AIF360) [4]. *Fairea* is the

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] We use "bias" and "unfairness" interchangeably to refer to the difference in ML behaviours towards protected attributes. Section 3.1 introduces the specific metrics that can be used to define and measure the concept.

[2] In this paper, the term "accuracy" refers to the standard accuracy in machine learning, which is the number of correct predictions against the total number of predictions.

first quantitative approach to benchmarking the fairness-accuracy trade-off for bias mitigation methods. Our empirical study is also the first large-scale systematic study to evaluate the effectiveness of existing bias mitigation methods.

Our results reveal that, surprisingly, in 49% of the cases, bias mitigation methods have a poor bias mitigation effectiveness. In particular, 15% which reduce bias exhibit worse trade-offs than the baseline provided by *Fairea*, while 34% lead to a decrease in accuracy and an increase in bias. Furthermore, our observations reveal the following **limitations** among the existing bias mitigation methods: 1) it is challenging to achieve a good trade-off between fairness and accuracy; 2) methods designed to optimise one fairness metric often decrease the values of other fairness metrics; 3) the effectiveness of a method is often dataset- and model-dependent. Only rarely does an approach work well on all datasets and ML models.

To conclude, this paper makes the following primary contributions:

- A **baseline** approach that enables evaluating the fairness-accuracy trade-off of ML bias mitigation methods through model behaviour mutation.
- A **quantitative measurement** for comparing different ML bias mitigation methods and trade-off parameters.
- A **large-scale study** on widely-studied bias mitigation methods in regards to their bias mitigation effectiveness as well as their achieved fairness-accuracy trade-offs.
- An open-source implementation of *Fairea* that has been made publicly available [32] for ML software developers and researchers to evaluating their bias mitigation methods.

The rest of the paper is organised as follows. Section 2 provides the current state of fairness research. Section 3 introduces the preliminaries. Section 4 introduces our approach. The experimental design is described in Section 5. Experiments and results are presented in Section 6. Section 7 concludes.

2 CURRENT STATE OF FAIRNESS RESEARCH

This section introduces the progress of fairness research in software engineering (in Section 2.1), the existing studies on the fairness-accuracy trade-offs in bias mitigation methods and how the effectiveness and trade-offs are evaluated in the literature (in Section 2.2).

For a more intuitive overview, Table 2 summarises the related works we introduce. The top rows show the research in the software engineering domain. The remaining rows are about the research on fairness-accuracy trade-off in other domains.

2.1 Software Engineering for ML Fairness

Fairness is an important non-functional testing property of ML software [66]. The testing and improvement of fairness has been regarded as a critical part in the life cycle of software development [18].

Brun and Meliou [10] published a vision paper on the fairness of ML software (where they call this software fairness). They stated that ensuring software fairness is a software engineering problem, which can be tackled from multiple directions, including requirements, architecture and design, testing, verification, and maintenance. Harrison et al. [29] studied the perceived fairness of humans in regards to ML models. Biswas and Hridesh [8] studied the fairness of ML models on crowd-sourced platforms. Finkelstein et al. [25] explored fairness in requirement analysis, and showed different needs among customers.

Several techniques have been proposed to conduct fairness testing. Themis [2, 27] used random test generation techniques to evaluate the degree of fairness. AEQUITAS [58] combined random generation and local search to explore the presence of discriminatory inputs. Aggarwal et al. [1] used symbolic execution and local explainability to generate test inputs for fairness testing. Zhang et al. [67] uses gradient computation and clustering to detect individual discriminatory instances of DNN. Sun et al. [56] proposed TransRepair combining mutation testing and metamorphic relation [54, 63], which can be adopted to automatically test and repair fairness bugs in machine translators.

The design of software can also support the reduction of bias. For example, Tramer provided a framework for detecting fairness bugs [57]. Burnett et al. [11] proposed GenderMag to identify gender bias in interfaces and respective workflows. Chakraborty et al. [17] explained the effect of bias on ML, and proposed two approaches to combat this. Zhang and Harman [64] studied the influence of enlarging feature set and training data set when building fair ML models, and found that a richer feature set could effectively improve ML fairness, which is also observed in the work of Biswas and Hridesh later on [9].

Different from these existing research, *Fairea* applies mutation on ML model behaviours to compose a baseline for evaluating bias mitigation methods. Mutation analysis has been well studied in traditional software engineering [34, 49, 65]. Recently, mutation analysis has drawn attention and been proved to be effective in automatically testing and improving ML software [33, 45, 56, 66]. As far as we know, *Fairea* is the first mutation analysis approach for fairness evaluation targeting ML software.

2.2 Fairness-Accuracy Trade-off

There have been numerous works studying the fairness-accuracy trade-off of bias mitigation methods [35]. Kamishima et al. [40] proposed a regularisation approach that adjusts the fairness-accuracy trade-off based on parameter η. Larger values of η improve fairness, but also cause a higher loss in accuracy. Berk et al. [5] normalised the loss of accuracy to study the severity of the fairness-accuracy trade-off. They call the decrease of accuracy brought by bias mitigation “Price of Fairness”. Corbett-Davies et al. [22] analysed the trade-off of public safety and racial disparities. Similar to Berk et al. [5], they showed that trade-offs can be very common in practice. Kamiran and Calders [36] gave a theoretical analysis of the trade-off. A classifier achieves an optimal trade-off if it is not dominated by another classifier (i.e., with larger accuracy and less bias).

To compare the fairness-accuracy trade-off achieved by bias mitigation methods, practitioners either observe the fairness and accuracy changes in separate graphs, or visualise them in a 2-dimensional graph (one dimension is accuracy, the other dimension is fairness) [12, 14–16, 24, 36, 37, 39, 41, 42, 51, 60]. The proposed mitigation methods are often compared with previous methods [16, 17, 37–41, 51, 61, 69], different configurations [12, 14, 24, 38–40], the original non-optimised classifier [12, 15, 36, 61, 62], or a classifier trained without using protected attributes [12, 15, 36, 69].

Table 1: State of the art of fairness research. The last column shows the section where each study is introduced in this paper. Fairness has been widely studied in both Software Engineering (the top rows) and ML literature (the bottom rows).

