A comparative study of fairness-enhancing interventions in machine learning*

Sorelle A. Friedler
Haverford College
sorelle@cs.haverford.edu

Carlos Scheidegger
University of Arizona
cscheid@cs.arizona.edu

Suresh Venkatasubramanian
University of Utah
suresh@cs.utah.edu

Sonam Choudhary†
University of Utah
sonam@cs.utah.edu

Evan P. Hamilton‡
Haverford College
evanphamilton@gmail.com

Derek Roth‡
Haverford College
derek.roth17@gmail.com

ABSTRACT

Computers are increasingly used to make decisions that have significant impact on people's lives. Often, these predictions can affect different population subgroups disproportionately. As a result, the issue of *fairness* has received much recent interest, and a number of fairness-enhanced classifiers have appeared in the literature. This paper seeks to study the following questions: how do these different techniques fundamentally compare to one another, and what accounts for the differences? Specifically, we seek to bring attention to many under-appreciated aspects of such fairness-enhancing interventions that require investigation for these algorithms to receive broad adoption.

We present the results of an open benchmark we have developed that lets us compare a number of different algorithms under a variety of fairness measures and existing datasets. We find that although different algorithms tend to prefer specific formulations of fairness preservations, many of these measures strongly correlate with one another. In addition, we find that fairness-preserving algorithms tend to be sensitive to fluctuations in dataset composition (simulated in our benchmark by varying training-test splits) and to different forms of preprocessing, indicating that fairness interventions might be more brittle than previously thought.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Software and its engineering** → *Software libraries and repositories*;

KEYWORDS

Fairness-aware machine learning, benchmarks

*This work was partially supported by National Science Foundation under grants IIS-1633387, IIS-1513651, and IIS-1633724, as well as by a grant from the Ethics and Governance of AI Initiative. Source code, including instructions for adding your own algorithm or dataset, can be found at https://github.com/algofairness/fairness-comparison and installed via `pip3 install fairness`.

†Work done while the author was a student at the University of Utah.

‡ Work done while the author was a student at Haverford College.

FAT '19, January 2019, Atlanta, Ga*

ACM ISBN 978-1-4503-6125-5/19/01.
https://doi.org/10.1145/3287560.3287589

ACM Reference Format:
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 Conference on Fairness, Accountability, and Transparency (FAT* '19).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3287560.3287589

1 INTRODUCTION

As the use of machine learning to make decisions about people has increased, so has the drive to make fairness-aware machine learning algorithms. A considerable body of research over the past ten years has produced algorithms for accurate yet fair decisions, under varying definitions of fair, for goals such as non-discriminatory hiring, risk assessment for sentencing guidance, and loan allocation. And yet we have not yet seen extensive deployment of these algorithms in the pertinent domains. The primary technical obstacle appears to be our ability to compare methods effectively across different evaluation measures and different data sets with consistent data preprocessing and testing methodologies. Such comparisons would not just reveal "best-in-class" methods; they would also suggest which measures are robust and how different algorithms are sensitive to different kinds of preprocessing. As pointed out by Lehr and Ohm [24], such considerations of the data processing *pipeline* are not just important for efficient implementation but also have legal ramifications for the resulting automated decision-making process.

In this paper, we present a test-bed to facilitate direct comparisons of *algorithms* with respect to *measures* on a variety of *datasets*. Our open-source framework allows for the easy addition of new methods, measures and data for the purpose of evaluation. We show how to use our test-bed for determining not only which specific algorithm has the best performance under a fairness or accuracy measure, but what types of algorithmic interventions[1] tend to be the most effective. In addition to the impact of these algorithmic choices, we examine the impact of different preprocessing techniques and different measures for accuracy and fairness that have an important, and previously obscured, impact on the results of these algorithms. Our goal is to provide a comprehensive comparative analysis of existing approaches that is currently lacking in the literature.

[1]In this paper, we use the term 'intervention' to refer to how the choice of algorithm used impacts the fairness of the overall system. We are not studying causal definitions of fairness.

1.1 Our results

Our evaluation yields the following major findings.

Fairness-accuracy tradeoffs depend on preprocessing (Section 5). Different algorithms tend to have slightly different requirements in terms of input: how are sensitive attributes encoded? Are multiple sensitive attributes supported? Does the algorithm directly support categorical attributes or are attribute transformations required? Choices for these requirements directly affect the accuracy and fairness of a fairness-aware classifier. This is significant because prior formal studies of fairness-accuracy tradeoffs typically focused on hyperparameter tuning, rather than preprocessing.

Measures of discrimination correlate with each other (Section 6). Even though there has been a proliferation of measures designed to highlight discrimination instances by machine learning algorithms, we find that a large number of these measures tend to strongly correlate with one another. As a result, techniques optimizing for one measure could perform well for a different measure (and similarly for poor performance).

Algorithms make significantly different fairness-accuracy tradeoffs (Section 7). The specific mechanisms that different algorithms employ to increase fairness are quite varied, but surprisingly, the actual predictions made by these algorithms tend to vary significantly as well. As a result, no algorithm's performance (as of the latest state of our benchmark) appears to dominate, either in accuracy or fairness measures.

Algorithms are fragile: they are sensitive to variations in the input (Section 7). We find surprising variability in fairness measures arising from variations in training-test splits; this appears to not have been previously mentioned in the literature.

2 BACKGROUND

Fairness-aware machine learning algorithms seek to provide methods under which the predicted outcome of a classifier operating on data about people is fair or non-discriminatory for people based on their *protected class status* such as race, sex, religion, etc., also known as a *sensitive attribute*. Broadly, fairness-aware machine learning algorithms have been categorized as those *preprocessing* techniques designed to modify the input data so that the outcome of any machine learning algorithm applied to that data will be fair, those *algorithm modification* techniques that modify an existing algorithm or create a new one that will be fair under any inputs, and those *postprocessing* techniques that take the output of any model and modify that output to be fair [28]. Many associated metrics for measuring fairness in algorithms have also been explored. These are detailed further in Section 6 and are also surveyed in [31]. This description of fairness-aware machine learning methods is limited to batch-learning-based interventions. We do not consider interventions that focus on sequential or reinforcement learning such as [8, 9, 15–17]

Preprocessing algorithms. The motivation behind preprocessing algorithms is the idea that training data is the cause of the discrimination that a machine learning algorithm might learn, and so modifying it can keep a learning algorithm trained on it from discriminating. This could be because the training data itself captures historical discrimination or because there are more subtle patterns in the data, such as an under-representation of a minority group, that makes errors on that group both more likely and less costly under certain accuracy measures. One such algorithm that we will analyze in this paper is that of Feldman et al. [10] that modifies each attribute so that the marginal distributions based on the subsets of that attribute with a given sensitive value are all equal; it does not modify the training labels. Additional preprocessing approaches include [5, 19].

