Fighting the Fair Fight: An Overview of the Black-Box Fairness Auditing Techniques

Shubham Singh
University of Illinois at Chicago
ssing57@uic.edu

ABSTRACT

Today's decision-making systems rely heavily on machine learning (ML) models. The ML models are often trained on a biased dataset and too complex to explain their output leading to discrimination against a group of individuals. Furthermore, the training dataset and the model parameters are not publicly available for scrutiny. This paper presents an overview of the recent techniques that aim to inspect the black-box models to disparate outcome and present the probable causes for the discrimination. We explain the core principles behind the techniques and provide critical analysis of their effectiveness and shortcomings. We also discuss the future directions of the application of machine learning and the need for more comprehensive notions of fairness. This paper emphasizes the importance of fairness in machine learning and provides an introduction of the current techniques that the auditors and the social-welfare organizations can use to test any veiled disparate impact on the members of the society, and ensure that all of them are treated equitably.

1 INTRODUCTION

The introduction of machine learning powered systems to make decisions that affect individuals in society has been steadily rising over the past decade. The machine learning models are trained on huge amounts of data that can be collected when you are browsing the internet [29], walking on the road [20], and even when watching TV [28]. The authorities responsible for the data collection use the data to either gain money by selling them to the third-party data brokers for advertising or use them to create machine learning models to gain more insights about their users. However, it is hard to account for any bias that might be present in the collected data. And even if the authorities make a conscious effort to de-bias the data, sometimes it is observable by humans, but the models can pick them up. The models can introduce bias themselves if their objective is not defined clearly. Such models exhibit discriminatory behavior towards an individual or a group of individuals. Some of the examples include the biased recidivism scores towards black individuals [2], the use of AI for predictive policing [27], and the biased performance of facial machine learning services offered by the cloud providers [38].

Often, the users and the auditors of the model do not have direct access to them, and they are provided as "out-of-the-box" service for the users, making it impossible to catch any bias that may have occurred during the data collection or the model training stage. Such a setting presents a strong case for developing techniques that can exploit the input and the outputs of a model to audit for any potential discrimination, as the auditors could not directly investigate the model and its parameters. This paper describes and analyzes three recently proposed approaches that provide a mechanism to audit models in the black-box setting and present a report listing the most probable reasons for the disparate impact One of the earlier attempts to do so was proposed by Tramer et al. [42] called *FairTest*. Their efforts divide the user population into the smaller populations to identify the groups that might be harmed the most by the model. *Gradient Feature Auditing (GFA)* is another proposed technique that constructs an obscured version of the dataset, which is free of any proxy effects related to the sensitive attributes. They leverage the concept that the drop in model accuracy on the obscured version is evidence for unfair consequences. And finally, we elaborate on the approach proposed by Black et al. [8] called *FlipTest* which finds an appropriate mapping of an individual from one protected class to the other and constructs a set of individuals for whom the model outcome would change. The size of the set for which the decision was flipped indicates the disparity embedded in the model.

Along with an extensive overview of the three black-box auditing techniques, we also underscore the techniques' similarities and differences. Even though all the techniques are designed to test for unfair treatment, they are based on different concepts, applicable in different settings, and present the probable cause in different manners. We highlight the strengths of each of the techniques and draw attention towards the areas where they fall short. We also discuss the future research directions and the feasibility of such techniques to broader areas. The rest of the paper is organized as follows: §2 elaborates on the concepts related to machine learning and fairness in machine learning that are referred to throughout this paper; §3 presents an overview of FairTest; §4 explains the GFA algorithm, §5 analyses FlipTest. We present a comparison and describe the strengths and the limitations of the techniques in §6 and finally, discuss the need for such techniques and possible future research directions in §7.

2 PRELIMINARIES

Before we describe the fairness auditing techniques, we want to introduce some terms and their definitions which would be used throughout the paper.

2.1 Machine Learning Model

A *machine learning model* is defined as an algorithm or function that takes in a dataset in the of rows and columns and outputs a classification. The model "learns" to output the classification label using a training set. In a typical machine learning process, the input data is divided into a training and a testing set. The model randomly initializes a set of weights of each column in the dataset and updates them based on a loss function. The weights or coefficients signify the importance of the column in assigning a classification label. The process during which the weight get

updated is called model training. Once the training is completed, the model tests its weights-based output against the test set during model testing. The difference in the model output and the ground-truth labels of the test set is used to compute the loss. Once the loss has reached the desired minimum value, we consider the model to be trained, freeze the updated weights, and use a new dataset to "predict" their classification label.

2.2 Sensitive Attributes

A dataset can have multiple columns or attributes. Sometimes, they are also called features. We use columns, attributes, and features interchangeably throughout this paper. Each row in the dataset has a value for each attribute that defines the row based on an individual's characteristics. Now, specific attributes are considered *sensitive* or *protected* if the values of the attributes are used to determine the outcome of a decision, and the decision impacts the individuals with that attribute value disproportionately. For example, when considering an application for a house, making a decision based on an individual is prohibited by law [31].

2.3 Proxy Attributes

Given the definition of sensitive attributes, *proxy attributes* are defined as the attributes whose values explicitly do not appear to be signaling towards an individual's sensitive attribute, however, they do exhibit an implicit relationship to the sensitive attributes. Such a relationship can be caused by historical actions or societal contexts, which might be visible to the users, and sometimes also by the model, which is much harder to detect. For example, let's say we do not have access to an individual's race for the house application, but we know their income and the current postal code. Due to the redlining, people have been segregated into separate neighborhoods based on their race in the United States. The information about an individual's income and the current postal code can reveal their race, so income and the current postal code are proxy attributes in this example.

2.4 Disparity

Disparity or *Discrimination* is defined as the case when an individual's sensitive attribute or a proxy attribute is responsible for a different outcome when the values of the remaining attributes are the same, and the decision-maker, a human or a model, is the same. For example, if two individuals have the same credit score, same qualification and same income, but one individual is white and the another one is black. Imagine both of them are applying for a house, and if one individual gets it and the other one does not, then this constitutes as disparity.

The disparity could be broken down into two parts on a granular level: *Disparate Treatment* and *Disparate Impact*. We want to distinguish between the two. Disparate Treatment is when the disparity occurs at the input or model stage of the decision-making process, i.e., when the decision-maker explicitly takes the individual's sensitive attributes into account. Disparate Impact is when the disparity occurs at the outcome stage of the decision-making process, and it arises when the relationship between the sensitive and proxy variables is not explicitly observable. However, the model can to leverage it, and its outcome disproportionately impacts one group of individuals [25]. The remedy for disparate treatment is often called *Equality*, and that for disparate impact is called *Equity*.

3 FAIRTEST

3.1 Motivation

More and more applications used today collect the personal information on their users. Some of this data is used to sell for first-party or third-party advertising, and some of the data is used to train machine learning (ML) models for a prediction task. The data collected in such a manner is not free of bias, and there are cases when such a model has led to harmful and discriminatory behavior against a certain set of its users [21, 41]. For example, Staples deployed a variable pricing algorithm for its online buying users to attract more customers. However, the algorithm was later found to be discriminating against people living in lower-income neighborhoods by showing them higher prices. The company called the situation an "unintended consequence" of their algorithm. The authors of the paper [42] intend to tackle such problems by considering them as *bugs* and providing the developers of such models with a debugging tool. They term such bugs as *unwarranted associations*.

Although a general way to indicate an unwarranted association would be the presence of strong statistical dependency of an algorithm output on the protected class, the authors find such a definition to be fuzzy. It lacks in outlining wide-scope applicability, a method to provide scalable assessment, and including any natural explanatory factors to justify the perceived bias. Therefore, they informally define *unwarranted associations* as any strong associations between the algorithm output and the attributes of a protected user group, where the associations arise in a meaningful subset of users, have no explanatory factors, and can be used in a testing toolkit for wide-variety of tasks and datasets.