Authors [Ref]	Year	Venue	Description	Section
Finkelstein et al. [25]	2009	RE	Multi-objective optimisation to improve requirements for fairer software.	2.1
Burnett et al. [11]	2016	Interact Comput	Evaluation of problem-solving software for gender-inclusiveness.	2.1
Galhotra et al. [27]	2017	ESEC/FSE	Automatic test suite generation for fairness testing ("Themis").	2.1
Tramer et al. [57]	2017	EuroS&P	Framework to discover unfair treatment in data-driven applications.	2.1
Brun and Meliou [10]	2018	ESEC/FSE	Vision paper on the power of software engineering to combat fairness issues.	2.1
Udeshi et al. [58]	2018	ASE	Automated approach to discover inputs that highlight fairness violations.	2.1
Angell et al. [2]	2018	ESEC/FSE	Automated test suite generation for two types of discrimination.	2.1
Aggarwal et al. [1]	2019	ESEC/FSE	Detection of individual discrimination with black-box testing.	2.1
Friedler et al. [26]	2019	FAT	Benchmarking of various bias mitigation methods, datasets, and metrics.	2.1
Harrison et al. [29]	2020	FAT	Empircal study about the perceived fairness of machine learning models.	2.1
Biswas and Hridesh [8]	2020	ESEC/FSE	Investigation of bias in crowd-sourced machine learning models.	2.1
Zhang et al. [66]	2020	TSE	Survey on testing for machine learning systems.	2.1
Chakraborty et al. [17]	2020	ESEC/FSE	Effect of biased training data on ML fairness, and proposal of two approaches to combat this.	2.1 , 2.2
Zhang et al. [67]	2020	ICSE	A lightweight search approach to detect individual discriminatory instances.	2.1
Zhang and Harman [64]	2021	ICSE	Effect of the richness of feature/data set on ML fairness.	2.1
Biswas and Hridesh [9]	2021	ESEC/FSE	Effect of data pre-processing on ML fairness.	2.1
Calders et al. [12]	2009	ICDM	Modelling of classification models with independence constraints on attributes.	2.2
Kamiran and Calders [35]	2009	CCCT	"Massaging" of dataset to apply changes with little intrusion.	2.2
Calders and Verwer [14]	2010	DMKDFD	Three approaches to make Naive Bayes discrimination free.	2.2
Kamiran et al. [37]	2010	ICDM	Adaptation of splitting criterion and pruning rules for Decision Tree fairness.	2.2
Žliobaite et al. [69]	2011	ICDM	Developed techniques to allow for conditional discrimination if explanatory attributes are responsible.	2.2
Kamiran and Calders [36]	2012	KAIS	Three pre-processing data to remove discrimination before a classifier is learned.	2.2
Kamishima et al. [40]	2012	ECML PKDD	Discussion of causes for unfairness in ML. Proposal of regularisation to achieve fairness during training.	2.2
Kamiran et al. [38]	2012	ICDM	Relabeling of predictions with high uncertainty.	2.2
Zemel et al. [61]	2013	ICML	Encoding of data to obfuscate protected attributes. Achieves group and individual fairness	2.2
Feldman et al. [24]	2015	SIGKDD	Investigation of disparate impact (difference in classification among groups).	2.2
Corbett-Davies et al. [22]	2017	KDD	Fairness as a constrained optimisation problem.	2.2
Berk et al. [5]	2017	FAT	Fairness regularisation for linear and logistic regression with variable weights.	2.2
Zafar et al. [60]	2017	AISTATS	Fair classifiers based on a novel notion of decision boundary (un)fairness.	2.2
Calmon et al. [15]	2017	NIPS	Convex optimisation to learn fair data transformations.	2.2
Pleiss et al. [51]	2017	NIPS	Investigation of calibration while minimising for error constraints.	2.2
Kamiran et al. [39]	2018	Inf. Sci.	Framework to handle predictions with high uncertainty.	2.2
Zhang et al. [62]	2018	AIES	Bias mitigation with adversarial learning.	2.2
Kearns et al. [41]	2018	PMLR	Fairness across exponentially many subgroups to avoid gerrymandering.	2.2
Kearns et al. [42]	2019	FAT	Empirical evaluation of rich subgroup fairness (fairness constraints over a large collection of groups).	2.2
Celis et al. [16]	2019	FAT	A meta algorithm to achieve fairness based on a given fairness metric.	2.2

In all of these works, the loss of accuracy and improvement of fairness are measured and visualised separately. It is unclear whether the improved fairness is simply the consequence of the loss in accuracy. There is no unified baseline or quantitative measurement to evaluate and compare the fairness-accuracy trade-off throughout different studies.

Fairea aims to provide a unified standard to evaluate bias mitigation methods. The baseline *Fairea* provides enables developers to classify the fairness-accuracy trade-offs of a bias mitigation method into good or poor. The quantitative measurement *Fairea* provides enables developers to compare different mitigation methods in a more fine-grained way, and help tune fairness penalty parameters.

3 PRELIMINARIES

In this section, we introduce two widely-used metrics to define and measure fairness for binary classification problems in Section 3.1, and the widely-studied bias mitigation methods in Section 3.2.

3.1 Fairness Metrics

Fairness metrics are designed to define and quantitatively measure ML fairness. There are two primary types of fairness as indicated by Speicher et al. [55]: individual fairness and group fairness. *Individual fairness* is satisfied when similar individuals (according to a distance function) receive the same prediction [23]. *Group fairness* requires that the predictive performance of a classification model is equal across different groups [21], which are divided by the values of protected attributes (i.e., race, age, sex). Groups are either *privileged* (more likely to get an advantageous outcome), or *unprivileged* (more likely to get a disadvantageous outcome).

In this paper, we adopt two group fairness metrics widely-studied in the literature: Statistical Parity Difference, and Average Odds Difference. We choose group fairness metrics for two reasons. First, these metrics are widely adopted in the literature [17, 21, 22, 40]; second, most mitigation methods are designed to optimise group fairness.

In the following, we use $\hat{y}$ to denote the predictions of a classification model. We use D to denote a group (privileged or unprivileged). We use Pr to denote probability.

The *Statistical Parity Difference (SPD)* is a fairness metric requiring that decisions are made independently of protected attributes [60]. Positive and negative classifications for each demographic group should be identical over the whole population [23]:

$$\begin{aligned} SPD = Pr(\hat{y} = 1 \mid D = unprivileged) \\ -Pr(\hat{y} = 1 \mid D = privileged) \end{aligned} \tag{1}$$

The *Average Odds Difference (AOD)* is a group fairness metric that averages the differences in True Positive Rate (TPR) and False Positive Rate (FPR) among privileged and unprivileged groups [28]:

$$AOD = \frac{1}{2}((FPR_{D=unprivileged} - FPR_{D=privileged}) + (TPR_{D=unprivileged} - TPR_{D=privileged})) \tag{2}$$

Following previous work [17], we are interested in the absolute values of these metrics, thus a minimal value of zero indicates no bias detected by the corresponding metric. Larger metric values correspond to a higher bias.

3.2 Bias Mitigation Methods

In order to improve machine learning fairness, researchers have proposed three primary types of bias mitigation methods: pre-processing, in-processing and post-processing [26] methods.

3.2.1 Pre-Processing. Pre-processing methods aim at processing the training data to reduce bias in the data.

Reweighing (RW) is a pre-processing method that applies weights to different groups in the training data to achieve fairness [12, 36]. Optimised Pre-processing (OP) is a method to learn probabilistic transformations to edit labels and features of the dataset [15]. Learning Fair Representations (LFR) encodes data into an intermediate representation with the aim of obfuscating protected attribute information, while minimising the overall information disruption [61]. Other pre-processing approaches that have been proposed include the removal of data points [17, 69], and editing values of non-protected features [24].

3.2.2 In-processing. In-processing methods aim to mitigate bias during training by directly optimising algorithms.

Adversarial Debiasing (AD) is a technique that trains a classifier while simultaneously minimising the ability to predict protected attributes [62]. Prejudice Remover (PR) learns a classifier with a regularisation term to optimise fairness [40]. A fair classifier, in regards to gerrymandering, is proposed by Kearns et al. [41]. This approach applies a two-player game between a Learner and Auditor to adjust subgroups. Celis et al. [16] introduced a meta algorithm that creates an optimised classifier for a given input fairness metric. Other in-processing approaches include training process fairness constraints [5, 13], adaptation of split rule for decision trees [37], decision boundary (un)fairness [60], and latent-unbiased variables [14].

3.2.3 Post-processing. Post-processing methods change the prediction outcomes of a model to mitigate bias after the model has been trained.

Reject Option based Classification (ROC) exploits predictions with high uncertainty [38]. In particular, favourable outcomes are assigned to unprivileged groups and unfavourable outcomes to privileged groups. Post-processing can also be applied in regards to the Equal Opportunity Difference (EOD) [28, 51]. Two post-processing methods that optimise for EOD, are Equalised Odds (EO) [28, 51] and Calibrated Equalised Odds [51]. Other post-processing approaches include the modification of the probability of positive decisions for Naive Bayes (NB) [14], leaf relabelling for Decision Trees (DT) [37], and further investigation of uncertain labels [39].

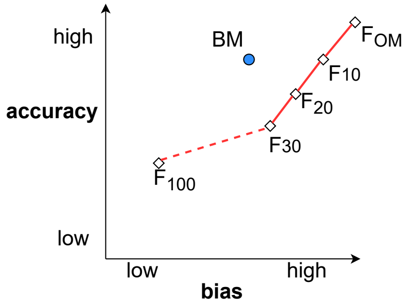


Figure 1: The *Fairea* fairness-accuracy trade-off baseline is represented by the F_{OM} trade-off point and the $F_{10}...F_{100}$ points obtained by model behaviour mutation. A bias mitigation method *BM* is effective if it exhibits a better trade-off than the *Fairea* baseline (i.e., if it is above the red line).

4 THE FAIREA APPROACH

There are three primary steps in *Fairea* to benchmarking and quantitatively evaluating bias mitigation methods.