Algorithm modifications. Modifications to specific learning algorithms, e.g., in the form of additional constraints, have been by far the most common approach. We study three such methods in this paper. Kamishima et al. [21] introduce a fairness focused regularization term and apply it to a logistic regression classifier. Zafar et al. [33] observe that standard fairness constraints are nonconvex and hard to satisfy directly and introduce a convex relaxation for purpose of optimization. Calders and Verwer [4] build separate models for each value of a sensitive attribute and use the appropriate model for inputs with the corresponding value of the attribute.

Another method that combines preprocessing and algorithm modification is the work by Zemel et al. [35]. Their approach is to learn a modified representation of the data that is most effective at classification while still being free of signals pertaining to the sensitive attribute.

Postprocessing techniques. A third approach to building fairness into algorithm design is by modifying the results of a previously trained classifier to achieve the desired results on different groups. Kamiran et al. [20] designed a strategy to modify the labels of leaves in a decision tree after training in order to satisfy fairness constraints. Recent work by Hardt et al. [13] and Woodworth et al. [32] explored the use of post-processing as a way to ensure fairness with respect to error profiles (see Section 6 for more on this).

In this paper we focus on *group fairness* approaches that aim to ensure non-discrimination across protected groups where the goal is to optimize metrics such as disparate impact. Another line of thought, known as *individual fairness*, is detailed in [7]. In this work, we do not study algorithms that seek to optimize individual fairness: our goal is to focus on methods that explicitly deal with group-based discrimination and there are (to the best of our knowledge) no publicly available implemented algorithms that optimize solely for individual fairness, although [5] does use individual fairness (approximately) as a distortion constraint in its pre-processing.

2.1 Related Work

Three prior efforts are relevant to our work. FairTest [30][2] provides a general methodology to explore potential biases or feature associations in a data set, as well as a way to identify regions of the input space where an algorithm might incur unusually high errors. THEMIS[11][3] takes a blackbox decision-making procedure and designs test cases automatically to explore where the procedure might be exhibiting group-based or causal discrimination. Fairness Measures [34] occupies a different point in the design space. Given a particular algorithm that one wishes to evaluate, they provide

[2] https://github.com/columbia/fairtest
[3] https://github.com/LASER-UMASS/Themis

a framework to test the algorithm on a variety of datasets and fairness measures. This approach on the one hand is more general than our framework, because it works with any algorithm. On the other hand, it is less effective for a comparative evaluation of different algorithms especially if they have different preprocessing and training methods.

There are other software packages that audit black box software to determine the influence of individual variables. We omit a detailed description of these approaches as they are out of the scope of the investigation presented here. For more information, the reader is referred to the excellent new survey on explainability by Guidotti et al. [12].

3 BENCHMARK STRUCTURE

In order to provide a platform for clear comparison of results across fairness-aware machine learning algorithms, we separate each stage of the learning and analysis process (see Figure 1) and ensure that each algorithm is compared using the same dataset (including the same preprocessing), the same set of training / test splits, and all desired fairness and accuracy measures. Much previous work has combined the preprocessing for a specific dataset with the code for the fairness-aware algorithm, which makes comparisons with other algorithms and other datasets difficult. Similarly, algorithms have often been analyzed only under one or two measures. Here, we distinguish preprocessing, algorithms, and measures, and create a pipeline in which all algorithms are analyzed under a standard preprocessing of datasets and a large set of measures.

In order to encourage easy adoption of this codebase as a platform for future algorithmic analysis, each of these choices is modularized so that adding new datasets, measures, and/or algorithms to the pipeline is as easy as creating a new object. The pipeline will then ensure that all existing algorithms are evaluated under the new dataset and measure. More details and instructions for adding to the code base can be found at the repository.[4]

4 DATA

We perform all experiments based on five real-world data sets that have been previously considered in the fairness-aware machine learning literature and preprocess each consistently depending on the needs of the algorithm.[5] The real-world datasets come from some of the domains impacted by questions of fairness in machine learning: hiring and promotion, credit-worthiness and loans, and recidivism prediction.

Ricci. The Ricci dataset comes from the case of Ricci v. DeStefano [29], a case before the U.S. Supreme Court in which the question at issue was an exam given to determine if firefighters would receive a promotion. The dataset has 118 entries and five attributes, including the sensitive attribute `Race`. The original promotion decision was made by a threshold of achieving at least a score of 70 on the combined exam outcome [26]. The goal in a fair learning context is to predict this original promotion decision while achieving fairness with respect to the sensitive attribute, `Race`.

Adult Income. The Adult Income dataset [25] contains information about individuals from the 1994 U.S. census. It is pre-split into a training and test set; we use only the training data and re-split it. There are 32,561 instances and 14 attributes, including sensitive attributes `race` and `sex`. 2,399 instances with missing data are removed during the preprocessing step. The prediction task is predicting whether an individual makes more or less than $50,000 per year.

German. The German Credit dataset [25] contains 1,000 instances and 20 attributes describing individuals along with a classification of each individual as a good or bad credit risk. Sensitive attribute `sex` is not directly included in the data, but can be derived from the given information. Sensitive attribute `age` is included, and is discretized into values `adult` (age at least 25 years old) and `youth` based on an analysis by [18] showing this discretization provided for the most discriminatory possibilities.

ProPublica recidivism. The ProPublica data includes data collected about the use of the COMPAS risk assessment tool in Broward County, Florida [2]. It includes information such as the number of juvenile felonies and the charge degree of the current arrest for 6,167 individuals, along with sensitive attributes `race` and `sex`. Data is preprocessed according to the filters given in the original analysis [2]. Each individual has a binary "recidivism" outcome, that is the prediction task, indicating whether they were rearrested within two years after the charge given in the data.

ProPublica violent recidivism. The violent recidivism version of the ProPublica data [2] describes the same scenario as the recidivism data described above, but where the predicted outcome is a rearrest for a violent crime within two years. 4,010 individuals are included after preprocessing is applied, including 652 instances of rearrest, and the sensitive attributes are `race` and `sex`. Note that while the individuals in this data set are a subset of the overall recividism set from above, their labels might be different, i.e., the same individual might have different recidivism labels in the two data sets.

5 PREPROCESSING

Each algorithm we will analyze has certain requirements for the type of data it will operate over, and these necessitate different preprocessing techniques. However, in order to provide a consistent comparison across algorithms, it's important that each algorithm receive the same input. We reconcile these needs by creating types of inputs that multiple algorithms can handle. Algorithms that handle the same input can be directly compared to each other. Algorithms can also be compared across different preprocessing strategies for the same dataset, even though in this setting conclusions are less clear, since the two sources of variability might interfere with one another.

