Fairway: A Way to Build Fair ML Software

Joymallya Chakraborty
jchakra@ncsu.edu
North Carolina State University
Raleigh, USA

Suvodeep Majumder
smajumd3@ncsu.edu
North Carolina State University
Raleigh, USA

Zhe Yu
zyu9@ncsu.edu
North Carolina State University
Raleigh, USA

Tim Menzies
timm@ieee.org
North Carolina State University
Raleigh, USA

ABSTRACT

Machine learning software is increasingly being used to make decisions that affect people's lives. But sometimes, the core part of this software (the learned model), behaves in a biased manner that gives undue advantages to a specific group of people (where those groups are determined by sex, race, etc.). This *"algorithmic discrimination"* in the AI software systems has become a matter of serious concern in the machine learning and software engineering community. There have been works done to find "algorithmic bias" or "ethical bias" in software system. Once the bias is detected in the AI software system, mitigation of bias is extremely important. In this work, we **a)** explain how ground truth bias in training data affects machine learning model fairness and how to find that bias in AI software, **b)** propose a method **Fairway** which combines pre-processing and in-processing approach to remove ethical bias from training data and trained model. Our results show that we can find bias and mitigate bias in a learned model, without much damaging the predictive performance of that model. We propose that (1) testing for bias and (2) bias mitigation should be a routine part of the machine learning software development life cycle. Fairway offers much support for these two purposes.

CCS CONCEPTS

• **Software and its engineering → Software creation and management**; • **Computing methodologies → Machine learning**.

KEYWORDS

Software Fairness, Fairness Metrics, Bias Mitigation

ACM Reference Format:
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 European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20), November 8–13, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3368089.3409697

1 INTRODUCTION

Software plays an important role in many high-stake applications like finance, hiring, admissions, criminal justice. For example, software generates models that decide whether a patient gets released from hospital or not [1, 2]. Also, software helps us to choose what products to buy [3]; which loan applications are approved [4]; which citizens get bail or sentenced to jail [5]. Further, self-driving cars are run by software which may lead to damage of property or human injury [6]. These all are examples of software systems where the core part is machine learning model.

One problem with any machine learning (ML) model is they are all a form of statistical discrimination. Consider, for example, the discriminatory nature of decision tree learners that deliberately selects attributes to divide that data into different groups. Such discrimination becomes unacceptable and unethical when it gives certain privileged groups advantages while disadvantaging other unprivileged groups (e.g. groups divided by age, gender, skin color, etc). In such situations, discrimination or bias is not only objectionable, but illegal.

Much recent SE researchers presume that the construction of fairer, less biased AI systems is a research problem for software engineers [7, 8]. We assert that modern principles for software engineering should encompass principles for building AI/ML software. This paper mainly focuses on improving AI software to satisfy an important and specific non-functional requirement - **fairness**. In the age of agile software development, requirements gathering, architectural design, implementation, testing, verification - in any step, bias may get injected into software system. So, test and mitigation is now a primary concern in any SE task that uses AI.

Many researchers agree that fairness is a SE problem worthy of SE research. For example, entire conference series are now dedicated to this topic: see the "Fairware" series[1]; the ACM FAT conference FAT [9] ("FAT" is short for fairness, accountability, and transparency); and the IEEE ASE EXPLAIN [10] workshop series. Nevertheless, when discussing this work with colleagues, we are still (sometimes) asked if this problem *can* or *should* be addressed by software engineers. We reply that:

ESEC/FSE '20, November 8–13, 2020, Virtual Event, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7043-1/20/11...$15.00
https://doi.org/10.1145/3368089.3409697

[1] http://fairware.cs.umass.edu

- SE researchers *can* address bias mitigation. As shown below, technology developed within the SE community can be applied to reduce ML bias.
- As to whether or not this community *should* explore ML bias mitigation, that is no longer up to us. When users discover problems with software, it is the job of the person maintaining that software (i.e. a software engineer) to fix that problem.

For all these reasons, this paper explores ML bias mitigation. In the recent software engineering literature, we have found some works to identify bias in machine learning software systems [7, 11]. But there is no prior work done to explain the reason behind the bias and also removing the bias from the software. We see some recent works from ML community to mitigate ML model bias. All of these works trust the ground truth or the original labels of the training data. But any human being or algorithm can make biased decisions and introduce biased labels. For example, white male employees were given higher priority to be selected for company leadership by human evaluators [12]; COMPAS Recidivism algorithm was found biased against black people[5]. If these kind of biased data is used for machine learning model training, then trusting the ground truth could introduce unfair decisions in future. So, training data validation, testing model for bias and bias mitigation are equally important. This paper covers all the concerns. The idea of *Fairway* comes from two research directions:

- Chen et al. mentioned that a model acquires bias from training data[13]. They bolstered on data collection process and training data sampling. Their work motivated us to find bias in the training data rather than model.
- Berk et al. have stated that achieving fairness has a cost [14]. Most of the bias mitigation algorithms damage the performance of the prediction model while making it fair. This is called *accuracy-fairness trade-off*. When trading off competing goals, it is useful to apply multiobjective optimization. While doing so one objective is to reduce bias or achieve fairness and another objective is to keep the performance of the model similar.

Drawing inspiration from both these works, we propose a new algorithm, "Fairway", which is a combination of pre-processing and in-processing methods. Following the motivation of Chen et al, we evaluate the original labels of the training data and identify biased data points which can eventually make the machine learning model biased. Then following the idea of Berk et al, we apply multiobjective optimization approach to keep the model performance same while making it fair. The combination of these two approaches makes Fairway a handy tool for bias detection and mitigation. Overall, this paper makes the following contributions:

- We explain how a machine learning model acquires bias from training data.
- We find out the specific data points in training data which cause the bias. Thus, this work includes finding bias in AI software.
- We are first to combine two bias mitigation approaches - pre-processing (before model training) and in-processing(while model training). This combined method, *Fairway*, performs better than each individual.
- Our results show that we can achieve fairness without much damaging the performance of the model.
- We comment on the shortcomings of broadly used fairness metrics and how to overcome that.
- We describe how concept of ethical bias depends on various applications and how we can use different fairness definitions in different domains.
- Our Fairway replication package is publicly available on GitHub[2] and figshare[15]. This last point is not so much a research contribution but a systems contribution since it enables other researchers to repeat/confirm and perhaps even refute/improve our results.

The rest of this paper is structured as follows- Section 2 provides an overview of software fairness and generates the motivation of this work. Two subsections summarize the previous works. Section 3 explains some fairness terminology and metrics. Section 5 describes the five datasets used in our experiment. Section 6 describes our methodology to make fairer software. Section 7 shows the results for six research questions. In section 8, we have stated the threats to validity of our work. Finally Section 9 concludes the paper.