Step1: Baseline Creation with Model Behaviour Mutation. First, *Fairea* builds the baseline by simulating the behaviours of a series of naive bias mitigation models. *Fairea* does this via model behaviour mutation. The accuracy and fairness of these simulated models, together with the original classification model, are adopted to construct the fairness-accuracy trade-off baseline.

Step2: Bias mitigation effectiveness region division. Second, *Fairea* maps the effectiveness of a bias mitigation method into five mitigation regions with the *Fairea* baseline constructed in the first step. The division of such regions helps to classify bias mitigation effectiveness into different levels, providing an intuitive overview of the changes in accuracy and fairness of a mitigation method.

Step3: Quantitative Evaluation of Trade-off Effectiveness. Third, *Fairea* quantifies the effectiveness of fairness-accuracy trade-off by measuring the gap between its effectiveness and the *Fairea* baseline. This step focuses on the bias mitigation methods that improve fairness but decrease accuracy, and enables the quantitative comparison among their trade-offs.

The details for each step are explained below.

4.1 Baseline Creation

When presenting the fairness and accuracy of a bias mitigation method in a two-dimensional coordinate system, the baseline that *Fairea* provides can be viewed as a line, as shown by Figure 1. The line is constructed by connecting the fairness-accuracy points of the original model (i.e., the model obtained by using the original classifier without applying any mitigation method) and a series of naive mitigation models constructed by model behaviour mutation. In the following, we explain how we obtain these points.

Trade-off points Collection: The starting trade-off point is based on the accuracy and fairness of the original model (i.e., the model without applying any bias mitigation method), as shown by point F_{OM} in Figure 1. The remaining points are based on the accuracy and fairness of a series of pseudo models whose behaviours are mutated from the original model. The hypothesis is that these models could improve the fairness of the original model in a naive way: by "blindly" sacrificing its accuracy with model behaviour mutation. For example, when *Fairea* mutates the original model into a random guessing model, the fairness will be greatly improved (because the predictive performance are equally worse among different protected groups), yet the accuracy is largely sacrificed. The fairness-accuracy trade-offs of such mutated models are expected to be surpassed by any reasonable bias mitigation methods. This hypothesis holds unless the original model performs even worse than a random guess model. Moreover, the bias measured by fairness metrics should monotonically decrease with an increased mutation degree. As far as we know, widely-adopted fairness metrics such as SPD, AOD, and EOD all satisfy this condition.

Mutation Degree: To obtain mutated model behaviours, we copy the original model predictions, then mutate the predictions made by this model (i.e., instead of returning the original predicted label, a random subset of the predictions is replaced by other labels). We consider different mutation degrees (i.e., the fraction of predictions to mutate) from 10% to 100%, with a step-size of 10%. For example, when the mutation degree is 10%, we randomly choose 10% of the predictions made by the original model to mutate.

Mutation Strategy: There are different mutation strategies we can choose to mutate the prediction behaviours, such as random mutation or mutating all the chosen predictions into the same label. In this paper, we choose the second strategy following the zero-normalisation principle introduced by Speicher et al. [55], which states that fairness metrics are minimised when each individual receives the same label. For an n-class classification problem, there are n labels that one can choose to conduct mutation, therefore n mutation strategies are possible, one for each label. We choose the label that will yield the highest accuracy when 100% of the predictions are mutated, in order to provide a tighter trade-off baseline. We explore the influence of different mutation strategies in RQ4 (see more details in Section 6.4).

Example: Table 2 illustrates an example of the mutation process and its corresponding fairness-accuracy trade-off for binary classification. There are 10 instances in this example (ID from 1 to 10) belonging to two groups ($g1$ and $g2$). The column "Bias" shows the absolute False Positive Rate (FPR) difference between group g_1 and g_2. A larger absolute FPR difference indicates more bias in the model towards the two groups. The original model achieves an accuracy of 0.80, with a bias of 0.5. When the mutation degree is 40%[3], the accuracy is reduced to 0.6, the fairness is improved, with a bias of 0.17. Finally, mutating 100% of the labels achieves the best fairness with a bias of 0.0, but also leads to a low accuracy of 0.50.

Baseline Construction: As shown by Table 2, each mutation degree corresponds to one mutated model, whose accuracy and fairness will form a point for constructing the baseline of *Fairea*. For example, in Figure 1, F_{10}, F_{20}, F_{30}, ..., F_{100} illustrate the fairness and accuracy of mutated models with mutation degree of 10%, 20%, ..., 100%, respectively. These points, together with the initial fairness and accuracy of the original model, are connected to form the baseline of *Fairea*. The shape of the baseline is not necessarily linear. Different fairness metrics may have different baseline shapes.

Table 2: An example of the mutation procedure in *Fairea*. Bias is represented by the absolute False Positive Rate difference (*Bias*). From the table, bias can be reduced by simply "sacrificising" accuracy through mutating model predictions.

ID	1	2	3	4	5	6	7	8	9	10		
Group	g1	g1	g1	g1	g1	g1	g2	g2	g2	g2	Accuracy	Bias
True label	0	1	0	1	0	1	1	1	0	0		
Original model	0	1	0	1	0	0	1	1	1	0	0.80	0.50
mutation degree: 40%	**1**	**1**	**1**	**1**	0	0	1	1	1	0	0.60	0.17
mutation degree: 100%	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0.50	0.00

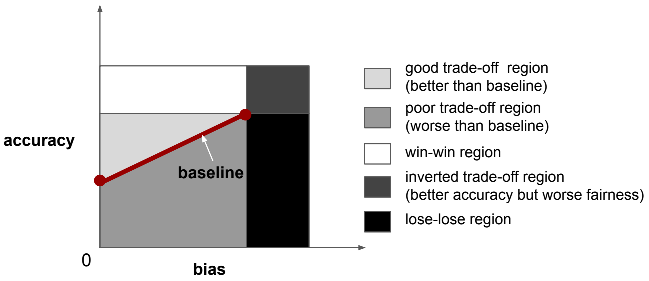


Figure 2: Mitigation regions of bias mitigation methods based on changes in accuracy and fairness. The baseline is created following the procedure we introduced in Section 4.1

Both accuracy and bias values are re-scaled to a range between 0 and 1[4] for ease of presentation, which does not affect the relative comparison results among different bias mitigation methods.

4.2 Bias Mitigation Outcome Categorisation

After obtaining a baseline, *Fairea* categorises the bias mitigation method's effectiveness into several regions, with different regions representing different categories of bias mitigation effectiveness.

As shown by Figure 2, there are five mitigation regions. If a bias mitigation method improves the accuracy and reduces the bias of the original model, it belongs to the *win-win* region. This win-win region is challenging to achieve, but is still possible [59]. A bias mitigation method falls in the *lose-lose* region if it reduces the accuracy but at the same time increases the bias of the original model (i.e., it produces worse results for both measures). If a bias mitigation improves accuracy but introduces more bias it falls in the *inverted* trade-off region. The *trade-off* region means that a bias mitigation method reduces bias but decreases accuracy. There are two types of trade-off regions: the *good trade-off* region indicates that the bias mitigation method achieves better trade-off than the baseline of *Fairea*; otherwise, it belongs to the *poor trade-off* region.

This five-region categorisation of *Fairea* helps provide an overview of the overall effectiveness of a bias mitigation method. In the following, we introduce how *Fairea* quantitatively measures the goodness of fairness-accuracy trade-off.

[3] In this example, mutating the predictions to label 1 and 0 have equal effects on the baseline strictness. We thus demonstrate only the results of mutating the predictions into 1.

[4] Given a list of values x, each element $x_i \in x$ is re-scaled given the minimum (x_{min}) and maximum (x_{max}) in x: $x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$.