Our first preprocessing step is to modify the input data according to any data-specific needs: removing features that should not be used for classification, removing or imputing any missing data, and potentially removing items or adding derived features. In order to allow the analysis of fairness based on multiple sensitive attributes (e.g., not just ensuring fairness based on race or sex alone, but based on both someone's race and sex) we also add a combined

[4] https://github.com/algofairness/fairness-comparison

[5] All raw datasets, preprocessing code, and resulting processed datasets are available in the repository: https://github.com/algofairness/fairness-comparison. Preprocessing described here can be reproduced by running: `python3 preprocess.py`

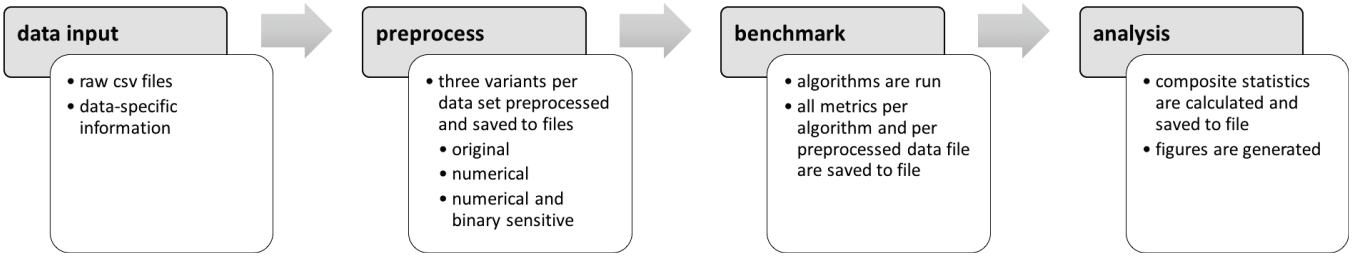

Figure 1: The stages of the fairness-aware benchmarking program: data input, preprocessing, benchmarking, and analysis. Intermediate files are saved at each stage of the pipeline to ensure reproducibility.

sensitive attribute (e.g., attribute "race-sex" with values like "White-Woman") to each dataset that contains multiple sensitive attributes. All algorithms will receive versions of the dataset with this same preprocessing applied.

While some algorithms are able to handle the datasets for training with only the described initial preprocessing (we'll call this version of the processed data `original`), most algorithms considered here have additional constraints.[6] For algorithms that can only handle numerical training data as input, we modify the data to include one-hot encoded versions of each categorical variable and call this version of the data `numerical`. Some algorithms additionally require that the sensitive attributes be binary (e.g., "White" and "not White" instead of handling multiple racial categorizations) - for this version of the data (`numerical+binary`) we modify the given privileged group to be 1 and all other values to be 0. The three data tags should be interpreted as indicating constraints on the *algorithms* that use them.

5.1 Analysis

With these preprocessed versions of each data set in place, we can compare how a single algorithm performs relative to all versions of the dataset on which it can run. The most common form of input for the algorithms we consider here is `numerical`, and all these algorithms can additionally handle the `numerical+binary` version of the dataset. This gives an opportunity to determine the effect, per algorithm and per dataset, of allowing an algorithm access to full information about sensitive attribute categorization or only a binary summary.

Figure 2 illustrates this analysis on the impact of the `numerical+binary` version of the preprocessed data on the algorithm proposed by Feldman et al. [10]. In the left figure we examine the relation between the accuracy on `numerical` preprocessing versus `numerical+binary` binary-encoded sensitive attributes. Each algorithm was run over ten random $\frac{2}{3} : \frac{1}{3}$ splits and the result on each split is shown as a single point on the figure. As discussed in Section 7, Feldman et al. use a generic classifier after running a preprocessing "fairness-enhancing" filter on the data, and the different algorithms reflect the different classifiers used. We also automate the parameter tuning for λ, the fairness-accuracy tradeoff parameter for this algorithm (more about parameter tuning specifics can be found in Section 7), for both accuracy and the disparate impact value. As we can see, for most variants of the algorithm the resulting accuracy is higher when using the `numerical+binary` representation than when using the `numerical` representation. We speculate that this is because the Feldman et al. algorithm conditions on the sensitive value in its preprocessing on the data, and this step better preserves accuracy when a larger number of people are in each sensitive group – as is the case when the unprivileged groups are grouped together in the binary preprocessing variant.

We can do a similar analysis on the fairness achieved by the methods, as seen in the right side of Figure 2. Again, we compare the fairness measure (in this case DI – see Section 6) achieved for different data representations. First, we see that the fairness achieved varies across runs, an issue we will return to when we discuss measure stability. Second, we notice that the algorithm variants achieve greater fairness when using the `numerical` preprocessing of the data, likely because each group's fairness is separately ensured, while in the `numerical+binary` variant all unprivileged classes are grouped together. Note that this indicates the presence a fairness-accuracy tradeoff which arises not from hyperparameters, but from *choice of preprocessing*.

6 MEASURES

There are many ways to evaluate the accuracy and fairness of a model. Rather than be exhaustive,[7] we will focus on representative measures for each aspect. Let $D = (\mathbb{X}, S, Y)$ be a dataset where $\mathbb{X}$ is the data subset that can be used for training (whether categorical or numerical), S is the sensitive attribute where 1 is the privileged class, and Y is the binary classification label where 1 is the positive outcome and 0 is the negative outcome. Let $\hat{Y}$ be the predicted outcomes of some algorithm. We can define accuracy and fairness measures in terms of conditional probabilities of outcome variables $(Y, \hat{Y})$ with respect to variables like $\hat{Y}$ and S.

6.1 Accuracy measures

We consider the standard accuracy measures: the (uniform) accuracy ($P[\hat{Y} = Y]$), the true positive rate (TPR) ($P[\hat{Y} = 1 \mid Y = 1]$), and the true negative rate (TNR) ($P[\hat{Y} = 0 \mid Y = 0]$). We also consider

[6] For example, `scikit-learn` classifiers only handle numerical data, even for classifiers like decision trees where this is not inherently a requirement. As a consequence, some of the tested algorithms that would otherwise handle original data require numerical data since these algorithms internally call `scikit-learn` procedures.

[7] A recent tutorial puts the number of fairness measures at 21 [27]!