2 BACKGROUND

2.1 About Software Fairness

There are many instances of a machine learning software being biased and generating arguably unfair decisions. Google's sentiment analyzer model is used to determine positive or negative sentiment. It gives negative score to some sentences like 'I am a Jew', and 'I am homosexual' [16]. Google's photo tagging software mis-categorizes dark-skinned people as animals [17]. Translation engines inject social biases, like, "She is an engineer, He is a nurse" translates into Turkish and back into English becomes "He is an engineer, She is a nurse" [18]. A study was done on YouTube's automatically-generated captions across two genders. It is found that YouTube is more accurate when automatically generating captions for videos with male than female voices [19]. A popular facial-recognition software shows error rate of 34.7% for dark-skinned women and 0.8% for light-skinned men [20]. Recidivism assessment models that are used by the criminal justice system have been found to be more likely to falsely label black defendants as future criminals at almost twice the rate as white defendants [21]. Amazon scraped automated recruiting tool that showed bias against women [22].

In 2018, Brun et al. first commented that it is now time that software engineers should take these kinds of discrimination as a major concern and put effort to develop fair software [8]. A software is called fair if it does not provide any undue advantage to any specific group (based on race, sex) or any individual. This paper represents a method **Fairway** which specifically tries to detect and mitigate ethical bias in a binary classification model used in many AI software.

2.2 Previous Work

Bias in Machine Learning models is a well-known topic in ML community. Recently, SE community is also showing interest in this area. Large SE industries have started putting more and more importance

[2] https://github.com/joymallyac/Fairway

on ethical issues of ML model and software. IEEE [23], the European Union [24] and Microsoft [25] recently published the ethical principles of AI. In all three of them, it is stated that an intelligent system or machine learning software must be fair when it is used in real-life applications. IBM has launched a software toolkit called AI Fairness 360 [26] which is an extensible open-source library containing techniques developed by the research community to help, detect and mitigate bias in machine learning models throughout the AI application lifecycle. Microsoft has created a research group called FATE [27] which stands for Fairness, Accountability, Transparency, and Ethics in AI. Facebook announced they developed a tool called Fairness Flow [28] that can determine whether a ML algorithm is biased or not. ASE 2019 has organized first International Workshop on Explainable Software [10] where issues of ethical AI were extensively discussed. German et al. have studied different notions of fairness in the context of code reviews[29]. In summary, the importance of fairness in software is rising rapidly. So far, the researchers have concentrated on two specific aspects -

- Testing AI software model to **find ethical bias**
- Making the model prediction fair by **removing bias**

2.3 Finding Ethical Bias

Angell et al. [7] commented that software fairness is part of software quality. An unfair software is considered as poor quality software. Tramer and other researchers proposed several ways to measure discrimination [30]. Galhotra et al. created THEMIS [31], a testing-based tool for measuring how much a software discriminates, focusing on causality in discriminatory behavior. THEMIS selects random values from the domain for all the attributes to determine if the system discriminates amongst the individuals. Udeshi et al. have developed AEQUITAS [32] tool that automatically discovers discriminatory inputs which highlight fairness violation. It generates test cases in two phases. The first phase is to generate test cases by performing random sampling on the input space. The second phase starts by taking every discriminatory input generated in the first phase as input and perturbing it to generate furthermore test cases. Both techniques THEMIS and AEQUITAS aim to generate more discriminatory inputs. The researchers from IBM Research AI India have proposed a new testing method for black-box models [11]. They combined dynamic symbolic execution and local explanation to generate test cases for non-interpretable models.

These all are test case generation algorithms that try to find bias in a trained model. We did not use these methods because along with the model, we also wanted to find bias in the training data. We developed our own testing method based on the concept of situation testing[33].

2.4 Removing Ethical Bias

The prior works in this domain can be classified into three groups depending on the approach applied to remove ethical bias.

- **Pre-processing algorithms**: In this approach, before classification, data is pre-processed in such a way that discrimination or bias is reduced. Kamiran et al. proposed *Reweighing* [34] method that generates weights for the training examples in each (group, label) combination differently to achieve fairness. Calmon et al. proposed an *Optimized pre-processing* method [35] which learns a probabilistic transformation that edits the labels and features with individual distortion and group fairness.
- **In-processing algorithms**: This is an optimization approach where the dataset is divided into three sets - train, validation and test set. After learning from training data, the model is optimized on the validation set and finally applied on the test set. Zhang et al. proposed *Adversarial debiasing* [36] method which learns a classifier to increase accuracy and simultaneously reduce an adversary's ability to determine the protected attribute from the predictions. This leads to generation of fair classifier because the predictions cannot carry any group discrimination information that the adversary can exploit. Kamishima et al. developed *Prejudice Remover* technique [37] which adds a discrimination-aware regularization term to the learning objective of the classifier.
- **Post-processing algorithms**: This approach is to change the class labels to reduce discrimination after classification. Kamiran et al. proposed *Reject option classification* approach [38] which gives favorable outcomes to unprivileged groups and unfavorable outcomes to privileged groups within a confidence band around the decision boundary with the highest uncertainty. *Equalized odds post-processing* is a technique which particularly concentrate on the Equal Opportunity Difference(EOD) metric. Two most cited works in this domain are done by Pleiss et al. [39] and Hardt et al [40].

Fairway combines both *Pre-processing* and *In-processing* approach. Further, post-processing is not needed after using Fairway. Changing a misclassified label requires domain knowledge based on the type of application. That kind of knowledge can be difficult to collect (since it requires access to subject matter experts). Hence, post-processing is not explored in this paper.

3 FAIRNESS TERMINOLOGY

In this section some specified terminology from the field of fairness in machine learning are described. This paper is limited to the binary classification models and tabular data(row-column format). Each dataset used has some attribute columns and a class label column. A class label is called *favorable label* if its value corresponds to an outcome that gives an advantage to the receiver. Examples include - being hired for a job, receiving a loan. *Protected attribute* is an attribute that divides a population into two groups (privileged & unprivileged) that have difference in terms of benefits received. An example of such attribute could be "sex" or "race". These attributes are not universal but are specific to the application. *Group fairness* is the goal that based on the protected attribute, privileged and unprivileged groups will be treated similarly. *Individual fairness* is the goal of similar individuals will receive similar outcomes.

4 FAIRNESS MEASURES

Martin argues (and we agree) that "bias is a systematic error" [41]. Our main concern is unwanted bias that puts privileged groups at a systematic advantage and unprivileged groups at a systematic disadvantage. A *fairness metric* is a quantification of unwanted bias in models or training data [42]. We used two such fairness metrics in our experiment-

- **Equal Opportunity Difference(EOD)**: Difference of True Positive Rates(TPR) for unprivileged and privileged groups [42].

Table 1: Combined Confusion Matrix for Privileged(P) and Unprivileged(U) Groups.

	Predicted No	Predicted Yes	Predicted No	Predicted Yes
	Privileged		Unprivileged	
Actual No	TN_P	FP_P	TN_U	FP_U
Actual Yes	FN_P	TP_P	FN_U	TP_U

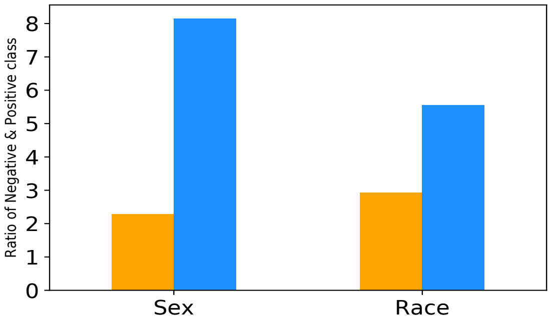

Figure 1: Ratio of negative and positive class for two protected attributes - sex and race for "Adult" dataset. "Orange" column is for Privileged group(Male,White) and "Blue" column is for unprivileged group(Female,Non-white).