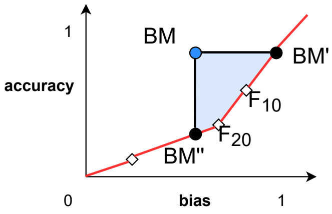


Figure 3: Quantifying the fairness-accuracy trade-off of a given bias mitigation method by measuring the area between *Fairea* baseline and the mitigation method. BM **represents the accuracy and bias of the mitigation method; the red line represents *Fairea*'s baseline; the area is constructed by connecting** BM **horizontally (**BM'**) and vertically (**BM''**) to the *Fairea* baseline.**

4.3 Trade-off Quantitative Evaluation

The *win-win*, *lose-lose*, and *poor trade-off* regions provide sufficiently clear signals on the effectiveness of the bias mitigation method. Thus, in this section, we focus on providing a quantitative measurement on the trade-off goodness of bias mitigation methods that fall into the *good trade-off* region, to facilitate a more fine-grained comparison for different bias mitigation methods.

Fairea measures the goodness of such a trade-off by calculating the area encompassed by a mitigation method and the *Fairea* baseline. Figure 3 illustrates the area obtained by connecting the bias mitigation trade-off point to the *Fairea* baseline, vertically and horizontally. The vertical line and horizontal line, together with the *Fairea* baseline, form a closed area. For example, for the case in Figure 3, the closed area is shown by the filled blue area, which is formed by five points: BM, BM', BM'', F_{10}, and F_{20}.

When comparing the area of two bias mitigation methods, the method with a larger area is regarded to have a better fairness-accuracy trade-off. Using the area as a trade-off measurement, instead of other criterion such as the distance to the baseline, ensures a reasonable comparison when the baseline is curved.

5 EXPERIMENTAL SETUP

In this section, we describe the design of the experiments we carry out to evaluate *Fairea*. We first introduce the research questions, then introduce the subjects and the experimental procedure. The implementation code and the results are available at our homepage [32] to support reproducibility and future studies.

5.1 Research Questions

Our evaluation answers the following research questions:

RQ1: Which mitigation regions do the existing bias mitigation methods fall into according to *Fairea*?

This research question evaluates the overall performance of state-of-the-art bias mitigation methods by checking how they are matched into the five mitigation regions shown by Figure 2, according to *Fairea*. To answer this question, we analyse the effectiveness of 12 popular state-of-the-art bias mitigation methods when used with three classification models, by mapping their accuracy-bias trade-off into mitigation regions as illustrated in Figure 2. We show the proportion of bias mitigation cases that fall into each mitigation region.

RQ2. What fairness-accuracy trade-off do state-of-the-art bias mitigation methods achieve based on *Fairea*?

This research question compares the methods that fall into the *good trade-off* region with the quantitative measurement *Fairea* provides. To answer this question, we calculate the area for the target method under each mitigation task (with different ML models, datasets, and fairness metrics). This allows us to quantitatively compare the methods and determine which bias mitigation method achieves the best fairness-accuracy trade-off under each task.

RQ3. Can *Fairea* be used to tune trade-off parameters for in-processing bias mitigation methods?

For in-processing methods, there are usually trade-off parameters for controlling the degree of bias mitigation. A larger trade-off parameter mitigates more bias, thus may sacrifice more accuracy. The quantitatively measurement of *Fairea* naturally enables automatic tuning of such parameters for the purpose of achieving the best trade-off. To answer the question, we investigate the in-processing methods (Prejudice Remover [40] with fairness trade-off parameter η, and Adversarial Debiasing [62] with the *adversary_loss_weight*), then check whether our measurement helps to easily spot parameters that yield good fairness-accuracy trade-off.

RQ4. How does the mutation strategy influence *Fairea*?

As explained in Section 4, different mutation strategies can be used to build *Fairea*. This question evaluates the difference among mutation strategies in providing the baseline. To answer this question, we compare the baselines created by the different strategies, to motivate the choice of the most suitable mutation strategy.

5.2 Datasets

We perform our experiments on the three[5] mostly widely-studied, real-world datasets in the fairness literature: the Adult, German, and COMPAS datasets.

The **Adult Census Income (Adult)** [44] contains financial and demographic information about individuals from the 1994 U.S. census. A classification is made to determine whether individuals have an income above 50 thousand dollars a year.

The **COMPAS** (Correctional Offender Management Profiling for Alternative Sanctions) [52] dataset contains criminal history and demographic information of criminal offenders in Broward County, Florida. Each individual is assigned with a *recidivism* label, indicating whether they were caught re-offending within two years.

The **German Credit Data (German)** [30] dataset contains credit information of 1,000 people with a classification of good or bad credit risk. Based on the given features, the protected attribute *sex* can be derived.

These datasets are the most widely-explored in the fairness literature. For example, Galhotra et al. [27] used two datasets: Adult and German; Chakraborty et al. [18] used the same three datasets.

Table 3 provides more information about these three datasets. This includes the size of the dataset (Column "Size"), the number of attributes (Column "Attri."), the favourable label, and the majority label. For each dataset, we present the protected attributes that are

[5] The number of datasets we used align with the fairness literature. According to our collection, 90% of fairness papers use no more than three datasets in their evaluation.

Table 3: Dataset information.

Dataset	Size	Attri.	Favour Label	Majority Label	Prot.Attrib	Privileged
Adult	48,842	14	1 (income >50k)	0 (75%)	Sex Race	male white
COMPAS	7,214	28	0 (no recidivism)	0 (54%)	Sex Race	female caucasian
German	1,000	20	1 (good credit)	1 (70%)	Sex	male

present in the dataset (Column "Prot.Attrib"). Privileged groups are outlined for protected attributes (Column "Priviledged").

5.3 Bias Mitigation Methods

We explore all the three types of bias mitigation methods during our evaluation (see more details in Section 3.2). Under each type, we choose widely-studied methods, which have been implemented in the IBM AIF360 library:

- **Pre-processing**: Optimised Pre-processing (OP), Learning Fair Representations (LFR), Reweighing (RW);
- **In-processing**: Prejudice Remover (PR), Adversarial Debiasing (AD);
- **Post-processing**: Reject Option Classification (ROC), Calibrated Equalised Odds (CO), Equalised odds (EO).

In AIF360, ROC and CO are implemented with three different fairness metrics to guide the bias mitigation process. ROC offers a choice between SPD, AOD, and EOD; CO offers a choice between False Negative rate (FNR), False Positive Rate (FPR), and a weighted metric to combine both. We implemented and evaluated every of the three methods for ROC and CO. All together, we study 12 bias mitigation methods.

5.4 Experimental Configuration

Pre-processing and post-processing methods are model independent. We implement them using three traditional classification models, which have been widely adopted in previous works that study fairness: Logistic Regression (LR) [17, 24, 38–40, 60], Decision Tree (DT) [38, 39], and Support Vector Machine (SVM) [24, 39, 60]. As in previous work [17, 38, 39], we use the default configuration for each classifier, as provided by Scipy.[6]

The two in-processing methods studied in this paper have their own model with different trade-off parameters. In this case, to build *Fairea*, when getting the original model, we turn off the trade-off parameters (so that such a model does not use any bias mitigation function); when evaluating the effectiveness of a in-process method in RQ1 and RQ2, we use its default trade-off parameter. In RQ3, we explore the trade-off performance of different parameters and investigate whether *Fairea*'s quantitative measurement helps to tune the parameters to get the best trade-off.

We apply each of the bias mitigation methods to the three datasets and their protected attributes, with three ML models and two fairness metrics. Thus, for each mitigation method, it will be evaluated per *(dataset, protected attribute, ML model, fairness metric)* combination. We call such as a combination a mitigation **task**.

Each optimisation process is repeated 50 times, each time with a random re-spilt of the data based on a fixed train-test split ratio 7:3. We use the mean value of these multiple runs to represent the method's average performance, as a common practice in the fairness literature [8, 16]. We treat each single run as an individual **mitigation case**, and present the proportion of cases that fall into each bias mitigation region for a bias mitigation method (to answer RQ1). The baseline is also obtained by repeating the label model behaviour mutation procedure 50 times for each mutation degree (10%, 20%, ..., 100%).

The source code containing the implementation of *Fairea* and the implementation/configuration of each bias mitigation method, as well as the results, are available in our project repository [32].