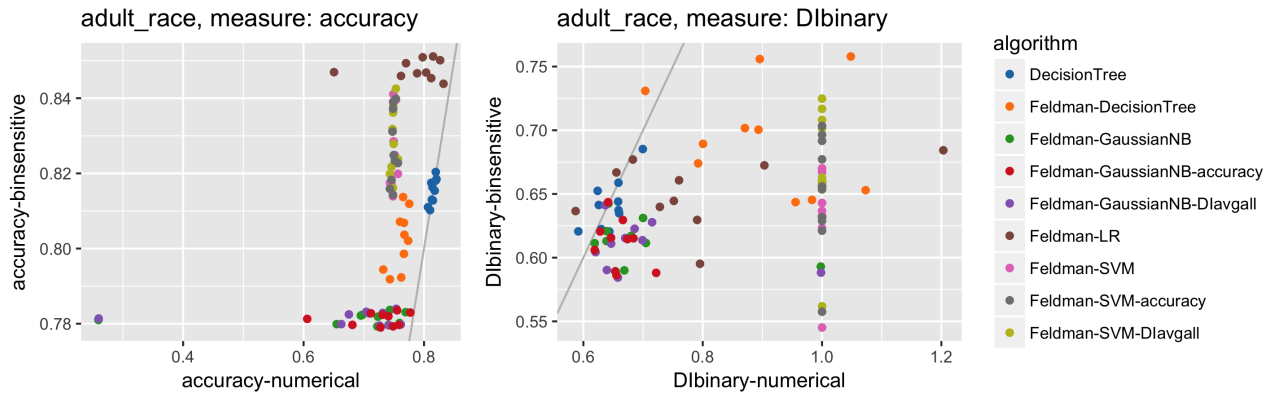

Figure 2: Examining the results of the Feldman et al. [10] algorithm under different preprocessing choices: `numerical` versus `numerical+binary`. Each dot plots the result of a single split of the data in terms of the labeled metric under both preprocessing choices. The gray line shows equality between the preprocessing choices. The model used within the Feldman algorithm is listed, and some variants of the algorithm had the tradeoff parameter optimized for either accuracy or disparate impact value.

the balanced classification rate (BCR), a version of accuracy that is unweighted per class:

Definition 6.1 (BCR).

$$\frac{P[\hat{Y} = 1 \mid Y = 1] + P[\hat{Y} = 0 \mid Y = 0]}{2}$$

All of these measures lie in the range $[0, 1]$.

6.2 Fairness measures

Fairness measures can be divided into three broad categories, in all cases conditioned on values of the sensitive attribute S. In what follows, we normalize measures to make comparisons easier. In all cases, the measures lie in the range $[0, \infty)$ or $[0, 2]$ where in both cases perfect fairness is achieved at 1. We note that some of these measures have appeared in the literature not as something to be optimized (to be close to 1) but as a constraint to be satisfied (i.e, that the appropriate value must equal 1).

6.2.1 Measures based on base rates.

Definition 6.2 (Disparate Impact (DI) [10, 33]).

$$\frac{P[\hat{Y} = 1 \mid S \neq 1]}{P[\hat{Y} = 1 \mid S = 1]}$$

This measure is inspired by one of the two tests for disparate impact in the legal literature in the United States [3]. In the cases where there are more than two values for a given sensitive attribute, we consider two variants of DI (which are equivalent in the case when there are only two sensitive values): binary and average. In the binary case, all unprivileged classes are grouped together into a single value $S \neq 1$ (e.g., "non White") that is compared as a group to the privileged class $S = 1$ (e.g., "White"). In the average case, pairwise DI calculations are done against the privileged class (e.g., "White" compared to "Black", "White" compared to "Asian", etc.) and the average of these calculations is taken. This is analogous to the one-vs-all and all-vs-all methodology in multi-class classification.

Definition 6.3 (CV [4]).

$$1 - \left(P[\hat{Y} = 1 \mid S = 1] - P[\hat{Y} = 1 \mid S \neq 1]\right)$$

This measure is the same as DI, but where the difference is taken instead of the ratio; such a measure has been used for example to measure discrimination in the United Kingdom [28]. A binary grouping strategy (described above for DI) is used in the case where there is more than one sensitive value, and the averaging method can also be used. Note that we do not take the absolute value of the difference so that skew in favor of one group versus another can be detected. We note that requiring $CV = 1$ is sometimes called a *demographic parity* constraint.

6.2.2 Measures based on group-conditioned accuracy. In general, we can think of fairness measures based on group-conditioned accuracy as asking whether the error rates for each group are similar. This yields the following definitions.

Definition 6.4. (*Group-conditioned accuracy measures.*)

s-Accuracy.

$$P[\hat{Y} = y \mid Y = y, S = s]$$

s-TPR.

$$P[\hat{Y} = 1 \mid Y = 1, S = s]$$

s-TNR.

$$P[\hat{Y} = 0 \mid Y = 0, S = s]$$

s-BCR.

$$\frac{P[\hat{Y} = 1 \mid Y = 1, S = s] + P[\hat{Y} = 0 \mid Y = 0, S = s]}{2}$$

We note that these measures have been studied under different names. For example, error rate balance [6] is the aim of achieving equal 1− s-TPR and 1− s-TNR values across sensitive groups and, equivalently, equalized odds [13] is the aim of achieving equal s-TPR and 1− s-TNR across sensitive groups.

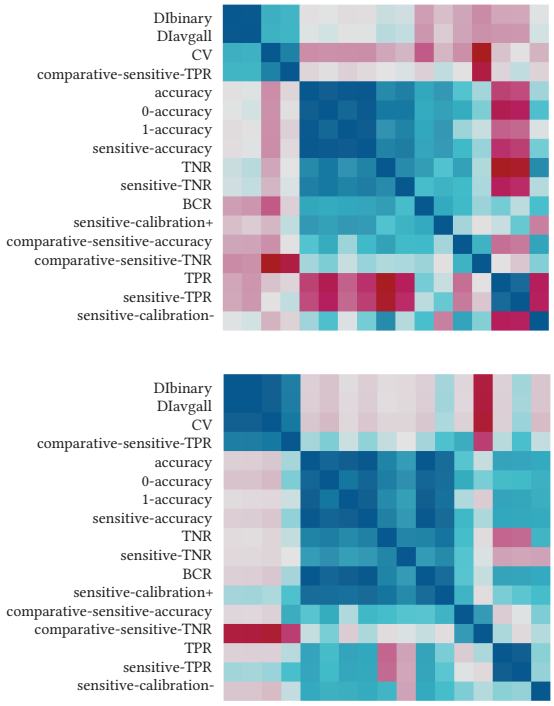

Figure 3: Examining the relationships between different measures of accuracy and fairness when considered across all examined datasets and algorithms (including both baseline and fairness-aware algorithms) under `numerical+binary` preprocessing. A simple sample correlation statistic is then computed for each set of pairs of measurements. Strongly positively correlated metric pairs are shown in blue and strongly negatively correlated pairs are shown in red. The top figure shows the results when run on the original datasets and the bottom when the datasets are downsampled to be balanced by class and sensitive attribute.