The authors use the definition of unwarranted associations to provide a framework called *unwarranted associations (UA) framework* to discover and analyze association bugs at the *data collection* stage of an ML pipeline, identify a *semantically meaningful subpopulation* that is affected, provide any *explanatory factors*, allow selecting the *suitable statistical measures* to support the bug discovery, and providing a *debugging mechanism* to the users. They package the proposed UA framework into a testing toolkit, which is called *FairTest* testing toolkit. The debugging applications of FairTest involve creating a digestible *debug report* for more in-depth inspections related to the association bugs. To substantiate the harm of association bugs, FairTest employs a decision-tree based approach named *association-guided tree construction* that splits the user space into subgroups to observe the effects of decreased population size and increased unwarranted association. The proposed approaches are evaluated over a diverse set of tasks and experiments. They also provide a publicly available[1] implementation of the FairTest framework.

3.2 Approach

This section describes the proposed methodology in two parts. First, we begin by exploring the conceptual UA framework, and

[1] https://github.com/columbia/fairtest

then we go over the details of the core components of the FairTest architecture based on the UA framework.

3.2.1 The UA Framework. The authors define an *unwarranted association* as any statistically significant association, in a semantically meaningful user subpopulation, between a protected attribute and an algorithmic output, where the association has no accompanying explanatory factor. They claim that a benefit of such a flexible definition over a mathematical expression is the ability to apply and extend it to broad areas, some of which could not be expressed in terms of mathematical relationships. Moreover, the statistical association encapsulates any relationship between two quantifiable entities such that they are statistically dependent.

Consider an algorithm that uses the data collected on the users, including sensitive features like location or age. The output of the algorithm that needs to be inspected is denoted as $\boldsymbol{O}$. The accuracy of $\boldsymbol{O}$ could have different *user utility* depending on the task. The attributes of the dataset can be characterized into three categories.

(1) *Protected attributes*, denoted as $\boldsymbol{S}$, are the primary attributes along which discrimination can occur. Typically, a group of individuals with certain $\boldsymbol{S}$ values constitutes a sensitive group and is protected by the law and policies.
(2) *Contextual attributes*, denoted as $\boldsymbol{X}$, are the attributes along which a population can be split to highlight hidden unwarranted associations. $\boldsymbol{X}$ is usually a proxy attribute for $\boldsymbol{S}$ that can be directly used by the algorithm and unwittingly reveal $\boldsymbol{S}$'s values to the algorithm.
(3) *Explanatory attributes*, denoted as $\boldsymbol{E}$, are the attributes whose values can justify a seemingly discriminatory behavior by the algorithm.

The categorization of $\boldsymbol{S}$, $\boldsymbol{X}$, and $\boldsymbol{E}$ is subjective to the task and is independent of the operations of the UA framework. The subjectiveness of the attributes can be explained by considering a hiring decision as an example. A company would want to hire candidates with more experience, even though, more experience can be a proxy for a candidate's age or gender.

Based on the values that $\boldsymbol{O}$ and $\boldsymbol{S}$ can take, the authors classify the choice of metrics that can be used to assess the strength of the association between $\boldsymbol{O}$ and $\boldsymbol{S}$ below:

- *Frequency Distribution Metrics:* When $\boldsymbol{O}$ and $\boldsymbol{S}$ are binary attributes, they can be written as $\boldsymbol{O} = \{o_1, o_2\}$ and $\boldsymbol{S} = \{s_1, s_2\}$. The *ratio metric* is defined as $\Pr(o_1|s_1)/\Pr(o_2|s_2) - 1$, and the *difference metric* is given as $\Pr(o_1|s_1) - \Pr(o_2|s_2)$. They are often useful when examining the algorithm output for unwarranted associations.
- *Mutual Information:* When the values of $\boldsymbol{O}$ and $\boldsymbol{S}$ are non-binary, inspired by information theory, the authors suggest leveraging the notion of *mutual information (MI)*, given by $\sum_{o,s} \Pr(o,s)\ ln(\frac{\Pr(o,s)}{\Pr(o)\Pr(s)})$. The *normalized MI (NMI)* can be computed by dividing the measure by the minimum of Shannon entropies of $\boldsymbol{O}$ and $\boldsymbol{S}$. They are also used when testing for associations between $\boldsymbol{O}$ and $\boldsymbol{S}$.
- *Correlation:* Although MI works well for the continuous values of $\boldsymbol{O}$ and $\boldsymbol{S}$, it is expensive to compute. The authors turn to Pearson's correlation to quantify the relationship between $\boldsymbol{O}$ and $\boldsymbol{S}$, when they are linearly dependent. It is used when the users want to profile the algorithm for errors.
- *Regression:* Regression is employed when the outputs of the algorithm are not known *a priori* or when the domain of output values is very large. The *regression coefficient* for each output value can provide evidence for the strength of association.
- *Conditional Metric:* It is used when looking for explanatory factors for a possible unwarranted association. For any given association metric, $\mathcal{M}(\boldsymbol{S};\boldsymbol{O})$ and explanatory attribute $\boldsymbol{E}$, the conditional association is given as the expectation, $\mathbb{E}_{\boldsymbol{E}}(\mathcal{M}(\boldsymbol{S};\boldsymbol{O})|\boldsymbol{E})$.

The authors point out that looking for associations across the full user population is not useful, as discrimination takes place in specific user groups. Therefore, they need to search for smaller but meaningful subpopulation that exhibits higher association. The framework uses *association-guided tree construction* to accomplish this, whose details are given later. And finally, since the associations can be justified in the presence of explanatory factors, discovering the association bugs is not a one-shot process. The framework is designed for multiple subsequent inspections, supported by statistical validity.

The core investigation primitives of the framework can be summarized as:

Testing: The users of the framework should be able to test for any suspected association between the algorithm outputs and protected attributes, in the presence of explanatory attributes.

Discovery: The framework should allow for the identification of algorithm outputs even when their values are not known *a priori*. This primitive is valuable when dealing with a large space of output values.

Error Profiling: The utility of an algorithmic output is dependent on how many times it is accurate for a subgroup. If an algorithm is more frequently accurate for one subgroup, compared to the other, it may be discriminating against a subgroup. The UA framework's compatibility with multiple metrics to measure the association make it capable of error profiling.

3.2.2 The FairTest Design. The authors preface the details of the FairTest architecture, by providing an example of the *association report* that simulates the Staples' pricing algorithm by giving discounts to customers who live within 20 miles radius of a competing store. As shown in Figure 1, the report highlights the statistically significant associations between the protected attribute of `income` and the algorithm output, `price`. For each population size, *NMI* is used to express the strength of the association, and the *contingency table* for output values and contextual attributes shows the frequency distribution. We observe that the global population does not experience a difference in the percentage of people shown a high price, and the NMI values are low. As we zoom into the subpopulation of white individuals living in California (CA), we notice that 8% of low-income individuals are offered high-price. In contrast, only 4% of the high-income individuals are offered high-price. Similarly, when looking at black male individuals living in New York (NY), 4% of low-income individuals were advertised high-price compared to

Report of associations of O=Price on S_i=Income:
Assoc. metric: norm. mutual information (NMI).

Global Population of size 494,436
p-value=3.34e-10 ; NMI=[0.0001, 0.0005]

Price	Income <$50K	Income >=$50K	Total
High	**15301 (6%)**	**13867 (6%)**	29168 (6%)
Low	234167(94%)	231101(94%)	465268 (94%)
Total	249468(50%)	244968(50%)	494436(100%)

1. Subpopulation of size 23,532
Context={State: CA, Race: White}
p-value=2.31e-24 ; NMI=[0.0051, 0.0203]

Price	Income <$50K	Income >=$50K	Total
High	**606 (8%)**	**691 (4%)**	1297 (6%)
Low	7116(92%)	15119(96%)	22235 (94%)
Total	7722(33%)	15810(67%)	23532(100%)

2. Subpopulation of size 2,198
Context={State: NY, Race: Black, Gender: Male}
p-value=7.72e-05 ; NMI=[0.0040, 0.0975]

Price	Income <$50K	Income >=$50K	Total
High	**52 (4%)**	**8 (1%)**	60 (3%)
Low	1201(96%)	937(99%)	2138 (97%)
Total	1253(57%)	945(43%)	2198(100%)

...more entries (sorted by decreasing NMI)...

