Fair Enough: Searching for Sufficient Measures of Fairness

SUVODEEP MAJUMDER, JOYMALLYA CHAKRABORTY, GINA R. BAI, KATHRYN T. STOLEE, and TIM MENZIES, North Carolina State University, USA

Testing machine learning software for ethical bias has become a pressing current concern. In response, recent research has proposed a plethora of new fairness metrics, for example, the dozens of fairness metrics in the IBM AIF360 toolkit. This raises the question: How can any fairness tool satisfy such a diverse range of goals? While we cannot completely simplify the task of fairness testing, we can certainly reduce the problem. This article shows that many of those fairness metrics effectively measure the same thing. Based on experiments using seven real-world datasets, we find that (a) 26 classification metrics can be clustered into seven groups and (b) four dataset metrics can be clustered into three groups. Further, each reduced set may actually predict different things. Hence, it is no longer necessary (or even possible) to satisfy all fairness metrics. In summary, to simplify the fairness testing problem, we recommend the following steps: (1) determine what type of fairness is desirable (and we offer a handful of such types), then (2) lookup those types in our clusters, and then (3) just test for one item per cluster.

For the purpose of reproducibility, our scripts and data are available at https://github.com/Repoanonymous/Fairness_Metrics.

CCS Concepts: • **Social and professional topics** → **User characteristics**; • **Software and its engineering** → *Software design tradeoffs*; *Software reliability;*

Additional Key Words and Phrases: Software fairness, fairness metrics, clustering, theoretical analysis, empirical analysis

ACM Reference format:
Suvodeep Majumder, Joymallya Chakraborty, Gina R. Bai, Kathryn T. Stolee, and Tim Menzies. 2023. Fair Enough: Searching for Sufficient Measures of Fairness. *ACM Trans. Softw. Eng. Methodol.* 32, 6, Article 134 (September 2023), 22 pages.
https://doi.org/10.1145/3585006

1 INTRODUCTION

The issue of bias in the **Artificial Intelligence (AI)** and **machine learning (ML)** community has gained much momentum in the last few years. Increasingly, the software is being used for critical automated decision-making processes, such as patient release from hospitals [14, 80], credit card applications [45], hiring [78], and admissions [17]. According to guidelines from the European Union [12] and IEEE, the software cannot be used in real-life applications if it is found to be discriminatory toward an individual based on any sensitive attribute such as gender, race, or age. Hence

The work was partially funded by LAS and NSF Grant No. 1908762.
Authors' address: S. Majumder, J. Chakraborty, G. R. Bai, K. T. Stolee, and T. Menzies, North Carolina State University, Raleigh, USA; emails: {smajumd3, jchakra, rbai2, ktstolee}@ncsu.edu, timm@ieee.org.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1049-331X/2023/09-ART134 $15.00
https://doi.org/10.1145/3585006

"fairness testing" is now an open and pressing problem in software engineering. Researchers are working on topics such as fairness and bias in ML, and the explainability of "black-box" models, which focus on the risks and drawbacks of using ML in automated decision-making. Research institutions, private companies, and public sector organizations are formulating ethical principles and guidelines for the responsible use of AI systems. To create and establish such principles and guidelines, we need definitions of fairness. Thus, over the past few years, to measure fairness, researchers have proposed a plethora of fairness metrics that try to formalize different perspectives from which to assess and monitor fairness in decision-making processes. That number is growing (e.g., see all the metrics proposed in References [18, 23, 59]). Given that, it is somewhat strange to report that researchers in this area only use a few metrics in their papers [35, 50, 56, 61, 72, 87]. For example, in our literature review of papers from the past 3 years, we see only a handful of papers (13 of 60, to the best of our knowledge) using more than five fairness metrics to evaluate their method. This is surprising, since all of them ignore more than half the available metrics. Is that wise? Can only a few fairness metrics help us detect all types of bias, or do we need to use all of them? Moreover, if only a few can satisfy all fairness measures, then which few do we choose?

Recently, the authors faced a similar methodological issue where reviewers challenged the validity of the metrics they used to assess that work. Prompted by that experience, we examined how the current SE research community selects metrics for assessing the *fairness of algorithmic decision-making* from an empirical point of view. Verma et al. [86] mentions that statistical definitions of metrics are often insufficient, and it is often unclear how metrics will perform when applied in real data. They also said that these theoretical definitions could be biased given the implicit biases of the expert. Thus an empirical analysis of these metrics is necessary, along with a theoretical analysis. Also, their theoretical results were not consistent with empirical observations (specifically, they found that theoretically similar metrics proved to show different empirical performances). For example, our reading of the literature is that it often contains what might be called an anti-pattern:

- While the literature proposes a plethora of metrics[1]
- We could not find a principled argument (across a large space of known metrics) that it was necessary/unnecessary to report some metric X.

This raises various methodological questions:

- Should we reject papers that "only" use (e.g.,) five metrics? Or should researchers always use dozens of metrics?
- When we use automatic tools to optimize for fairness, should we optimize for dozens of goals? Or is optimizing for a smaller set sufficient?

To resolve these methodological concerns, we made the following conjecture. Given the large space of known metrics (such as the 30 studied in this article), perhaps *many of these metrics are measuring the same thing*. As shown by the experiments of this article, this is indeed the case, since we can cluster these 30 metrics into around half a dozen. While our results pertain to a particular domain, there is nothing in principle stopping this methodology from being applied to any domain where researchers keep proposing new metrics without first checking if the new metric is not just "old wine in new bottles."

The conjecture of this paper test is that *too many spurious metrics all measure very similar things*. If that were true, then it should be possible to simplify the fairness assessment as follows:

[1]For example, the Fairlearn [18] tool lists 16 metrics; the Fairkit-learn tool [59] comes with its own 16 metrics; IBM AIF360 toolkit [23] offers 45 fairness metrics.

Run metrics on real-world data. Find clusters of correlated metrics. Prune "insensitive clusters."[2] *Only use one metric per surviving cluster.*

This article experiments with seven datasets and finds that (a) 26 classification fairness metrics can be clustered into just seven groups, (b) four dataset metrics can be clustered into three groups, and (c) these clusters actually predict for different things. It is no longer necessary (or even possible) to satisfy all these fairness metrics. Hence, to simplify fairness testing, we recommend (a) determining what type of fairness is desirable (and we offer a handful of such types), then (b) looking up those types in our clusters, and then (c) testing for one item per cluster.

This article is structured around these research questions.

RQ1: *Do current fairness metrics agree with each other?* Our experiments show that current fairness metrics often disagree with each other.

RQ2: *Can we group (cluster) fairness metrics based on similarity?* Based on our experimental framework with agglomerative clustering [5], we could find seven meaningful clusters for 26 classification metrics and three clusters for four dataset metrics. Each of the resultant clusters measures different types of bias, and selecting one metric from each should be representative enough to measure an increase or decrease in bias in other metrics in the same cluster.

RQ3: *Are some fairness metrics more sensitive to change than others?* Our result shows that while most metrics are sensitive to the changes in bias in the model, some metrics (specifically between group individual fairness metrics) are not.

RQ4: *Can we achieve fairness based on all the metrics at the same time?* Our results show that while achieving fairness based on some metrics is possible, achieving fairness based on all the metrics is challenging, since some are competing goals and some are contradictory based on definitions.

In terms of research contributions, this study is important, since the art of software fairness testing is evolving rapidly. Studies like this one are essential to documenting what methods are "best" (as opposed to those that might distract from core issues). Accordingly,

- This article proposes a novel metric assessment tactic that can clarify and simplify future research reports in this field (run metrics on real-world data; find clusters of correlated metrics; prune "insensitive clusters1"; only use one metric per surviving cluster).
- This article tests that tactic in an *extensive case study* applying 30 fairness metrics and groups them into clusters (RQ1 and RQ2). This study is extensive, since it is far more detailed than prior work. All our empirical results were repeated 100 times. Our study explores multiple bias mitigation algorithms on seven datasets (than prior work [34, 36–38, 55] was tested on far fewer metrics and far fewer datasets).
- To the best of our knowledge, this study is the first to perform such a *sensitivity meta-analysis* of fairness testing and to warn that some metrics are unresponsive to data changes (RQ3).
- This study also presents a *meta-analysis of metrics ability* to achieve fairness after applying the bias mitigation technique (RQ4).
- To support replication and reproduction of our results, all our datasets and scripts are publicly available at https://github.com/Repoanonymous/Fairness_Metrics.

1.1 Preliminaries

Before beginning, we digress to make four points.

First, mitigating the untoward effects of AI is a much broader problem than just exploring bias in algorithmic decision-making (as done in this article). The general problem of fairness is that

[2]Note: Here, by "insensitive" clusters, we mean those where changes to the data do not change the fairness scores.