Letting any of the above measures be denoted $f(Y, \hat{Y}, s)$, the values can then be aggregated for comparison by taking the mean[8] directly $\sum_{s \in S} f(Y, \hat{Y}, s)/|S|$ or by taking the mean over comparisons analogous to DI and CV: $f(Y, \hat{Y}, s)/f(Y, \hat{Y}, 1)$ or $1 - (f(Y, \hat{Y}, 1) - f(Y, \hat{Y}, s))$. In each of these cases, as we saw above, the unprivileged sensitive values could be grouped together or handled separately in the ratio or difference. For example, consider a dataset where race is the sensitive attribute and which has been preprocessed so that the sensitive attribute takes binary values. In this case, the accuracy conditioned on having a sensitive value of 1 (e.g., "White") is denoted as the 1−accuracy. We will denote the average of the 1−accuracy and 0−accuracy in this case as the *race-accuracy* (or in general as the *sensitive-accuracy*) and the mean of the per-race differences, i.e., $\sum_{s \in S}[1 - (\text{1-accuracy} - \text{s-accuracy})]/|S|$, as the *comparative-race-accuracy* (or in general as the *comparative-sensitive-accuracy*). We'll use the same naming scheme for other accuracy measures and other sensitive attributes.

[8]Worst-case notions are not considered in this analysis and paper, but can be easily added to the code repository. Future work will consider these measures as well.

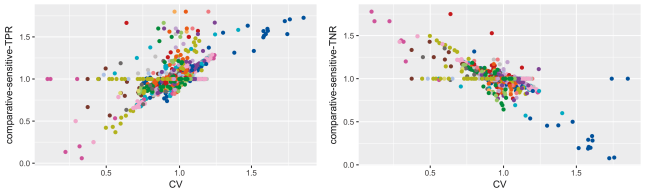

Figure 4: A comparison between the CV and comparative-sensitive-TPR (left) and comparative-sensitive-TNR (right) metrics across all datasets and fairness-aware algorithms considered. Each dot represents one out of 10 random train-test splits. Dots are colored by algorithm. The color legend is the same as that of Figure 5.

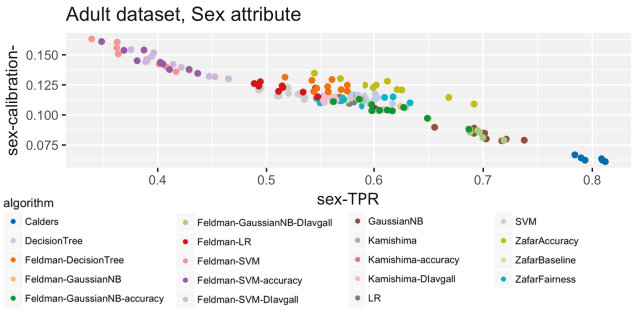

Figure 5: An illustration of the tradeoff between `sensitive-calibration-` and `sensitive-TPR` for all algorithms on the `Adult` dataset with sensitive value `sex`. Each dot represents one run out of 10 random train-test splits.

6.2.3 Measures based on group-conditioned calibration. A predictor that outputs a probability $\hat{Y}$ for an event is said to be *well-calibrated* if $P[Y = 1 \mid \hat{Y} = p] = p$. Traditionally, calibration measures are used for measuring the consistency of confidence scores. For instance, a well-calibrated predictor should be 75% accurate on all predictions it issued with confidence at least 75%. Motivated by this, we define fairness measures by conditioning the "calibration function" $p \mapsto P[Y = 1 \mid \hat{Y} = p]$ on a group.

Definition 6.5 (s-Calibration+).

$$P[Y = 1 \mid \hat{Y} = 1, S = s]$$

Definition 6.6 (s-Calibration-).

$$1 - P[Y = 1 \mid \hat{Y} = 0, S = s]$$

Calibration has been introduced previously with the goal of equalizing across sensitive value [6, 23]. Note that we define *s*-Calibration- so that "good" values are close to 1, consistent with the other measures in this paper.

6.3 Analysis

Most of the algorithms considered here (discussed in more detail in Section 7) were analyzed with respect to the single fairness measure being introduced in the paper, or with respect to a subset of the

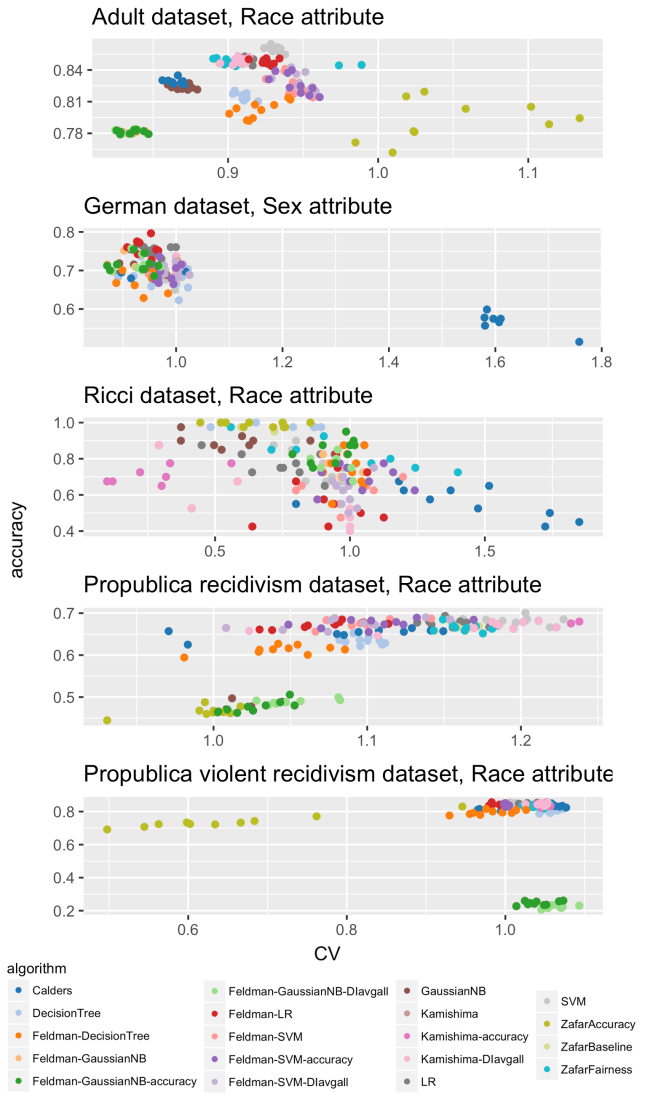

Figure 6: The performance of all algorithms on each dataset with the goal of removing discrimination on a specific attribute. From top to bottom, the algorithms and sensitive attributes considered are: Adult Income on race, German Credit on sex, Ricci on race, ProPublica recidivism on race, and ProPublica violent recidivism on race. Each point is the result of a single algorithm running on a single training / test split - each algorithm is shown for ten such splits.

measures. Incorporating all of the above accuracy and fairness measure variations into our software framework allows us to examine measure trends across multiple measures and multiple algorithms. While these measures are often presented as opposing, here we are interested in analyzing the extent to which this is true in practice.

There are many variations on these and other measures, but we find many of these are correlated on these algorithms and datasets. This is not entirely surprising as these measures are definitionally related. For example, DI takes the ratio of two probabilities while CV takes the difference. However, by analyzing resulting measures across many algorithms, we find correlations that are less obvious. In fact, it appears that there are a few main clusters of measures.