Figure 1: A sample association report on Staples's simulated pricing algorithm, taken from [42].

only 1% of high-income individuals. The sample association report provides clear insights into the discrimination against subgroups and could flag the behavior before a biased algorithm is deployed in-the-wild.

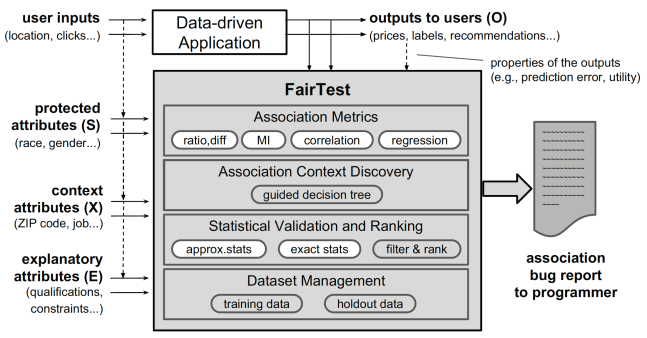

Figure 2: Architecture components for the FairTest illustrating the user inputs, compute mechanisms, and association report as the output. The image is taken from [42].

Figure 2 shows the architecture of FairTest, consisting of four major components, namely *Association Metrics, Association Context Discovery, Statistical Validation and Ranking, and Dataset Management*. Each of the components could be described as:

- The input to FairTest is a dataset, denoted as $D = \{(\boldsymbol{S}, \boldsymbol{X}, \boldsymbol{E}, \boldsymbol{O})\}$ that is split into a training set D_{train} and a test set D_{test}.
- The *Association Metric* is the module that is responsible for computing the association metrics described in §3.2.1.
- The *Association Context Discovery* module splits D_{train} for each $\boldsymbol{S}_i \in S$ to create meaningful subpopulation groups, based on $\boldsymbol{X}$, with the objective to maximize the association between $\boldsymbol{S}_i$ and O.
- For each subpopulation, *Statistical Validation and Ranking* module validates the bug on D_{test} using appropriate test statistic.
- *Dataset Management* module is responsible for managing copies of the test set to ensure the validity of statistical tests across multiple investigations.

The *Association Context Discovery* is driven by a heuristic partitioning technique called *guided decision-tree construction*. The algorithm is inspired by the way decision-trees work for classification. In this algorithm, $\boldsymbol{X}_i \in \boldsymbol{X}$ is chosen as the root node along which the dataset D is split into smaller sets, $\mathbb{D} = \{D_1, D_2, ...\}$. The number of splits depends on the values $\boldsymbol{X}_i$ can take. If $\boldsymbol{X}_i$ is categorical, D is split along each unique value of $\boldsymbol{X}_i$. If $\boldsymbol{X}_i$ is continuous, the authors choose a threshold t along which binary splits are created, $\mathbb{D} = \{D_1, D_2\}$, such that $\boldsymbol{X}_i < t$ for all rows in D_1. Threshold t is chosen based on testing the unique values of $\boldsymbol{X}_i$ that maximizes the association in the two splits. A *valid split* is characterized by at least one of the resulting subpopulation D_i showing a higher association than the current population D.

Statistical Validation and Ranking are an important component of FairTest as the objective of the Association Context Discovery module is to maximize the associations over a finite population, D_{train}. Therefore, an independent test set, D_{test} is required to validate the association bugs. The module employs *p-value* tests for hypothesis testing and *confidence intervals* for association metrics.

Since with each iteration, the train set is split into smaller subpopulation, FairTest needs to "track" such changes in the test set as well to validate the hypothesis across multiple investigations. To this end, FairTest takes a budget B as a user input along with the dataset at the beginning, so that FairTest can keep B test sets aside, one for each investigation.

3.3 Evaluation

The evaluation of FairTest attempts to answer the following three questions:

Q1 Is FairTest effective at detecting association bugs?
Q2 Is it fast enough to be practical?
Q3 Is it useful to identify and debug association bugs in a variety of applications?

3.3.1 Detective Effectiveness (Q1). The authors generate around 1M synthetic users using the US Census [9] for gender, income, and race. They begin by using a fair algorithm that assigns an output of $\{0, 1\}$ to individuals, without considering the income. Then, they introduce disparity in subpopulations, such that income level is related to a difference in output proportions of size 2Δ. As an example, if the value of $\Delta = 10\%$, the algorithm assigns 1 to 60% of the high-income individuals and 40% of the low-income individuals, among the white users living in California. They evaluate FairTest for different Δ values and population sizes. Their findings show that FairTest can detect disparities even for the low value of $\Delta = 2.5\%$ in larger contexts and the high value of $\Delta = 15\%$ in the context of a few hundred users. They also run FairTest on a set of simulated

and real-world datasets [7, 12, 13, 19, 35] and report the association bugs discovered. The detection of discrimination in the context of different population sizes appear to be accurate, but since there is no ground-truth for the datasets, the results are left to be further investigated by the domain experts.

3.3.2 Performance (Q2). The authors conduct runtime tests on all the datasets using a modern laptop, and they find that the total time taken ranges from 1-5 seconds for the smallest datasets to 60 seconds for the largest datasets (~1M individuals). They also compare FairTest to an approach like brute-force suggested by Pedreschi et al. [34]. Since the brute-force approach would generate exponentially large subpopulation size, they are limited by the number of smallest and largest subpopulation sizes created by FairTest. They used the Adult Census data [13] with protected attribute `gender` and target attribute `income`. Their findings show that both the approaches find a similar association, but FairTest does that with 4× to 8× less potential contexts.

Report of assoc. of O=Admitted on S_i=Gender, conditioned on attribute E=Department:

Global Population of size 2,213
p-value=7.98e-01 ; COND-DIFF=[-0.0382, 0.1055]

Admitted	Female	Male	Total
No	615(68%)	680(52%)	1295 (59%)
Yes	**295(32%)**	**623(48%)**	918 (41%)
Total	910(41%)	1303(59%)	2213(100%)

*** Department A: Population of size 490:**
p-value=4.34e-03 ; DIFF=[0.0649, 0.3464]

Admitted	Female	Male	Total
No	9(15%)	161(37%)	170 (35%)
Yes	51(85%)	269(63%)	320 (65%)
Total	60(12%)	430(88%)	490(100%)

*** Department B: Population of size 279:**
p-value=1.00e+00 ; DIFF=[-0.4172, 0.3704]

Admitted	Female	Male	Total
No	3(30%)	93(35%)	96 (34%)
Yes	7(70%)	176(65%)	183 (66%)
Total	10 (4%)	269(96%)	279(100%)

*** ... Departments C-F, with high p-values ...**

Figure 3: Association Report on the Berkeley graduate admissions dataset generated when investigating with the explanatory attribute, E = `Department`. The image is taken from [42].

3.3.3 Investigation Experience (Q3). In this section, we show the evaluation of FairTest on the Berkeley graduate admissions dataset [7]. We direct the user to the full paper [42] for the insights gained by experiments on the other real-world datasets. As shown in Figure 3, the initial investigations on the complete dataset disclose disparate impact, where only 32% of female applicants get admitted, whereas, 48% of male applicants are granted admission. The authors then define `department` as an explanatory attribute that tells FairTest to look for associations only within the applicants of a department. When looking at applicants in `Department A`, it is evident that the department has much higher admission rates for female applicants. The trend is also exhibited in `Department B`. This trent is called *Simpson's Paradox* when the effects exhibited by the overall population are different and sometimes opposite to that exhibited by smaller population.

4 GRADIENT FEATURE AUDITING

4.1 Motivation

Many machine learning services are available today as black-boxes, in the form of an application programming interface (API) both for individual and enterprise purposes. In such an environment, a user's interaction is limited to sending the inputs to the model and getting back the prediction outcome. Since the model-building processes are opaque to the user, it is not only hard to detect any bias in the model, but it is also impossible to retrain the model to rectify them. Adler et al. [1] propose a *Gradient Feature Auditing* (GFA) algorithm to identify the *indirect* influence of proxy variables in the decision the outcome of a black-box model. The *indirect* influence, as opposed to the *direct* influence, is termed as the influence of a non-sensitive attribute on the model outcome that is not classified as sensitive alone, but has a veiled relationship to the sensitive attribute. For example, when deciding to hire a candidate for a job, we do not want the sensitive attribute, `gender` to have a *direct* influence on our decision, however, `height` can cause an *indirect* influence, which can be linked back to the candidate's `gender`.