5.5 Threats to Validity

The primary threat to internal validity lies in the implementation of *Fairea*. To reduce this threat, the authors independently reviewed the implementation code. The adoption of IBM AIF360 framework [4], a widely adopted fairness tool in software fairness [8, 17], also reduces such threat. The threats to external validity lie primarily with the subjects investigated. To reduce this threat, we use the three most widely adopted datasets in fairness research. We study 12 bias mitigation methods, with different classification models, to obtain more generalised conclusions. Moreover, we make our scripts and data publicly available, to allow for reproductions, replications and its adoption in future bias mitigation studies [32].

6 EMPIRICAL STUDY RESULTS

This section presents the results of our experiments to answer the research questions explained in Section 5.1.

6.1 RQ1: Mitigation Region Distribution

The first research question checks the mitigation region distribution of the existing bias mitigation methods. We apply bias mitigation methods to the three datasets to evaluate their region distribution (Section 4) according to the baseline provided by *Fairea*.

We apply each pre- and post-processing bias mitigation method on three classification models (LR, DT, SVM) used for five bias mitigation tasks (i.e., *Adult-sex*, *Adult-race*, *COMPAS-sex*, *COMPAS-race*, *German-sex*). Each task is repeated for 50 times with different training-test splits. DT achieves a prediction accuracy below the majority class for the German dataset. Therefore, it does not meet our baseline requirement (as introduced in Section 4.1) and is disregarded in the subsequent experiments. Thus, for each bias mitigation method, there are 5*3*50-50 = 700 evaluations.

For each in-processing method, as we introduced in Section 5.4, we build the baseline upon an original model without applying bias mitigation (with the trade-off parameter set to 0). For Prejudice Remover, its accuracy on the COMPAS/German dataset is too low to be reduced by mutation, we thus only present its results on the Adult dataset. Therefore, our experiment conducts 50 evaluates on Prejudice Remover, and 250 evaluations on Adversarial Debiasing.

We then calculate the percentage of evaluations that fall into each region. We use the proportion as a high-level indication of the bias mitigation performance of each method.

6.1.1 *Overall results.*

Table 4 shows the results of the region classification of bias mitigation methods. Each row represents a bias mitigation method. Each cell contains a percentage of scenarios

[6] https://www.scipy.org/.

Table 4: RQ1: Proportion of mitigation cases that fall into each mitigation region. We observe that half of the existing bias mitigation methods either decrease accuracy and increase bias (*lose-lose*) of the original model, or have a worse trade-off than the *Fairea* baseline (*poor trade-off*).

Bias mitigation method		Statistical Parity Difference (SPD)					Average Odds Difference (AOD)				
		Lose-Lose	Poor Trade-off	Inverted	Good Trade-off	Win-Win	Lose-Lose	Poor Trade-off	Inverted	Good Trade-off	Win-Win
Pre	LFR	19%	48%	0%	20%	13%	33%	38%	0%	17%	13%
	OP	11%	16%	14%	40%	18%	20%	11%	13%	36%	20%
	RW	5%	14%	4%	54%	23%	12%	12%	3%	49%	24%
In	PR	1%	6%	0%	85%	8%	11%	0%	1%	81%	7%
	AD	29%	5%	12%	44%	10%	55%	5%	15%	17%	8%
Post	CO_{fnr}	52%	2%	15%	30%	2%	52%	5%	14%	26%	2%
	CO_{fpr}	58%	20%	7%	7%	8%	66%	13%	7%	6%	8%
	$CO_{weighed}$	64%	3%	21%	6%	7%	64%	2%	20%	6%	8%
	ROC_{SPD}	19%	26%	0%	45%	9%	28%	25%	0%	37%	9%
	ROC_{AOD}	45%	16%	4%	26%	9%	26%	28%	3%	34%	9%
	ROC_{EOD}	47%	15%	4%	26%	9%	43%	14%	3%	31%	9%
	EO	11%	6%	6%	69%	8%	14%	4%	7%	67%	8%
	Mean	33%	16%	7%	33%	10%	36%	15%	7%	31%	11%

that fall into corresponding regions for a mitigation method. The last row shows the overall ratios for each mitigation region.

We make the following primary observations from Table 4. **First**, to our surprise, a large proportion of bias mitigation performance falls into the *lose-lose* trade-off region. For example, for the CO_{fnr} post-processing method, the proportion is as high as 52% for AOD. The mean value of the *lose-lose* proportion is 33% for SPD and 36% for AOD, which means that those bias mitigation methods perform worse than the original model. For SPD, 49% of the bias mitigation methods perform worse than *Fairea* while 43% perform better. Similarly, 51% of the bias mitigation methods achieve worse trade-offs than *Fairea* for AOD, while being better among 42% of the evaluations.

One possible reason for this is that mitigation methods are often designed to optimise one fairness metric, but such kind of one-target optimisation usually affects other fairness metrics [20, 43]. For example, CO_{fnr} and CO_{fpr} are designed to optimise the difference of false negative/positive rate between privileged and unprivileged groups. Their *lose-lose* percentages measured by SPD and AOD are over 50%. Nevertheless, we observe that when using the same metric to optimise and measure mitigation performance, the *lose-lose* percentages are still high (i.e., 19% for ROC_{SPD} measured by SPD, and 26% for ROC_{AOD} measured by AOD).

Second, a notable proportion of evaluations fall into the poor trade-off region (16% for SPD and 15% for AOD). While this means that they achieve more fairness than the original model, their fairness-accuracy trade-off is worse than the baseline of *Fairea*.

We also observe a small ratio of evaluations falling into the *win-win* region (10%) or *inverted* trade-off region (7%). A larger proportion of pre-processing methods belong to the *win-win* region, in comparison to in- and post-processing methods. This may indicate that optimising training data has more promises in providing solutions to optimise both accuracy and fairness.

Third, pre-processing methods are more likely to fall into the *win-win* region with both accuracy and fairness being improved. For example, for SPD, the average proportion of pre-processing methods that fall into the *win-win* region is 18%, which is only 7% for post-processing methods. This suggests that, if one pursues improving both accuracy and fairness, it might be favourable to pre-process the training data and prevent the bias from reaching the model, than to mitigate the bias after the model has learned the bias from the data.

Table 5: RQ1: Averaged proportion of mitigation cases that fall into each mitigation region organised by different ML models (top three rows) and datasets (bottom five rows). The differences across models and datasets indicate that the effectiveness of the methods we studied are model and dataset dependent.

	Lose-Lose	Poor	Inverted	Good	Win-Win
LR	30%	20%	3%	41%	6%
DT	43%	8%	12%	24%	12%
SVM	28%	18%	6%	36%	13%
Adult - Sex	49%	17%	1%	32%	1%
Adult - Race	43%	15%	3%	37%	2%
COMPAS - Sex	18%	13%	10%	35%	23%
COMPAS - Race	26%	8%	15%	34%	16%
German - Sex	34%	27%	9%	19%	12%

6.1.2 Comparison among different models and datasets. We further analyse the region distribution based on ML models (for pre- and post-processing methods) and datasets. The purpose is to investigate whether the performance of different bias mitigation methods are influenced by ML models or datasets.

Table 5 shows the results. Among the three classification models, we observe that different models have different results, which indicates that the effectiveness of pre- and post-processing methods are model dependent. Overall, LR and SVM have a better effectiveness (higher percentage of *good trade-offs*) than DT.

Among different datasets and protected attributes, the differences are also notable. We observe that for the COMPAS dataset, there are more scenarios in the win-win region and fewer scenarios in the lose-lose region. We suspect that this is because COMPAS dataset

Table 6: RQ2: Trade-off assessment results for pre-processings and post-processing methods. For each method in the *good trade-off* region, a trade-off measurement value provided by *Fairea* is given; for other regions the region type is displayed. The values in bold indicate the best mitigation method for each mitigation task. From this table, we observe that *Fairea* provides distinguishable measurements for trade-off comparison, and helps to detect the best mitigation method under each bias mitigation task.