In Figure 3 we show the correlations between measures across all algorithms (both baseline and fairness-aware) and datasets when each fairness-aware algorithm is run for each sensitive attribute (including the combined sensitive attributes such as "race-sex"). The top of the figure shows the results when run on the full datasets. There appear to be four main clusters: DI and CV, accuracy-related measures, TPR-related measures, and sensitive-calibration-.

To determine the extent how this clustering was impacted by the skew in the data (in terms of both class and sensitive attribute), the datasets were downsampled uniformly with replacement to contain 1000 items that were balanced so that 500 have the positive classification and 500 the negative, and within each of those 250 have the privileged sensitive attribute and 250 have an unprivileged value. The bottom of Figure 3 shows the correlations between metrics on this balanced sample. This clarifies the clusterings and five are found: DI-related measures, accuracy-related measures, comparative-accuracy measures, TPR measures, and sensitive-calibration-. On this balanced sample there also appear to be two weaker but larger clusterings: DI-related measures versus all accuracy, comparative-accuracy, and calibration-related measures.

Within this clustering there are some clear patterns. Pairs of accuracy measures and their sensitive counterparts (e.g., accuracy and sensitive-accuracy, TPR and sensitive-TPR, and TNR and sensitive-TNR) are always clustered together. Recall that sensitive-accuracy is the average of the accuracy on the privileged group (the 1-accuracy) and the accuracy on the unprivileged groups (the 0-accuracy), so it makes sense that improving the overall accuracy would improve this average as well. Perhaps more surprisingly, the 0-accuracy and 1-accuracy are also strongly positively correlated, i.e., improving the accuracy on the privileged group also improves the accuracy on the unprivileged groups on these algorithms and datasets.

A caveat to the strength of these clusterings is that these results only consider the measures when assessed on these algorithms. Algorithms might exist or be created that focus on optimizing one specific measure that change these clusterings, especially in cases where the rationale for the clusterings is less obvious (e.g., the clustering of accuracy and TNR together). But this experiment does allow us to assess *in practice* how optimizing for one fairness measure affects other fairness measures.

The Calders, Feldman, Kamishima, and Zafar algorithms were all designed to optimize DI, CV, or similarly motivated measures. Since DI and CV are analytically closely related to each other, optimizing for one can be reasonably expected to optimize for the other. But does optimizing for these base rate focused fairness measures also optimize for the group-conditioned accuracy focused fairness measures? When considering only these fairness-aware algorithms, the clusterings presented in Figure 3 still hold, i.e., optimizing for DI and CV does not tend to increase accuracy and the other measures in that cluster. Interestingly though, DI and CV do have a strong positive correlation with comparative-sensitive-TPR and a strong negative correlation with comparative-sensitive-TNR. Figure 4 demonstrates these correlations empirically for CV with

comparative-sensitive TPR (correlation of 0.65) and comparative-sensitive-TNR (correlation of −0.76). Similar results (correlations of 0.76 and −0.71 respectively) are found on the datasets when sampled to be balanced. Note that these numbers are not completely reflective of the correlation due to outliers that have a fixed value of CV. We retained them in spite of this in the interest of transparency.

Additionally, in some cases we expect would be tradeoffs between measures. Assuming unequal base rates across populations, impossibility results show it is impossible to achieve both calibration and error rate balance (both the same false positive rate and the same false negative rates across groups) [6, 23]. In Figure 5 we empirically examine this tradeoff. As before, each colored point represents one instance of train-test split for an algorithm. As Figure 5 shows, there is a clear tradeoff between sensitive-calibration- and sensitive-TPR for each dataset. Interestingly, different algorithms situate themselves in different parts of the tradeoff line.

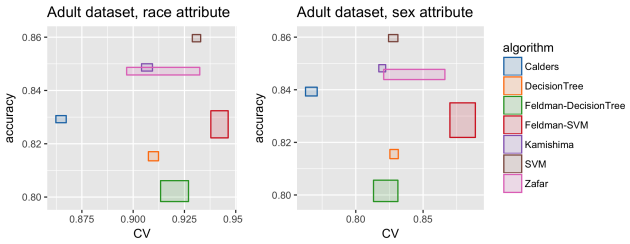

Figure 7: The stability of algorithms on `Adult`. Each algorithm is tested on ten random train / test splits and a rectangle centered on the mean and with a width and height equal to the standard deviation along that measure is plotted. On the left, the algorithms attempt to remove race discrimination, and on the right, sex discrimination.

7 ALGORITHMS

We choose a selection of existing fairness-aware algorithms to assess, based on availability of source code and diversity of fairness interventions (e.g., preprocessing versus algorithm modification). Each algorithm is run on each dataset and each metric is calculated on the predicted results.[9] Synthesis statistics (such as stability) are then calculated and comparison graphs are produced.[10] We analyze the following algorithms along with non-fairness-aware algorithms chosen for a baseline comparison: SVM, decision trees, Gaussian naive Bayes, and logistic regression (LR).

Calders and Verwer [4]. Calders and Verwer introduce a fairness-aware algorithm modification called Two Naive Bayes. Their approach trains separate models for the values and iteratively assesses the fairness of the combined model under the CV measure, makes small changes to the observed probabilities in the direction of reducing the measure, and retrains their two models. This algorithm can handle both categorical and numerical input data, but requires that the given sensitive attribute be binary. We use the Kamishima et al. [21] implementation of this algorithm.[11] The algorithm has a β hyperparameter specifying a prior probability for the features. We follow the original implementation and use a default of $\beta = 1.0$.

Feldman et al. [10]. Feldman et al. give a preprocessing approach that modifies each attribute so that the marginal distributions based on the subsets of that attribute with a given sensitive value are all equal; it does not modify the training labels. Any algorithm can then be trained on the resulting "repaired" data. The algorithm can handle both categorical and numerical input data, but since we train scikit-learn classifiers based on this preprocessed data, our implementation can only handle numerical input. Both binary and non-binary sensitive attributes can be handled. A tuning parameter λ is provided to tradeoff between fairness and accuracy, where $\lambda = 0$ gives the fairness of a regular non-fairness aware classifier and $\lambda = 1$ maximizes fairness. $\lambda = 1$ is used as the default, and all values of λ at increments of 0.05 in $[0, 1]$ are included when the algorithm is optimized using a grid search over the parameters. The implementation comes from Feldman et al. [10] and [1].[12]

Kamishima et al. [21]. Kamishima et al. introduce a fairness-focused regularization term and apply it to a logistic regression classifier. Their approach requires numerical input and a binary sensitive attribute. A tuning parameter η is provided to tradeoff between fairness and accuracy, where $\eta = 1$ is the default. When optimizing the parameter we use values between 0 and 300, with a finer grid used for the lower values of that range; these parameter choices are based on the experimental exploration of this parameter given in [21]. As above, we use Kamishima et al.'s implementation.