To study the indirect influence of proxy attributes, the authors suggest creating a modified dataset with minimal indirect influence and observing the black-box models' performance. A naive way to create the modified dataset would be to add random perturbations to the features, but doing so can affect the model performance arbitrarily and result in loss of information important to the model's outcome. The authors propose a deterministic approach to obscure the indirect influence attributes, while preserving the task-specific signals in the dataset. The contributions of this work can be listed as:

- An algorithm to construct a modified dataset with obscured indirect influence, with theoretical support.
- Formal definition of indirect influence in terms of black-box model outcomes.
- Evaluation of the approach on multiple publicly available datasets.

4.2 Approach

Consider a black-box classifier, $f : \mathbb{X} \rightarrow \mathbb{Y}$, where $\mathbb{X}$ is a d-dimensional feature space and $\mathbb{Y} = \{-1, 1\}$ for binary classification. A dataset drawn from the feature spaces can be denoted as (X, Y), where the i-th coordinate of X is a vector, $X_i = (x_{i1}, x_{i2}, ..., x_{id})$, such that $x_{ij} \in \mathbb{X}_j, 1 \leq j \leq d$. Given the notations, the accuracy of the model is given as:

$$\mathrm{acc}(X, Y, f) = \frac{1}{n} \sum \mathbf{1}_{y_i \neq f(X_i)} \tag{1}$$

And, the l_p norm is defined as:

$$||x||_p = (\sum_{i=1}^{d} |x_i|^p)^{1/p}) \tag{2}$$

Now, in order to measure the indirect influence of a feature j quantitatively, we would introduce perturbations to the values of $x_{ij}, \forall i \in X_i$. We call the new dataset with the perturbed values, X_{-j}, and now we calculate the difference between $\mathsf{acc}(X, Y, f)$ and $\mathsf{acc}(X_{-j}, Y, f)$ to get the indirect influence of j on the model outcome.

If the perturbations are introduced randomly, they can cause a two-fold adverse effect on the model evaluation. First, it can take away any useful information present in the feature that might be crucial for the model outcome. Second, it does not quantify the feature's proxy effects in a clear relationship to the outcome. On the contrary, the authors' method adds directed and deterministic perturbations to the feature which overcomes the problems with random perturbations. The modifications should be minimal such that they obscure the information to prevent predicting the values of feature j using the remaining features, thereby removing j from the dataset and obscuring its influence on X.

4.2.1 Notations. To test the feature predictability, the authors suggest using the *balanced error rate* as it is more robust against the class imbalance than the standard misclassification rate. The *balanced error rate*, BER of f on (X, Y) is defined as:

$$\mathsf{BER}(X, Y, f) = \frac{1}{|\text{supp}(y)|}\left(\sum_{j \in \text{supp}(y)} \frac{\sum_{y_i=j} \mathbf{1}_{f(X_i)\neq j}}{|\{i|y_i = j\}|}\right) \tag{3}$$

where $\text{supp}(Y) = y \in \mathbb{Y}|y \in Y$ is the set of elements of $\mathbb{Y}$ appearing in the dataset. Given the predictability measure, we want to provide the notation for the obscurity of a feature, where a feature is obscured if it can not be predicted using the remaining features. We define the *ϵ-obscure* version of X with respect to the feature space $\mathbb{X}_i$, as $X \setminus_\epsilon \mathbb{X}_i$. This can mathematically be written as:

$$\mathsf{BER}(X \setminus_\epsilon \mathbb{X}_i, X^{(i)}, f) > \epsilon \tag{4}$$

where $X^{(i)})$ is a feature drawn from $\mathbb{X}_i$. The definition of obscurity has been derived from one of the earlier works in [14]. And finally, we need a notation for the indirect influence of a feature. Indirect influence, II is defined in terms of the difference in model accuracies on the original and the modified dataset. It is given as:

$$\text{II}(i) = acc(X, Y, f) - acc(X \setminus_\epsilon \mathbb{X}_i, Y, f) \tag{5}$$

4.2.2 GFA Algorithm. After stating the notations, we now describe the authors' algorithm called *Gradient Feature Auditing (GFA)* to compute the indirect influence of a feature. Let us consider $O = \mathbb{X}_i$ as a categorical feature that needs to be removed from the dataset, and another feature $W = \mathbb{X}_j$ that is numerical and needs to be obscured. The marginal probability distribution of W conditioned on $O = x$ can be written as $W_x = \Pr(W|O = x)$ and the cumulative distribution is given as $F_x(w) = \Pr(W \geq w|O = x)$. The algorithm is designed in a way that it works feature by feature, by obscuring the influence of one feature at a time.

The authors define a *median distribution A* with the cumulative distribution is given as F_A and it's inverse function $F_A^{-1} = \text{median}_{x \in O} F_x^{-1}(u)$, where F_A^{-1} is also known as the quantile function. Inspired by the work in [14], they suggest that modifying the distribution of the feature being obscured, W, to the median distribution A changes W minimally and obscures the influence of O on W maximally. The modification is achieved by changing the values of W to mimic the median distribution, A, such that $\hat{W} = F_A^{-1}(F_x(w)), \forall w \in W$, where $\hat{W}$ is the modified version of W. They argue that A also minimizes the earth-mover distance [39], $d(.,.)$ in $\sum_{x \in O} d(W_x, A)$ between the two distributions using l_2 distance $d(p, q) = ||p - q||_2$ as the base metric.

In the description of the algorithm above, the authors assumed that the W is numerical and O is categorical. They elaborate on how does the algorithm work when O is numerical, and W is categorical. For the first case, they suggest removing the higher-order bits of the numerical values and using the lower order bits to bin the numerical feature, and then the rest of the algorithm would work as described before, substituting bins as categorical features. The second case is when the feature to obscure is categorical, making it infeasible to compute the cumulative distributions that are crucial to the algorithm. They introduce the exact metric $\mathbf{1}$ such that $\mathbf{1}(x, w) = 1 \iff x = w$. Like in the algorithm stated earlier, the exact metric is used as the base metric to define A as the distribution that minimizes the distance function $\sum_{x \in o} d(W_x, A)$. If we have two distributions p and q, then the earthmover distance between them using the exact metric $\mathbf{1}$ is given as $d(p, q) = ||p - 1||_1$ using the l_1 norm. The minimizing distribution A in this case, can be found taking a component-wise median for each value $w \in W$ and can be written as $p_A(w) = \text{median}_w W_x(w)$. The obscured, but minimally modified version $\hat{W}$ can be computed by using the minimum cost flow solution over the earthmover distance between each W_x and A.

The algorithm also provides a 0-1 scale of obscurity for the features where 0 represents an unchanged dataset, and 1 represents complete obscurity. It is important to provide such a scale as sometimes complete removal of the attributes can render the model useless, and by allowing *partial* obscurity, we can explore the fairness-utility tradeoff.

4.3 Evaluation

The authors perform an extensive evaluation of their proposed approach on multiple combinations of datasets and models they trained to treat as black-boxes. Once they have trained the models using the original dataset, they do not retrain the models but only use the modified datasets to observe the change in model accuracies. The code and the datasets used in the evaluation are publicly available[2].

4.3.1 Datasets and Models. They create a two-class *Synthetic dataset* with 6,000 items and five features distributed equally to each class. They create a feature (P) that encodes the row number, and two features (Q and R) are multiples of this feature. They also add a random and constant feature.

They use the *Adult Income* and *German Credit dataset* from the UCI Machine Learning Repository [13]. The Adult Income dataset contains 48,842 people, and 14 US Census attributes with a binary classification for each individual if they make more or less than $50K per year. The German Credit dataset contains data for 1,000

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

people and 20 attributes with a binary classification for good or bad credit score.