Table 1. Mathematical Definitions of the Classification and Dataset Metrics Used in This Research

Metric Id (MID)	Metric Name	Description	Ideal Value	AIF360	Fairkit	Fairlearn
Classification Metrics						
C0	true_positive_rate_difference	$TPR_{D=unprivileged} - TPR_{D=privileged}$	0	✓	✓	✓
C1	false_positive_rate_difference	$FPR_{D=unprivileged} - FPR_{D=privileged}$	0	✓	✓	✓
C2	false_negative_rate_difference	$FNR_{D=unprivileged} - FNR_{D=privileged}$	0	✓	✓	✓
C3	false_omission_rate_difference	$FOR_{D=unprivileged} - FOR_{D=privileged}$	0	✓	✓	
C4	false_discovery_rate_difference	$FDR_{D=unprivileged} - FDR_{D=privileged}$	0	✓	✓	
C5	false_positive_rate_ratio	$FPR_{D=unprivileged}/FPR_{D=privileged}$	1	✓	✓	✓
C6	false_negative_rate_ratio	$FNR_{D=unprivileged}/FNR_{D=privileged}$	1	✓	✓	✓
C7	false_omission_rate_ratio	$FOR_{D=unprivileged}/FOR_{D=privileged}$	1	✓	✓	
C8	false_discovery_rate_ratio	$FDR_{D=unprivileged}/FDR_{D=privileged}$	1	✓	✓	
C9	average_odds_difference	$\frac{1}{2}$ (*false_positive_rate_difference* + *true_positive_rate_difference*)	0	✓	✓	
C10	average_abs_odds_difference	$\frac{1}{2}$ (|*false_positive_rate_difference*| + |*true_positive_rate_difference*|)	0	✓	✓	
C11	error_rate_difference	$ERR_{D=unprivileged} - ERR_{D=privileged}$	0	✓	✓	
C12	error_rate_ratio	$ERR_{D=unprivileged}/ERR_{D=privileged}$	1	✓	✓	
C13	selection_rate	$Pr(\hat{Y} = favorable)$	0	✓	✓	
C14	disparate_impact	$Pr(\hat{Y} = 1\|D = unprivileged)/Pr(\hat{Y} = 1\|D = privileged)$	1	✓	✓	✓
C15	statistical_parity_difference	$Pr(\hat{Y} = 1\|D = unprivileged) - Pr(\hat{Y} = 1\|D = privileged)$	0	✓	✓	✓
C16	generalized_entropy_index	$\frac{1}{n\alpha(\alpha-1)} \sum_{i=1}^{n} [(\frac{b_i}{\mu})^{\alpha} - 1]$ where $b_i = \hat{y_i} - y_i + 1$	0	✓		
C17	between_all_groups_generalized _entropy_index	generalized_entropy_index between all groups	0	✓		
C18	between_group_generalized _entropy_index	generalized_entropy_index between privileged and unprivileged groups	0	✓		
C19	theil_index	$\frac{1}{n} \sum_{i=1}^{n} \frac{b_i}{\mu} \ln \frac{b_i}{\mu}$	0	✓		
C20	coefficient_of_variation	$2 * \sqrt{generalized\_entropy\_index}$	0	✓		
C21	between_group_theil_index	theil_index between privileged and unprivileged groups	0	✓		
C22	between_group_coefficient _of_variation	coefficient_of_variation privileged and unprivileged groups	0	✓		
C23	between_all_groups_theil _index	theil_index between all groups	0	✓		
C24	between_all_groups_coefficient _of_variation	coefficient_of_variation between all groups	0	✓		
C25	differential_fairness_bias _amplification	Smoothed EDF between the classifier and the original dataset	0	✓		
Dataset Metrics						
D0	consistency	$1 - \frac{1}{n * n\_neighbors} \sum_{i-1}^{n} \|\hat{y_i} - \sum_{j \in N_{nneighbors}(x_i)} \hat{j_y}\|$	1	✓		
D1	smoothed_empirical _differential_fairness	Smoothed EDF	0	✓		
D2	mean_difference	$Pr(\hat{Y} = 1\|D = unprivileged) - Pr(\hat{Y} = 1\|D = privileged)$	0	✓		
D3	disparate_impact	$Pr(\hat{Y} = 1\|D = unprivileged)/Pr(\hat{Y} = 1\|D = privileged)$	1	✓		

Definitions are collected from IBM AIF360 [23], Fairkit-learn [59], and Fairlearn [18]. For definitions of the terms used here, see Table 3.

influential groups in our society might mandate systems that (deliberately or unintentionally) disadvantage sub-groups within that society. An algorithm might satisfy all the metrics of Table 1 and still perpetuate social inequities. For example,

- Its license feeds might be so expensive that only a small monitory of organizations can boast they are "fair";
- The skills required to use a model's API might be so elaborate that only an elite group of programmers can use it, even if the model is fair.

Gebru et al. [19, 30] argue that inequities arise from the core incentives that drive the organizations building an AI model, e.g., tools funded by the Defence Department tend to support damage to property or life. She argues, "There needs to be regulation that specifically says that corporations need to show that their technologies are not harmful before deploying them." In terms of her work, this article addresses the technical issue of how to measure "harm." As shown in Table 1, there are dozens of ways we might call software "biased" (and, hence, harmful). However, we can also show that many measures are relatively uninformative. Hence, if some organization wishes to

follow the recommendations of Gebru et al., then with the methods of this article, they can make their case of "harmless" via a smaller and simpler report.

Second, Table 1 lists dozens of metrics currently seen in the SE fairness testing literature. This article makes an *empirical argument* that this list is too long, since many of these metrics offer similar conclusions. One alternative to our *empirical argument* is an *analytical argument* that, for example, metric X is equivalent to metric Y. Later in this article (see Section 5.1), we make the case that to reduce the space of metrics to be explored, that kind of analytical argument may be misleading.

Third, to be clear, while we can reduce dozens of metrics down to 10, there will still be issues of how to trade off within this reduced set. That said, we assert our work is valuable, since debating the merits of, say, 10 metrics is a far more straightforward task than trying to resolve all the conflicts between 30. Further, and more importantly, our methods could be used as a litmus test to prune away spurious new metrics that merely report old ideas but in a different way.

Fourth, even after our mitigation algorithms, some fairness metrics still can contradict each other regarding the presence of bias. Hence, in Section 5.3, we offer an extensive discussion on what to do in that situation.

2 BACKGROUND

2.1 The Problem of Algorithmic Fairness

As software developers, we cannot turn a blind eye to the detrimental social effects of our software. While no single paper can hope to fix all social inequities, this article shows how to reduce the complexity involved in assessing one particular kind of unfairness (algorithmic decision-making bias). There is much evidence of ML software showing discriminatory behavior. For example, language processing tools are more accurate in English written by Anglo-Saxons than written by people of other races [28]. An Amazon hiring tool was biased against women [11]. YouTube makes more mistakes while generating closed captions for videos with female voices than males [68, 81]. A popular risk-score predicting algorithm was found to be heavily biased against African Americans, showing a higher error rate while predicting future criminals [8]. Gender bias is also prevalent in Google [31] and Bing [59] translators.

Due to so many undesirable events, academic researchers and big industries have started giving immense importance to ML software fairness. Microsoft has launched ethical principles of AI where "fairness" has been given the topmost priority [16]. IBM has built a toolkit called AI Fairness 360 [23] containing the most notable works in the fairness domain. The software engineering research community has also started exploring this topic in recent years. ICSE'18 held a special workshop for "software fairness" [13]. ASE'19 held another workshop called EXPLAIN, where fairness and explainability of ML models were discussed [15]. Johnson et al. have created a public GitHub repository for data scientists to evaluate ML models based on quality and fairness metrics simultaneously [59].

As to technology developed to detect and fix these issues of fairness, we can see three groups: *fairness testing*, *model-based mitigation*, and *fairness metrics*.

Fairness Testing: The idea here is to generate discriminatory test cases and find whether the model shows discrimination. The first work on this was THEMIS, done by Galhotra et al. [54]. THEMIS generates test cases by randomly perturbing attributes. AEQUITAS [83] improves the way of test case generation to become more efficient. Aggarwal et al. combined local explanation and symbolic execution to generate a better black-box testing strategy [20].

Model Bias Mitigation: Three techniques are used to remove bias from model behavior. The first one is "pre-processing," where bias is removed from training data before model training. Some popular prior work includes optimized pre-processing [32], Fair-SMOTE [37], and reweighing [62].

The second one is "in-processing," where the trained model is optimized for fairness after model training. Popular prior work includes prejudice remover regularizer [65] and meta fair classifier [33]. The last one is "post-processing," where model output is changed to remove discrimination while making predictions. Some noted works include reject option classification [64] and calibration [72]. Some work combines two or more of these techniques, such as Fairway [38], a combination of "pre-processing" and "in-processing."

While fairness testing and model bias mitigation are essential areas, we note that *before* we can declare success in those two areas, we *first* need some way to measure that success.

Accordingly, this article focuses on the third area called:

Fairness Metrics: Early work in this area was done by Verma et al. [86] who divided 20 fairness metrics into five groups based on the theoretical definitions. They say in their paper that although statistical definitions of fairness metrics are easy to measure, they are often insufficient, and it is often unclear how metrics will perform when applied to real data. They also said that these theoretical definitions could be biased given the implicit biases of the expert. Hinnefeld et al. made a comparative empirical analysis of six fairness metrics [57] on one dataset (with artificially introduced bias in the dataset). They showed that not all metrics similarly distinguish bias, and the sensitivity of metrics differs from metric to metric and is dependent on types of bias. Wang et al. did a user study to find a relation between fairness metrics and human judgments [88]. There are also some papers coming from the industry on the topic. LinkedIn has created a toolkit called LiFT for scalable computation of fairness metrics as part of large ML systems [85]. Recently, Amazon internally published an empirical study based on 18 fairness metrics [49].

The above work has now generated a plethora of metrics—so many that we are left to speculate about overlaps and redundancies in all those different measures. Hence, in this article, we check if we can simplify the current space of metrics by performing an empirical analysis of fairness metrics that includes verifying their fairness agreement, grouping, and sensitivity.

2.2 Metrics Used in This Study

In our work, we collected all the metric definitions from the IBM AI Fairness 360 GitHub repository. Table 1 lists the metrics studied in this article. The *Fairkit* and *Fairlearn* columns in Table 1 show the metrics that are common among the IBM AIF360 metrics and metrics from Fairkit [59] (16 of 16 available metrics) and Fairlearn [18] (7 of 16 metrics) toolkit.

Before explaining fairness metrics, we need to understand some terminology. Table 2 contains seven binary classification datasets. The binary outcomes are *favorable* if it gives an advantage to the receiver (i.e., being hired for a job, getting a credit card approved). Each of these datasets has at least one *protected attribute* that divides the population into two groups (*privileged & unprivileged*) that have differences in terms of benefits received. "sex," "race," and "age" are examples of protected attributes. The goal of group fairness is that privileged and unprivileged groups will be treated similarly based on the protected attribute. In contrast, individual fairness tries to provide similar outcomes to similar individuals.