- **Average Odds Difference(AOD)**: Average of difference in False Positive Rates(FPR) and True Positive Rates(TPR) for unprivileged and privileged groups [42].

$$TPR = TP/P = TP/(TP + FN) \tag{1}$$

$$FPR = FP/N = FP/(FP + TN) \tag{2}$$

$$EOD = TPR_U - TPR_P \tag{3}$$

$$AOD = [(FPR_U - FPR_P) + (TPR_U - TPR_P)] * 0.5 \tag{4}$$

EOD and AOD are computed using the input and output datasets to a classifier. A value of 0 implies that both groups have equal benefit, a value lesser than 0 implies higher benefit for the privileged group and a value greater than 0 implies higher benefit for the unprivileged group. In this study, absolute value of these metrics have been considered.

Depending upon the notion of fairness, there are various fairness metrics also. The statistical notion of fairness in binary classification mainly comes from the confusion matrix - a table that is often used to describe the accuracy of a classification model. If there are two confusion matrices for two groups - privileged and unprivileged (see Table 1), all the fairness metrics try to find the difference of True Positive Rate and False Positive Rate for those two groups from those two matrices [40, 42–46]. Beutel et al. commented that all of these fairness metrics suffer from three shortcomings [47]-

- These metrics ignore the class distribution for privileged and unprivileged groups. As a case study, Figure 1 shows the ratio of negative(low income) and positive(high income) class for two protected attributes - sex and race for "Adult" dataset. "Orange" column is for Privileged group(sex- male, race - white) and "Blue" column is for unprivileged group(sex- female, race - non-white). The figure shows the uneven distribution of positive and negative classes for unprivileged and privileged groups.
- These metrics do not consider the sampling of the data. But incorrect sampling creates data imbalance which may lead to incorrect measurement of bias.
- These metrics ignore the cost of misclassification. For example, in case of credit card approval software, assigning bad credit score to an applicant who has actual good credit score is less costlier than assigning good credit score to an applicant who has actual bad credit score.

In this work, several steps are taken to overcome those shortcomings. Most of the prior works have either used AOD or EOD, we have used both of them for our study as we compared our approach with previous works [42]. Instead of depending on only those two metrics, the concept of *situation testing* was used to find discrimination [31]. In the context of binary classification, *situation testing* is the process of verifying whether model prediction changes for same data point with changed protected attribute value [33]. While measuring the performance of Fairway, we used random sampling of data for ten times to overcome the sampling problem. Cost of misclassification is not solved because that is application specific and requires domain knowledge.

5 DATASET DESCRIPTION

In this experiment, five datasets from UC Irvine Machine Learning Repository have been used. All the datasets are quite popular in fairness domain and used by previous SE researchers[31, 32, 48]. A brief description of the datasets are given -

- Adult Census Income - This dataset contains records of 48,842 people. The class label is yearly income [49]. It is a binary classification dataset where the prediction task is to determine whether a person makes over 50K a year. There are fourteen attributes among them two are protected attributes.
- Compas - This is a dataset containing criminal history, demographics, jail and prison time, and COMPAS (which stands for Correctional Offender Management Profiling for Alternative Sanctions) risk scores for defendants from Broward County [50]. The dataset contains 7,214 rows and twenty-eight attributes. Among them there are two protected attributes.
- German Credit Data - This dataset contains records of 1,000 people and binary class labels (Good Credit or Bad Credit) [51]. There are twenty attributes among them one is protected.
- Default Credit - There are 30,000 records of default payments of people from Taiwan [52]. Binary class label is Default Payment "Yes" or "No". There are twenty-three attributes among them one is protected.
- Heart Health - The Heart Dataset from the UCI ML Repository contains fourteen features from 297 adults [53]. The goal is to accurately predict whether or not an individual has a heart condition.

Table 2: Description of the datasets used for the experiment.

Dataset	#Rows	#Features	Protected Attribute		Label	
			Privileged	Unprivileged	Favorable	Unfavorable
Adult Census Income	48,842	14	Sex-Male Race-White	Sex-Female Race-Non-white	High Income	Low Income
Compas	7,214	28	Sex-Female Race-Caucasian	Sex-Male Race-Not Caucasian	Did not reoffend	Reoffended
German Credit Data	1,000	20	Sex-Male	Sex-Female	Good Credit	Bad Credit
Default Credit	30,000	23	Sex-Male	Sex-Female	Default Payment - Yes	Default Payment - No
Heart Health	297	14	Age-Young	Age-Old	Not Disease	Disease

Table 2 gives an overall description of all five datasets. These are binary classification datasets. Like most of the prior research[35, 37, 40], we used *Logistic Regression* model on these datasets . But our approach is applicable for any classification model.

6 THE "FAIRWAY" METHOD

As stated above, the Fairway algorithm is a combination of the preprocessing and in-processing approach to make machine learning software fairer.

6.1 Why not Remove the Protected Attributes?

This section describes one of the methods we explored, before arriving at Fairway.

When we think of prediction model discriminating over a protected attribute, the first solution which comes to mind is that why not train the model without that protected attribute. Being novice in fairness domain, we tried that for the five datasets. Two of the datasets have two protected attributes (Adult, Compas - Sex, Race) and other three datasets have only one protected attribute. We removed the protected attribute column from the train and test data so that the model has no information about that attribute. Surprisingly, there was almost no change in bias metrics even after that.

Brun et al. have mentioned one reason behind this surprising result. They mentioned that if there is high correlation between attributes of the dataset, then even after removing the protected attribute, the bias stays [54]. In 2016, Amazon created a model for same-day delivery service offered to Prime users around the major US cities[55]. But the model turned out to be highly discriminatory against black neighborhood. While training this model, "Race" attribute was not used but the model became biased against a certain "Race" because the "Zipcode" attribute highly correlates with "Race". The training data had "Zipcode" and the model induced "Race" from that. Initially, we also thought maybe correlation is the reason for our datasets also. But when we checked for the correlation between attributes, we found that bias is not coming from the correlation.

For the datasets we are using here, the bias mainly comes from the class label. The data have been historically captured over the years. The classification was done by several human beings or algorithms - whether credit card gets approved or a person having a disease. Human bias or Algorithmic bias against certain sex or race reflected on predictions. In some cases, people of specific race or sex were unfairly treated. Thus the historical records have improper labels for some portion of data.

This is to say that even if we remove the "protected attribute" column, bias still remains. For removal of bias, we need to find out those data points having improper labels.

Finally, we can summarize different ways of a model acquiring bias from training data -

- If in the training data, the class labels are related to any of the protected attributes, while training, a model can acquire that bias. If there is no protected attribute but other correlated attributes which affect the decision, then also model may become biased.
- Kamishima et al. reported a reason for unfairness called "Underestimation" [37]. It happens when a trained model is not fully converged due to the finiteness of the size of the training data set. They defined a new metric called the underestimation index (UEI) based on the Hellinger distance to find "Underestimation". According to them, this occurs very rarely. So, we did not try to find UEI for our datasets.
- Bias may come from unfair sampling of training data or unfair labeling of the training data. For the five datasets used in this study, the main reason of bias is unfair labeling of some data points. In this work, data has been randomly sampled ten times to make sure bias does not come from improper sampling.