Zafar et al. [33]. Zafar et al. re-express fairness constraints (which can be nonconvex) via a convex relaxation. This allows them to define efficient versions of fairness-aware logistic regression and support vector machines. They propose two related optimization problems, one that maximizes accuracy subject to fairness constraints (in our experiments, we call this "ZafarAccuracy"), and another to maximize fairness subject to accuracy constraints (we call this "ZafarFairness"). They use two parameters: c is a parameter that controls the degree of independence of the outcome and the sensitive attribute via a covariance calculation: setting $c = 0$ forces complete independence (and therefore fairness). The second parameter γ fixes the degree of approximation they are willing to tolerate: the algorithm is only required to find an answer that is within a $1+\gamma$ factor of the optimal solution. In their experiments they set $\gamma = 0.5$ and vary c as a linear function of the corresponding covariance estimate for an unconstrained classifier. When optimizing, we use values between 0.001 and 1 in 10 logarithmic steps.

Figure 6 shows a basic summary of the performance of each algorithm considered on each data set. [13] Since each algorithm focuses on creating a fair outcome with respect to a specific attribute in the data, we have chosen a single sensitive attribute to consider per

[9] All algorithm implementations can be found in the repository (https://github.com/algofairness/fairness-comparison), along with all resulting metric calculations. The full set of results can be reproduced by running: `python3 benchmark.py`

[10] Algorithm analysis code can be found in the repository, and can be reproduced by running: `python3 analysis.py`

[11] https://github.com/tkamishima/kamfadm/releases/tag/2012ecmlpkdd

[12] https://github.com/algofairness/BlackBoxAuditing

[13] For `propublica-violent-recidivism` and `propublica-recidivism`, the results for `GaussianNB` and `Feldman-GaussianNB` are drawn behind `Feldman-GaussianNB-accuracy`. In addition, `propublica-violent-recidivism` is unbalanced, with only 16% of the data representing rearrests.

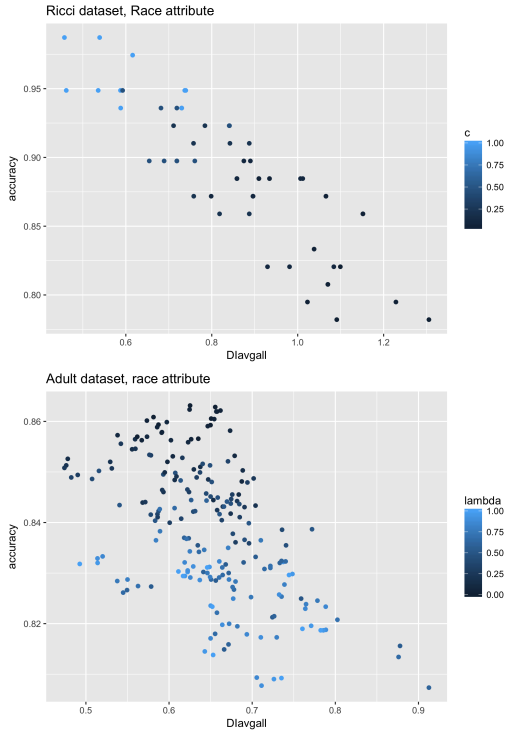

Figure 8: The results of the Zafar et al. [33] algorithm on `Ricci` (top) and the Feldman et al. [10] algorithm on `Adult Income` (bottom) when the provided parameter to tradeoff between fairness and accuracy is used. The parameter is varied and each split and each new parameter value is shown.

dataset in these overall results. It is clear that there is no one "winner" - no algorithm that is both more fair and more accurate than the others on all datasets. It is also clear that there is tremendous variation even within a single algorithm over the random splits it receives. We examine this point in more detail next.

7.1 Stability

When analyzing algorithms, we are additionally concerned with *stability* - will the algorithm still perform well if the training data is slightly different? To assess this, we considered the standard deviation of each metric over 10 random splits, where for each split the algorithm is trained and evaluated on a different train/test partition. The results are shown in Figure 7 for `Adult Income` for all algorithms when focusing on non-discrimination in terms of race (left) and sex (right) using `numerical+binary` preprocessing. These results give perhaps the clearest indication of the quality of an algorithm on a given data set. It is also easy to see that each algorithm occupies a slightly different place on the trade-off between fairness (measured here by CV when taken over binary sensitive attributes) and accuracy. For example, when focusing on non-discrimination in either sex or race on the Adult dataset, Zafar et al.'s algorithm is potentially the best choice in terms of a balance between fairness and accuracy, but the large standard deviation over CV may make it a less desirable option.

7.2 Parameters

Many fairness-aware learning algorithms provide a parameter to allow manually trading off fairness and accuracy. We automate the search for this balance and present results for all algorithms optimizing accuracy or fairness. This provides an additional means of testing the algorithm, as well as the possibility for further optimizing the tradeoff between the two. Figure 8 shows different results based on parameter tuning for the Zafar et al. [33] algorithm on `Ricci` (left) and the Feldman et al. [10] algorithm on `Adult Income`. A clear tradeoff between fairness and accuracy in these algorithms can be seen; the parameters are appropriately allowing exploration of the possible solution space.

7.3 Multiple sensitive attributes

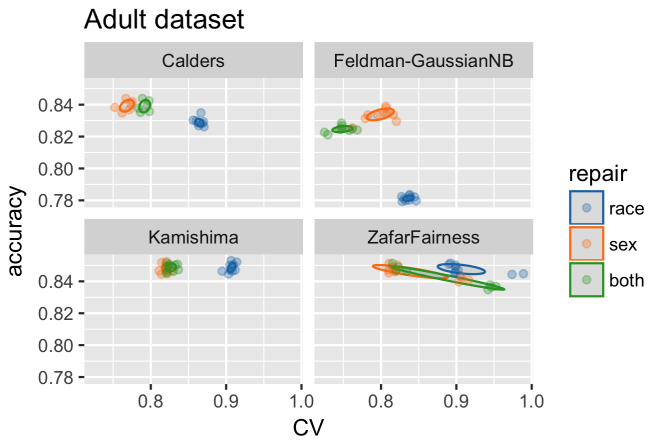

Figure 9: Four algorithms making predictions while accounting for different protected attributes (race, sex, and a composite attribute). These not only behave quite differently from one another, but their performance varies significantly depending on which specific attribute is being considered.