			Logistic Regression (LR)					Decision Tree				SVM				
			Adult		COMPAS		German	Adult		COMPAS		Adult		COMPAS		German
			Sex	Race	Sex	Race	Sex	Sex	Race	Sex	Race	Sex	Race	Sex	Race	Sex
Statistical Parity Difference	Pre	LFR	poor	poor	poor	poor	poor	poor	poor	poor	poor	poor	poor	poor	poor	poor
		OP	poor	0.002	0.076	0.011	lose-lose	0.008	0.111	**win-win**	inverted	0.000	0.002	**win-win**	inverted	lose-lose
		RW	0.001	0.007	0.195	0.138	poor	**0.029**	**0.176**	**win-win**	**win-win**	0.001	0.029	**win-win**	**win-win**	lose-lose
	Post	CO_{fnr}	0.014	0.019	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	0.011	0.012	lose-lose	lose-lose	lose-lose
		CO_{fpr}	lose-lose	lose-lose	poor	lose-lose	**0.115**	lose-lose	lose-lose	0.000	lose-lose	lose-lose	lose-lose	poor	lose-lose	**0.063**
		$CO_{weighed}$	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose
		ROC_{SPD}	0.006	poor	**0.274**	**0.273**	poor	lose-lose	lose-lose	0.112	0.043	poor	poor	0.264	0.258	poor
		ROC_{AOD}	lose-lose	lose-lose	0.185	0.185	poor	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	poor	0.172	0.180	poor
		ROC_{EOD}	lose-lose	lose-lose	0.149	0.093	poor	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	poor	0.126	0.108	poor
		EO	**0.024**	**0.067**	0.104	0.159	0.038	poor	lose-lose	0.002	0.000	**0.021**	**0.054**	0.118	0.166	0.018
Average Odds Difference	Pre	LFR	lose-lose	lose-lose	poor	poor	poor	lose-lose	lose-lose	poor	poor	poor	poor	poor	poor	poor
		OP	poor	0.028	0.108	0.027	lose-lose	poor	lose-lose	**win-win**	inverted	0.028	0.041	**win-win**	inverted	lose-lose
		RW	0.041	0.039	0.213	0.153	poor	**0.016**	lose-lose	**win-win**	**win-win**	0.009	0.026	**win-win**	**win-win**	lose-lose
	Post	CO_{fnr}	0.000	0.066	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	0.037	0.087	lose-lose	lose-lose	lose-lose
		CO_{fpr}	lose-lose	lose-lose	poor	lose-lose	**0.054**	lose-lose	lose-lose	0.000	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	**0.038**
		$CO_{weighed}$	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose
		ROC_{SPD}	lose-lose	poor	**0.281**	**0.201**	poor	lose-lose	lose-lose	0.140	0.040	poor	poor	0.240	0.215	lose-lose
		ROC_{AOD}	poor	poor	0.229	0.187	poor	lose-lose	lose-lose	lose-lose	lose-lose	poor	0.001	0.204	0.201	lose-lose
		ROC_{EOD}	lose-lose	lose-lose	0.197	0.112	poor	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose	0.003	0.154	0.141	lose-lose
		EO	**0.169**	**0.198**	0.111	0.159	0.029	lose-lose	lose-lose	0.003	0.000	**0.158**	**0.087**	0.120	0.160	0.010

is more balanced than Adult and German (54% of majority labels v.s. 75% and 70% majority labels according to Table 3).

To conclude, for RQ1, we have the following answer:

> Answer to RQ1: Surprisingly, approximately 50% of the bias mitigation scenarios have a poor mitigation effectiveness, with 34% of them decreasing accuracy and increasing bias (*lose-lose*), and 15% of them exhibiting a *poor trade-off* according to *Fairea*.

6.2 RQ2: Quantitative Measurement for Fairness-accuracy Trade-off

To answer RQ2, we present the quantitative measurement results of the fairness-accuracy trade-off achieved by different bias mitigation methods with *Fairea*. We quantify results that fall into the *good trade-off* region, as the other regions are either strictly dominating the original model (*win-win*), dominated by the *Fairea* baseline (*lose-lose* and *poor trade-off*, or do not improve fairness (*inverted*). We use the arithmetic mean results of the 50 runs to indicate the average level of mitigation effectiveness.

Table 6 shows the results for pre- and post-processing bias mitigation methods. The values in bold indicate the best mitigation method for each mitigation task (i.e., the combination of dataset, protected attribute, ML model, and fairness metric). From the table, the quantitative trade-off measurement *Fairea* provides helps to compare different the trade-offs among different mitigation method, and to choose the best one under each mitigation task.

The same as RQ1, we observe that the trade-offs of bias mitigation methods are highly dataset dependent. For example, the best trade-off on the Adult dataset is achieved by EO (highest scores for both AOD and SPD). The best trade-off on German is achieved by CO_{fpr}.

We also explore whether the protected attribute considered under each dataset impacts the performance of bias mitigation methods. From Table 6, for the same dataset, different protected attributes have very similar patterns. Specifically, in 85% (102/120) of the cases, bias mitigation methods are classified into the same mitigation region with different protected attributes. This suggests, that the protected attribute has a limited impact on the trade-off performance of bias mitigation methods.

Table 7: RQ2: Trade-off assessment results for in-processing methods.

		Adult		COMPAS		German
		Sex	Race	Sex	Race	Sex
Statistical Parity Difference	PR	0.042	0.003	NA	NA	NA
	AD	0.176	0.042	lose-lose	lose-lose	lose-lose
Average Odds Difference	PR	0.090	0.011	NA	NA	NA
	AD	lose-lose	lose-lose	lose-lose	lose-lose	lose-lose

Due to the different characteristics of in-processing methods, we provide their quantitative results separately in Table 7. Prejudice Remover is not applicable to the COMPAS and German dataset (see Section 6.1.1 for more details) so we mark the results as "NA". Adversarial Debiasing is applicable for all three datasets, however only achieves *good trade-offs* on the Adult dataset for SPD. All the other trade-offs are in the *lose-lose* region. However, when comparing the two in-processing methods on Adult dataset measured by

SPD, Adversarial Debiasing has a better trade-off than Prejudice Remover.

These observations lead to the following answer to RQ2:

> Answer to RQ2: The quantitative measurement of *Fairea* allows us to determine and compare fairness-accuracy trade-offs achieved by different bias mitigation methods. For example, *Fairea* measures that the EO method achieves a 71.4% better trade-off than CO_{fnr} (i.e., 0.024 vs. 0.014) for the case LR-Adult-Sex under Statistical Parity Difference. Different datasets have different bias mitigation methods that achieve the best trade-off (i.e., EO for Adult; CO_{fpr} for German).

6.3 RQ3: Parameter Tuning

In RQ3, we investigate the effectiveness of *Fairea* in evaluating the parameter tuning for in-processing methods. For this purpose, we apply *Fairea* on the original model of Prejudice Remover with $\eta = 0$, and Adversarial Debiasing with an $adversary\_loss\_weight = 0$.

As in previous experiments, we perform 50 train-test splits for all numerical values of η between 1-100 for PR, with a step size of 1. We evaluate *adversary_loss_weights* in a range of 0.05-1, with a step size of 0.05. Due to limited space, we choose the Adult dataset as an example to illustrate experiments on parameter tuning. Full results are available in our project repository [32].

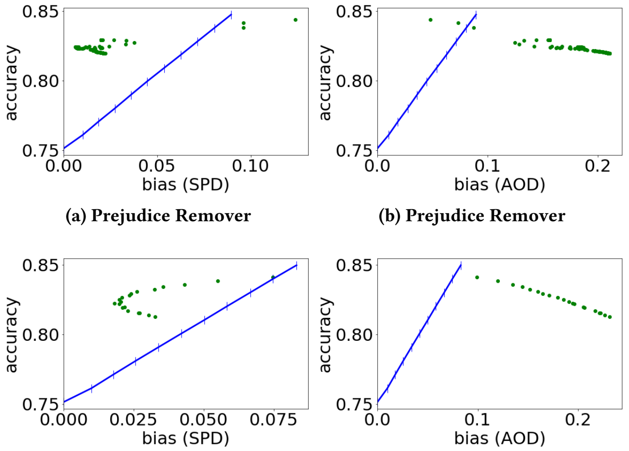
We first plot the accuracy and fairness achieved by each parameter setting, shown in Figure 4. For both methods, all parameter settings for SPD achieves better trade-off than the original model. However, the bias mitigation effectiveness for AOD is much worse.