6.2 Removal of Ambiguous (Biased) Data Points

Depending upon the protected attribute, there is a privileged group and an unprivileged group in each dataset. Which group is privileged and which group is unprivileged depend on the application. For example:

- In credit card applications, "Male" might be considered privileged and "Female" as unprivileged;
- In criminal prediction, "White" people might be considered privileged and "non-white" as unprivileged.

In this step, we try to find and remove the data points which are responsible for creating the bias based on the protected attribute. We call these data points the **ambiguous** data points.

Fig. 2 describes the approach we applied to find out the ambiguous data points depending on the protected attribute. We divide

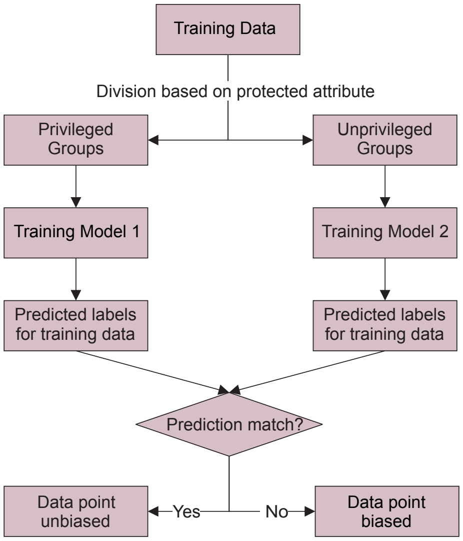

Figure 2: Pre-processing technique for bias removal from training data

the training data into two groups based on the protected attribute - privileged and unprivileged. Then we train two separate models on those two groups. Once we get the two trained models, for all the training data points, we check the prediction of these two models:

- If the prediction matches in both cases, the data point being examined is unbiased.
- If two models contradict each other for a data point, there is a possibility of this data point being biased, this is an ambiguous data point. We remove that data point from training data. Later we will describe why this works and how to validate.

We call this data cleaning process as "Bias Removal" from training data. Once we are done removing the probable biased data points, we train a new model on the rest of the training data and make prediction using that model. Table 3 shows the total number of rows in each dataset and the number of rows we removed. We see that at most we lose 15% of training data after bias removal step. Later we will show that this does not affect much the performance of the prediction model.

We remove the ambiguous(bias causing) data points by constructing two separate logistic regression models conditioned upon the protected attribute of the dataset. Let's assume the original data points are denoted as X where $x_1, x_2, x_3, ..., x_n$ are the attributes of the dataset and the protected attribute is denoted as s (s = x_k, where k is a number between 1 to n) and $\hat{y}$ is the model prediction. The original dataset is further divided into subsets based on the values of a protected attribute, in this case, $X_1 \subset X \forall s = 1$ and $X_2 \subset X \forall s = 0$. We use these two subsets to build two logistic regression models such as -

$$p(\hat{y} = 1|s = 1) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_{n-1} x_{n-1} \tag{5}$$

$$p(\hat{y} = 1|s = 0) = \beta'_0 + \beta'_1 x_1 + \beta'_2 x_2 + ... + \beta'_{n-1} x_{n-1} \tag{6}$$

$$f_1(x) = \log_e \frac{p(\hat{y} = 1|s = 1)}{p(\hat{y} = 0|s = 1)} \tag{7}$$

$$f_2(x) = \log_e \frac{p(\hat{y} = 1|s = 0)}{p(\hat{y} = 0|s = 0)} \tag{8}$$

Next, we use these logistic regression models to check for each training data point, by retaining the data points where

$$\forall x \in X \; (f_1(x_1) == f_2(x_1))$$

This results in retaining only the data points where there is no contradiction about the models' outcome irrespective of data distribution conditioned upon the protected attribute, thus removing the data points which add ambiguity to the model and introduce bias into the model's prediction.

Table 3: #Rows = Total number of Rows, #Dropped Rows = Total number of rows detected as ambiguous(biased)

Dataset	Protected Attribute	#Rows	#Dropped Rows	% of Rows Dropped
ADULT	Sex Race	48,842	6,178 2,315	12.6 4.7
COMPAS	Sex Race	7,214	1,128 724	15.6 10.0
DEFAULT CREDIT	Sex	30,000	505	1.7
HEART HELTH	Age	297	32	10.8
GERMAN	Sex	1,000	38	3.8

In the five datasets we used, due to the pre-processing step we do not lose much of training (see Table 3). But in case of other datasets or real-world scenarios, if too many data points are found biased and model prediction gets damaged due to this loss, then we would suggest relabeling of data points instead of removal. In such relabeling, any majority voting technique like k-NN can be used. Biased data points will be assigned a new class label depending on k nearest neighbor data points. Such relabeling comes with an extra cost (finding distance for all the data points), so we recommend it to use only if model prediction is affected due to the removal of biased data points. This study does not include that experiment, but this could be an interesting direction for future work.

6.3 What if there are two protected attributes?

Fig. 2 shows the approach we applied for one protected attribute. But in some cases, there are more than one protected attribute in a dataset. Like - Adult and Compas datasets (Sex and Race). If we have two protected attributes, we divide the training data based on those two attributes into four groups (two privileged and two unprivileged groups). Then we apply the similar logic to find the biased data points. We train four different models on those four groups and check their predictions match or not. These models are not used for prediction, they are used to find biased data points only. In the two datasets, we did not lose more than 16% of training data with this approach.

As to handling more than two protected attributes, we do not explore it here, for the following reason. With our data sets, such ternary (or more) protection divides the data into unmanageable small regions. Future research in this area would require case studies with much larger data sets.

6.4 Model Optimization

IBM has created a GitHub repo to combine some promising prior works on fairness domain[26]. The results show that most of the prior methods damage the performance of the model while making it fair. So, prediction performance and fairness are competitive goals[14]. When there is a trade-off between competing performance goals, multi-objective optimization is the way to explore the goal space. In our case, the goal of such optimizer would be to make the model as fair as possible while also not degrading other performance measures such as recall or false alarm.

To explore such multiobjective optimization, we divided the dataset into three groups - Training (70%), Validation (15%) and Test (15%)[56]. During the pre-processing step, we removed biased data points from the training set. After that Logistic Regression model is trained on the training set with the standard default parameters[3]. Then we used the FAIR_FLASH algorithm (discussed below) to find out the best set of parameters to achieve optimal value of four metrics (Higher Recall, Lower False Alarm, Lower AOD, and Lower EOD) on the validation set. Finally, the tuned model is applied on the test set.

Nair et al. proposed FLASH [57], a novel optimizer, that utilizes sequential model-based optimization(SMBO). The concept of SMBO is very simple. It starts with "What we already know about the problem" and then decides "what should we do next". The first part is done by a machine learning model and the second part is done by an acquisition function. Initially, a few points are randomly selected and measured. These points along with their performance measurements are used to build a model. Then the model is used to predict the performance measurements of other unevaluated points. This process continues until a stopping criterion is reached. FLASH improves over traditional SMBO as follows:

- FLASH models each objective as a separate Classification and Regression Tree (CART) model. Nair et al. report that the CART algorithm can scale much better than other model constructors (e.g. Gaussian Process Models).
- FLASH replaces the actual evaluation of all combinations of parameters(which can be a very slow process) with a *surrogate evaluation*, where the CART decision trees are used to guess the objective scores (which is a very fast process). Such guesses may be inaccurate but, as shown by Nair et al., such guesses can rank guesses in (approximately) the same order as that generated by other, much slower, methods [58].