While there are still few fairness-aware algorithms that can explicitly handle multiple sensitive attributes ([14, 22]), all algorithms discussed can handle them if preprocessed as described earlier so that they are combined into a single sensitive attribute (e.g., race-sex). However, we might expect combining the attributes in this way to degrade performance under some metrics, especially when such algorithms can only handle binary sensitive attributes, or when too many combinations cause imbalance issues for some of the new combined sensitive values. Looking at the `Adult` dataset when fairness-aware algorithms are run focusing on non-discrimination in terms of race, sex, and both, we find varying results for each of the algorithms in Figure 9. Sex is especially predictive on the Adult Income data set, so the CV value for sex is low, even on these fairness-aware algorithms. Race generally receives a higher CV value from these algorithms. When correcting for both at once, most of the algorithms find that the CV value is somewhere in between that for race and that for sex, but the Zafar et al. [33] algorithm has a much larger variance over race and sex than over

either individually. While it might have been the case that looking at a combined sensitive value would cause these algorithms to drastically lower in accuracy and/or fairness, encouragingly this does not appear to be the case.

8 DISCUSSION

Besides providing a central point of access to existing fairness-enhancing interventions and classification algorithms, our benchmark also highlights a number of gaps in the current practice and reporting of fairness issues in machine learning. We conclude with the following recommendations for future contributions to the area:

Emphasize preprocessing requirements. If there are multiple plausible ways in which a dataset can be processed to generate training data for an algorithm, provide performance metrics for more than one of the possible choices. If algorithms are being compared to each other, ensure they are compared based on the same preprocessing.

Avoid proliferation of measures. New fairness measures should only be introduced if they behave fundamentally differently from existing metrics. Our study indicates that a combination of group-conditioned accuracy and either DI or CV is a good minimal set.

Account for training instability. Showing the performance of an algorithm in a single training-test split appears to be insufficient. We recommend reporting algorithm success and stability based on a moderate number of randomized training-test splits.

One limitation of our benchmark is the number of methods it currently provides implementation for. We hope other researchers will contribute their implementations to the repository. It would be particularly interesting to see how our conclusions above evolve as the number and variety of methods increases.

Additionally, while we frame some of the differences in algorithm performance as fairness versus accuracy tradeoffs, this can be misleading since it makes many assumptions about the data and social context, including, e.g., that the labels represent desired outcomes. We leave the examination of how the algorithmic choices interact with the social context for other work.

9 ACKNOWLEDGEMENTS

We thank Shira Mitchell and anonymous reviewers for helpful comments on an earlier version of the paper.

REFERENCES

[1] Philip Adler, Casey Falk, Sorelle A Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. 2018. Auditing black-box models for indirect influence. *Knowledge and Information Systems* 54, 1 (2018), 95–122.

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. *ProPublica* (May 23, 2016).

[3] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Cal. L. Rev.* 104 (2016), 671.

[4] Toon Calders and Sicco Verwer. 2010. Three Naive Bayes Approaches for Discrimination-Free Classification. *Data Mining and Knowledge Discovery* 21, 2 (2010), 277–292.

[5] 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 30*. 3995–4004.

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

[7] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness Through Awareness. In *Proc. of Innovations in Theoretical Computer Science*.

[8] Danielle Ensign, Sorelle A. Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. 2018. Decision Making with Limited Feedback: Error bounds for Recidivism Prediction and Predictive Policing.. In *Algorithmic Learning Theory (ALT)*.

[9] Danielle Ensign, Sorelle A. Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. 2018. Runaway Feedback Loops in Predictive Policing. In *Conf. on Fairness, Accountability and Transparency in Computer Science (FAT*)*.

[10] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. *Proc. 21st ACM KDD* (2015), 259–268.

[11] 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.

[12] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. A Survey Of Methods For Explaining Black Box Models. *arXiv preprint arXiv:1802.01933* (2018).

[13] Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *Adv. in Neural Inf. Processing Systems*. 3315–3323.

[14] Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. 2017. Calibration for the (Computationally-Identifiable) Masses. *arXiv:1711.08513* (Nov. 2017).

[15] Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. 2017. Fairness in Reinforcement Learning. In *PMLR*. 1617–1626.

[16] Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2016. Fair Algorithms for Infinite and Contextual Bandits. *arXiv:1610.09559 [cs]* (Oct. 2016).

[17] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. 2016. Fairness in Learning: Classic and Contextual Bandits. In *Advances in Neural Information Processing Systems 29*. 325–333.

[18] Faisal Kamiran and Toon Calders. 2009. Classifying without Discriminating. In *Proc. of the IEEE International Conf. on Computer, Control and Communication.*

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

[20] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *Proc. of IEEE Intl. Conf. on Data Mining*. 869–874.

[21] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware Classifier with Prejudice Remover Regularizer. *Machine Learning and Knowledge Discovery in Databases* (2012), 35–50.

[22] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2017. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. *arXiv preprint arXiv:1711.05144* (2017).

[23] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent trade-offs in the fair determination of risk scores. In *Proceedings of Innovations in Theoretical Computer Science (ITCS)*.

[24] David Lehr and Paul Ohm. 2017. Playing with the Data: What Legal Scholars Should Learn About Machine Learning. *UC Davis Law Review* 51, 2 (2017), 653–718.

[25] M. Lichman. 2013. UCI Machine Learning Repository. (2013). http://archive.ics.uci.edu/ml

[26] Weiwen Miao. 2011. Did the Results of Promotion Exams Have a Disparate Impact on Minorities? Using Statistical Evidence in Ricci v. DeStefano. *J. of Stat. Ed.* 19, 1 (2011).

[27] Arvind Narayanan. 2018. 21 Fairness Definitions and Their Politics. (Feb. 23 2018). Tutorial presented at the Conf. on Fairness, Accountability, and Transparency.

[28] Andrea Romei and Salvatore Ruggieri. 2013. A Multidisciplinary Survey on Discrimination Analysis. *The Knowledge Engineering Review* (April 3 2013), 1–57.

[29] Supreme Court of the United States. 2009. Ricci v. DeStefano. 557 U.S. 557, 174. (2009), 2658 pages.

[30] Florian Tramèr, Vaggelis Atlidakis, Roxana Geambasu, Daniel J. Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. 2015. Discovering Unwarranted Associations in Data-Driven Applications with the FairTest Testing Toolkit. *CoRR* abs/1510.02377 (2015). arXiv:1510.02377

[31] Indrė Žliobaitė. 2017. Measuring discrimination in algorithmic decision making. *Data Mining and Knowledge Discovery* 31, 4 (July 2017), 1060–1089.

[32] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. 2017. Learning non-discriminatory predictors. *arXiv preprint arXiv:1702.06081* (2017).

[33] 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.

[34] Meike Zehlike, Carlos Castillo, Francesco Bonchi, Ricardo Baeza-Yates, Sara Hajian, and Mohamed Megahed. 2017. Fairness Measures: A Platform for Data Collection and Benchmarking in discrimination-aware ML. http://fairness-measures.org. (Jun 2017).

[35] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning Fair Representations. In *Proc. of Intl. Conf. on Machine Learning*. 325–333.