A *fairness metric* quantifies unwanted bias in training data or models. Table 1 shows a sample of such metrics. When selecting these particular metrics, we skipped over the following:

- Metrics for which we could not access precise definitions and implementations in IBM AIF360 toolkit [23];
- Metrics for which we could not find publications to use as baselines in this article.

These two selection rules resulted in the 30 metrics of Table 1, which divide as follows:

Classification Metrics: These measure fairness based on classification results and are labeled in Table 1 using a *Metric Id* beginning with *C*. Two inputs are needed to measure this: The first

Table 2. Details of the Datasets Used in This Research

Dataset	#Rows	#Features	Protected Attribute		Class Label	
			Privileged	Unprivileged	Favorable	Unfavorable
Adult Census Income [2]	48,842	14	Sex-Male Race-White	Sex-Female Race-Non-white	High Income	Low Income
Compas [7]	7,214	28	Sex-Male Race-Caucasian	Sex-Female Race-Not Caucasian	Did not reoffend	Reoffended
German Credit [3]	1,000	20	Sex-Male	Sex-Female	Good Credit	Bad Credit
Heart Health [4]	297	14	Age-Young	Age-Old	Not Disease	Disease
Bank Marketing [9]	45,211	16	Age-Old	Age-Young	Term Deposit - Yes	Term Deposit - No
Student Performance [6]	1,044	33	Sex-Male	Sex-Female	Good Grade	Bad Grade
Titanic ML [10]	891	10	Sex-Male	Sex-Female	Survived	Not Survived

Table 3. Mathematical Definition of Various Confusion Matrix-based Rates

	Actual Positive	Actual Negative
Predicted Positive	TP PPV = TP/(TP+FP) TPR = TP/(TP+FN)	FP FDR = FP/(TP+FP) FPR = FP/(FP+TN)
Predicted Negative	FN FOR = FN/(TN+FN) FNR = FN/(TP+FN)	TN NPV = TN/(TN+FN) TNR = TN/(TN+FP)

These are used to calculate fairness metrics defined in Table 1.

one is the original dataset with true labels, and the second one is the predicted dataset. In the case of binary classification, classification metrics can be calculated from the confusion matrix. Table 3 shows a combined confusion matrix where every cell is divided based on the protected attribute.

Dataset Metrics: While classification metrics relate to predictions made from models, *dataset metrics* discuss learner-independent properties of the data. These are labeled in Table 1 using a *Metric Id* beginning with *D*. Only one input is needed to compute this: the original dataset or transformed (by some bias mitigation algorithm) dataset. It can be applied for both *group and individual fairness*.

Distortion Metrics: For completeness, we note that AIF360 includes a third set of metrics called *distortion metrics*. While these metrics are not seen extensively in the current literature, they would be a worthy target for future research.

In Table 1, each metric has an *ideal value* representing the best-case scenario. This means that at an ideal value, according to the metric privileged and unprivileged groups are treated equally. For most metrics, the ideal value is zero, while in some cases where the metric is a ratio, the ideal value is one. If the ideal value for a metric is zero, then a positive value denotes an advantage for the unprivileged group, while a negative value denotes an advantage for the privileged group. However, if the ideal value for a metric is one, then a value <1 denotes an advantage for the privileged group, and >1 denotes an advantage for the unprivileged group.

To use these metrics, some threshold must be applied to report "fair" or "unfair":

- For metrics with ideal value 0: The IBM AIF360 toolkit [23] uses the following definition of "fair": ranges between −0.1 to 0.1 as "fair" (so "unfair" means values outside that range).
- For metrics with ideal value 1: The IBM AIF360 toolkit [23] uses the following definition of "fair": ranges between 0.8 to 1.2 as "fair" (so "unfair" means values outside that range).

3 METHODS

All our research questions use data collected from the methods described in this section. We first describe the experimental methodology followed by detailed descriptions of the component used.

3.1 Experimental Setup

We summarize our experimental setup as follows.

3.1.1 Data Pre-processing. Three different pre-processing steps are performed before using the data [53, 69, 77] for model building. At first, each categorical value in the dataset is converted either using a label encoder or one hot encoder, as most ML algorithms cannot handle categorical values directly. Then the protected attributes are changed into ones and zeros from their original values. Here we denote the privileged attribute as one and the unprivileged as zero. Finally, we use min-max normalization in the datasets to normalize the data before building the models.

3.1.2 Model Training. We used fivefold cross-validation repeated 20 times with random seeds build training/test sets (as recommended in References [63, 77, 84, 86]). This step divides the data into multiple subsets of data with various degrees of bias. We train three models in each iteration. (a) *Baseline model:* Here we use the training data to build a logistic regression model. (b) *Reweighing model:* Here we first train the reweighing method and then use the learned model to transform the training data to achieve group fairness. Using the transformed data, we train a logistic regression from scikit-learn with "l2" regularization, "lbfgs" solver, and maximum iteration of 1,000. (c) *Meta Fair Classifier model:* Here to train the meta fair classifier model, we use the training data to build multiple meta fair classifier model with different values of τ (a hyperparameter for fairness penalty in the model) and measure the bias in the model using the validation set. Then to build the final model, we select the τ for which the model had the lowest bias in the validation set and build the final meta fair classifier model.

3.1.3 Fairness Metric Calculation. We collect the performance of each model based on 26 classification and four dataset metrics for each iteration of the cross-validation. So for each iteration, we use the test data for prediction, and then the predicted values, along with the ground truth, are used for calculating the 26 classification metrics. Similarly, we collect the four dataset metrics on the baseline and reweighing method. The meta fair classifier is not applicable in the case of dataset metrics.

3.1.4 Measure for Fairness. Data Pre-processing, Model Training, and Fairness Metric Calculation steps are performed for each of the seven datasets with fivefold repeat cross-validation. Then, to measure if the model built on a dataset is fair or unfair according to a metric, we selected a threshold for each metric. As mentioned in Section 2.2, that threshold is the *fair range*. If a metric value falls in that range, then we say it "fair" otherwise "unfair."

3.1.5 Building Clusters. One of the main goals of this study is to group a set of metrics together that perform similarly and measure similar kinds of bias. We use 26 classification metrics calculated on seven datasets with three different methods to calculate metric-to-metric correlation based on the Spearman rank correlation coefficient. We do the same for the four dataset metrics as well. This provides us with two correlation matrices: one 26×26 and one 4×4. After that, to build the clusters using agglomerative clustering, we convert the similarity matrix into a dissimilarity matrix [46, 58] using Equation (1). We use this dissimilarity matrix to create the clusters. The agglomerative clustering process creates a dendrogram, as shown in Figure 1. To select the number of clusters, we cut the dendrogram at a height where the clusters will remain unchanged with the most increase/decrease of the cutting threshold. For classification metrics, we cut the dendrogram

(Figure 1) at 0.57 as the clusters will remain unchanged between the cutoff values 0.49 and 0.64. Finally, we get the clusters containing classification metrics measuring similar kinds of bias. We perform the same process for dataset metrics and cut the dendrogram at the height of 0.4,

$$d(x, y) = 1 - |sim(x, y)|. \tag{1}$$

3.1.6 Calculating Sensitivity. Research question four asks about the consistency of the metric values for three cases: (a) raw data, (b) after applying **Reweighing (RW)**, and (c) after applying **Meta Fair Classifier (MFC)**. As we are using fivefold cross-validation with 20 repeats for all the datasets, we get 100 results for each dataset and report for all seven datasets:

- the median value: the 50th percentile (or Q_2);
- the IQR: the (75–25)th percentile (or $Q_3 - Q_1$)

3.2 Models

This article analyzes the 30 fairness metrics in Table 1 using the seven datasets described in Table 2. In that work, we use one baseline model and two models tuned by pre-processing and in-processing algorithms to generate predictions that will be used for measuring the bias based on the 30 different fairness metrics. We decided to use one baseline model (logistic regression) to have a model that shows bias in different fairness measures, one pre-processing and one in-processing model to remove that bias from the model based on two different types of techniques. We decided not to use post-processing algorithms, as these do not create an unbiased model; instead, they modify the output of the biased model to make the models fairer. We will use those metric values based on the three models to answer all four research questions in this article:

- **Baseline:** We used a logistic regression model for creating baseline results. Logistic regression is widely used in the fairness domain as baseline model [32, 38, 40, 41, 65]. We will be using this model in all four research questions as the baseline model to create clusters, check fairness agreement between metrics, or to identify the sensitivity of metrics. We used scikit-learn implementation with "l2" regularization (which helps to prevent over-fitting), "lbfgs" solver (which is a quasi-Newton optimization algorithm), and maximum iteration of 1,000 (although the default value is 100 for scikit-learn logistic regression, we used 1,000 as we tuned the model for convergence).
- **Reweighing:** This is a widely used [21, 23, 36, 60, 75] pre-processing method proposed by Kamiran et al. [62]. Here, before model training, examples in each group and label are given different weights to ensure fairness. We use this method to build the clusters, identify if some fairness measures are more sensitive to changes than others, and verify if we can achieve fairness for a model based on all metrics when the model is built using Reweighing.
- **Meta Fair Classifier:** This is an in-processing method proposed by Celis et al. [33], which is a widely used meta-algorithm in the fairness research community [24, 34, 55, 71]. The optimization algorithm is developed to improve 11 fairness metrics with minimal loss in accuracy. Like Reweighing, we use this method to build the clusters, identify if some fairness measures are more sensitive to changes than others, and verify if we can achieve fairness for a model based on all metrics when the model is built using Meta Fair Classifier.

The last two bias mitigation algorithm implementations are taken from IBM AIF360 [23].