FLASH was invented to solve software configuration problem and it performed faster than more traditional optimizers such as Differential Evolution[59] or NSGA-II[60]. For our work, we modified FLASH and generated FAIR_FLASH that seeks best parameters for *Logistic regression* model with four goals - higher recall, lower false alarm, lower AOD, lower EOD. Algorithm 1 shows the pseudocode of FAIR_FLASH. It has two layers - one learning layer and one optimization layer. When training data arrives, the estimator in the learning layer is being trained, and the optimizer in optimizing layer provides better parameters to the learner to help improve the performance of estimators. Such trained learner is evaluated on the validation data afterward. Once some stopping criteria is met, the generated learner is then passed to the test data for final testing.

Algorithm 1 Pseudocode of FAIR_FLASH inspired from [57]

```
def FAIR_FLASH():
    # pick a number of data into build_pool, evaluate the build_pool,
    # and put the rest into rest_pool
    while life > 0:
        # build CART model by using build_pool
        next_point = max(model.predict(rest_pool))
        build_pool += next_point
        rest_pool −= next_point
        if model.evaluate(next_point) < max (build_pool):
            life −= 1
    return max(build_pool)
```

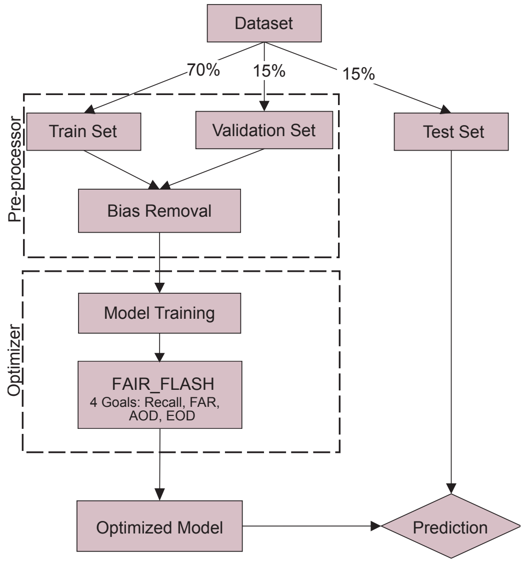

Figure 3: Block diagram of Fairway. For details on FAIR_FLASH, see Algorithm 1.

In summary, Fairway consists of two parts - bias removal from training data and model optimization to make trained model fair. Fig. 3 shows an overview of the method.

7 RESULTS

Our results are structured around six research questions. For all the results, we repeated our experiments ten times with data shuffling and we report the median.

RQ1. What is the problem with just using standard learners?

The premise of the paper is our methods offer some improvement over common practices. To justify that we first need to show that there are open issues with standard methods. We trained a logistic regression model with default scikit-learn parameters and tested on

[3] In Scikit-Learn, those details are C=1.0, penalty='l2', solver='liblinear', max_iter=100.

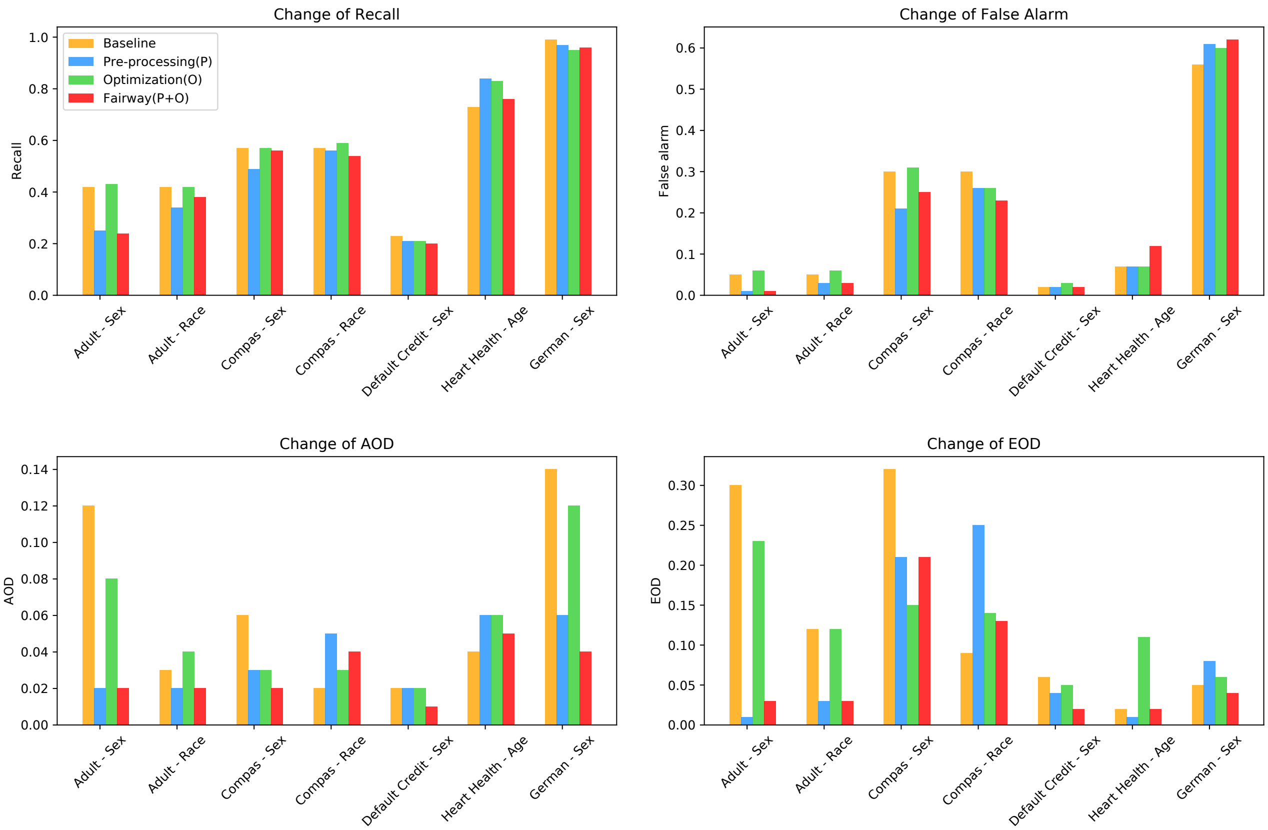

Figure 4: Performance and fairness metrics for (a) default state in Orange; (b) after pre-processing in Blue; (c) after just optimization in Green; and (d) after performing pre-processing + optimization in Red. In these charts, *higher* recalls are *better* while for all other scores, *lower values* are *better.*

the five datasets. The "Orange" column in Fig. 4 shows the results achieved using that model. The recall is higher the better, and false alarm, AOD, EOD are lower the better. Recall and False alarm are showing the prediction performance of the model. The high value of fairness metrics(AOD, EOD) in all five datasets signifies that model prediction is not fair means depending upon protected attribute, privileged group is getting advantage over unprivileged group. We treat this results as baseline for our experiment. We need to make the prediction fair without much damaging the performance.