Although the different parameters all belong to the good trade-off region, it is difficult to determine which parameter setting achieves the best fairness-accuracy trade-off. We therefore investigate whether our quantitative measurement in *Fairea* helps spot the parameter that achieves the best trade-off.

Figure 5 shows these results. Sub-Figures 5 (a) and (b) show the trade-off measurement results provided by *Fairea* with different trade-off parameters. The remaining sub-figures show the accuracy and fairness changes separately without *Fairea*.

From sub-Figure 5.(a) and sub-Figure 5.(b), we observe that, when the trade-off parameter changes, our trade-off measurement first increases, then decreases, with a turning point indicating the parameter with the best trade-off. However, from the remaining sub-figures, without the support from *Fairea*, it is difficult to choose a parameter with accuracy and fairness changing at the same time.

Of course, in practice, the desired trade-offs may depend on the application scenario and the specific requirement. Some applications may demand a higher degree of fairness, with the capability of enduring more accuracy loss. However, the quantitative measurement in *Fairea* provides an engineering solution for finding the best trade-off as a reference for developers.



(a) Prejudice Remover **(b) Prejudice Remover**

(c) Adversarial Debiasing **(d) Adversarial Debiasing**

Figure 4: RQ3: Accuracy and fairness achieved by Prejudice Remover (sub-figure a and b), and Adversarial Debiasing (sub-figure c and d) with different parameters on Adult-sex. Each green point represents a trade-off parameter, the blue line represents the *Fairea* baseline.

> Answer to RQ3: Our trade-off measurement helps to quantify the fairness-accuracy trade-offs achieved by in-processing methods with different trade-off parameter settings, and to identify parameters that achieve the best fairness-accuracy trade-offs.

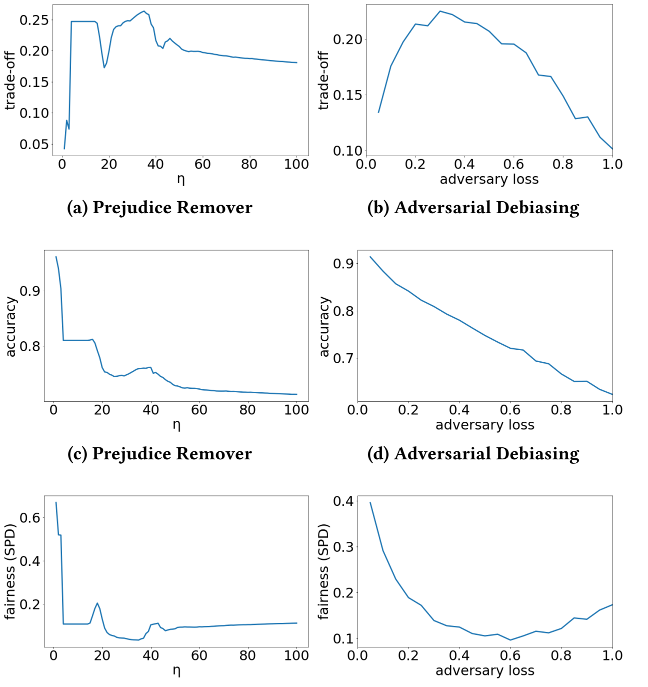
6.4 RQ4: Influence of Mutation Strategies

This research question is designed to investigate how the mutation strategy for simulating naive mitigation methods affects the construction of the *Fairea* baseline. We show and compare three different mutation strategies: replace labels with "0", replace labels with "1", and replace labels at random.

Figure 6 shows the accuracy and fairness (SPD, AOD) of the three mutation strategies. We analysed all three datasets, the conclusions are identical, so we only present results for the Adult-sex task (full results are available in our project repository [32]).

As can be seen, when we mutate the labels with the majority class label (0 in the case for Adult-sex), its baseline is on top of the other two strategies. This means that overwriting with the *majority* label provides a more strict baseline than the other two strategies. Mutation with the *minority* class label (1 in the case for Adult-sex) instead leads to a baseline with lower accuracy on the same level of fairness. Using such a baseline would provide weaker conditions when checking the trade-off of bias mitigation methods. Replacement with random labels leads to a baseline in-between the other two strategies, but with 100% labels replaced, the fairness values are not minimised at zero because of the imbalanced data distribution.

In this paper, we adopted the strategy of mutating predictions with the majority class label in the training data. Although this is



(a) Prejudice Remover **(b) Adversarial Debiasing**

(c) Prejudice Remover **(d) Adversarial Debiasing**

(e) Prejudice Remover **(f) Adversarial Debiasing**

Figure 5: RQ3: In-processing trade-off parameter tuning with *Fairea*. The horizontal axis in each sub-figure shows different parameter values. Figure (a) and (b) show the trade-off measurement changes provided by *Fairea*. Figure (c), (d), (e), (f) show the changes of accuracy and fairness separately.

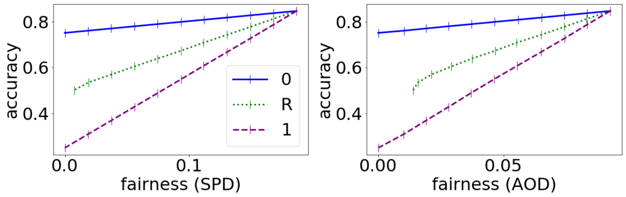


Figure 6: RQ4: Comparison of three mutation strategies (mutate the original prediction into 0**,** 1**, or randomly (***R***)) on the Adult dataset with the protected attribute *sex*.**

the most strict among the three strategies, it is still a naive bias mitigation method achieved simply by label overwriting, which we expect that a reasonably effective bias mitigation method should outperform.

> Answer to RQ4: Among the different mutation strategies we explored, replacing labels with the majority class label for a dataset leads to the strictest baseline.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed Fairea, a novel approach to evaluating and quantitatively measuring the fairness-accuracy trade-off. There are three primary questions that previous work could not answer without Fairea: 1) The Fairea baseline tells whether a bias mitigation method trades accuracy for fairness (or even worse than that). The qualitative approach used by previous work is not able to differentiate "good trade-off" and "poor trade-off" like Fairea does; 2) Fairea provides extra information for developers by telling whether bias mitigation method A outperforms method B when they both achieve a "good trade-off"; 3) Fairea helps to tune the fairness mitigation parameter for in-processing methods.

We performed a large scale empirical study to evaluate our baseline *Fairea* on three widely used datasets and 12 bias mitigation methods. We found that half of the bias mitigation methods are not able to achieve a reasonable bias mitigation effectiveness (either achieving a worse trade-off than our baseline, or decreasing accuracy and increasing bias). In addition, few methods perform well on all datasets and all models. These results show the limitations and challenges of the existing bias mitigation methods, suggesting the need for further research effort on improving ML software fairness. In future, we plan to involve *Fairea* into the bias mitigation process to guide mitigation optimisation and develop new bias mitigation methods.

ACKNOWLEDGEMENTS

The work of Max Hort, Jie Zhang, Federica Sarro and Mark Harman at UCL is funded by the ERC advanced grant no. 741278 (EPIC: Evolutionary Program Improvement Collaborators).

REFERENCES

[1] Aniya Aggarwal, Pranay Lohia, Seema Nagar, Kuntal Dey, and Diptikalyan Saha. 2019. Black box fairness testing of machine learning models. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 625–635.

[2] Rico Angell, Brittany Johnson, Yuriy Brun, and Alexandra Meliou. 2018. Themis: Automatically testing software for discrimination. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 871–875.

[3] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias. ProPublica. *See https://www. propublica. org/article/machine-bias-risk-assessments-in-criminal-sentencing* (2016).

[4] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943* (2018).

[5] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth. 2017. A Convex Framework for Fair Regression. *FAT-ML Workshop* (2017).

[6] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2018. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research* (2018), 0049124118782533.