And finally, they use the *Dark Reaction dataset* consisting 3,955 experiments, 273 attributes, and the classification attribute is indicating the successful production of an ionic crystal.

The authors train multiple models on all the datasets that include SVMs [11], feedforward neural networks (FNNs) [15], C4.5 decision trees [36].

4.3.2 Auditing White-box vs. Black-box Models. In a black-box setting, we can not retrain the model with the obscured dataset. However, the black-box model could have been trained on a dataset, in presence of a sensitive attribute O. The authors claim that after the attribute W is obscured with respect to O, the modified dataset has no information related to O. Therefore, even if the model was originally trained with O, there is no information related to O in the test data that can be leveraged to make the decision. To validate this claim, they use the synthetic dataset and use decision tree and SVM models. For one case, they do retrain the model after obscuring each feature, and for the other case, they do not retrain the model, but only test the model accuracy on the obscured test data. They observe that the relative drop in the model accuracies was very close for both the cases, which validates their claim. They also observe that in the case of decision trees, when the model is fixed and the root split node is P, obscuring the values of Q and R do not affect the data relative to the split feature instantly. On the other hand, when the model is retrained on the obscured feature dataset, and it can change the split node feature based on the new information in the obscured dataset.

4.3.3 Black-box Feature Auditing. Now the authors evaluate their GFA algorithm on all the models and all the datasets. They start with the original dataset and increase the partial obscurity in 0.1 intervals, all the way till it reaches the value of 1, which is complete obscurity.

While running the experiment on the *synthetic dataset*, obscuring any of the encoding features (P, Q, or R) reduces the model accuracy to 50%. Although obscuring the constant feature does not affect the model performance, obscuring the random feature suffers from a slight drop as the random feature also uniquely identifies each row.

Based on the definition of indirect influence in Equation 5, the feature's importance to the model is directly related to the drop in model accuracy. In the experiments on the *Adult Income* dataset, they find that `age` was one of the most important features for SVM and FNN models, but less important for the decision tree model. They also note that counter to the intuition `race` is the least important feature. The results show that obscuring some of the features on the FNN model increases the model accuracy, which the authors attribute to the reduction in noise for a suboptimal model. They notice a similar but more pronounced noisy behavior on the *German Credit dataset*. On the contrary, the important features highlighted in the experiments are infact the most discerning features for the reaction outcome. The outcome is similar for SVM, FNN, and decision tree models and is corroborated by [37].

4.3.4 Auditing for Consistency. The authors want to investigate the noisy model accuracy results on *German Credit* and *Adult Income* datasets. To this end, they define *model consistency* as the change in the model accuracy with the increasing obscurity relative to the predicted label, rather than the original label. In other words, they replace the original class label with the model predicted label and calculate the difference in the accuracy. Therefore, they begin a 100% consistency and observe the decrease in the consistency with respect to the obscurity of each feature, which gives them insight into the feature importance and makes the findings independent of a suboptimal (overfitted) model. They note a fairly smooth drop in the consistency with slight noise for categorical features. They find `credit amount`, `checking status`, and `existing credits` as the top-ranked features across all models. They run a similar experiment on the *Adult Income* dataset with the FNN model. The results show that the top and bottommost feature ranks do not change, but features like *occupation* and *marital.status* are ranked higher in the consistency ranking.

5 FLIPTEST

5.1 Motivation

The classic approach to measure group fairness is using metrics like demographic parity [4] and equalized odds [18]. However, the approach fails to capture the discriminatory behavior towards individuals [10, 22] or even subgroups [26]. To study the discrimination at an individual level, prior work [16] have proposed changing the individual protected class. Although this process prevents the use of protected attributes directly, the attributes correlated with the protected attributes could still be a cause of harm. A more nuanced approach is suggested by Kusner et al. in [24]. They study the causal relationship with the protected attribute leading to a more granular understanding of the model discrimination against individuals. Nonetheless, the *disparate impact* can arise in the absence of such a causal relationship as well. Black et al. [8] propose another black-box fairness testing approach called *FlipTest* that has similar applications to Adler et al.'s approach [1] as elaborated in section 4.

FlipTest is a comprehesive and interpretable technique motivated by the question: *had an individual been of a different protected status, would the model have treated them differently?* FlipTest leverages optimal transport mapping [43] to transform the distribution between protected class labels and observe the shift in the model outcome. A change in model outcome indicates discrimination based on the protected attribute. The authors also highlight that the mapping does not depend on causal relationships to capture the discrimination caused at the outcome stage without considering any assumptions about the underlying data. Computing an optimal transport map is computationally expensive, and the cost grows with the increase in the size and dimensions of the dataset. To that end, the authors also introduce and validate a faster and efficient approximation method to compute optimal transport maps using Generative Adversarial Networks (GANs) [17]. The model output is assumed to be binary, positive and negative, and the results of optimal transport map from one class to the other are termed as *flipsets*. The authors create a *transparency report* that highlights the differences between the flipsets and provides an insight into what features might be responsible for the discrimination. Finally, FlipTest is a framework to examine a machine learning model for discriminatory behavior towards the protected groups. It can assist the well-intentioned

model creators and the external auditors who do not have direct access to the model.

5.2 Approach

Before describing their approach, the authors draw our attention to an example to show the applications of flipsets in action. They use a synthetic dataset created in [26], containing two features - hair length and work experience, and a classification label on whether the candidate was hired or not. They emphasize that flipping the gender attribute is not sufficient to study any discriminatory behavior as the model can learn the original protected attribute using the information present in the proxy attributes. Instead, for each female in the dataset, they need to find an appropriate map in the male category and vice versa, such that the mapped *counterpart* also accounts for the shift in the remaining attribute values. For example, due to the historical trend in the dataset for male candidates, a female candidate with no experience could be mapped to her male counterpart with two years of experience. Also, the decision of not hiring a female candidate with no experience might be based more on the work experience rather than the gender itself. This example persuades us to think about how *disparate treatment* might not always be a cause for disparate impact.

5.2.1 Optimal Transport Mapping. The authors then introduce the notation used throughout this section of the paper. Consider the two distributions $\mathcal{S}$ and $\mathcal{S}'$ for two classes over the feature space $\mathcal{X}$. Let n be the number of samples drawn from these two distributions, such that the set $S = \{x_1, ..., x_n\}$ and set $S' = \{x'_1, ..., x'_n\}$, where $n = |S| = |S'|$. Although the two sets are of equal size here, it is not a hard requirement for the proposed approximation method. The cost function is defined as $c : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty]$, such that $c(x, x')$ denotes the cost of moving a point from S to S'. An optimal transport map accomplishes moving between two points by minimizing the cost function. It is given as a bijection $f : S \rightarrow S'$ such that the expected cost is minimized:

$$\mathbb{E}[c(x, f(x))] = \frac{1}{n} \sum_{i=1}^{n} c(x_i, f(x_i)) \tag{6}$$

The existing methods of solving optimal transport map like Hungarian algorithm [23] do not scale well for large values of n. Therefore, the authors propose using a GAN to find robust approximations for computing the optimal transport map. The authors use the implementation of Wasserstein GAN (WGAN) provided in [3] to train a generator G to learn the optimal transport mapping. While training, the generator's loss function is given as $\frac{1}{n} \sum_{x \in S} D(G(x))$, where D denotes the discriminator. The loss function for the discriminator D is written as $\frac{1}{n} \sum_{x' \in S'} D(x') - \frac{1}{n} \sum_{x \in S} D(G(x))$. They modify the generator's loss function to include the cost function specified in Equation 6 and the new generator cost function is given as:

$$L_G = \frac{1}{n} \sum_{x \in S} D(G(x)) + \frac{\lambda}{n} \sum_{x \in S} c(x, G(x)) \tag{7}$$

λ here is the parameter to control the relative importance of the cost function. The authors provide the following proposition to motivate the use of generator G to compute optimal transport.