3.3 Agglomerative Clustering

Our metrics selection strategy requires a clustering algorithm. Two classes of such clustering algorithms are (a) partitioning clustering and (b) hierarchical clustering. Here we are grouping

fairness metrics based on similarity, not on distance, and we have no prior idea about the number of clusters. Thus, in this case, the ideal choice is *hierarchical clustering*. Agglomerative clustering [5] is a hierarchical bottom-up clustering approach that is widely used in the ML community [22, 46–48, 51, 70, 74, 79, 90]. In this approach, the closest pairs of items are grouped. The closest of these groups are then grouped into a higher-level group. This repeats until everything falls into one group. We used the agglomerative clustering method provided by scikit-learn with the ward linkage method. Instead of measuring the distance directly, this method analyzes the variance of clusters. The ward linkage method is based on merging clusters that minimize the increase in sum-of-square errors; thus, clusters will only be merged when they are of similar type. To achieve this, we used the average pairwise dissimilarity between objects in two different clusters as linkage criteria between groups. This process creates a dendrogram, a hierarchical structure of the groups/clusters obtained by between-cluster distance or dissimilarity. From this tree of groupings, we use the within-cluster similarity from the dendrogram and use elbow method [1, 44, 82] to select the number of clusters to be formed. We extract the clusters at the largest change in dissimilarity (which is similar to Sum of Squared Error).

3.4 Spearman Rank Correlation

To build the clusters and dendrograms, we measure the similarity of the two metrics. In this article, by "similarity" we mean they measure the similar bias in the models/dataset. Similar metrics will show a similar pattern of changes in bias when models are built using different parts of the data or different bias removal algorithms. To compute this similarity, we sample from our model training procedure (see Section 3.1.2) that computes our metrics 100 times, using different train/validation/test samples of the data. Next, for each dataset, for those 100 numbers, we use correlation to assess similarity.

Two widely used definitions of correlation [42, 46–48, 58, 73, 79, 90] are the (a) Pearson correlation (which evaluates the linear relationship between two continuous variables) and the (b) Spearman rank correlation (which is a non-parametric measure of rank correlation that evaluates the monotonic relationship between two continuous or ordinal variables). We choose Spearman rank correlation, as it measures the monotonic relationship between two variables and is less affected by outliers.

4 RESULTS

Our results are organized based on four research questions.

RQ1: Do current fairness metrics agree with each other?

First, we need to verify our motivation. In real life, do the fairness metrics contradict? Table 4 contains results for 26 classification metrics; Table 5 contains results for four dataset metrics. The learner here is logistic regression. The last row contains the percentage of metrics marking the specific dataset as unfair in both tables. If we combine last rows of Tables 4 and 5 and sort them in ascending order, then we get the following list:

$$\{23, 34, 50, 50, 50, 54, 58, 65, 75, 75, 75, 75, 77, 100\}\%.$$

The median value here is 62%; i.e., nearly half the time, the metrics make *different* conclusions about the *same* data. This means that researchers and practitioners will be spending much effort trying to understand their systems using disagreeing oracles (a result that motivates this entire article).

Table 4. Cluster-based Results for 26 Classification Metrics on Seven Datasets for Models Trained on All Three Models

Cluster Id	MID	Metrics	Datasets							Metric Type
			Adult	Compas	German	Health	Bank	Student	Titanic	
0	C3	false_omission_rate_difference	Unfair	Fair	Fair	Unfair	Fair	Fair	Unfair	Mis-classification
0	C7	false_omission_rate_ratio	Unfair	Fair	Fair	Unfair	Fair	Unfair	Unfair	
0	C11	error_rate_difference	Unfair	Fair	Fair	Unfair	Fair	Fair	Fair	
0	C12	error_rate_ratio	Unfair	Fair	Fair	Unfair	Fair	Fair	Fair	
		fair/Unfair ratio	**0/4**	**4/0**	**4/0**	**0/4**	**4/0**	**3/1**	**2/2**	
		Percentage of agreement	**100%**	**100%**	**100%**	**100%**	**100%**	**75%**	**50%**	
1	C10	average_abs_odds_difference	Unfair	Unfair	Unfair	Unfair	Unfair	Fair	Unfair	Differential Fairness
1	C25	differential_fairness_bias_amplification	Unfair	Unfair	Unfair	Unfair	Unfair	Fair	Unfair	
		fair/Unfair ratio	**0/2**	**0/2**	**0/2**	**0/2**	**0/2**	**2/0**	**0/2**	
		Percentage of agreement	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	
2	C16	generalized_entropy_index	Fair	Unfair	Unfair	Fair	Fair	Fair	Unfair	Individual Fairness
2	C19	theil_index	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	
2	C20	coefficient_of_variation	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	
		fair/Unfair ratio	**1/2**	**0/3**	**0/3**	**1/2**	**1/2**	**1/2**	**0/3**	
		Percentage of agreement	**67%**	**100%**	**100%**	**67%**	**67%**	**67%**	**100%**	
3	C4	false_discovery_rate_difference	Fair	Fair	Fair	Fair	Fair	Fair	Unfair	Mis-classification
3	C8	false_discovery_rate_ratio	Fair	Fair	Fair	Fair	Fair	Unfair	Unfair	
		fair/Unfair ratio	**2/0**	**2/0**	**2/0**	**2/0**	**2/0**	**1/1**	**0/2**	
		Percentage of agreement	**100%**	**100%**	**100%**	**100%**	**100%**	**50%**	**100%**	
4	C0	true_positive_rate_difference	Unfair	Unfair	Fair	Unfair	Unfair	Fair	Unfair	Confusion Matrix Based Group Fairness
4	C1	false_positive_rate_difference	Fair	Unfair	Unfair	Unfair	Unfair	Fair	Unfair	
4	C2	false_negative_rate_difference	Unfair	Unfair	Unfair	Unfair	Unfair	Fair	Unfair	
4	C5	false_positive_rate_ratio	Fair	Unfair	Unfair	Unfair	Unfair	Fair	Unfair	
4	C6	false_negative_rate_ratio	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	
4	C9	average_odds_difference	Unfair	Unfair	Unfair	Unfair	Unfair	Fair	Unfair	
4	C14	disparate_impact	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	
4	C15	statistical_parity_difference	Unfair	Unfair	Unfair	Unfair	Unfair	Fair	Unfair	
		fair/Unfair ratio	**2/6**	**0/8**	**1/7**	**0/8**	**0/8**	**6/2**	**0/8**	
		Percentage of agreement	**75%**	**100%**	**88%**	**100%**	**100%**	**75%**	**100%**	
5	C17	between_all_groups_generalized_entropy_index	Fair	Fair	Fair	Fair	Fair	Fair	Fair	Between Group Individual Fairness
5	C18	between_group_generalized_entropy_index	Fair	Fair	Fair	Fair	Fair	Fair	Fair	
5	C21	between_group_theil_index	Fair	Fair	Fair	Fair	Fair	Fair	Fair	
5	C22	between_group_coefficient_of_variation	Fair	Fair	Fair	Fair	Fair	Fair	Fair	
5	C23	between_all_groups_theil_index	Fair	Fair	Fair	Fair	Fair	Fair	Fair	
5	C24	between_all_groups_coefficient_of_variation	Fair	Fair	Fair	Fair	Fair	Fair	Unfair	
		fair/Unfair ratio	**6/0**	**6/0**	**6/0**	**6/0**	**6/0**	**6/0**	**5/1**	
		Percentage of agreement	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**83%**	
6	C13	selection_rate	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	Intermediate Metric
		fair/Unfair ratio	**0/1**	**0/1**	**0/1**	**0/1**	**0/1**	**0/1**	**0/1**	
		Percentage of agreement	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	
Percentage of metrics marking dataset as unfair			**58%**	**54%**	**50%**	**65%**	**50%**	**27%**	**73%**	

For a metric with ideal an value of zero, anything below −0.1 and above 0.1 is "unfair." For a metric with an ideal value of one, anything <0.8 or >1.2 is "unfair."

Table 5. Cluster-based Results for Four Dataset Metrics on Seven Datasets for Models Trained on Logistic Regression

Cluster Id	MID	Metrics	Datasets							Metric Type
			Adult	Compas	German	Health	Bank	Student	Titanic	
0	D0	consistency	Fair	Unfair	Fair	Unfair	Fair	Unfair	Fair	Individual Fairness
1	D1	smoothed_empirical_differential_fairness	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	Unfair	Differential Fairness
2	D2	mean_difference	Unfair	Unfair	Unfair	Fair	Unfair	Fair	Unfair	Confusion Matrix Based Group Fairness
2	D3	disparate_impact	Unfair	Unfair	Unfair	Fair	Unfair	Fair	Unfair	
Percentage of metrics marking dataset as unfair			**75%**	**100%**	**75%**	**50%**	**75%**	**50%**	**75%**	

For a metric with ideal value of zero, anything below −0.1 and above 0.1 is "unfair." For a metric with ideal value of one, anything <0.8 or >1.2 is "unfair."

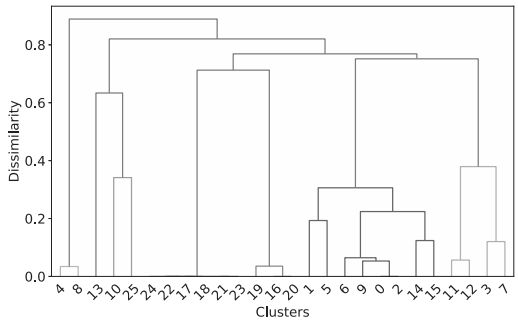

Fig. 1. Agglomerative clustering of classification metrics (using Spearman rank correlation). Here x-axis shows the classification metric Ids from Table 1 and y-axis shows the dissimilarity measure between clusters.

RQ2: Can we group (cluster) fairness metrics based on similarity?