[7] Peter J Bickel, Eugene A Hammel, and J William O'Connell. 1975. Sex bias in graduate admissions: Data from Berkeley. *Science* 187, 4175 (1975), 398–404.

[8] Sumon Biswas and Rajan Hridesh. 2020. Do the Machine Learning Models on a Crowd Sourced Platform Exhibit Bias? An Empirical Study on Model Fairness. *arXiv preprint arXiv:2005.12379* (2020).

[9] Sumon Biswas and Hridesh Rajan. 2021. Fair Preprocessing: Towards Understanding Compositional Fairness of Data Transformers in Machine Learning Pipeline. In *The ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE) (to appear)*. IEEE.

[10] Yuriy Brun and Alexandra Meliou. 2018. Software fairness. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 754–759.

[11] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. 2016. GenderMag: A method for evaluating software's gender inclusiveness. *Interacting with Computers* 28, 6 (2016), 760–787.

[12] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. 2009. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*. IEEE, 13–18.

[13] Toon Calders, Asim Karim, Faisal Kamiran, Wasif Ali, and Xiangliang Zhang. 2013. Controlling attribute effect in linear regression. In *2013 IEEE 13th international conference on data mining*. IEEE, 71–80.

[14] Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292.

[15] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*. 3992–4001.

[16] L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. 2019. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 319–328.

[17] Joymallya Chakraborty, Suvodeep Majumder, Zhe Yu, and Tim Menzies. 2020. Fairway: a way to build fair ML software. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 654–665.

[18] Joymallya Chakraborty, Tianpei Xia, Fahmid M. Fahid, and Tim Menzies. 2019. Software Engineering for Fairness: A Case Study with Hyperparameter Optimization. arXiv:1905.05786 [cs.SE]

[19] Jiahao Chen, Nathan Kallus, Xiaojie Mao, Geoffry Svacha, and Madeleine Udell. 2019. Fairness under unawareness: Assessing disparity when protected class is unobserved. In *Proceedings of the conference on fairness, accountability, and transparency*. 339–348.

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

[21] Sam Corbett-Davies and Sharad Goel. 2018. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023* (2018).

[22] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 797–806.

[23] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[24] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 259–268.

[25] Anthony Finkelstein, Mark Harman, S Afshin Mansouri, Jian Ren, and Yuanyuan Zhang. 2009. A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. *Requirements engineering* 14, 4 (2009), 231–245.

[26] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 329–338.

[27] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. 2017. Fairness testing: testing software for discrimination. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ACM, 498–510.

[28] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*. 3315–3323.

[29] Galen Harrison, Julia Hanson, Christine Jacinto, Julio Ramirez, and Blase Ur. 2020. An empirical study on the perceived fairness of realistic, imperfect machine learning models. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 392–402.

[30] Dr. Hans Hofmann. [n.d.]. Statlog (german credit data) data set. http://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

[31] Jennifer Horkoff. 2019. Non-functional requirements for machine learning: Challenges and new directions. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*. IEEE, 386–391.

[32] Max Hort, Jie Zhang, Federica Sarro, and Mark Harman. [n.d.]. On-line Appendix to the paper Fairea: A Model Behaviour Mutation Approach to Benchmarking Bias Mitigation Methods. https://github.com/maxhort/Fairea/

[33] Gunel Jahangirova and Paolo Tonella. 2020. An empirical evaluation of mutation operators for deep learning systems. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*. IEEE, 74–84.

[34] Yue Jia and Mark Harman. 2010. An analysis and survey of the development of mutation testing. *IEEE transactions on software engineering* 37, 5 (2010), 649–678.

[35] Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*. IEEE, 1–6.

[36] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2012), 1–33.

[37] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *2010 IEEE International Conference on Data Mining*. IEEE, 869–874.

[38] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. 2012. Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 924–929.

[39] Faisal Kamiran, Sameen Mansha, Asim Karim, and Xiangliang Zhang. 2018. Exploiting reject option in classification for social discrimination control. *Information Sciences* 425 (2018), 18–33.

[40] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 35–50.

[41] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness *(Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, Stockholmsmässan, Stockholm Sweden, 2564–2572. http://proceedings.mlr.press/v80/kearns18a.html

[42] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2019. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 100–109.

[43] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807* (2016).

[44] Ron Kohav. [n.d.]. Adult data set. http://archive.ics.uci.edu/ml/ datasets/adult.

[45] Lei Ma, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Felix Juefei-Xu, Chao Xie, Li Li, Yang Liu, Jianjun Zhao, et al. 2018. Deepmutation: Mutation testing of deep learning systems. In *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 100–111.

[46] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* (2019).

[47] Jakub Mikians, László Gyarmati, Vijay Erramilli, and Nikolaos Laoutaris. 2012. Detecting price and search discrimination on the internet. In *Proceedings of the 11th ACM workshop on hot topics in networks*. 79–84.

[48] Amitabha Mukerjee, Rita Biswas, Kalyanmoy Deb, and Amrit P Mathur. 2002. Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management. *International Transactions in operational research* 9, 5 (2002), 583–597.

[49] Mike Papadakis, Marinos Kintis, Jie Zhang, Yue Jia, Yves Le Traon, and Mark Harman. 2019. Mutation testing advances: an analysis and survey. In *Advances in Computers*. Vol. 112. Elsevier, 275–378.

[50] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 560–568.

[51] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. In *Advances in Neural Information Processing Systems*. 5680–5689.

[52] propublica. [n.d.]. data for the propublica story 'machine bias'. https://github.com/propublica/compas-analysis/.

[53] Andrea Romei and Salvatore Ruggieri. 2011. A multidisciplinary survey on discrimination analysis.

[54] Sergio Segura, Gordon Fraser, Ana B Sanchez, and Antonio Ruiz-Cortés. 2016. A survey on metamorphic testing. *IEEE Transactions on software engineering* 42, 9 (2016), 805–824.

[55] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. 2018. A unified approach to quantifying algorithmic unfairness: Measuring individual &group unfairness via inequality indices. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2239–2248.

[56] Zeyu Sun, Jie M Zhang, Mark Harman, Mike Papadakis, and Lu Zhang. 2019. Automatic Testing and Improvement of Machine Translation. *arXiv preprint arXiv:1910.02688* (2019).

[57] Florian Tramer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. 2017. FairTest: Discovering unwarranted associations in data-driven applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 401–416.

[58] Sakshi Udeshi, Pryanshu Arora, and Sudipta Chattopadhyay. 2018. Automated directed fairness testing. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. 98–108.

[59] Michael Wick, Jean-Baptiste Tristan, et al. 2019. Unlocking Fairness: a Trade-off Revisited. In *Advances in Neural Information Processing Systems*. 8783–8792.

[60] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*. 962–970.

[61] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International Conference on Machine Learning*. 325–333.

[62] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 335–340.

[63] Jie Zhang, Junjie Chen, Dan Hao, Yingfei Xiong, Bing Xie, Lu Zhang, and Hong Mei. 2014. Search-based inference of polynomial metamorphic relations. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*. 701–712.

[64] Jie Zhang and Mark Harman. 2021. "Ignorance and Prejudice" in Software Fairness. In *2021 IEEE/ACM 43th International Conference on Software Engineering (ICSE)*. IEEE.

[65] Jie Zhang, Lingming Zhang, Mark Harman, Dan Hao, Yue Jia, and Lu Zhang. 2018. Predictive mutation testing. *IEEE Transactions on Software Engineering* 45, 9 (2018), 898–918.

[66] J. M. Zhang, M. Harman, L. Ma, and Y. Liu. 2020. Machine Learning Testing: Survey, Landscapes and Horizons. *IEEE Transactions on Software Engineering* (2020), 1–1.

[67] Peixin Zhang, Jingyi Wang, Jun Sun, Guoliang Dong, Xinyu Wang, Xingen Wang, Jin Song Dong, and Ting Dai. 2020. White-box fairness testing through adversarial sampling. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. 949–960.

[68] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496* (2018).

[69] Indre Žliobaite, Faisal Kamiran, and Toon Calders. 2011. Handling conditional discrimination. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, 992–1001.