Proposition 1. *Suppose that $G*$ is a minimizer of L_G among all G such that $G(S) = S'$. If $\lambda > 0$, $G*$ is an exact optimal transport mapping from S to S'.*

The proof for the proposition is provided in the supplementary material of [8]. The authors then conduct experiments to show the stability of the GAN approximated mapping over the exact mapping.

Consider a fixed point $x \in S$, $n - 1$ other points randomly drawn from the distribution S, and n points randomly drawn from the distribution S'. They repeat the random draw multiple times to get different sets S and S' each time and study the variance of the point $f(x)$. They use the square of the L_1 distance as the cost function. The linear optimal transport mapping is implemented in Python 3 and WGAN is implemented using Keras[3], a Python library with the Tensorflow[4] backend. To measure the stability, they want to map the distribution to itself. For the first set of experiments, they set x to one vector and calculate the mean of $f(x)$. Because $f(x)$ is mapped to the same distribution as x, over multiple iterations, f should be equivalent to the identity function, and the mean of $f(x)$ should be close to x. The authors indeed observe the expected effect for GAN approximation, but they also observe a significant increase in the size of the linear program as the number of dimensions increased in the feature space. In the second set of experiments, they set x to be the zero vector, and just like in the first set of experiments, they study the variance of $f(x)$. They find the variance of $f(x)$ remains low and more stable using the GAN approximation compared to the exact mapping and another method suggested by Seguy et al. in [40].

5.2.2 Flipsets and Transparency Reports. The authors describe *flipsets* as the sets of individuals whose outcome changes for their mapped counterparts. A formal definition is given as:

Definition 1 (Flipset). *Let $h : \mathcal{X} \rightarrow \{0, 1\}$ be a classifier and $G : \mathcal{S} \rightarrow \mathcal{S}'$ be an optimal transport mapping (or an approximation). The flipset $F(h, G)$ is the set of points in S whose mapping into $\mathcal{S}'$ under G changes classification.*

$$F(h, G) = \{x \in S \mid h(x) \neq h(G(x))\} \tag{8}$$

The positive *and* negative *partitions of $F(h, G)$ are denoted by $F^+(h, G)$ and $F^-(h, G)$.*

$$F^+(h, G) = \{x \in S \mid h(x) > h(G(x))\}$$
$$F^-(h, G) = \{x \in S \mid h(x) < h(G(x))\}$$

For a better understanding, the authors use an example where $\mathcal{S}$ denotes the female candidates, and $\mathcal{S}'$ denotes the male candidates, and h is the binary decision function whose output determines if the candidate should be hired or not. Then $F^+(h, G)$ would be the set of female candidates who are hired, but their mapped male

[3] https://keras.io/
[4] https://www.tensorflow.org/

counterparts are not. Similarly, $F^-(h, G)$ would be the female candidates who are not hired, but their male counterparts are hired. Conversely, the mapping could be reversed to become $G' : S' \to S$ to generate male flipsets.

In an ideal scenario, when the input data is independent of any information about the protected attributes, the distributions $\mathcal{S}$ and $\mathcal{S}'$ would be equal, G would become an identity function, and the positive and negative flipsets would be empty. On the other hand, if the two flipsets are nonempty but of equal size, it can indicate demographic parity but does not necessarily mean individual fairness. Proposition 2 elaborates on this phenomenon:

Proposition 2. *Let h be a binary classification and $G : S \to S'$ with $|S| = |S'|$, be the exact optimal transport mapping. Then, $|F^+(h, G)| = |F^-(h, G)|$ if and only if the model satisfies demographic parity on the observed points, i.e.,*

$$|\{x \in S \mid h(x) = 1\}| = |\{x' \in S' \mid h(x') = 1\}|$$

The proof for the proposition is again provided in the supplementary material of [8].

Finally, the authors define *transparency reports* as the set of features that experience the highest amount of change between the members of a flipset under G. It is formally stated as:

Definition 2 (Transparency Report). *Let $h : \mathcal{X} \to \{0, 1\}$ be a classifier, $G : \mathcal{S} \to \mathcal{S}'$ be an optimal transport mapping (or an approximation), and $F(h, G)$ be the corresponding flipset. If $\mathcal{X} \subseteq \mathbb{R}^d$, we can compute the following vectors, each of whose coordinate corresponds to a feature in $\mathcal{X}$:*

$$\frac{1}{|F^\star(h, G)|} \sum_{x \in F^\star(h,G)} x - G(x), \text{ and}$$

$$\frac{1}{|F^\star(h, G)|} \sum_{x \in F^\star(h,G)} sign(x - G(x))$$

Here, $\star \in \{+, -\}$. Together, these vectors define a transparency report, which consists of two rankings of the features in $\mathcal{X}$, each sorted by the absolute values of each coordinate.

The highest-ranked features are responsible for the highest amount of difference in the model outcome on the flipset.

5.3 Evaluation

In this subsection, we go over the authors' evaluation of their approach using real and synthetic datasets. They begin by showing experiments that entrench the use of GAN approximation for the optimal transport mapping. Then they conduct case studies on two datasets - Chicago Strategic Subject List (SSL) [30] and the candidate hiring dataset created in [26].

5.3.1 GAN Validation. The authors train a GAN on the samples from two identical distributions. They state that as the number of training samples increases, the size of the flipsets gets closer to zero. Since the GAN generator is supposed to learn and generate the data close to the training samples, this experiment provides an empirical bound on the number of flipsets that may arise due to the noise in the GAN approximation. To compensate for mapping the distribution itself, they add random features *dependent* on the protected attribute to simulate close to in-the-wild GAN training behavior. To this end, they draw 10,000 samples from each distribution S and S' and train a complex SVM model with RBF kernel over random labels. Even though these measures are introduced as a way to increase the flipsets, the size of the flipsets remains as low as 3% of the dataset size, which helps establish a lower threshold for experiments described further.

The authors conduct another experiment to validate GAN approximation. They create a dataset with 2,000 members ($x \in S'$) and compute the optimal transport map using the exact method ($f(x)$) and the GAN approximation method ($G(x)$). They calculate the square of L_1 distance between x and $f(x)$ and compare it to that between x and $G(x)$. Furthermore, they employ Kolmogorov–Smirnov(KS) statistic between S' and $G(S)$. And finally, they train linear regression models over the real target data S' to predict each feature from the remaining ones. The mean squared error between the regression model output and $G(S)$ provides a measure of how well the GAN captures the feature correlation. The results of the all the experiments provide evidence for the validation of the proposed GAN approximation approach.

5.3.2 Testing a Biased Model. The authors use the SSL dataset [30] that contains records of arrested individuals having attributes like: the age during the arrest, number of prior arrests for violent offenses, number of prior narcotic arrests, gang affiliation, number of times as a victim of a shooting incident. The classification attribute is a score on the scale of 0 (low) to 500 (high) based on the likelihood of the individual being involved in a shooting incident either as a victim or an offender. Since prior work [44] has shown that the models trained on this dataset do not exhibit discriminatory behavior, the authors introduce bias in the model by giving more weight to the attribute of the number of prior narcotic arrests, which is correlated with the individual's race, and is less predictive than the other attributes. The final model classifies an individual as high risk if the individual satisfies $-53 \cdot age + 25 \cdot narc > 65$, which disparately assigns high-risk scores to black individuals.

Of the 3,683 high-risk black individuals classified by the model, the positive flipset $F^+(h, G)$ has a size of 1,290 black individuals whose white counterparts are classified as low-risk. On the contrary, out of 37,877 black individuals that low risks, the negative flipset $F^-(h, G)$ consists of only four white counterparts who are classified as high-risk. Based on the size of the flipsets, it is evident that the model discriminates based on race, which is a part of the model design.

To look at the model outcomes from the lens of subgroup fairness, the authors use the histograms of the marginal distribution of the features in the flipset and compare them to the marginal distribution of the entire black population in the dataset. The normalized feature values for age are lower and higher for narcotic arrests in the flipsets than the full black population, reflecting the introduced model bias. Conversely, the marginals for gang affiliation did not show any difference, which was not weighed by the model. For a granular analysis, they turn to the transparency report to gain insight into which features are responsible for the discrimination. The feature that exhibits the most change in $F^+(h, G)$ is narcotic arrests, which again consistent with the introduced bias in the model.