RQ2. How well does Pre-processing improve the results?

Fairway is a two-part procedure- data pre-processing (ambiguity removal) and learner optimization(FAIR_FLASH). It is reasonable to verify the contribution of both parts. Accordingly **RQ2** tests the effects of just doing ambiguity removal.

Before training Logistic regression model, training data was cleaned to remove ambiguous data points(having improper labels) using the approach mentioned in section 6.2. Table 3 shows this step causes loss of maximum 15% of the training data. After that logistic regression model was trained on remaining data points and tested. The "Blue" column in Fig. 4 shows the results achieved using that model. We see minor damage in recall for some cases and significant improvement in case of fairness metrics (lower AOD, EOD). It is evident that pre-processing the data before model training makes the model prediction fairer.

RQ3. How well does Optimization improve the results?

Moving on from **RQ2**, the third research question is to check the effect of just optimization(no pre-processing).

To do that, we tuned the Logistic regression model parameters using FAIR_FLASH to optimize the model for higher recall, lower false alarm and lower fairness metrics(AOD, EOD). Then the tuned model was used for prediction. The "Green" column in Fig. 4 shows the results achieved using that model. We see that in cases of prediction performance(recall, false alarm) it performs similar or better than pre-processing but in case of fairness metrics(AOD, EOD), preprocessing does better. So, optimized learner is significantly better than baseline learner but combining pre-processing may perform even better.

Table 4: Comparison of Fairway with prior algorithms. Recall is higher the better. False alarm, AOD and EOD are lower the better. "Gray" cells show improvement and "Black" cells show damage. "White" cells show no change.

Algorithm	Dataset	Protected Attribute	Recall		False alarm		AOD		EOD	
			Before	After	Before	After	Before	After	Before	After
Optimized Preprocessing[35]	Adult	Sex	0.42	0.38	0.05	0.07	0.12	0.03	0.30	0.04
		Race	0.42	0.42	0.05	0.08	0.03	0.03	0.12	0.09
	Compas	Sex	0.57	0.56	0.30	0.32	0.06	0.03	0.32	0.07
		Race	0.57	0.59	0.30	0.32	0.02	0.01	0.09	0.03
	German	Sex	0.99	0.97	0.56	0.61	0.14	0.12	0.04	0.03
Reweighing(Pre-processing)[34]	Adult	Sex	0.42	0.41	0.05	0.07	0.12	0.02	0.30	0.03
		Race	0.42	0.40	0.05	0.08	0.03	0.01	0.12	0.02
	Compas	Sex	0.57	0.55	0.30	0.34	0.06	0.03	0.32	0.12
		Race	0.57	0.57	0.30	0.37	0.02	0.01	0.09	0.03
	German	Sex	0.99	0.94	0.56	0.60	0.14	0.10	0.04	0.03
Adversial Debiasing[36] (In-processing)	Adult	Sex	0.42	0.41	0.05	0.04	0.12	0.01	0.30	0.02
		Race	0.42	0.42	0.05	0.07	0.03	0.01	0.12	0.02
	Compas	Sex	0.57	0.53	0.30	0.35	0.06	0.04	0.32	0.06
		Race	0.57	0.52	0.30	0.36	0.02	0.02	0.09	0.06
	German	Sex	0.99	0.94	0.56	0.60	0.14	0.12	0.04	0.04
Reject Option Classification[38] (Post-processing)	Adult	Sex	0.42	0.21	0.05	0.02	0.12	0.03	0.30	0.04
		Race	0.42	0.23	0.05	0.02	0.03	0.02	0.12	0.10
	Compas	Sex	0.57	0.62	0.30	0.34	0.06	0.01	0.32	0.03
		Race	0.57	0.61	0.30	0.34	0.02	0.02	0.09	0.07
	German	Sex	0.99	0.94	0.56	0.61	0.14	0.04	0.04	0.01
Fairway (Pre-processing + In-processing)	Adult	Sex	0.42	0.24	0.05	0.01	0.12	0.02	0.30	0.03
		Race	0.42	0.38	0.05	0.03	0.03	0.02	0.12	0.03
	Compas	Sex	0.57	0.57	0.30	0.25	0.06	0.02	0.32	0.21
		Race	0.57	0.54	0.30	0.23	0.02	0.04	0.09	0.13
	German	Sex	0.99	0.96	0.56	0.62	0.14	0.04	0.04	0.04

RQ4. How well does Fairway improve the results?

Our fourth research question explores the effect of Fairway which is a combination of pre-processing and optimization.

The "Red" column in Fig. 4 shows the results achieved after applying Fairway. Fairway is performing better than pre-processing and optimization in most of the cases. For example:

- In case of Adult dataset, for the protected attribute race, Fairway achieves almost similar recall with optimization but much better in the other three metrics.
- In case of Default Credit dataset, for the protected attribute sex, Fairway is providing best results for all four metrics.

In some cases, recall is slightly damaged. But overall, Fairway is making the model fair without much affecting the performance. So, pre-processing the data before model training and tuning the model while training both are important.

RQ5. How well does Fairway perform compared to previous fairness algorithms?

We have decided to compare our approach Fairway with some popular previous algorithms described in section 2.4. We chose five such algorithms (all from IBM AIF360) which we thought could be representative of the works done before. Table 4 shows the results for three datasets - Adult, Compas, and German. It shows the change of recall, false alarm and two fairness metrics AOD, EOD before and after the algorithms are applied. In most of the cases, Fairway is performing better or the same with prior algorithms in case of reducing ethical bias (AOD,EOD). In case of false alarm, Fairway has less number of black cells showing damage. Like Fairway, previous algorithms also slightly damage the recall metric. In some situations, this may become a matter of concern. We see a scope of improvement here where future researchers should focus.

We have performed scott-knott significance test and A12 effect size test for comparison. For AOD, Fairway performs better in 2/5 cases and for EOD, in 3/5 cases. Here better means result is statistically significantly better. For the rest of the cases, although having the same rank, improvement is between 10%-25%. Also, Fairway wins on false alarm for all cases and keeps the same recall in 3/5 cases and damages in 2/5 . And when Fairway loses in recall, it does not lose by much (10%-12%).

Fairway is not just another bias mitigation approach. It differs from prior works in several ways -

- The first part of Fairway is finding bias in training data. So, even before model training, Fairway shows which data points in the training data have improper/biased labels and can affect prediction in future. If labeling was done by human reviewers, it leads to finding bias in human decisions. Instead of blindly trusting the ground truth of training data, Fairway can be used to find bias in the ground truth.
- Prior bias mitigation algorithms come from the core concepts of machine learning. Software practitioners having little ML knowledge may face difficulties to use these algorithms[61]. In case of Fairway, users can clearly see how two different models trained on privileged and unprivileged groups give different predictions on biased data points. This makes Fairway much comprehensible. FAIR_FLASH gives user the flexibility to choose which parameters to optimize. In this paper, Logistic regression

model is used. But FAIR_FLASH is easily extensible for other classification models. So, FAIR_FLASH is adjustable too.