Figure 1 shows the dendrogram created for the classification metrics using the method described in Section 3.4. Based on this dendrogram, using the agglomerative clustering process, we created seven clusters from the 26 classification metrics, as can be seen in Table 4. Table 5 shows that four dataset metrics can be divided into three clusters using a similar process. These clusters are formed using the spearman correlation results from all three models with fivefold cross-validation repeated 20 times with random seed. More importantly, we note that

- RQ1 reported intra-project disagreement on "fair"-vs-"unfair";
- We note that there is much intra-cluster agreement for each dataset in Tables 4 and 5.

As evidence, we note that the majority fairness decision is always the same within the clusters for each dataset. In Table 4, the row *Percentage of agreement* comments on the uniformity of decisions within each cluster (for each dataset). Note that uniformity is very high (often 100%). That means metrics inside each cluster agree with each other for every dataset. Among the seven clusters, we see six clusters (except cluster two) show 100% agreement considering the median value across seven datasets. For example, in the case of cluster zero, the percentage of agreement is 100% for five datasets, 75% for one, and 50% for one. The majority is 100%. That is true for clusters 1, 3, 4, 5, 6, and 7. We see similar agreement pattern inside clusters in Table 5 also.

For reference purposes, the last column of Tables 4 and 5 offers names for those clusters:

- **Misclassification (cluster 0, 3)**: These metrics try to measure the difference or ratio of misclassification errors between groups;
- **Differential fairness (cluster 1)**: These metrics try to measure if probabilities of the outcomes are similar regardless of the combination of protected attributes [52];
- **Individual Fairness (cluster 2)**: It measures if two similar individuals with respect to the classification task receive the same outcome or not;
- **Confusion matrix based group fairness (cluster 4)**: These metrics measure difference or ratio between groups based on confusion matrix;
- **Between group individual fairness (cluster 5)**: Measures the difference or ratio of individual fairness between groups;
- **Intermediate metrics (cluster 6)**: These are intermediate metrics.

Table 6. This Table Shows Sensitivity of the Classification Metrics on the Three Different Models Used in This Study: (a) Baseline, (b) RW, and (c) MFC

	Compas						Health						German					
MID	Baseline		RW		MFC		Baseline		RW		MFC		Baseline		RW		MFC	
	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR
C3	−0.077	0.081	−0.133	0.016	−0.058	0.041	−0.14	0.032	−0.211	0.062	−0.142	0.137	0	0.49	−0.503	0.676	0	0.587
C7	0.617	0.219	0.682	0.031	0.784	0.102	0.357	0.094	0.158	0.348	0.363	0.346	2.32	0.712	0.002	0.45	1.003	0.524
C11	−0.033	0.062	−0.023	0.032	−0.047	0.032	−0.117	0.01	−0.133	0.014	−0.089	0.121	0.061	0.069	0.059	0.102	0.047	0.056
C12	0.972	0.118	0.881	0.122	0.887	0.062	0.488	0.214	0.339	0.031	0.441	0.512	1.149	0.273	1.172	0.409	1.161	0.225
C10	0.292	0.052	0.03	0.023	0.181	0.042	0.141	0.094	0.106	0.076	0.161	0.062	0.224	0.163	0.045	0.048	0.031	0.119
C25	0.561	0.384	−0.223	0.128	0.361	0.153	0.173	0.25	−0.094	0.392	0.121	0.431	2.402	3.291	1.159	0.445	1.498	2.107
C16	0.209	0.001	0.185	0.012	0.183	0.007	0.103	0.004	0.087	0.014	0.089	0.027	0.073	0.016	0.069	0.021	0.057	0.014
C19	0.262	0.002	0.253	0.017	0.269	0.008	0.137	0.023	0.142	0.049	0.139	0.034	0.083	0.019	0.071	0.019	0.059	0.011
C20	0.908	0.001	0.872	0.037	0.876	0.018	0.592	0.009	0.589	0.061	0.598	0.079	0.561	0.037	0.532	0.039	0.483	0.041
C4	0.044	0.019	0.138	0.057	0.042	0.061	−0.092	0.123	−0.007	0.192	−0.018	0.149	0.061	0.129	0.059	0.109	0.052	0.063
C8	1.009	0.062	1.376	0.203	1.103	0.172	0	0.937	0.898	1.521	0.934	1.287	2.543	0.537	1.173	0.462	1.147	0.213
C0	−0.263	0.102	−0.004	0.102	−0.198	0.052	−0.116	0.119	0.132	0.237	−0.102	0.412	−0.077	0.089	0	0.036	−0.018	0.059
C1	−0.136	0.053	−0.018	0.026	−0.173	0.037	−0.194	0.058	−0.126	0.183	−0.108	0.129	−0.303	0.243	0	0.029	−0.053	0.176
C2	0.183	0.087	0.005	0.052	0.22	0.052	0.113	0.169	−0.131	0.241	0.118	0.392	0.076	0.083	0	0.038	0.017	0.062
C5	0.378	0.036	0.896	0.069	0.464	0.071	0.002	0.219	0.249	0.584	0.162	0.332	0.691	0.232	1.003	0.029	0.923	0.162
C6	1.631	0.251	1.009	0.128	1.421	0.152	1.397	0.493	0.389	1.283	1.429	2.043	3.387	0.52	0.002	5.529	11.362	3.345
C9	−0.182	0.052	−0.028	0.062	−0.172	0.042	−0.142	0.103	−0.052	0.162	−0.139	0.159	−0.219	0.167	0	0.038	−0.031	0.121
C14	0.472	0.076	0.882	0.143	0.571	0.061	0.238	0.14	0.435	0.282	0.367	0.281	0.842	0.123	1	0.045	0.922	0.112
C15	−0.281	0.053	−0.049	0.059	−0.205	0.031	−0.367	0.079	−0.289	0.169	−0.258	0.179	−0.159	0.121	0	0.043	−0.029	0.110
C17	0.003	0.002	0.001	0.003	0.001	0.002	0.001	0.003	0.002	0.002	0.002	0.001	0.001	0.002	0.002	0.001	0.001	0.001
C18	0.003	0.003	0.002	0.002	0.002	0.001	0.003	0.001	0.004	0.001	0.001	0.002	0.002	0.002	0.002	0.002	0.001	0.003
C21	0.002	0.002	0.002	0.004	0.001	0.001	0.002	0.001	0.002	0.002	0.001	0.001	0.003	0.002	0.003	0.001	0.002	0.001
C22	0.078	0.049	0.054	0.005	0.055	0.018	0.042	0.037	0.015	0.054	0.045	0.03	0.027	0.063	0.031	0.051	0.029	0.038
C23	0.002	0.005	0.001	0	0.002	0.001	0.003	0.003	0.004	0.002	0.001	0.001	0.001	0.002	0.001	0.001	0.004	0.001
C24	0.068	0.049	0.049	0.007	0.061	0.019	0.038	0.037	0.015	0.039	0.045	0.03	0.024	0.065	0.028	0.053	0.036	0.038
C13	0.385	0.019	0.441	0.013	0.413	0.017	0.397	0.05	0.391	0.131	0.411	0.056	0.921	0.015	0.955	0.031	1.001	0.041

The table shows the median and IQR values of three datasets. Here the cells in IQR columns are marked with "red" those that change by more than a small amount (35th percentile of the standard deviation of the IQR values). The insensitive metrics are those that usually have white IQR values.

From a practitioner's viewpoint, this clustering is useful because

- The clustering reduces the confusion of having too many metrics and not knowing their similarity.
- As the metrics inside the same cluster measure the same kind of bias and behave in the same manner, we can choose just one metric from each cluster. Thus we measure a few metrics but can cover a much more comprehensive range of fairness notions.
- If we see agreement among all the metrics inside a cluster for a particular dataset, then one metric can be chosen as representative of the whole cluster.
- In case of intra-cluster conflicts, choosing only one metric can be risky. Practitioners must conduct a proper risk assessment before selecting metrics in these cases. That said, if there is intra-cluster conflict among metrics, then policymakers can choose one from the "fair" group and one from the "unfair" group to mitigate that risk, i.e., if a cluster shows two metrics are "fair" and one "unfair," then we select two metrics from this cluster, one of the metrics that is from "fair group" and select the metric that shows "unfair."

As part of this study, we further analyzed each cluster mathematically to verify if our cluster of metrics and their mathematical definitions coincide. A detailed analysis of these clusters and their mathematical analysis has been discussed in Section 5.1.

RQ3: Are some fairness metrics more sensitive to change than others?

Table 7. This Table Is Similar to Table 6, Showing the Sensitivity of the Dataset Metrics on (a) Baseline and (b) RW

	Compas						Health						German					
MID	Baseline		RW		MFC		Baseline		RW		MFC		Baseline		RW		MFC	
	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR	Med	IQR
D1	0.548	0.023	0.568	0.021	—	—	0.814	0.02	0.803	0.02	—	—	0.636	0.008	0.641	0.007	—	—
D2	0.257	0.043	0	0	—	—	0.867	0.325	0.001	0	—	—	0.301	0.095	0.002	0	—	—
D3	−0.112	0.016	0	0	—	—	−0.325	0.058	0	0	—	—	−0.087	0.032	0	0	—	—
D4	0.778	0.033	1	0	—	—	0.413	0.128	1	0	—	—	0.856	0.042	1	0	—	—

Table 8. The Number of Classification Metrics That Move toward or away from the Ideal Value when Either Reweighing or Meta Fair Classifier Is Used to Remove Bias in the Models

Dataset	Reweighing (RW)			Meta Fair Classifier (MFC)		
	UF	FU	NC	UF	FU	NC
Adult	13	13	0	11	15	0
Compas	15	7	4	16	6	4
Health	17	5	4	17	7	1
German	19	6	1	19	7	0
bank	16	6	4	15	7	4
Titanic	11	15	0	17	9	0
Student	15	7	4	12	10	4

Here "UF" shows the number of metrics that moved toward the ideal metric value, while "FU" shows the opposite. Finally, "NC" shows the number of metrics that did not change at all.

An ideal metric is responsive to the dataset it examines. An "insensitive" metric is one that delivers the same conclusions, no matter what data are being examined. An "insensitive" cluster is one containing mostly insensitive metrics. Such insensitive clusters could be ignored, since they are not informative.

We measure sensitivity by looking at the variability of our metrics scores using the intra-quartile range (IQR=$Q_3 - Q1$). For each dataset, we found the IQR across all clusters. Next, we highlight the sensitive results; i.e., those with an IQR greater than d^* standard deviation. The remaining unhighlighted results are the insensitive metrics.

As to what value of d to use in this analysis, we take the advice of a widely cited paper by Sawilowsky [76] (this 2009 paper has 1,100 citations). That paper asserts that "small" and "medium" effects can be measured using $d = 0.2$ and $d = 0.5$ (respectively). We will analyze this data by splitting the difference looking for differences larger than $d = (0.5 + 0.2)/2 = 0.35$.

Turning now to Tables 6 and 7, we see that most clusters have highlight IQR results. However, in Table 6, we see the clusters formed by metrics C16, C19, C20 (individual fairness) and C17, C18, C21, C22, C23, C24 (between group individual fairness) are insensitive. This, in turn, means that we should *not* criticize a fairness analysis that ignores these metrics.

RQ4: Can we achieve fairness based on all the metrics at the same time?

Different fairness metrics measure different kinds of bias. If any of the metrics complain about the fairness of the test results, then we cannot trust the model blindly, and it should go through further scrutiny and improvement. Bias mitigation algorithms try to make unfair models fairer. Here we are verifying whether, even after applying bias mitigation algorithms, we can achieve fairness based on all the metrics We have chosen two highly cited algorithms from IBM AIF360: RW by Kamiran et al. [62] and Meta Fair Classifier by Celis et al. [33].

Table 8 shows the results collected for seven datasets after using the RW and MFC algorithms. For every dataset (row-wise), we show the number of metrics changed toward or away from its ideal value. In that table:

- FU denotes the metrics that changed toward ideal value;
- UF denotes the metrics that moved away from the ideal value,
- NC means the metrics that did not change.

Note that majority of the metrics move toward "fair," but there are some metrics that move toward "unfair." For Reweighing, some metrics show "no change," but we have verified they always remain in the *fair range*.

The main takeaway is that it is no longer necessary (or even possible) to satisfy all these fairness metrics. While our analysis can reduce dozens of metrics down to 10, there will still be issues of how to trade off within this reduced set. Even after applying bias mitigation approaches, some metrics still conflict with others. This finding is similar to the claim made by others:

- Berk et al. [25] offer an "Impossibility Theorem" that says there is no way to satisfy all kinds of fairness together.
- As Yuriy Brun said at his keynote at ICSSP'2020 "*we need to work the system in a biased way sometimes*" [29].

In terms of the Brun quote, we would say agree that some biases are necessary (to guide a search), and too many biases mean we cannot make a conclusion (since what satisfies one bias will not satisfy another). We have shown that dozens of seemingly different biases can be resolved to a much smaller set, making subsequent reasoning simpler and more straightforward.

5 DISCUSSION

We have described all of our results. Here we summarize the results comprehensibly to reach a stable conclusion. The main idea of this work is to reduce the complexity of measuring fairness. That said, it is imperative that we narrate our conclusions in a straightforward way. We discuss three major concerns arising from Section 4 and try to simplify fairness measurement to our best.

5.1 Why Not Group Metrics via Their Analytical Structure?

This article has offered an empirical analysis that many of the metrics in Table 4 are synonymous, since, when clustered, they fell together into just a few similar groups. In this section, we check if the same conclusions can be achieved from a more analytical analysis that looks at the structure of the equations for the fairness metrics.

Sometimes, a group generated by the formula's analytical structure is similar to the clusters we generated above. For example:

- In cluster three (from Table 4), all metrics are based on *FDR*, which suggests that both from an empirical and analytical point of view, they should be similar.
- Also, in cluster zero, we see that all those metrics are based on *FOR* and error rate. Intuitively, this seems sensible, since metrics try to measure the amount of misclassification here.

That said, as shown by the following three examples, there are many examples where an equation's analytical structure does *not* predict for its empirical cluster.

- ***EXAMPLE #1:*** If we look at cluster five, then all six metrics inside this cluster are related to "between group individual fairness." This metric is based on the same benefit function:

$$y = \hat{y} - y + 1 \tag{2}$$

(for more details on that, see Table 1 metric id C16.) We note that cluster two is also based on Equation (2), but the metrics inside this cluster represent individual fairness for each group separately. That means

Although all metrics inside cluster two and cluster five are based on the same benefit function, they measure different definitions of fairness.

That is, a formal analysis of the analysis might combine these clusters, whereas a data-oriented empirical analysis would argue for their separation.

- ***EXAMPLE #2:*** In cluster four from Table 1, the metrics C0, C1, C2, C5, C6, and C9 dependent on *TPR*, *FPR*, and *FNR*. Recall that *FPR* and *FNR* report type one and type two errors (misclassification on fairness). Now *TPR* can be expressed as *1 - FPR*, which means the change in *TPR* will mirror changes in *FPR*. In contrast, in this cluster, the other two metrics, C14 and C15, are based on selection rate (ratio of the number of predicted positives and number of instances). Although there is not much similarity in the formula between these two and other metrics in this cluster, we can see they perform similarly when measuring fairness. That is:

An analytical analysis does not always reflect the measurement of fairness in the real-world scenario.

Verma et al. [86] notice a similar phenomenon where they observe that *Equal Predictive parity (a measure they explore) should also have equal* FDR *... but when measured from an empirical point of view, they showed they are not the same.*

- ***EXAMPLE #3:*** In cluster one, metrics C10 and C25 have very different mathematical formulas. C10 is based on *FPR* while C25 is based on smoothed **empirical differential fairness (EDF)**. EDF is calculated based on Dirichlet smoothed base rates for each intersecting group in the dataset, based on the count of predicted positives. Here as well, we see that

Two formulas with a different analytical structure can have a similar performance w.r.t. fairness.

To summarize the above, we quote Alfred Korzybski, who warned:

A map is not the territory.

While the analytical structure of the formula offers intuitions about the nature of fairness, those intuitions had better be checked via empirical analysis.

5.2 Is Our Empirical Analysis Useful?

We have established the requirement of empirical analysis, and we have also done that analysis. We need to determine whether this analysis would be helpful in real-life applications. Here we describe various scenarios of fairness contradiction and how our study helps to remove that.

Imagine a college admission decision scenario where the system might be seen as biased against group B if applicants from group A are accepted more than group B. Here group A and group B are divided based on different values of a protected attribute. The college applies a bias mitigation approach to solve this problem using a group fairness metric by changing group A's or B's scoring threshold. Now if a member of group A is rejected, while a member of group B has been accepted with an equal or lower score, then the system might be seen as biased against that individual. The

main takeaway from this story is that there is a conflict between "individual fairness" and "group fairness" [26].

The concept of fairness is very much application specific and choosing the appropriate metric is the sole responsibility of the policymaker. An ideal scenario will be building a machine learning model that does not show any kind of bias. However, that is too good to be true. Brun et al. found out that if a model is adjusted to be fair based on one protected attribute (e.g., sex), then in some cases model becomes more biased based on another protected attribute (e.g., race) [13]. Kleinberg and other researchers argue that different notions of fairness are incompatible, and hence it is impossible to satisfy all kinds of fairness simultaneously [67]. One thing to remember while making a prediction is that fairness is not the only concern. Prediction performance is the most important goal. Berk et al. found that accuracy and fairness are competing goals [25]. This tradeoff makes the job even more complicated, since damaging model performance while making it fair may be unacceptable.

As researchers, we know that satisfying all kinds of fairness together is not possible. A policymaker has to choose which fairness definitions are most important for a particular domain and ignore the rest. Our work of dividing fairness tries to make the choice easier, as choosing metrics from a group of 10 options is much simpler than choosing from 30 options. Using our results of Tables 4 and 5, in a specific domain, if group fairness is more important than individual fairness, then cluster four will be given more priority than clusters two and five (Table 4). Once a cluster is given priority, one or two metrics can be chosen to represent the whole cluster. That means our whole work boils down to minimizing the number of metrics to look at and covering a wide range of fairness. We believe future researchers and industry practitioners will use our work as a guide and that will be the fulfillment of this study.

5.3 What to Do When the Metrics Contradict Each Other?

We have seen that there are scenarios where fairness metrics contradict each other. According to some metrics, the prediction is fair, whereas others disagree. Fairness metrics find out how critical the errors of a prediction model are. It is the decision of the policymaker or the domain expert to choose appropriate fairness metrics based on what kind of bias is more important for the specific domain. For example, consider scenarios where models of health outcomes or student performance have been built, which are unfair to certain protected groups (e.g., student progress models can be unnecessarily biased by zip code of the student; patient health outcomes can be unnecessarily biased by the income of the patient):

- Suppose we are predicting whether a patient has cancer or not, depending on the symptoms. Here predicting a benign case as malignant is not very dangerous but predicting a malignant case as benign is extremely dangerous. A wrong diagnosis for an actual cancer patient will delay the treatment, and the patient may die. That means *false negative* is more important here.
- Suppose we are predicting future performance of a student based on previous records. Here, if we predict a good student as bad, then that is not that fatal. However, if a student who needs special attention and help from teachers is given a good rating, then that student will be miserable. That means *false positive* is more important here.

Now, based on the metric clusters defined in Table 4 and their definitions, we can say that if researchers just care about bias in misclassification, that is, if the difference in performance between two groups based on one protected attribute when the wrong classification is being made, then researchers should focus more attention to cluster 0 and 3. While if they care about both correct and misclassification between two groups based on one protected attribute, then researchers should

focus on cluster 4. Now researchers need to focus on cluster 1 if multiple protected attributes are being used in the model, and the model needs to be fair based on the inter-sectional AI fairness criterion; that is, the bias in the model is not based on each protected attribute separately but based on the combination of all. Now, if individual fairness is essential in a system, that is, similar individuals are treated similarly regardless of their protected attributes, then researchers should focus on metrics from cluster 2. Finally, researchers should focus on cluster 5 if the system needs to be fair for each group based on a protected attribute when the fairness is measured for each individual.

Now that we know which metrics look at what kind of bias, it will be easier for the decision-maker to choose. That said, based on the guidance we have provided, one metric over another will be given priority in case of contradiction among metrics.

6 THREATS TO VALIDITY AND FUTURE WORK

This article explores machine learning methods for software engineering. One issue with any paper like this is a few selection and evaluation biases along with construct and external validity based on the choice of models, datasets, and methods. In the future, we plan to address the apparent threats to validity that this article has not fully addressed.

Construct Validity: Here we have used popular *hierarchical* clustering called *agglomerative* approach, as the number of clusters was not known beforehand. In the future, we need to experiment with other clustering techniques to check for conclusion stability. This analysis used **logistic regression (LR)**, as much prior work on fairness has also used LR [23, 38].

This research also does not explore **hyperparameter optimization (HPO)** as part of fine-tuning the models. This is an important point, since some analyses may be biased because of poorly tuned classification models. That said, we argue that the results of this study are still valid, despite the lack of HPO, for the following two reasons:

- It is not correct to characterize the models used here as "poorly tuned classification models." We say this since this is not the first time we have analyzed this data. We have generated predictions from these modes in prior publications [39, 41]. There, the observed precision, recall, and false alarms were healthy (false alarms usually under 20%, never more than a third; previsions and recalls often over 70 and never less than two-thirds).
- It is not easy to see how HPO could be applied in this context. If HPO was applied to the metrics of Table 1, then some different results might arise. But given current limitations in optimization technology, we would doubt the legitimacy of that study. There are 30+ listed in Table 4. Given our current optimization technology, we cannot tune for 30+ goal problems–which means to use HPO in this study, we would have to tune for some small subset of the total set. This seems (at least to us) to be a somewhat perverse experiment, since it would not result in an "apples-to-apples" comparison.

In the future, we plan to address the apparent threats to validity that this article has not fully addressed. Also, in other future work, we need to explore some other classification models, including DL models. Also, the metric clusters found in Tables 4 and 5 are created using the results of our choice ML models, dissimilarity measures, and cutting point in the dendrogram. Thus, choosing one metric from each cluster may contain some risk, and researchers must be careful while making informed choices about metric selection.

Evaluation Bias: We have used 30 metrics taken from IBM AIF360 [23]. We have also covered most of the metrics from Fairkit-learn [59] and Fairlearn [18]. There are other metrics and definitions of fairness; thus, the results of this study may not generalize to all available metrics.

Nevertheless, the 30 metrics covered in this study are widely used in the fairness domain [27, 43, 53, 66, 89]. In future work, we will need to run more metrics.

External Validity: We have used seven datasets. In the fairness domain, one big challenge is the availability of adequate datasets. In future work, it would be insightful to re-run this study on new datasets and also on other domains.

Sampling Bias: In this work, we used thresholds recommended by IBM AIF360 ("fair" means −0.1,0.1 or 0.8,1.2 for different kinds of metrics). Future work should explore the sensitivity of our conclusions to changes in those thresholds.

Another issue with sampling bias is that our analysis is based on the data of Table 2. We recommend that when new data becomes available, we test the conclusions of this article against that new data. That would not be an arduous task (and to simplify that, we have placed all our scripts online in order).

7 CONCLUSION

From these results, we argue that

- There are many spurious fairness metrics, i.e., metrics that measure very similar things.
- To simplify fairness testing, (a) determine what type of fairness is desirable (for a list of types, see Tables 4 and 5); then (b) look up those types in our clusters; then (c) just test for one item per cluster.
- While this approach does not entirely remove all issues with fairness testing, it does reduce a very complex problem of (say) 30 metrics to a much smaller and manageable set.
- Also, the methods of this article could be used as a litmus test to prune away spurious new metrics that merely report the same thing as existing metrics.

REFERENCES

[1] 1953. Stanford hlab. Retrieved from https://hlab.stanford.edu/brian/number_of_clusters_.html.
[2] 1994. UCI:Adult Data Set. Retrieved from http://mlr.cs.umass.edu/ml/datasets/Adult.
[3] 2000. UCI:Statlog (German Credit Data) Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data).
[4] 2001. UCI:Heart Disease Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Heart+Disease.
[5] 2011. sklearn.cluster.AgglomerativeClustering. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html.
[6] 2014. Student Performance Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Student+Performance.
[7] 2015. propublica/compas-analysis. Retrieved from https://github.com/propublica/compas-analysis.
[8] 2016. Machine Bias: There's Software Used across the Country to Predict Future Criminals. And It's Biased against Blacks. Retrieved from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.
[9] 2017. Bank Marketing UCI. Retrieved from https://www.kaggle.com/c/bank-marketing-uci.
[10] 2017. Titanic: Machine Learning from Disaster. Retrieved from https://www.kaggle.com/c/titanic/data.
[11] 2018. Amazon Scraps Secret AI Recruiting Tool That Showed Bias against Women. Retrieved from https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G.
[12] 2018. Ethics Guidelines for Trustworthy Artificial Intelligence. Retrieved from https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai.
[13] 2018. FAIRWARE 2018: International Workshop on Software Fairness. Retrieved from http://fairware.cs.umass.edu/.
[14] 2018. Health Care Start-up Says A.I. Can Diagnose Patients Better Than Humans Can, Doctors Call That 'Dubious'. Retrieved from https://www.cnbc.com/2018/06/28/babylon-claims-its-ai-can-diagnose-patients-better-than-doctors.html.
[15] 2019. EXPLAIN 2019. Retrieved from https://2019.ase-conferences.org/home/explain-2019.
[16] 2019. Microsoft AI Principles. Retrieved from https://blogs.microsoft.com/eupolicy/artificial-intelligence-ethics/.
[17] 2020. Improving the Enrollment Process through Machine Learning. Retrieved from https://www.ellucian.com/insights/improving-enrollment-process-through-machine-learning.
[18] 2021. Fairlearn. Retrieved from https://fairlearn.org/.

[19] K. Adams and T. Gebur. 2021. Timnit Gebru Envisions a Future for Smart, Ethical AI; Podcast 'MarketPlaceTech'. Retrieved from https://www.marketplace.org/shows/marketplace-tech/timnit-gebru-envisions-a-future-for-smart-ethical-ai/.

[20] Aniya Aggarwal, Pranay Lohia, Seema Nagar, Kuntal Dey, and Diptikalyan Saha. 2019. Black box fairness testing of machine learning models. In *Proceedings of the 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'19)*. ACM, New York, NY, 625–635. https://doi.org/10.1145/3338906.3338937

[21] Ashrya Agrawal, Florian Pfisterer, Bernd Bischl, Jiahao Chen, Srijan Sood, Sameena Shah, Francois Buet-Golfouse, Bilal A. Mateen, and Sebastian J. Vollmer. 2020. Debiasing classifiers: Is reality at variance with expectation? (unpublished).

[22] Gabriele Bavota, Sebastiano Panichella, Nikolaos Tsantalis, Massimiliano Di Penta, Rocco Oliveto, and Gerardo Canfora. 2014. Recommending refactorings based on team co-maintenance patterns. In *Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering*. 337–342.

[23] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. https://arxiv.org/abs/1810.01943

[24] Suman K. Bera, Deeparnab Chakrabarty, Nicolas J. Flores, and Maryam Negahbani. 2019. Fair algorithms for clustering. arXiv preprint arXiv:1901.02393 (2019).

[25] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2017. Fairness in criminal justice risk assessments: The state of the art. arXiv:1703.09207 [stat.ML].

[26] Reuben Binns. 2019. On the apparent conflict between individual and group fairness. arXiv:1912.06883 [cs.LG].

[27] Sumon Biswas and Hridesh Rajan. 2020. Do the machine learning models on a crowd sourced platform exhibit bias? An empirical study on model fairness. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 642–653.

[28] Su Lin Blodgett and Brendan O'Connor. 2017. Racial disparity in natural language processing: A case study of social media African-American English. arXiv:1707.00061 [cs.CY].

[29] Yuriy Brun. 2020. Preventing undesirable behavior of intelligent machines (ICSSP and ICGSE 2020 keynote). Retrieved from https://www.youtube.com/watch?v=6M2Y3EG4fik&start=835s&ab_channel=YuriyBrun.

[30] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*. PMLR, 77–91.

[31] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186. https://doi.org/10.1126/science.aal4230

[32] 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*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3992–4001.

[33] L. Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi. 2020. Classification with fairness constraints: A meta-algorithm with provable guarantees. arXiv:1806.06055 [cs.LG].

[34] L. Elisa Celis and Vijay Keswani. 2019. Improved adversarial learning for fair classification. arXiv preprint arXiv:1901.10443 (2019).

[35] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2018. Ranking with fairness constraints. arXiv:1704.06840 [cs.DS].

[36] Juliana Cesaro and Fabio Gagliardi Cozman. 2019. Measuring unfairness through game-theoretic interpretability. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 253–264.

[37] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. 2021. Bias in machine learning software: Why? How? What to do? In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'21)*. Association for Computing Machinery, New York, NY, 429–440. https://doi.org/10.1145/3468264.3468537

[38] Joymallya Chakraborty, Suvodeep Majumder, Zhe Yu, and Tim Menzies. 2020. Fairway: A way to build fair ML software. 654–665. https://doi.org/10.1145/3368089.3409697

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

[40] J. Chakraborty, K. Peng, and T. Menzies. 2020. Making fair ML software using trustworthy explanation. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE'20)*. 1229–1233.

[41] Joymallya Chakraborty, Tianpei Xia, Fahmid M. Fahid, and Tim Menzies. 2019. Software engineering for fairness: A case study with hyperparameter optimization. arXiv:1905.05786 [cs.SE].
[42] Tse-Hsun Chen, Mark D. Syer, Weiyi Shang, Zhen Ming Jiang, Ahmed E Hassan, Mohamed Nasser, and Parminder Flora. 2017. Analytics-driven load testing: An industrial experience report on load testing of large-scale systems. In *Proceedings of the IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP'17)*. IEEE, 243–252.
[43] Andrew Cotter, Heinrich Jiang, Maya R. Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. 2019. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *J. Mach. Learn. Res.* 20, 172 (2019), 1–59.
[44] David Dann, Matthias Hauser, and Jannis Hanke. 2017. Reconstructing the giant: Automating the categorization of scientific articles with deep learning techniques. In *Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik*, 1538–1549.
[45] R. H. Davis, D. B. Edelman, and A. J. Gammerman. 1992. Machine-learning algorithms for credit-card applications. *IMA J. Manage. Math.* 4, 1 (01 1992), 43–51.
[46] Patrik D'haeseleer. 2005. How does gene expression clustering work? *Nat. Biotechnol.* 23, 12 (2005), 1499–1501.
[47] William Dickinson, David Leon, and A. Fodgurski. 2001. Finding failures by cluster analysis of execution profiles. In *Proceedings of the 23rd International Conference on Software Engineering (ICSE'01)*. IEEE, 339–348.
[48] Jin Hwan Do, D. Choi, et al. 2008. Clustering approaches to identifying gene expression patterns from DNA microarray data. *Molecules Cells* 25, 2 (2008), 279.
[49] Sanjiv Das et al. 2020. Fairness measures for machine learning in finance. *AWS Cloud* (October 2020).
[50] Michael Feldman, Sorelle Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. arXiv:1412.3756 [stat.ML].
[51] Marios Fokaefs, Nikolaos Tsantalis, Eleni Stroulia, and Alexander Chatzigeorgiou. 2011. JDeodorant: Identification and application of extract class refactorings. In *Proceedings of the 33rd International Conference on Software Engineering (ICSE'11)*. IEEE, 1037–1039.
[52] J. Foulds, Rashidul Islam, Kamrun Keya, and Shimei Pan. 2019. Differential fairness. *UMBC Faculty Collection* (2019).
[53] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 329–338.
[54] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. 2017. Fairness testing: Testing software for discrimination. *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE'17)*. https://doi.org/10.1145/3106237.3106277
[55] Vincent Grari, Boris Ruf, Sylvain Lamprier, and Marcin Detyniecki. 2019. Fair adversarial gradient tree boosting. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'19)*. IEEE, 1060–1065.
[56] Ben Green and Yiling Chen. 2019. Disparate interactions: An algorithm-in-the-loop analysis of fairness in risk assessments. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*'19)*. Association for Computing Machinery, New York, NY, 90–99. https://doi.org/10.1145/3287560.3287563
[57] J. Henry Hinnefeld, Peter Cooman, Nat Mammo, and Rupert Deese. 2018. Evaluating fairness metrics in the presence of dataset bias. arXiv:1809.09245 [cs.LG].
[58] Bipul Hossen, Hoque A. Siraj-Ud-Doulah, and Aminul Hoque. 2015. Methods for evaluating agglomerative hierarchical clustering for gene expression data: A comparative study. *Comput. Biol. Bioinf.* 3, 6 (2015), 88–94.
[59] Brittany Johnson, Jesse Bartola, Rico Angell, Katherine Keith, Sam Witty, Stephen J. Giguere, and Yuriy Brun. 2020. Fairkit, Fairkit, on the wall, who's the fairest of them all? Supporting data scientists in training fair models. arXiv:2012.09951 [cs.LG].
[60] Gareth P. Jones, James M. Hickey, Pietro G. Di Stefano, Charanpal Dhanjal, Laura C. Stoddart, and Vlasios Vasileiou. 2020. Metrics and methods for a systematic comparison of fairness-aware machine learning algorithms. arXiv preprint arXiv:2010.03986 (2020).
[61] Nathan Kallus and Angela Zhou. 2018. Residual unfairness in fair machine learning from prejudiced data. arXiv:1806.02887 [stat.ML].
[62] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* 33, 1 (01 October 2012), 1–33. https://doi.org/10.1007/s10115-011-0463-8
[63] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* 33, 1 (2012), 1–33.
[64] Faisal Kamiran, Sameen Mansha, Asim Karim, and Xiangliang Zhang. 2018. Exploiting reject option in classification for social discrimination control. *Inf. Sci.* (2018). https://doi.org/10.1016/j.ins.2017.09.064
[65] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases*, Peter A. Flach, Tijl De Bie, and Nello Cristianini (Eds.). Springer, Berlin, 35–50.

[66] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*. PMLR, 2564–2572.
[67] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. arXiv:1609.05807 [cs.LG].
[68] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R. Rickford, Dan Jurafsky, and Sharad Goel. 2020. Racial disparities in automated speech recognition. *Proc. Natl. Acad. Sci. U.S.A.* 117, 14 (2020), 7684–7689. https://doi.org/10.1073/pnas.1915768117
[69] Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. 2019. ifair: Learning individually fair data representations for algorithmic decision making. In *Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE'19)*. IEEE, 1334–1345.
[70] Qingwei Lin, Hongyu Zhang, Jian-Guang Lou, Yu Zhang, and Xuewei Chen. 2016. Log clustering based problem identification for online service systems. In *Proceedings of the IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C'16)*. IEEE, 102–111.
[71] Kirtan Padh, Diego Antognini, Emma Lejal Glaude, Boi Faltings, and Claudiu Musat. 2020. Addressing fairness in classification with a model-agnostic multi-objective algorithm. arXiv preprint arXiv:2009.04441 (2020).
[72] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q. Weinberger. 2017. On fairness and calibration. arXiv:1709.02012 [cs.LG].
[73] Pablo D. Reeb, Sergio J. Bramardi, and Juan P. Steibel. 2015. Assessing dissimilarity measures for sample-based hierarchical clustering of RNA sequencing data using plasmode datasets. *PLoS One* 10, 7 (2015), e0132310.
[74] Pedro Pereira Rodrigues, Joao Gama, and Joao Pedroso. 2008. Hierarchical clustering of time-series data streams. *IEEE Trans. Knowl. Data Eng.* 20, 5 (2008), 615–627.
[75] Prasanna Sattigeri, Samuel C. Hoffman, Vijil Chenthamarakshan, and Kush R. Varshney. 2019. Fairness GAN: Generating datasets with fairness properties using a generative adversarial network. *IBM J. Res. Dev.* 63, 4/5 (2019), 3–1.
[76] Shlomo S. Sawilowsky. 2009. New effect size rules of thumb. *J. Mod. Appl. Stat. Methods* 8, 2 (2009), 26.
[77] Sebastian Schelter, Yuxuan He, Jatin Khilnani, and Julia Stoyanovich. 2019. Fairprep: Promoting data to a first-class citizen in studies on fairness-enhancing interventions. arXiv preprint arXiv:1911.12587 (2019).
[78] Kumba Sennaar. 2019. Machine Learning for Recruiting and Hiring—6 Current Applications. Retrieved from https://emerj.com/ai-sector-overviews/machine-learning-for-recruiting-and-hiring/.
[79] Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, and Teh Ying Wah. 2015. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS One* 10, 12 (2015), e0144059.
[80] E. Strickland. 2016. Doc bot preps for the O.R. *IEEE Spectrum* 53, 6 (June 2016), 32–60. https://doi.org/10.1109/MSPEC.2016.7473150
[81] Rachael Tatman. 2017. Gender and dialect bias in YouTube's automatic captions. In *Proceedings of the 1st ACL Workshop on Ethics in Natural Language Processing*. Association for Computational Linguistics, 53–59. https://doi.org/10.18653/v1/W17-1606
[82] Robert L. Thorndike. 1953. Who belongs in the family? *Psychometrika* 18, 4 (1953), 267–276.
[83] Sakshi Udeshi, Pryanshu Arora, and Sudipta Chattopadhyay. 2018. Automated directed fairness testing. *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE'18)*. https://doi.org/10.1145/3238147.3238165
[84] Inês Valentim, Nuno Lourenço, and Nuno Antunes. 2019. The impact of data preparation on the fairness of software systems. In *Proceedings of the IEEE 30th International Symposium on Software Reliability Engineering (ISSRE'19)*. IEEE, 391–401.
[85] Sriram Vasudevan and Krishnaram Kenthapadi. 2020. LiFT. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. https://doi.org/10.1145/3340531.3412705
[86] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness (FairWare'18)*. Association for Computing Machinery, New York, NY, 1–7. https://doi.org/10.1145/3194770.3194776
[87] Christina Wadsworth, Francesca Vera, and Chris Piech. 2018. Achieving fairness through adversarial learning: An application to recidivism prediction. arXiv:1807.00199 [cs.LG].
[88] Hanchen Wang, Nina Grgic-Hlaca, Preethi Lahoti, Krishna P. Gummadi, and Adrian Weller. 2019. An empirical study on learning fairness metrics for COMPAS data with human supervision. arXiv:1910.10255 [cs.CY].
[89] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*. PMLR, 962–970.
[90] Feng Zhang, Quan Zheng, Ying Zou, and Ahmed E. Hassan. 2016. Cross-project defect prediction using a connectivity-based unsupervised classifier. In *Proceedings of the IEEE/ACM 38th International Conference on Software Engineering (ICSE'16)*. IEEE.

Received 22 January 2022; revised 29 December 2022; accepted 23 January 2023