Now, the authors investigate the use of optimal transport mapping towards the application in fairness notion of equalized odds. *Equalized odds* states that the model outcome is independent of the protected attribute given the ground-truth. To that end, they train two optimal transport maps: one to map black ground-truth negatives to white ground-truth negatives and the other one to map black ground-truth positives to white ground-truth positives. For the ground-truth negatives, $F^+(h, G)$ comprises 499 individuals mapped from 5,002 black ground-truth negative individuals predicted high-risk by the model. On the other hand, $F^-(h, G)$ is an empty set. The histogram analysis of the $F^+(h, G)$ highlights the lower age and higher narcotics arrest compared to the rest of the black population. The transparency report corroborates these findings. For the ground-truth positives, the fliptest $F^+(h, G)$ has 261 individual members out of 1,568 black ground-truth positive individuals whose model outcome was high-risk. $F^-(h, G)$ contains 102 individuals out of 3,767 black ground-truth positive individuals predicted as low-risk. The findings are similar to that of the ground-truth negatives.

5.3.3 Testing a Group-Fair Model. The authors train an algorithm suggested by Zafar et al. [45] on the synthetic dataset created by Lipton et al. [26] that contains hair length and work experience as candidate features, with the classification attribute gesturing the decision to hire the candidate or not. In their study, Lipton et al. find out that the algorithm designed to equalize the hiring rates across the protected attribute, gender in this case, results in a linear model that uses hair length as proxy attribute and overcorrects by benefitting long-haired men and harm short-haired women. The authors of the FlipTest paper recreate the linear model and trained it over 10,000 individuals, each belonging to the male and female protected groups. The model outcome generates nearly identical hiring rates of 27% and 30% for male and female groups, respectively. Looking closely at the flipsets, they notice the individuals that are being discriminated against. When mapped to male individuals, 715 female individuals are rejected, while 1,215 males mapped to female individuals are hired. The transparency report confirms the findings in [26] that state that the model benefits feminine characteristics, longer hair-length in this case, to equalize the hiring rates. The disadvantaged women have short hair and a lot less experience compared to the advantaged women, who have long hair and slightly less experience.

6 COMPARISON

In this section, we present an extensive comparison of all the three approaches covered in §3, §4, and §5. The techniques provide a way to test for the fairness criteria of an ML model or an algorithm, which is why all of them are *post-processing* techniques. To elaborate, all the techniques require the output of the model, which is then used to infer a direct or indirect relationship with sensitive attributes. In most cases, the model does not directly use the sensitive attributes to make a decision, and in some cases, it is prohibited by the law. Even though the model creators do not explicitly use the sensitive attributes, proxy attributes often reveal the information which can identify the individuals and their sensitive attributes. Such a relationship is usually derived from the societal contexts and makes the *proxy attributes* the focus of all the techniques to investigate discriminatory behavior.

Gradient Feature Auditing (GFA) and FlipTest techniques are explicitly designed to work in the *black-box settings*, i.e. when the auditors do not have access to the model, and it can not be retrained. Although in the experiments used in the evaluation of FairTest, the authors had access to the models, the technique would also work if the models were locked in a black-box, as FairTest mainly relies upon the algorithmic output. Another shared property that comes when dealing with the model outcome rather than the input is that all the techniques test the model for *disparate impact*, rather than disparate treatment. The bias responsible for the disparate impact can be introduced at any stage, from historical trends in the data to the unintended objective function in the model, and varies a lot based on the prediction task. Consequently, the remedies for the observed disparate impact is left as further exploration for the domain experts.

Although the techniques seem quite similar, they do have many differences in terms of model assumptions and the core principles of their approach. FairTest uses decision-tree inspired, *association-guided tree construction* routine to divide the dataset into *smaller populations* that exhibit higher correlation with the algorithmic output. The *association metrics* determine the strength of the observed associations. The final output of FairTest is an *association report* that an auditor can read to gain insights into possible discrimination. GFA proposes creating a median distribution based on the similarly ranked pairs of individuals and moving the real values towards the median distribution by minimizing the *earth mover's distance (EMD)*. It preserves the ranks within the protected groups and minimizes the effect of the feature correlated with the protected attribute. The result is an *obscured version* of the original dataset, and the *drop in model accuracy* presents evidence for discrimination. The order of features that cause the decrease in model accuracy represents the *ranking of the features* responsible for disparate impact. Finally, FlipTest employs GAN approximation for *optimal transport mapping* between two protected classes. The resulting *flipsets* and their size is indicative of the model discrimination against a subgroup. The *transparency report* is the FlipTest's outcome for the domain expert to inspect the *features that change in the flipsets* compared to the remaining user subpopulation and narrow down the cause of potential discrimination.

FairTest allows the use of multiple model outputs and attribute values to compute the associations and works towards testing for subgroup fairness. GFA is agnostic to the model output and is compatible with numerical and categorical attributes. It is designed to audit for group fairness. Fliptest is only compatible with binary model output and numerical attributes, and it examines the model for subgroup and individual fairness. The similarities and differences between the three techniques are summarized in Table 1.

The differences also highlight the strengths and weaknesses of each of the techniques. Beginning with FairTest, the technique's major strength could be presented as the compatibility of use of regression to find association in the large output space. The ability to consider the explanatory factors into the observed association makes it a widely applicable testing toolkit. However, one shortcoming of the technique could be found in its association-guided tree mechanism. Since it is inspired by the decision tree model used

Technique	Access Setting	Inspection Surface	Discrimination Stage	Dataset Modification	Modification Mechanism
FairTest	Black-box & White-box	Proxy variables	Disparate Impact	User Subpopulation	Association-guided tree construction
GFA	Black-box	Proxy variables	Disparate Impact	Obscured dataset	EMD to minimize distribution transformation
FlipTest	Black-box	Proxy variables	Disparate Impact	Flipsets	GAN approximation for optimal transport mapping

Technique	Metrics	Audit Report	Model Outcome	Fairness Category
FairTest	Association Metrics	Association Report	Multiple	Subgroup Fairness
GFA	Accuracy Drop	Feature Ranking	Agnostic	Group Fairness
FlipTest	Size of flipsets	Transparency Report	Binary	Subgroup & Individual Fairness

Table 1: A comparison of all the techniques reviewed in this work.

for classification, it assumes that the disparity is monotonically related to the contextual attribute. In other words, it is limited by binary splits for the continuous values of the protected attribute. It also fails to provide any theoretical guarantee for the , and the discovery of the association is bound by the required minimum sample size for statistical tests. Although GFA is agnostic to the model output, one of the drawbacks of the technique is its inability to find discrimination based on multiple proxy attributes. It also assumes that the model outcome is dependent on the inherent ranking of individuals across the obscured attributes. Additionally, they only rely on the change in model accuracy to quantify the influence of a protected attribute. However, the model accuracy might not be equipped to capture such a relationship in the context of discriminatory behavior. The feature ranking presented as the audit report in GFA is not as comprehensive as that of the other two techniques. FlipTest's most significant contribution is the use of GANs to approximate an expensive but suitable algorithm to map individuals from one protected group to the other. On the other hand, one of the reasons the technique falls short is that it works only with the binary model output, limiting the applications of FlipTest to a wide variety of use-cases. Moreover, the discovery of the discrimination and the creation transparency report presented to the auditor do not scale well with the high-dimensional datasets.

7 DISCUSSION

In this paper, we presented the details of the three recent techniques that concentrate on testing the ML models deployed in-the-wild for discrimination and highlighting the reasons that might be responsible for such behavior. The techniques represent the current set of comprehensive auditing techniques, but as the applications of machine learning as rising in different areas, especially where the decisions can have a societal impact. For example, the recent studies by Asudeh et al. [5, 6] propose *coverage* as a fairness definition. To expand, coverage can be defined as *fairness in rows* or *fairness in representation* rather than the classic notion of fairness in columns or attributes. Suppose there are not enough samples representing a set of individuals belonging to a subgroup. In that case, even the most fair model can not learn a good decision boundary to distinguish the subgroup. Another set of recent studies done by Patro et al. [32, 33] propose methods to establish fairness in two-sided markets where the model should not exhibit any discriminatory behavior towards either the consumers or the producers. Such novel notions of fairness emphasize the need for more comprehensive, extensive, and robust fairness auditing techniques that would require redesigning from the ground up.

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 (Jan. 2018), 95–122. https://doi.org/10.1007/s10115-017-1116-3

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]* (Dec. 2017). http://arxiv.org/abs/1701.07875 arXiv: 1701.07875.

[4] Arvind Narayanan. 2018. Tutorial: 21 fairness definitions and their politics. https://www.youtube.com/watch?v=jIXIuYdnyyk

[5] Abolfazl Asudeh, Tanya Berger-Wolf, Bhaskar DasGupta, and Anastasios Sidiropoulos. 2020. Maximizing coverage while ensuring fairness: a tale of conflicting objective. arXiv:cs.CC/2007.08069

[6] A. Asudeh, Z. Jin, and H. V. Jagadish. 2019. Assessing and Remedying Coverage for a Given Dataset. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. 554–565. https://doi.org/10.1109/ICDE.2019.00056

[7] P. J. Bickel, E. A. Hammel, and J. W. O'Connell. 1975. Sex Bias in Graduate Admissions: Data from Berkeley. *Science* 187, 4175 (1975), 398–404. https://doi.org/10.1126/science.187.4175.398 Publisher: American Association for the Advancement of Science _eprint: https://science.sciencemag.org/content/187/4175/398.full.pdf.

[8] Emily Black, Samuel Yeom, and Matt Fredrikson. 2020. FlipTest: fairness testing via optimal transport. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM, Barcelona Spain, 111–121. https://doi.org/10.1145/3351095.3372845

[9] US Census Bureau. 2015. Census.gov. https://www.census.gov/en.html Section: Government.

[10] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (June 2017), 153–163. https://doi.org/10.1089/big.2016.0047

[11] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[12] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision*

and Pattern Recognition. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[13] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[14] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*. ACM Press, Sydney, NSW, Australia, 259–268. https://doi.org/10.1145/2783258.2783311

[15] Terrence L. Fine, S. L. Lauritzen, M. Jordan, J. Lawless, and V. Nair. 1999. *Feedforward Neural Network Methodology* (1st ed.). Springer-Verlag, Berlin, Heidelberg.

[16] 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, Paderborn Germany, 498–510. https://doi.org/10.1145/3106237.3106277

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc., 2672–2680. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[18] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of Opportunity in Supervised Learning. *arXiv:1610.02413 [cs]* (Oct. 2016). http://arxiv.org/abs/1610.02413 arXiv: 1610.02413.

[19] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[20] Shelley Kasli. 2018. Picture Intelligence Unit – Aadhaar Based Surveillance By Foreign Firms – Tune IN'23. https://web.archive.org/web/20180202190150/http://tunein23.com/EN/?p=2757

[21] Jana Kasperkevic. 2015. Google says sorry for racist auto-tag in photo app. http://www.theguardian.com/technology/2015/jul/01/google-sorry-racist-auto-tag-photo-app Section: Technology.

[22] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017) (Leibniz International Proceedings in Informatics (LIPIcs))*, Christos H. Papadimitriou (Ed.), Vol. 67. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 43:1–43:23. https://doi.org/10.4230/LIPIcs.ITCS.2017.43 ISSN: 1868-8969.

[23] H. W. Kuhn and Bryn Yaw. 1955. The Hungarian method for the assignment problem. *Naval Res. Logist. Quart* (1955), 83–97.

[24] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. (2017), 11.

[25] Google Machine Learning. 2020. Machine Learning Glossary: Fairness | Google Developers. https://developers.google.com/machine-learning/glossary/fairness

[26] Zachary Lipton, Julian McAuley, and Alexandra Chouldechova. 2018. Does mitigating ML's impact disparity require treatment disparity? *Advances in Neural Information Processing Systems* 31 (2018), 8125–8135. https://proceedings.neurips.cc/paper/2018/hash/8e0384779e58ce2af40eb365b318cc32-Abstract.html

[27] Vidushi Marda and Shivangi Narayan. 2020. Data in New Delhi's Predictive Policing System. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 317–324. https://doi.org/10.1145/3351095.3372865

[28] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W. Felten, Prateek Mittal, and Arvind Narayanan. 2019. Watching You Watch: The Tracking Ecosystem of Over-the-Top TV Streaming Devices. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*. Association for Computing Machinery, New York, NY, USA, 131–147. https://doi.org/10.1145/3319535.3354198

[29] David Nield. 2017. Here's All the Data Collected From You as You Browse the Web. https://gizmodo.com/heres-all-the-data-collected-from-you-as-you-browse-the-1820779304

[30] City of Chicago. 2017. Strategic Subject List - Historical | City of Chicago | Data Portal. https://data.cityofchicago.org/Public-Safety/Strategic-Subject-List-Historical/4aki-r3np Accessed on Dec 17, 2020.

[31] US Department of Justice. 1968. Fair Housing Act. https://www.justice.gov/crt/fair-housing-act-2

[32] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2020. FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms. In *Proceedings of The Web Conference 2020 (WWW '20)*. Association for Computing Machinery, New York, NY, USA, 1194–1204. https://doi.org/10.1145/3366423.3380196

[33] Gourab K. Patro, Abhijnan Chakraborty, Niloy Ganguly, and Krishna Gummadi. 2020. Incremental Fairness in Two-Sided Market Platforms: On Smoothly Updating Recommendations. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 01 (April 2020), 181–188. https://doi.org/10.1609/aaai.v34i01.5349

[34] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. 2009. Integrating Induction and Deduction for Finding Evidence of Discrimination. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law (ICAIL '09)*. Association for Computing Machinery, New York, NY, USA, 157–166. https://doi.org/10.1145/1568234.1568252

[35] Heritage Health Prize. 2012. Heritage Health Prize. https://kaggle.com/c/hhp

[36] J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[37] Paul Raccuglia, Katherine C. Elbert, Philip D. F. Adler, Casey Falk, Malia B. Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A. Friedler, Joshua Schrier, and Alexander J. Norquist. 2016. Machine-learning-assisted materials discovery using failed experiments. *Nature* 533, 7601 (May 2016), 73–76. https://doi.org/10.1038/nature17439 Number: 7601 Publisher: Nature Publishing Group.

[38] Inioluwa Deborah Raji and Joy Buolamwini. 2019. Actionable Auditing: Investigating the Impact of Publicly Naming Biased Performance Results of Commercial AI Products. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (AIES '19)*. Association for Computing Machinery, New York, NY, USA, 429–435. https://doi.org/10.1145/3306618.3314244

[39] Y Rubner, C Tomasi, and L J Guibas. 1998. A metric for distributions with applications to image databases. (1998), 8.

[40] Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. 2018. Large-Scale Optimal Transport and Mapping Estimation. arXiv:stat.ML/1711.02283

[41] Jennifer Valentino-DeVries Soltani, Jeremy Singer-Vine and Ashkan. 2012. Websites Vary Prices, Deals Based on Users' Information. *Wall Street Journal* (Dec. 2012). https://online.wsj.com/article/SB10001424127887323777204578189391813881534.html

[42] Florian Tramer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. 2017. FairTest: Discovering Unwarranted Associations in Data-Driven Applications. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, Paris, 401–416. https://doi.org/10.1109/EuroSP.2017.29

[43] Cédric Villani. 2009. *Optimal Transport: Old and New*. Springer-Verlag, Berlin Heidelberg. https://doi.org/10.1007/978-3-540-71050-9

[44] Samuel Yeom, Anupam Datta, and Matt Fredrikson. 2018. Hunting for Discriminatory Proxies in Linear Regression Models. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 4568–4578. http://papers.nips.cc/paper/7708-hunting-for-discriminatory-proxies-in-linear-regression-models.pdf

[45] 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. http://proceedings.mlr.press/v54/zafar17a.html ISSN: 2640-3498.