- Fairway is a combination of bias testing and mitigation. This is described in **RQ6**.

RQ6. Can Fairway be used as a combined tool for detection and mitigation of bias?

In section 2.2, it is shown that there are mainly two types of previous works done by researchers - finding the bias in AI software and mitigating the bias. As per our knowledge, we are the first one to combine these two. Fairway finds the data points which have unfair labeling in the training data and remove those data points so that prediction is not affected by protected attribute. We used *Situation testing* [33] to verify whether after bias removal, the role of a protected attribute on the prediction changes or not. we switched the protected attribute value for all the remaining data points (e.g. we changed Male to Female and Female to Male). Then we checked whether these changes lead to prediction changes or not. If the prediction changes for a data point, we say that it fails situation testing. Figure 5 shows the percentage of data points failing situation testing before and after pre-processing step of Fairway:

- The "orange" and "blue" columns show results before/after applying Fairway.
- In all cases, the values on the blue column are far smaller than orange column.

So, Fairway can find the data points responsible for bias in the training data. Now, it is an engineering decision to set the threshold of what percentage of training data can be ambiguous where prediction may change depending on the protected attribute value. Fairway provides the percentage and depending on the application, user can decide whether bias is present in the system or not. So, Fairway can be applied as a discrimination finder tool. If discrimination is above the tolerable threshold, then Fairway can be applied for removing bias from training data and optimizing model without damaging predictive performance. So, Fairway can be used as a combined tool for detection and mitigation of discrimination or ethical bias. One unique feature of Fairway is it is **model-agnostic**. It finds bias by verifying prediction of a model and mitigates bias by cleaning training data and tuning model parameters. So, it can work for any black box model. As Fairway only works on the output space of a model, it can be easily used in industrial purposes where revealing core algorithm of the underlying model is not possible.

So, to summarize the results, we say that we have explained the reasons of bias in the five datasets we used. We have developed a comprehensible method *Fairway* which can remove bias from training data and the model. Unlike prior works, *Fairway* is not just a bias mitigation approach, it is a combined tool for ground truth validation, bias detection and mitigation.

8 THREATS TO VALIDITY

- **Sampling Bias** - We have used five datasets from UCI machine learning repository where most of prior works in fairness domain use only one or two datasets. These are well-known datasets and used by previous researchers in ML and software fairness domain. It is an open issue if these data sets reflect an interesting range of fairness issues for other data sets. In future work, we would explore more data sets.

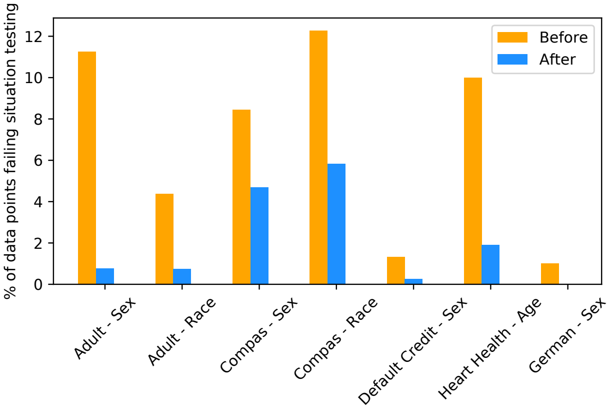

Figure 5: Percentage change of data points failing situation testing (showing bias) before and after pre-processing.

- **Evaluation Bias** - We have used two fairness metrics - EOD and AOD. We have mentioned the drawbacks of fairness metrics which only consider the TPR and FPR and neglect the class distribution. Recent work has deduced a new fairness metric called *Conditional Equality of Opportunity* to overcome this drawback [47]. Conditional Equality of Opportunity is defined for conditioning on every feature and finding the opportunity gap for privileged and unprivileged groups. In future work, we would explore more performance criteria.
- **Construct Validity** - In our work we trained different models on privileged and unprivileged groups. The datasets contained one or two protected attributes, so our method is feasible. All the prior works we have seen treated each protected attribute individually. We have shown how to deal with two protected attributes. In future work, we would explore larger data sets with more protected attributes.
- **External Validity** - Fairway is limited to classification models which are very common in AI software. We are currently working on extending it to Regression models. In future work, we would extend this work to other kinds of data mining problems; e.g. to text mining or video processing systems.

9 CONCLUSION

We have explained how a model acquires bias from improper labels of training data and have demonstrated an approach called "Fairway" which removes "ethical bias" from the training data and optimizes a trained model for fairness and performance. We have shown that Fairway is comprehensible and can be used as a combined tool for detection and mitigation of bias. Unlike some prior ML works, Fairway is not just a bias mitigation tool, it validates ground truth labels, finds bias and mitigates bias. We have made the source code of "Fairway" publicly available for software researchers and practitioners. To the best of our knowledge, we claim this is the first work in SE domain which concentrates on mitigating ethical bias from software and making software fair using optimization methods augmented with some data pre-processing. In future, we hope more and more software researchers will work on this domain and industries will consider publishing more datasets. When that data becomes available, it would be appropriate to rerun this study. [62]

REFERENCES

[1] “Health care start-up says a.i. can diagnose patients better than humans can, doctors call that ‘dubious’,” *CNBC*, June 2018. [Online]. Available: https://www.cnbc.com/2018/06/28/babylon-claims-its-ai-can-diagnose-patients-better-than-doctors.html

[2] E. Strickland, “Doc bot preps for the o.r.” *IEEE Spectrum*, vol. 53, no. 6, pp. 32–60, June 2016.

[3] “On orbitz, mac users steered to pricier hotels,” 2012. [Online]. Available: https://www.wsj.com/articles/SB10001424052702304458604577488822667325882

[4] “The algorithm that beats your bank manager,” 2011. [Online]. Available: https://www.forbes.com/sites/parmyolson/2011/03/15/the-algorithm-that-beats-your-bank-manager/#15da2651ae99

[5] “Machine bias: There’s software used across the country to predict future criminals. and it’s biased against blacks,” 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[6] “Can you program ethics into a self-driving car?” 2016. [Online]. Available: https://spectrum.ieee.org/transportation/self-driving/can-you-program-ethics-into-a-selfdriving-car

[7] R. Angell, B. Johnson, Y. Brun, and A. Meliou, “Themis: Automatically testing software for discrimination,” ser. ESEC/FSE 18.

[8] Y. Brun and A. Meliou, “Software fairness,” in *ESEC/FSE 2018*, 2018.

[9] “Acm conference on fairness, accountability, and transparency (acm fat*).” [Online]. Available: https://fatconference.org/

[10] “Explain 2019.” [Online]. Available: https://2019.ase-conferences.org/home/explain-2019

[11] A. Aggarwal, P. Lohia, S. Nagar, K. Dey, and D. Saha, “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*, ser. ESEC/FSE 2019. New York, NY, USA: ACM, 2019, pp. 625–635. [Online]. Available: http://doi.acm.org/10.1145/3338906.3338937

[12] “White men account for 72% of corporate leadership at 16 of the fortune 500 companies,” 2017. [Online]. Available: https://fortune.com/2017/06/09/white-men-senior-executives-fortune-500-companies-diversity-data/

[13] I. Chen, F. D. Johansson, and D. Sontag, “Why is my classifier discriminatory?” 2018.

[14] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth, “A convex framework for fair regression,” 2017.

[15] J. Chakraborty, “Fairway,” 6 2020. [Online]. Available: https://figshare.com/articles/software/Fairway/12521408

[16] “Google’s sentiment analyzer thinks being gay is bad,” *Motherboard*, Oct 2017. [Online]. Available: https://bit.ly/2yMax8V

[17] “Google apologizes for mis-tagging photos of african americans,” July 2015. [Online]. Available: https://cbsn.ws/2LBYbdy

[18] A. Caliskan, J. J. Bryson, and A. Narayanan, “Semantics derived automatically from language corpora contain human-like biases,” *Science*, vol. 356, no. 6334, pp. 183–186, 2017. [Online]. Available: https://science.sciencemag.org/content/356/6334/183

[19] R. Tatman, “Gender and dialect bias in YouTube’s automatic captions,” in *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 53–59. [Online]. Available: https://www.aclweb.org/anthology/W17-1606

[20] “Study finds gender and skin-type bias in commercial artificial-intelligence systems,” 2018. [Online]. Available: http://news.mit.edu/2018/study-finds-gender-skin-type-bias-artificial-intelligence-systems-0212

[21] “Machine bias,” *www.propublica.org*, May 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[22] “Amazon scraps secret ai recruiting tool that showed bias against women,” Oct 2018. [Online]. Available: https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G

[23] “Ethically-aligned design: A vision for prioritizing human well-begin with autonomous and intelligence systems.” 2019.

[24] “Ethics guidelines for trustworthy artificial intelligence.” 2018. [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

[25] “Microsoft ai principles. 2019.” 2019. [Online]. Available: https://www.microsoft.com/en-us/ai/our-approach-to-ai

[26] “Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias,” 10 2018. [Online]. Available: https://github.com/IBM/AIF360

[27] “Fate: Fairness, accountability, transparency, and ethics in ai,” 2018. [Online]. Available: https://www.microsoft.com/en-us/research/group/fate/

[28] “Facebook says it has a tool to detect bias in its artificial intelligence,” 2018. [Online]. Available: https://qz.com/1268520/facebook-says-it-has-a-tool-to-detect-bias-in-its-artificial-intelligence/

[29] D. M. German, G. Robles, G. Poo-Caamaño, X. Yang, H. Iida, and K. Inoue, ““was my contribution fairly reviewed?”: A framework to study the perception of fairness in modern code reviews,” in *Proceedings of the 40th International Conference on Software Engineering*, ser. ICSE ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 523–534. [Online]. Available: https://doi.org/10.1145/3180155.3180217

[30] F. Tramer, V. Atlidakis, R. Geambasu, D. Hsu, J.-P. Hubaux, M. Humbert, A. Juels, and H. Lin, “Fairtest: Discovering unwarranted associations in data-driven applications,” *EuroS&P17*, Apr.

[31] S. Galhotra, Y. Brun, and A. Meliou, “Fairness testing: testing software for discrimination,” *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2017*, 2017. [Online]. Available: http://dx.doi.org/10.1145/3106237.3106277

[32] S. Udeshi, P. Arora, and S. Chattopadhyay, “Automated directed fairness testing,” *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering - ASE 2018*, 2018. [Online]. Available: http://dx.doi.org/10.1145/3238147.3238165

[33] L. Zhang, Y. Wu, and X. Wu, “Situation testing-based discrimination discovery: A causal inference approach,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI’16. AAAI Press, 2016, p. 2718–2724.

[34] F. Kamiran and T. Calders, “Data preprocessing techniques for classification without discrimination,” *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, Oct 2012. [Online]. Available: https://doi.org/10.1007/s10115-011-0463-8

[35] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, and K. R. Varshney, “Optimized pre-processing for discrimination prevention,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3992–4001. [Online]. Available: http://papers.nips.cc/paper/6988-optimized-pre-processing-for-discrimination-prevention.pdf

[36] B. H. Zhang, B. Lemoine, and M. Mitchell, “Mitigating unwanted biases with adversarial learning,” 2018.

[37] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, “Fairness-aware classifier with prejudice remover regularizer,” in *Machine Learning and Knowledge Discovery in Databases*, P. A. Flach, T. De Bie, and N. Cristianini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 35–50.

[38] F. Kamiran, S. Mansha, A. Karim, and X. Zhang, “Exploiting reject option in classification for social discrimination control,” *Inf. Sci.*, 2018.

[39] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, “On fairness and calibration,” 2017.

[40] M. Hardt, E. Price, and N. Srebro, “Equality of opportunity in supervised learning,” 2016.

[41] J. Martin, “Bias or systematic error (validity),” 2010. [Online]. Available: https://www.ctspedia.org/do/view/CTSpedia/BiasDefinition

[42] R. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. Natesan Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, R. Kush, and Y. Zhang, “Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias,” 10 2018.

[43] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, “Algorithmic decision making and the cost of fairness,” 2017.

[44] M. Feldman, S. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, “Certifying and removing disparate impact,” 2014.

[45] A. Chouldechova, “Fair prediction with disparate impact: A study of bias in recidivism prediction instruments,” 2016.

[46] J. Kleinberg, S. Mullainathan, and M. Raghavan, “Inherent trade-offs in the fair determination of risk scores,” 2016.

[47] A. Beutel, J. Chen, T. Doshi, H. Qian, A. Woodruff, C. Luu, P. Kreitmann, J. Bischof, and E. H. Chi, “Putting fairness principles into practice: Challenges, metrics, and improvements,” 2019.

[48] J. Chakraborty, T. Xia, F. M. Fahid, and T. Menzies, “Software engineering for fairness: A case study with hyperparameter optimization,” 2019.

[49] “Uci:adult data set,” 1994. [Online]. Available: http://mlr.cs.umass.edu/ml/datasets/Adult

[50] “propublica/compas-analysis,” 2015. [Online]. Available: https://github.com/propublica/compas-analysis

[51] “Uci:statlog (german credit data) data set,” 2000. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

[52] “Uci:default of credit card clients data set,” 2016. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[53] "Uci:heart disease data set," 2001. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[54] "Fairware 2018:international workshop on software fairness." [Online]. Available: http://fairware.cs.umass.edu/

[55] "Amazon just showed us that 'unbiased' algorithms can be inadvertently racist." [Online]. Available: https://www.businessinsider.com/how-algorithms-can-be-racist-2016-4

[56] "Why split data in the ratio 70:30?" 2012. [Online]. Available: http://information-gain.blogspot.com/

[57] V. Nair, Z. Yu, T. Menzies, N. Siegmund, and S. Apel, "Finding faster configurations using flash," *TSE*, pp. 1–1, 2018.

[58] V. Nair, T. Menzies, N. Siegmund, and S. Apel, "Using bad learners to find good configurations," in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 2017, pp. 257–267.

[59] R. Storn and K. V. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[60] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[61] K. Holstein, J. Wortman Vaughan, H. Daumé, M. Dudik, and H. Wallach, "Improving fairness in machine learning systems," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 2019. [Online]. Available: http://dx.doi.org/10.1145/3290605.3300830

[62] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf