9-2023

TESTSGD: Interpretable testing of neural networks against subtle group discrimination

Mengdi ZHANG
Singapore Management University, mdzhang.2019@phdcs.smu.edu.sg

Jun SUN
Singapore Management University, junsun@smu.edu.sg

Jingyi WANG
Zhejiang University

Bing SUN
Singapore Management University, bing.sun.2020@phdcs.smu.edu.sg

Citation

ZHANG, Mengdi; SUN, Jun; WANG, Jingyi; and SUN, Bing. TESTSGD: Interpretable testing of neural networks against subtle group discrimination. (2023). *ACM Transactions on Software Engineering and Methodology*. 32, (6), 1-24.
Available at: https://ink.library.smu.edu.sg/sis_research/8144

TestSGD: Interpretable Testing of Neural Networks against Subtle Group Discrimination

MENGDI ZHANG and JUN SUN, Singapore Management University, Singapore
JINGYI WANG, Zhejiang University, China
BING SUN, Singapore Management University, Singapore

Discrimination has been shown in many machine learning applications, which calls for sufficient fairness testing before their deployment in ethic-relevant domains. One widely concerning type of discrimination, testing against *group discrimination, mostly hidden*, is much less studied, compared with identifying *individual discrimination*. In this work, we propose TestSGD, an interpretable testing approach that systematically identifies and measures hidden (which we call *"subtle") group discrimination* of a neural network characterized by *conditions over combinations of the sensitive attributes*. Specifically, given a neural network, TestSGD first automatically generates an interpretable rule set that categorizes the input space into two groups. Alongside, TestSGD also provides an estimated group discrimination score based on sampling the input space to measure the degree of the identified subtle group discrimination, which is guaranteed to be accurate up to an error bound. We evaluate TestSGD on multiple neural network models trained on popular datasets including both structured data and text data. The experiment results show that TestSGD is effective and efficient in identifying and measuring such subtle group discrimination that has never been revealed before. Furthermore, we show that the testing results of TestSGD can be used to mitigate such discrimination through retraining with negligible accuracy drop.

CCS Concepts: • **Software and its engineering** → *Extra-functional properties;*

Additional Key Words and Phrases: Fairness, machine learning, fairness testing, fairness improvement

ACM Reference format:
Mengdi Zhang, Jun Sun, Jingyi Wang, and Bing Sun. 2023. TestSGD: Interpretable Testing of Neural Networks against Subtle Group Discrimination. *ACM Trans. Softw. Eng. Methodol.* 32, 6, Article 137 (September 2023), 24 pages.
https://doi.org/10.1145/3591869

1 INTRODUCTION

Machine learning models, especially neural networks, are becoming ubiquitous in various real-life applications. For example, they are used in medical diagnosis [26], self-driving cars [9], and

This research is supported by the Ministry of Education, Singapore under its Academic Research Fund Tier 3 (Award ID: MOET32020-0004). This research is supported by the Key R&D Program of Zhejiang (2022C01018) and the NSFC Program (62102359).

Authors' addresses: M. Zhang, J. Sun, and B. Sun, Singapore Management University, Singapore; emails: mdzhang.2019@phdcs.smu.edu.sg, junsun@smu.edu.sg, bing.sun.2020@phdcs.smu.edu.sg; J. Wang, Zhejiang University, China; email: wangjyee@zju.edu.cn.

1049-331X/2023/09-ART137 $15.00
https://doi.org/10.1145/3591869

criminal sentencing [5]. Meanwhile, more attention has been paid to the fairness issues of these machine learning models [4, 11, 16, 17, 36, 37, 42, 49, 50], as discrimination has been discovered in many applications [18, 36, 41, 47]. For instance, machine learning models were used to predict recidivism risk for suspected criminals by computing the likelihood of committing a future crime [5]. Analysis results show that the prediction model was more likely to mislabel black defendants as high recidivism risk and mislabel white defendants as low risk. To minimize such ethical risks, it is crucial to systematically test the fairness of machine learning models, especially neural networks where such issues are typically "hidden" due to the lack of interpretability [32, 39].

Recently, multiple efforts have been made in the testing community to first search for (and then guide mitigating) discrimination of machine learning models spanning from traditional ones to neural networks [16, 41, 47, 49, 50]. For instance, state-of-the-art fairness testing work utilizes gradient information of the input sample to accelerate search/generation of discriminative samples [49, 50]. Despite being effective, existing research has mostly focused on *individual discrimination*, i.e., identifying or generating individual discriminatory instances of a machine learning model [16, 41, 47, 49, 50]. *Group discrimination*, which characterizes a model's discrimination against a certain group (whose sensitive attributes[1] satisfy certain conditions), is another concerning type of discrimination, which has been widely studied [16, 25, 40, 47]. However, testing against group discrimination has been much less studied so far. Compared to testing of individual discrimination, testing a machine learning model against group discrimination imposes new challenges. First, it is highly non-trivial to effectively enumerate all combinations of sensitive attributes (especially when the sensitive attributes have multiple or even continuous values). Second, *group discrimination can be hidden, i.e., there might be "subtle" group discrimination against those groups whose sensitive attributes satisfy certain unknown conditions, e.g., male-white of certain age group.* While a prior work [24] similarly addresses discrimination against subgroups defined over conjunctions of protected attributes in the learning phase, we propose an automatic testing approach to systematically identify such subgroups using interpretable rules and measure such discrimination before model deployment.

Specifically, in this work, we develop an effective method to systematically *test* a given machine learning model against such hidden *s*ubtle *g*roup *d*iscrimination, namely, TestSGD. An overview of TestSGD is shown in Figure 1, which consists of three main phases: (1) candidate rule set generation, (2) group fairness identification, and (3) discrimination mitigation. In the first phase, TestSGD will automatically generate a candidate set of rules concerning multiple sensitive attributes. Note that we only consider frequent rule set with sufficient support (which characterize a sufficiently large group). In the second phase, the rule set R effectively partitions the samples into two groups, i.e., $samples_r$, which satisfies the rules, and $samples_{\neg r}$, which does not. The key intuition behind is to develop effective criteria to automatically mine interpretable rules that are practical and relevant in the real-world applications. Then, we measure if the model suffers from group discrimination (against the groups partitioned by the rule set) by measuring the group discrimination score. Note that solely relying on the training samples might not be enough to accurately measure such a score. We thus propose to apply a standard data augmentation method, i.e., imposing minor perturbation on the available seed samples to generate new samples, and obtain an accurate estimation of the group discrimination score (with bounded errors). The testing results of the first two phases are thus the identified subtle group discrimination (characterized by the rules) and their corresponding group discrimination score (with bounded errors). For example, we test the model trained on the **Crime** [33] dataset, which predicts whether the violent crimes per population in a specific community is high. The interpretable rule set found by TestSGD shows that it discriminates

[1] We use "feature"/"attribute" interchangeably.

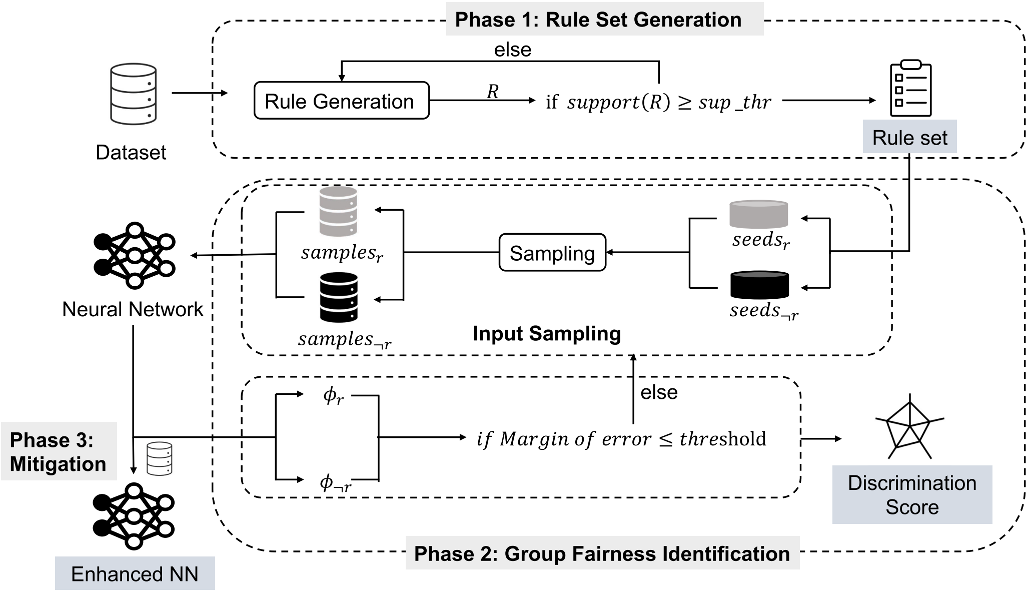

Fig. 1. An overview of TestSGD.

against communities in which the percentage of Caucasian population is lower than 80% and the percentage of females who are divorced is higher than 40%, with a 60.7% group discrimination score, i.e., it is 60.7% more likely to predict high crime rate for such a community. In the last phase (optional, depending on whether the identified discrimination is considered to be harmful), TestSGD leverages the testing results to mitigate the identified subtle group discrimination. That is, to improve group fairness, we generate new samples according to the condition under which discrimination exists and retrain the original model.

TestSGD is implemented as an open-source software [48]. We evaluate our TestSGD on eight models trained on widely adopted datasets including both structured data and text data. The experimental results show TestSGD is effective in identifying and measuring subtle group discrimination. The results also show that *subtle group discrimination does exist in all of these eight models and sometimes to a surprising level that has never been revealed before*. For instance, the model trained on the **COMPAS** [5] dataset is much less likely to predict Hispanic males older than 40 years old as criminals with high recidivism risk. Furthermore, our experiments show that the testing-guided discrimination mitigation is useful. That is, we can mitigate identified subtle group discrimination for all models without sacrificing the accuracy.

In a nutshell, we summarize our main contributions as follows:

- We propose a method to automatically generate an interpretable rule set to identify subtle group discrimination in neural networks, applicable for both structured and text data;
- We develop a theoretical bound for accurately sampling and estimating the group discrimination score against two groups.
- We show that we can generate samples systematically based on the interpretable rule set to mitigate subtle group discrimination.

2 BACKGROUND

Our goal is to develop a black-box method to identify subtle group discrimination in a user-provided neural network model. Our method supports neural networks trained on two different kinds of data, i.e., structure data and text data. Our method does not require the inner details of the neural network. That is, the neural network is viewed as a function $M : R^p \rightarrow R^q$ that maps an input $x \in R^p$ to an output $y \in R^q$. Furthermore, we focus on deep feed-forward neural networks and recurrent neural networks.

2.1 Input Type

First, we define two different data, i.e., structure data, text data, and their corresponding sensitive attributes that are used to evaluate the discrimination of the neural networks.

A sample of structured dataset is composed of a set of features, i.e., a feature vector. A feature can be categorical (i.e., with a fixed set of values) or continuous (i.e., with a certain value range). We define the structure data and the corresponding sensitive attributes as follows:

Definition 2.1 (Structured Data). A structured data x contains N features $\{x_1, x_2, \ldots, x_N\}$, where $\forall x_i, x_i \in L_i$, where L_i is a set of feature values. We write $S = \{s_1, s_2, \ldots, s_n\}$ to denote the set of sensitive attributes in x, where $n < N$.

The text data is composed of a set of terms. We define the sensitive attribute of text data based on the presence of sensitive terms. Note that there could be different categories of sensitive terms, e.g., terms referring to race, religion, or ethnicity. We define the text data and the corresponding sensitive attributes as follows:

Definition 2.2 (Text Data). A text data x contains a sequence of terms $\{x_1, x_2, \ldots, x_N\}$. We write $S = \{s_1, s_2, \ldots, s_n\}$ to denote a set of categories of sensitive terms, where $n < N$ and write T denote a set of sensitive terms $\{t_1, t_2, \ldots, t_k\}$, where $t_j \in s_i$ for some i, for all $j \in [1, k]$ and $t_j \in x$.

2.2 Fairness Definitions

To define our problem, we define fairness and the concept of group discrimination score. There are multiple definitions of fairness [12, 14, 16, 23, 25, 47]. Here, we briefly review two well-studied fairness definitions, i.e., individual fairness and group fairness. We adopt the term "discrimination score" to quantify the group fairness from existing studies [16]. Note that the higher the discrimination score is, the more serious the discrimination is. **Individual fairness** focuses on specific pairs of individuals. Intuitively, individual discrimination (unfairness) occurs when two individuals that differ by only certain sensitive attribute (such as gender or race) are treated differently, i.e., with a different predicted label. This notion is widely used to search discriminatory instances that differ only in those sensitive characteristics [41, 49, 50]. There are also plenty of works on learning models that are more likely to avoid individual discrimination [36].

Group fairness, also known as statistical fairness, focuses on sensitive groups such as ethnic minority and the parity across different groups based on some statistical measurements [7, 16, 25, 40, 47]. A classifier satisfies this metric if the samples in the sensitive group have a positive classification probability or true positive probability that is similar with or equal to that of the insensitive group.

In this work, we focus on group fairness for its relevance in many neural network applications. It is also a fairness notion that is easy to interpret. In the following, we provide a formal definition of group fairness based on positive classification rate measurement.

Definition 2.3. Let M be a neural network model; l be a (favorable) prediction; and ξ be a positive constant. Let G be a group identified by certain condition ϕ on sensitive attributes S. G can be defined as a set of samples $\{x|x \vDash \phi\}$, where $x \vDash \phi$ means x satisfies condition ϕ. We say M satisfies group fairness, with respect to ξ and G, if and only if

$$|\, P(M(x) = l \mid x \in G) - P(M(x) = l \mid x \notin G) \,| \;\leq \xi. \tag{1}$$

Note that, in some cases, the model may be fair overall but unfair under some specific "subtle" conditions. For example, the model is fair considering gender attribute if it approves half of the loans from female or male applicants. However, when we consider both gender and race, the model may show discrimination. For example, it approves loans for far less a percentage of Hispanic female individuals, compared to the remaining group. In this setting, we say that the model discriminates against Hispanic females (if we show that the testing results have sufficient statistical confidence).

3 PROBLEM DEFINITION AND RESEARCH METHOD

In the following, we define our research problem and briefly discuss how we aim to solve the problem and evaluate our approach.

Our problem is to develop a systematic method for identifying subtle group discrimination. That is, given a neural network model M, as well as a constant threshold ξ, we aim to generate a condition ϕ such that M is unfair with respect to the group identified by ϕ. The condition ϕ must satisfy the following conditions:

- It must be constituted by variables representing sensitive attributes.
- It must be human-interpretable, so our analysis result can be presented for human decision-making.
- It must identify a group of non-trivial size.

In addition, our method must support both structured data as well as text data. Furthermore, we would like our method to generate results with certain correctness guarantee, e.g., the chance of reporting non-existing discrimination is low.

Inspired by rule-based models, which are widely used to learn interpretable models [28, 34], we aim to solve the above problem by generating ϕ in the form of rules (a.k.a. constraints) that are understandable by human beings and also concrete enough to show model prediction differences between different groups. The rules should be constituted by the input features, without relying on any latent variables or representations. We define ϕ to be the conjunction of one or more rules, each of which is constituted by only one sensitive attribute. Furthermore, to limit the search space as well as to make sure the generated rules are interpretable, we limit each rule on continuous features to be of the form of a linear inequality, e.g., $age \geq 30$ is a possible rule but $age\ is\ multiples\ of\ 7$ is not. Last, to identify a discriminatory subgroup of non-trivial size, we use a support value to calculate the frequency of samples that satisfy the condition described by ϕ.

To test whether each generated rule set ϕ identifies certain group discrimination, we split the whole input space into two subgroups according to ϕ, i.e., samples satisfying ϕ and the rest. We evaluate the discrimination by calculating the discrimination score defined according to Definition 2.3. To make sure that the discrimination we discover is highly likely in the actual system, we propose a sampling-based approach to estimate the probability of predicting a certain label within a given group. That is, we sample inputs with a distribution that is similar to the data distribution of the training dataset. Such a method allows us to generate an estimation with certain level of statistical confidence, i.e., with a bound on the error. Note that it is not as simple as adopting existing techniques such as hypothesis testing [43, 44]. This is because the group discrimination score is the difference between two estimations (i.e., one for the individual in the group and the other for those not in the group). We solve this problem by establishing a conservative error bound on the difference based on the error bounds for the two estimations.

Our method is evaluated in multiple aspects. First, we aim to evaluate how effective our approach is in terms of identifying subtle group discrimination, measured using the identified discriminatory subgroups and their respective discrimination scores. Second, we aim to evaluate the efficiency of our approach by measuring the execution time. Third, since identifying subtle group discrimination is often not the end of the story in practice, we aim to evaluate whether our testing results can be used to mitigate the identified discrimination. That is, by generating additional samples to retrain the model, we evaluate whether we can mitigate the identified subtle discrimination by measuring the discrimination score and accuracy changes before and after the retraining.

4 THE STEPS OF TESTSGD

In this section, we describe how TestSGD works. There are three main steps, i.e., learning a rule set, identifying group discrimination based on the learned rule set, and discrimination mitigation. The first step is discussed in Section 4.1, while the second step is covered in Sections 4.2 and 4.3. Finally, the last step is discussed in Section 4.4. The inputs for our method include a machine learning model M, its training set D, and a set of sensitive attributes S. The output is the subtle group discrimination represented as a rule set characterizing the discriminated group and the corresponding group discrimination score. Last, based on identified subtle group discrimination, we adopt a data-augmentation-based approach to retrain the model to improve the fairness.

4.1 Generating Frequent Rule Sets

To identify discrimination against certain groups, we first need a way of characterizing a group. In this work, we characterize the groups based on a set of rules, each of which constrains one sensitive attribute. In the following, we present how we generate rules for sensitive attributes of both structured data and text data:

In terms of categorical features x_i in structured data such as gender or race, in general, a rule can be defined as a subset of the possible feature values L_i. For instance, given the sensitive attribute of race that has five values, i.e., Caucasian, Black, Hispanic, Asian, and other-race, a rule can be any set containing one to four of these five values. In total, we have 30 rules. For continuous features x_i in structure data such as age or percentage, there may be too many possible values to enumerate, i.e., the domain of L_i is too large. Thus, we apply techniques such as binning to turn continuous features into categorical features. Here, we divide the original value range into K intervals with equal width. We set K as 10 in our experiments. For example, we divide age attribute ranging from 0 to 99 into 10 equal intervals. Then, we consider each interval as a single value (e.g., 20–29 is considered the same value) and consider a set containing adjacent values as a rule, e.g., the two values representing $10 \leq age < 20$ and $20 \leq age < 29$ become $10 \leq age < 29$.

For textual dataset, defining rules is not that straightforward. In this work, we define the rules based on the presence of identity terms T, a.k.a. sensitive terms (refer to Definition 2.2) that refer to people with specific demographic characteristics. For each sensitive category s, we define a rule that intuitively means that the text sample contains a term t, where $t \in s$. In this work, we use a set of 48 terms created in Reference [13] as the sensitive terms that can be classified into four categories, i.e., gender, race, religion, and age. The sensitive terms are shown in Table 1. For example, when we consider the *gender* feature for text dataset, there are 14 sensitive terms and thus 14 rules are generated. Only when sensitive terms exist in the textual data do we consider that it has the corresponding sensitive attributes. Note that, similar to Reference [13], we consider gender-related terms and sexual orientation-related terms as one category and define all these terms as *gender* attribute for text dataset.

Table 1. Identity Sensitive Terms

Sensitive Attributes	Identity Terms
gender	bisexual, female, gay, heterosexual, homosexual, lesbian, lgbt, lgbtq, male, nonbinary, queer, straight, trans, transgender
race	african, african american, american, asian, black, canadian, chinese, european, hispanic, indian, japanese, latina, latino, latinx, mexican, middle eastern, white
religion	atheist, buddhist, catholic, christian, jewish, muslim, protestant, sikh, taoist
age	elderly, middle aged, millennial, old, older, teenage, young, younger

Once a set of rules is identified, we then characterize a group based on a rule set. Each element of a rule set is a rule constraining one sensitive attribute. Intuitively, a rule set partitions the input space of M into two disjoint groups, i.e., those who satisfy all the rules in the rule set and the rest. If these two groups have a significant different probability of being predicted favorably by the model M, then we successfully identify a subtle discrimination.

Note that a naive approach is to enumerate all possible rules based on one sensitive attribute and combine them arbitrarily. Such an approach is both infeasible and undesirable. First, there can be enormous combinations of the rules. Second, not all rule sets are interesting. For instance, a rule set may be $\{age \geq 100, gender = Male\}$. A discrimination found against the group identified by this rule set is likely to be due to the limited data.

We thus only consider frequent rule sets among all possible combinations of rules. A frequent rule set is a set of rules that is satisfied by a group with a size more than certain threshold. Formally, given a rule set R, the *support* for R is the frequency of the number of samples that satisfy all rules in rule set R. Given a *support threshold* θ (i.e., a percentage), we say that R is *frequent* if its *support* is no less than θ. In the following, we present how to identify a set of frequent rule sets for structured and text data:

For each sensitive attribute s, let R^s be the set of rules concerning s. A rule set R is composed of rules for each sensitive attribute, i.e., $R = \{r^{s_1}, r^{s_2}, \ldots, r^{s_n}\}$ where $r^{s_i} \in \{R^{s_i} \cup \emptyset\}$ and $R \neq \emptyset$. R is frequent if and only if $support(R) \geq \theta$ where $support(R)$ is defined as follows:

$$\text{support}(R) = \frac{\#\{d \in D | \forall r \in R.\ d \vDash r\}}{\#D}, \tag{2}$$

where $\#X$ of a set X is the number of elements in X; and $d \vDash r$ means that d satisfies r.

Example. Consider the structured dataset **Census Income** [35]. It has three sensitive attributes, i.e., gender, race, and age. Each feature has a set of values. The following constitutes a rule set:

$$\{gender = Male, race = White, 40 \leq age < 60\}.$$

Rule sets for text data are defined differently. Recall that each rule is a proposition on whether the text contains certain sensitive term. Formally, given the set of the categories of sensitive terms S, a rule set R is then a set of sensitive terms $\{r_1, r_2, \ldots, r_m\}$, where $r_k \in s_i$ for some i, for all $k \in [1, m]$ and $m \leq n$. The support of R is defined as follows:

$$\text{support}(R) = \frac{\#\{d \in D | \forall r \in R.\ contains(d, s_r)\}}{\#D}, \tag{3}$$

where s_r is the sensitive category referring to r and $contains(d, s_r)$ is a proposition that is true if and only if d contains at least one term in the category s_r.

ALGORITHM 1: $FrequentRuleSets(D, S, sup\_thr)$ where D is the training set, S is the sensitive attributes, θ is the support threshold

```
1: $single\_rules \leftarrow \{\}$, $rule\_sets \leftarrow \emptyset$
2: for each $s$ in $S$ do
3:     $rules \leftarrow \{r_1, r_2, \ldots\}$
4:     $single\_rule[s] = rules$
5: end for
6: $all\_single\_rules \leftarrow \{single\_rule[s] \cup \emptyset\}$ for all $s \in S$
7: $rule\_sets \leftarrow combinations(all\_single\_rules)$
8: $all\_rule\_sets \leftarrow \{R\ for\ R\ in\ rule\_sets\ if\ support(D, R) \geq \theta\}$
9: return $all\_rule\_sets$
```

Example. Consider the text dataset **Wikipedia Talk Pages** [45]. We have two categories of sensitive terms, e.g., gender and race. For each category, we have a set of corresponding sensitive terms as shown in Table 1. The following constitutes a rule set:

$$\{\text{“}bisexual\text{”}, \text{“}Caucasian\text{”}\}$$

Algorithm 1 shows the exact steps in generating all possible rule sets. At line 1, we first initialize a dictionary *single_rules* and an empty set *rules_sets*. During the loop from line 2 to 5, we generate all possible 1-feature rules for each sensitive attribute as discussed above. The set $single\_rule[s]$ contains all single rules r generated in line 3 for sensitive attribute s. At line 6, we generate a set of all 1-feature rules. Then, we generate all possible rule sets at line 7. Note that *rule_sets* generated in line 7 is restricted to contain only one single rule describing each feature from set *all_single_rules*. Last, at line 8, we only keep those rule sets that have a support value no less than θ.

In general, given a dataset with K sensitive attributes and at most N rules for each sensitive attribute, the number of rule sets is N^K in the worse case. For example, we have 2 gender-related single rules, 5 race-related 1-feature rules, and 10 age-related 1-feature rules, there are 17 rule sets when considering one sensitive attribute, 80 rule sets when considering two sensitive attributes, and 100 rule sets when considering all sensitive attributes. So, in total, there are 197 possible rule sets.

4.2 Identifying Group Fairness

For each group identified by a rule set, we then measure the discrimination against the group. That is, we aim to compute the probability of predicting certain label by M on those samples in the group, and that on those samples not in the group, and measure the difference. The score is the group discrimination score, which varies from 0 (i.e., no difference) to 1 (i.e., completely different). Formally,

Definition 4.1 (Group Discrimination Score). Let R be a rule set. Let l be a (favorable) label. The group discrimination score with respect to R and l is $|prob(R, l) - prob(\neg R, l)|$, where $prob(R, l)$ is the probability of predicting l given samples satisfying R, $\neg R$ identifies samples not satisfying R.

We remark that this definition is similar to the CV score [12] and multivariate group discrimination score [16]. However, the former is limited to binary input types and the latter is limited to categorical input types. In comparison, our discrimination score supports both structured data and text data.

Example. Take a model trained on the (structured) **Census Income** dataset as an example. The model predicts whether the income of an adult is above \$50,000 annually, i.e., “True” means above the threshold and "False" means otherwise. Assume the rule set is

$$\{gender = Male, race = White, 40 \leq age < 60\}.$$

Assume that the model predicts 28% of individuals in this group with "True" and 10% of the remaining population with "True." The model's group discrimination score, with respect to the rule set and the prediction, is 18%.

Given a rule set, measuring the group discrimination score requires us to measure $prob(R, l)$ and $prob(\neg R, l)$, which is non-trivial, since exhaustively enumerating all samples is infeasible due to the enormous input space. However, measuring it based on a limited number of samples may yield inaccurate results. In the following, we propose an approach to compute group discrimination scores with a statistical confidence guarantee. Formally, we would like to measure the group discrimination score f within a margin of error ϵ under a certain confidence δ, such that $prob(|f - \hat{f}| > \epsilon) < 1 - \delta$, where $\hat{f}$ is the real group discrimination score over all possible samples.

Algorithm 2 shows how we measure the group discrimination score. We maintain two sets of samples, i.e., $samples_r$, which contains samples satisfying R, and $samples_{\neg r}$, which contains samples not satisfying R. At line 1, we set both $samples_r$ and $samples_{\neg r}$ to be empty, error margin ϵ to be infinity, and the number of generated samples as 0. During the loop from line 2 to 16, we keep generating samples and calculating group discrimination score until the error margin ϵ is no more than the given error threshold $error\_thr$. From line 3 to line 6, we generate new samples for $samples_r$ and $samples_{\neg r}$, respectively, using a function $Sample$. We remark that the generated samples should follow the original data distribution (i.e., that of the training dataset). We present details on how we sample on structured and text dataset in the next subsection.

At line 7, we increase num by 1. After generating a sufficient number of samples, we check the error margin ϵ from line 9 to 15. We first calculate the probability of predicting l at line 9 and 10 for two sets of samples. Then, at line 11, we calculate the error margin ϵ on the group discrimination score. We explain why it is calculated this way below. If ϵ is less than or equal to $error\_thr$, then the stopping criteria is satisfied (as in lines 12 and 13). Last, at line 17, we return the absolute difference between ϕ_r and $\phi_{\neg r}$ as the group discrimination score.

In the above algorithm, we estimate the error margin of the group discrimination score based on an estimation of $prob(R, l)$ and $prob(\neg R, l)$. The complication is that both $prob(R, l)$ and $prob(\neg R, l)$ carry certain error margin, which may magnify the error margin for the group discrimination score. In the following, we prove that line 11 in the above algorithm allows us to conservatively estimate the error margin of the group discrimination score:

THEOREM 4.2. *Assume that ϕ_r satisfies the following:*

$$prob(|\phi_r - \hat{\phi_r}| > \epsilon_r) < 1 - \delta_r, \tag{4}$$

where ϵ_r and δ_r are constants. Similarly, $\phi_{\neg r}$ satisfies the following:

$$prob(|\phi_{\neg r} - \hat{\phi}_{\neg r}| > \epsilon_{\neg r}) < 1 - \delta_{\neg r}. \tag{5}$$

Then the following is satisfied:

$$prob(|f - \hat{f}| > \epsilon_r + \epsilon_{\neg r}) < 1 - \delta_r \delta_{\neg r}. \tag{6}$$

PROOF. Since $prob(|\phi_r - \hat{\phi_r}| > \epsilon_r) < 1 - \delta_r$ and $prob(|\phi_{\neg r} - \hat{\phi_{\neg r}}| > \epsilon_{\neg r}) < 1 - \delta_{\neg r}$, we have:

$$prob(|\phi_r - \hat{\phi_r}| \leq \epsilon_r) \geq \delta_r$$

$$prob(|\phi_{\neg r} - \hat{\phi}_{\neg r}| \leq \epsilon_{\neg r}) \geq \delta_{\neg r}.$$

ALGORITHM 2: $GroupFairnessScore(D, M, R, sample\_thr, error\_thr)$ where D is the training dataset; M is the machine learning model; R is a rule set, $sample\_thr$ is the number of generated inputs threshold; $error\_thr$ is error margin threshold

```
1:  $samples_r \leftarrow \emptyset$, $samples_{\neg r} \leftarrow \emptyset$, $\epsilon \leftarrow +\infty$, $num \leftarrow 0$
2:  while $\epsilon > error\_thr$ do
3:      $x \leftarrow Sample(D, R)$
4:      $x' \leftarrow Sample(D, \neg R)$
5:      $samples_r \leftarrow samples_r \cup x$
6:      $samples_{\neg r} \leftarrow samples_{\neg r} \cup x'$
7:      $num \leftarrow num + 1$
8:      if $num > sample\_thr$ then
9:          $\phi_r \leftarrow \#\{i \in samples_r | M(i) = l\}/num$
10:         $\phi_{\neg r} \leftarrow \#\{i \in samples_{\neg r} | M(i) = l\}/num$
11:         $\epsilon \leftarrow z \times \sqrt{\frac{\phi_r(1-\phi_r)}{num}} + z \times \sqrt{\frac{\phi_{\neg r}(1-\phi_{\neg r})}{num}}$
12:         if $\epsilon \leq error\_thr$ then
13:             break
14:         end if
15:     end if
16: end while
17: return $f \leftarrow |\phi_r - \phi_{\neg r}|$
```

Hence,

$$prob(|(\phi_r - \hat{\phi}_r) - (\phi_{\neg r} - \hat{\phi}_{\neg r})| \leq \epsilon_r + \epsilon_{\neg r}) \geq$$

$$prob(|\phi_r - \hat{\phi}_r| \leq \epsilon_r) \cdot prob(|\phi_{\neg r} - \hat{\phi}_{\neg r}| \leq \epsilon_{\neg r}) \geq \delta_r \delta_{\neg r}$$

and

$$prob(|(\phi_r - \hat{\phi}_r) - (\phi_{\neg r} - \hat{\phi_{\neg r}})| > \epsilon_r + \epsilon_{\neg r}) < 1 - \delta_r \delta_{\neg r}$$

$$prob(|(\phi_r - \phi_{\neg r}) - (\hat{\phi}_r - \hat{\phi}_{\neg r})| > \epsilon_r + \epsilon_{\neg r}) < 1 - \delta_r \delta_{\neg r}.$$

According to Definition 4.1, group discrimination score $f = \phi_r - \phi_{\neg r}$. Thus,

$$prob(|f - \hat{f}| > \epsilon_r + \epsilon_{\neg r}) < 1 - \delta_r \delta_{\neg r}. \qquad \square$$

The above theorem provides a theoretical guarantee on the statistical confidence for the group discrimination score estimation. That is, based on the Equation (6), the fairness level for discrimination score f is $\delta_r \delta_{\neg r}$ and the margin of error is the sum of two margin of errors as $\epsilon_r + \epsilon_{\neg r}$. Each ϵ is calculated by:

$$\epsilon = z \times \sqrt{\frac{\phi(1-\phi)}{num}}, \tag{7}$$

where z is the value from the standard normal distribution for a certain confidence level δ (e.g., for a 95% confidence level, $z = 1.96$). So, the final margin of error for discrimination score f is shown in line 11 of Algorithm 2. Based on the result, we derive the stopping criteria, as shown in lines 12 and 13 of Algorithm 2.

The above shows how we compute the group discrimination score for one rule set. Given multiple rule sets, we systematically compute the discrimination score for each rule set with Algorithm 2 and then rank the rule sets according to the resultant group discrimination score. If the group discrimination score of certain rule set is more than a given tolerable threshold, then we report that discrimination is identified.

Example. Take a model trained on the (structured) **Census Income** dataset as an example. We fixed the confidence level to 95% and the corresponding z-value is 1.96. We set the sampling threshold $sample\_thr$ as 1,000 and the error of margin threshold $error\_thr$ as 0.05. We are given a rule set

$$\{gender = Male, race = White, 40 \leq age < 60\}.$$

First, we sample 1,000 inputs as $samples_r$ using *Sample* function that represents white males who are older than 40 but younger than 60. Then, we sample another 1,000 inputs as $samples_{\neg r}$ using *Sample* function that represents the rest individuals. We observe that 283 samples in $samples_r$ are labeled as "True," while only 91 samples in $samples_{\neg r}$ are labeled as "True." So, ϕ_r is 28.3% and $\phi_{\neg r}$ is 9.1%. According to Algorithm 2, ϵ_r is 0.028 and $\epsilon_{\neg r}$ is 0.018. So, the margin of error ϵ for discrimination score is 0.046. Since ϵ is less than 0.05, we stop sampling. Finally, the group discrimination score is computed as 19.2% with 90.25% confidence.

4.3 Input Sampling

As discussed above, Algorithm 2 requires us to sample inputs with a distribution that is similar to the data distribution of the training dataset. As shown in Reference [19], modern machine learning models mostly rely on the i.i.d. assumptions. That is, the training and test set are assumed to be generated independently from an identical distribution. It is more likely for machine learning models to predict identically distributed data correctly.

While it is impossible to know the actual data distribution, we aim to generate additional samples from a distribution as close as possible to the distribution of the training set. For structured data, instead of generating feature vectors randomly, we generate new samples by adding tiny perturbations on original samples uniformly. The perturbation is added to one randomly selected non-sensitive attribute with randomly selected perturbation direction, and the perturbation size is 1 for integer variables or 0.01 for decimal variables. Formally, given the rule set R, we first search a seed instance from the dataset D as $seed = \{x_1, x_2, \ldots, x_N\}$, where $\forall r \in R.\ seed \vDash r$. Then, we randomly select a non-sensitive attribute x_k, where $x_k \notin S$. We perturb x_k as $x'_k = x_k + dir \cdot s\_pert$, where $dir \in [-1, 1]$ and s_pert is the perturbation size.

For text data, we generate new samples by replacing sensitive terms with a different term in the same sensitive term category. For example, when we test the machine learning model trained on the **Wikipedia Talk Pages** dataset, given a rule set {"*gay*"}, we need to generate additional comments containing the term "gay." First, we search all comments containing gender-related sensitive terms such as "lesbian" and "bisexual," as defined in Table 1. Then, we replace these terms in the original comments with the term "gay" to generate new comments. That is, we can generate "I am a gay" from an original comment "I am a lesbian." The reason why we use text replacement instead of text perturbation, as in the case of structured data, is that perturbing texts with synonyms (as proposed in Reference [38] for adversarial attacks) is ineffective to generate the texts in the desired group, since these models are much more complicated. We further remark that, for structured data, perturbing sensitive attributes risks shifting the original data distribution, especially for data with a small number of features. This is less a concern for text data that contains many features. Our text generation method also has the benefit of mitigating the influence of data imbalance that may cause unintended bias [13]. Formally, given the rule set $R = \{r_1, r_2, \ldots, r_m\}$, we first search a seed instance from the dataset D as $seed = \{x_1, x_2, \ldots, x_N\}$, where $\forall r \in R.\ contains(seed, s_r)$, where s_r is the sensitive category referring to r and $contains(d, s_r)$ is a proposition that is true if and only if d contains at least one term in the category s_r. Then, we replace the term x_i to term r_j, for all $r_j \in R$ and $x_i \in s_{r_j}$.

4.4 Discrimination Mitigation

After identifying group discrimination, we can mitigate the subtle group discrimination (represented by the given rule sets) through retraining. Here, we proposed a data-augmentation approach to improve fairness merely as an example to show the usefulness of our approach. We remark that there are alternative approaches that target reducing algorithmic bias as well, which are often referred to as in-processing methods [24, 37]. In fact, it is perceivable that our results can be integrated with those approaches as well, e.g., by training with additional loss functions based on the rule sets identified using our method.

In particular, we generate additional instances satisfying the rule set with the sampling approach described in Section 4.3. We only select those generated instances with the opposite label as the additional data for retraining. For example, the model trained on **COMPAS** is more likely to predict elderly males who are Hispanic or other race with "False" label. We thus apply the *Sample* function to generate instances satisfying the condition that are labeled as "True" according to the original model. Afterward, we retrain the original model with these additional instances. Note that we gradually increase the number of additional instances from 50 to 10% of original dataset size to achieve the lowest discrimination score without decreasing the accuracy of the retrained model. We evaluate the effectiveness of our discrimination mitigation approach by testing the subtle discrimination with respect to the same rule set to see the discrimination score improvement and accuracy changes.

5 IMPLEMENTATION AND EVALUATION

We have implemented TestSGD as a self-contained software toolkit based on Tensorflow [1] with about 6k lines of Python code. Our implementation is available at Reference [48].

5.1 Experiment Subjects

Our experiments are based on eight models trained with the following benchmark datasets. These datasets have been widely used as evaluation subjects in multiple previous studies on fairness [13, 16, 29, 36, 49, 50].

- **Census Income** [35]: The dataset contains more than 30,000 samples and is used to predict whether the income of an adult is above $50,000 annually. The attributes *gender*, *race*, and *age* are sensitive attributes.
- **Bank Marketing** [31]: The dataset contains 45,000+ samples and is used to train models for predicting whether the client would subscribe a term deposit. Its sensitive attribute is *age*.
- **German Credit** [21]: This is a small dataset with 600 samples. The task is to assess an individual's credit. The sensitive attributes are *gender* and *age*.
- **COMPAS** [5]: This dataset contains 7,000+ samples. The task is to predict whether the recidivism risk score for an individual is high. The sensitive attributes are *gender*, *race*, and *age*.
- **Crime** [33]: This dataset contains almost 2,000 data for communities within the US. The task is to predict whether the violent crimes per population in a specific community is high. Since this dataset records population statistics, their sensitive attributes are shown in multiple attributes with percentage values. Here, we extract all gender/race/age-related attributes to learn rule sets.

Table 2. Parameters of the Experiments

Parameters	Value	Description
θ	5%	support threshold
sample_thr	1,000	sampling threshold
δ	95%	confidence level
error_thr	0.05	error margin threshold
z	1.96	z value
s_pert	1	perturbation size for integer variables
s_pert	0.01	perturbation size for decimal variables

Table 3. Dataset and Models of Experiments

Dataset	Model	Accuracy	Favorable Prediction
Census Income	Six-layer Fully connected NN	86.13%	Income is above 50,000
Bank Marketing	Six-layer Fully connected NN	91.62%	Subscribe a Term Deposit
German Credit	Six-layer Fully connected NN	100%	Good Credit
COMPAS	Six-layer Fully connected NN	78.99%	High Recidivism Score
Law School	Six-layer Fully connected NN	95.19%	Pass Exam
Crime	Six-layer Fully connected NN	92.52%	High Crime Rate
Wikipedia Talk Pages	CNN Long Short-term memory network	93.89%	Toxic Comment
IMDB	CNN Long Short-term memory network	86.68%	Good Review

- **Law School** [6]: This dataset has more than 20,000 application records and is used to predict whether a student passes the bar exam. The attributes *race* and *gender* are sensitive attributes.
- **Wiki Talk Pages** [45]: This is a textual dataset containing more than 100,000 Wikipedia TalkPage comments. The task is to predict whether a given comment is toxic. Note that the sensitive attributes are defined in Table 1.
- **IMDB** [30]: IMDB dataset contains 50,000 movie reviews. The task is to predict whether a given sentence is a positive review. The sensitive attributes are defined in Table 1.

For the first six structured datasets, we train a six-layer feed-forward neural network using the exact same configuration as reported in the previous studies [49, 50]. For the last two textual datasets, we train a **convolutional neural network (CNN)** combined with **long short-term memory (LSTM)**. Table 2 shows the value of parameters used in our experiment to run TestSGD. The details of trained models are shown in Table 3. The accuracy of the trained models is expectedly similar to what is reported in the previous studies. All experiments are conducted on a server running Ubuntu 1804 with one Intel Core 3.10 GHz CPU, 32 GB memory, and two NVIDIA GV102 GPUs. To mitigate the effect of randomness, all the results are the average of three runs.

5.2 Research Questions

Then, we evaluate the performance and utility of TestSGD in identifying subtle group discrimination on neural networks trained on the above datasets. Specifically, we ask the following research questions:

- **RQ1: Effectiveness of Identifying Discrimination:** Is our method effective in identifying subtle group discrimination of a given machine learning model?
- **RQ2: Efficiency of Identifying Discrimination:** Is our method efficient?
- **RQ3: Effectiveness of Mitigating Discrimination:** Can we mitigate subtle discrimination using our testing results?

Table 4. Rule Sets and Discrimination Scores for Neural Networks

Dataset	top 1		top 2		top 3	
	Rule Set	Discrimination Score $(\phi_r, \phi_{\neg r})$	Rule Set	Discrimination Score $(\phi_r, \phi_{\neg r})$	Rule Set	Discrimination Score $(\phi_r, \phi_{\neg r})$
Census Income	gender = male, $40 \leq$ age < 80, race = White or Asian-Pac-Islander	20.2% (29.9%, 9.7%)	gender = male, $40 \leq$ age < 70, race = White or Amer-Indian-Eskimo	19.4% (28.9%, 9.5%)	gender = male, $40 \leq$ age < 80, race = White, Asian-Pac-Islander or Amer-Indian-Eskimo	18.4% (26.9%, 8.5%)
Bank Marketing	$10 \leq$ age < 90	38.2% (3.3%, 41.5%)	$10 \leq$ age < 70	22.8% (26.6%, 3.8%)	$10 \leq$ age < 60	20.5% (4.7%, 25.2%)
German Credit	gender = female, $60 \leq$ age < 70	21.9% (72.5%, 50.6%)	gender = female, $60 \leq$ age < 80	21.8% (70.5%, 48.7%)	gender = male, $40 \leq$ age < 80	15.5% (52.6%, 47.1%)
COMPAS	gender = male, age ≥ 40, race = Hispanic or other race	62.4% (20.7%, 83.1%)	gender = male, $40 \leq$ age < 60, race = Hispanic or other race	62.3% (20.3%, 82.6%)	gender = male, $50 \leq$ age < 60, race = Hispanic	62.3% (19.5%, 81.8%)
Law School	gender = male, race = Asian or Black	15.0% (84.5%, 99.5%)	gender = female, race = Asian or Black	11.1% (88.8%, 99.9%)	gender = female, race = Black	10.2% (89.7%, 99.9%)
Crime	FemalePctDiv ≥ 0.4, racePctWhite ≤ 0.8	60.7% (83.8%, 23.2%)	FemalePctDiv ≥ 0.5, racePctWhite ≤ 0.8	59.6% (87.0%, 27.4%)	FemalePctDiv ≥ 0.5, racePctWhite ≤ 0.6	59.5% (94.6%, 35.1%)
Wiki Talk Pages	"gay," "taoist"	6.5% (13.0%, 6.5%)	"gay," "protestant"	5.4% (12.9%, 7.5%)	"gay," "african american"	5.1% (12.5%, 7.4%)
IMDB	"european," "young"	6.6% (56.0%, 49.4%)	"white," "older"	6.6% (59.1%, 52.6%)	"lgbtq"	6.5% (7.5%, 14.0%)

RQ1: Is our method effective in identifying subtle group discrimination of a given machine learning model?

To answer the question, we systematically apply our approach to the above-mentioned models and measure the results. The results are summarized in Table 4. It shows results on the six models trained on structured data and results on the two models trained on text data. These four columns show datasets, rule sets, group discrimination scores and model accuracies, respectively. The favorable label is "True," the meanings of which are shown in Table 3. More details can be found in the above introduction on the corresponding dataset. Note that for each model, we rank the identified subtle discrimination according to the group discrimination score and we report the top three worst discrimination only. Similar to the example shown in Section 4.2, all results are computed with the same confidence (i.e., 90.25%) and error of bound (i.e., less than 0.05).

We can observe that subtle discrimination does exist in these models, which were never revealed in the previous studies [13, 16, 29, 49, 50]. For instance, the model trained on the **Bank Marketing** dataset predicts only 3.3% of the clients who are older than 10 but younger than 90 would subscribe a term deposit, while predicting 41.5% of clients older than 90 would subscribe a term deposit. All of the top three testing results show the model discriminates against young clients. We remark that although this is unfair according to the definition, it may have its underlying reasons and it is still up to human experts to decide whether it is actual discrimination.

For the models trained on the **Census Income** dataset, **German Credit** dataset, and the **Law School** dataset, they show relatively mild discrimination. In contrast, the model trained on the **COMPAS** dataset shows severe discrimination, with a discrimination score of 62.4%. That is, for Hispanic or other race male individuals who are older than 40, the model is much less likely to predict the recidivism risk as high. For the remaining individuals, the model predicts 83.1% of them have high recidivism risk. Top two and top three test results also show severe discrimination against older Hispanic or other race male individuals. Similarly, the model trained on the **Crime** dataset also shows high discrimination. Different from the first five structured datasets, samples in this dataset have 10 different sensitive attributes, each of which is a decimal ranging from 0.0 to 1.0 representing the percentage of certain population. As shown in the top 1 testing result, when the percentage of divorced females is above 40% and the percentage of Caucasians is below 80%, the model is much more likely to predict that the violent crimes per population in this community is high. All testing results on the model trained on **Crime** dataset suggest that the model discriminates against communities with high percentage of divorced females and low percentage of Caucasians.

In Table 4, the last two rows show the results on models trained on the text data. In general, we observe that the models trained on text dataset show less discrimination. The maximum discrimination score for the model trained on the **Wikipedia Talk Pages** dataset is 6.5%. That is,

Table 5. Time Taken to Identify the Subtle Discrimination

Dataset	Time (seconds)	#rule set
Census Income	869.35	880
Bank Marketing	141.52	34
German Credit	104.85	53
COMPAS	908.5	1,590
Law School	18.46	17
Crime	29,150.01	13,282
Wiki Talk Pages	34,982.28	732
IMDB	69,125.16	876

the model predicts 13.0% of comments containing both “gay” and “taoist” as toxic. For other comments (i.e., those without one of these two terms or both), the model predicts only 6.5% of them as toxic. Top two and top three testing results show that the model discriminates against comments containing both “gay” and “protestant” and comments containing both “gay” and “african american,” respectively. The model trained on the **IMDB** dataset shows a similar level of discrimination. It is more likely to predict reviewers containing “european” and “young” and reviews containing “white” and “older” as positive. It also shows discrimination against reviews containing “lgbtq.” Our conjecture on why the level of discrimination is considerably lower on these models is that each sample in these text datasets often has many features and, as a result, the influence of each term (including sensitive terms) is distributed.

Answer to RQ1: TestSGD is effective in identifying subtle group discrimination in neural networks.

RQ2: Is our method efficient?

To answer this question, we measure the amount of time required to identify the subtle discrimination for each model. The total execution time and the numbers of tested rule sets are shown in Table 5. For all models, the time required to identify the subtle discrimination is less than 20 hours. Furthermore, models trained on structured dataset take considerably less time than those trained on text dataset. That is, models trained on the **Census Income**, **Bank Marketing**, **German Credit**, **COMPAS**, and **Law School** take less than 16 minutes. One exception is the model trained on the **Crime** dataset, which takes more than 8 hours. The main reason is that it has a large number of rule sets, due to a large number of sensitive attributes (i.e., 10), all of which are continuous features. In contrast, both models trained on text dataset take more than 9 hours to finish. The main reason is that generating additional samples for such dataset takes much more time in general. We remark that the sampling procedure can be easily parallelized and, thus, we could significantly reduce the time if it is an issue.

Note that the support threshold θ is set to be 5% in all the above experiments. Intuitively, it means that each rule must be relevant to 5% of the population (although the rule set, which is a conjunction of multiple rules, may impact a smaller population). This hyper-parameter largely determines how many rule sets that we must examine and thus may have an impact on the execution time. We thus conduct additional experiments with different θ values, ranging from 1% to 50%, to evaluate the effect of θ on the execution time and the results. The results on two models, i.e., the model on **Law School** and the model on **COMPAS**, are detailed in Table 6.

Table 6. Effect of Different θ

Dataset	θ	Time (seconds)	#rule sets	Rule Set	Discrimination Score
Law School	1%	46.71	59	gender = male, race = Black	16.3%
	5%	18.46	17	gender = male, race = Asian or Black	15.0%
	10%	17.83	16	gender = male, race = Asian or White	1.0%
	20%	17.83	16	gender = male, race = Asian or White	0.9%
	50%	6.28	2	gender = male, race = other race	0.3%
COMPAS	1%	1,175.79	2,063	gender = male, age ≥ 40 race = Hispanic or other race	62.4%
	5%	908.50	1,590	gender = male, age ≥ 40 race = Hispanic or other race	62.4%
	10%	676.74	1,180	gender = male, age ≥ 20 race = Hispanic or other race	43.9%
	20%	0	0	NULL	NULL
	50%	0	0	NULL	NULL

The table shows the execution time, the number of rule sets, and the worst group discrimination score. We can observe that, the larger a θ we set, the fewer rule sets, the less execution time, and the smaller group discrimination score in general. If the threshold θ is too low, e.g., 1%, then we spend a lot of time testing a huge number of rule sets, which may not be interesting (one such example is $\{gender = Male, age \geq 100\}$). In contrast, if the threshold θ is too high, e.g., 20% or 50%, then there may only exist few or even none rule set (as in the case of the model trained on the **COMPAS** dataset).

We note that different θ may result in different discrimination being identified. For the model trained on **Law School**, the rule set shows that the model discriminates against black or Asian males the most when θ is 5%. However, when we set θ to be 1%, the model is shown to discriminate against black male individuals the most. For the model trained on the **COMPAS** dataset, the model discriminates against Hispanic or other race males who are older than 40 years old most when we set θ to be 5%. However, when we set θ higher (i.e., 10%), the age range is expanded to be over 20 years in the identified rule set. Such a result is expected, as a large θ requires us to find discrimination against a large group. What is considered to be a reasonable value for θ is a complicated question, which should probably be answered by lawmakers.

Answer to RQ2: TestSGD is reasonably efficient.

RQ3: Can we mitigate subtle discrimination using our testing results?

Table 7. Discrimination Mitigation for Neural Networks

Dataset	Rule Set	Before		After	
		Accuracy	Discrimination Score $(\phi_r, \phi_{\neg r})$	Accuracy	Discrimination Score $(\phi_r, \phi_{\neg r})$
Census Income	gender = male, $40 \leq$ age < 80, race = White or Asian-Pac-Islander	86.1%	20.2% (29.9%, 9.7%)	86.2%	10.1% (18.9%, 8.8%)
Bank Marketing	$10 \leq$ age < 90	91.6%	38.2% (3.3%, 41.5%)	90.6%	5.4% (6.9%, 12.3%)
German Credit	gender = female, $60 \leq$ age < 70	100%	21.9% (72.3%, 50.6%)	100%	7.3% (45.9%, 53.2%)
COMPAS	gender = male, age ≥ 40, race = Hispanic or other race	79.0%	62.4% (20.7%, 83.1%)	78.5%	4.2% (80.9%, 85.1%)
Law School	gender = male, race = Black	95.2%	15.0% (84.5%, 99.5%)	95.1%	7.5% (92.3%, 99.8%)
Crime	FemalePctDiv ≥ 0.4, racePctWhite ≤ 0.8	92.5%	60.7% (83.8%, 23.2%)	98.1%	51.4% (90.6%, 39.2%)
Wiki Talk Pages	“gay,” “taoist”	93.9%	6.5% (13.0%, 6.5%)	95.5%	0.4% (8.4%, 8.0%)
IMDB	“european,” “young”	86.7%	6.6% (56.0%, 49.4%)	84.0%	3.3% (43.7%, 40.4%)

To further show the usefulness of our approach, we evaluate whether we can mitigate the identified subtle discrimination using our testing results. The idea is to mitigate the discrimination by retraining, which is described in Section 4.4.

We only consider the top one worst rule sets to mitigate the discrimination. The results are shown in Table 7 for six models trained on additional structured data and two models retrained on additional textual data. We can observe that all models show reduced subtle discrimination and almost the same accuracy. The discrimination scores for retrained models on **Census Income**, **German Credit**, and **Law School** decrease by about half. For the most improvement, the model retrained on the **COMPAS** dataset shows much less subtle discrimination, as the discrimination score decreases by more than 10 times, i.e., from 57.7% to 4.2%. The discrimination score of the model trained on the **Crime** dataset decreases from 60.7% to 51.4%. Relatively, the fairness improvement is not obvious. We believe that it is due to its many continuous sensitive attributes and the large number of features (i.e., each input contains more than 100 attributes). That is, it would require a lot more additional data to improve fairness. In terms of CNN models, the discrimination score decreases from from 6.5% to 0.4% for the model retrained on **Wikipedia Talk Pages** and decreases from 6.6% to 3.3% for the model retrained on **IMDB**.

Answer to RQ3: TestSGD is useful in mitigating the identified subtle group discrimination through retraining.

5.3 Comparison with Baselines

In this section, we compare the effectiveness of TestSGD to the state-of-the-art group discrimination identification approaches, i.e., **THEMIS** [16] and **FairFictPlay** [24].

We identify the following two baselines from literature that can potentially identify similar group discrimination as our work: **(1) THEMIS [16]** calculates group discrimination scores over combinations of multiple features (subgroups) by measuring the difference between the maximum and minimum frequencies of two subgroups on randomly generated samples. Those subgroups can then be regarded as identified discrimination if the score is higher than a threshold. **(2) FairFictPlay [24]** proposed an in-processing algorithm aiming to improve subgroup fairness. The subgroups are identified with user-provided constraints in the form of conjunctions of Boolean attributes, linear threshold functions, or bounded degree polynomial threshold functions over

Table 8. Comparisons between TestSGD, THEMIS, and FairFictPlay

Dataset	TestSGD		THEMIS		FairFictPlay	
	Rule Set	Discrimination Score	Sensitive Attributes' values for Max/Min Proportion	Discrimination Score	Linear Threshold Function	Discrimination Score
Census Income	Gender = male, $40 \le age < 80$, race = White or Asian-Pac_islander	20.2%	[gender = Female, $60 \le age < 70$, race = Asian-Pac_islander] - [gender = Male, $10 \le age < 20$, race = White]	26.6%	Linear Threshold Function	13.9%
Bank Marketing	$10 \le age < 90$	38.2%	[$60 \le age < 70$] - [$10 \le age < 20$]	8.4%	Linear Threshold Function	7.6%
German Credit	gender = female, $60 \le age < 70$	21.9%	[gender = Female, $80 \le age < 90$] - [gender = Male, $10 \le age < 20$]	17.1%	Linear Threshold Function	7.0%
COMPAS	gender = male, $age \ge 40$, race = Hispanic or other race	62.4%	[gender = Female, $10 \le age < 20$, race = Native American] - [gender = Male, $60 \le age < 70$, race = other race]	67.3%	Linear Threshold Function	22.4%
Law School	gender = male, race = Asian or Black	15.0%	[gender = Male, race = White] - [gender = Female, race = Black]	13.5%	Linear Threshold Function	3.7%
Crime	$FemalePctDiv \ge 0.4$, $racePctWhite \le 0.8$	60.7%	-	-	Linear Threshold Function	38.8%

"-" means timeout.

multiple protected features. It supports multiple fairness metrics, i.e., false positive subgroup fairness, false negative subgroup fairness, and statistical parity subgroup fairness. Here, we consider statistical parity subgroup fairness, as it shares the same equality classification rates as our group fairness metrics.

In Table 8, we show the identified group discrimination with TestSGD, THEMIS, and FairFictPlay, respectively, along with the discrimination scores, on the same models trained on structured data (similarly to Table 3). All discrimination scores measure the frequency difference between two groups. We set the timeout as 24 hours.

In the following, we compare our method TestSGD with the alternative approaches from two aspects: First, we compare the effectiveness of the approaches in terms of the identified discriminatory subgroups. Note that THEMIS and FairFictPlay do not aim to identify discrimination subgroups in an interpretable way. Thus, we extract the discriminatory group based on their fairness definitions and the testing results. For example, based on the discrimination score defined in THEMIS, it split the whole input space based on all possible values of each sensitive attribute. Then it calculates the fraction of inputs that are predicted with favorable label in each subgroup. The discrimination score is then calculated as the difference between subgroup with the maximum and that with the minimum fraction. The sensitive attributes' values of the subgroup with the max/min fraction thus represent the discriminatory subgroup. FairFictPlay uses complex linear functions constituted by all the protected features to define discriminatory subgroups. Since these functions are complicated and hard to interpret, we do not show them in the table. Second, we compare the discrimination scores with the baselines. The higher the value is, the more serious the discrimination is.

We have the following observations: (1) TestSGD's testing results represented by rule sets are more interpretable. Similar to TestSGD, THEMIS is able to identify discriminatory subgroups automatically. However, THEMIS identifies two subgroups that are maximally different (in terms of being predicted favorably), while TestSGD identifies subgroups that are predicted differently from the rest. These two approaches thus produce results that are complementary to each other. However, FairFictPlay has no ability to identify discriminatory subgroups in an interpretable way. Note that both THEMIS and FairFictPlay do not support text data. (2) Compared to FairFictPlay, TestSGD identifies discrimination with higher scores (more discriminating). Moreover, TestSGD automatically identifies the discriminated subgroups without any prior knowledge. A further problem with FairFictPlay is that the linear threshold functions may split the whole input space into two extremely imbalanced subgroups, and thus the identified discrimination may be caused by data imbalance [13]. (3) In a close investigation, THEMIS may consume much more time for some specific cases, e.g., the **Crime** dataset. As THEMIS considers combinations of all possible values of sensitive attributes, it becomes inefficient and even infeasible with regard to datasets containing sensitive attributes with continuous values. For example, when testing against the model trained on the **Crime** dataset, THEMIS considers over 40,000,000 combinations.

Table 9. Discrimination Mitigation Comparisons between TestSGD and FairFictPlay

Dataset	Base Accuracy	TestSGD		FairFictPlay	
		Accuracy (Acc Change)	**Discrimination Score** (Score Change)	**Accuracy** (Acc Change)	**Discrimination Score** (Score Change)
Census income	86.1%	86.2% (0.1%↑)	10.1% (10.1%↓)	79.1% (7%↓)	2.9% (11%↓)
Bank Marketing	91.6%	90.6% (1.0%↓)	5.4% (32.8%↓)	89.5% (2.1%↓)	0.2% (7.4%↓)
German Credit	100%	100% 0%	7.3% (14.6%↓)	71.0% (29%↓)	4.1% (2.9%↓)
COMPAS	79.0%	78.5% (0.5%↓)	4.2% (58.2%↓)	73.3% (5.7%↓)	10.9% (11.5%↓)
Law School	95.2%	95.1% (0.1%↓)	7.5% (7.5%↓)	94.9% (0.3%↓)	0.02% (3.7%↓)
Crime	92.5%	98.1% (5.6%↑)	51.4% (9.3%↓)	85.2% (7.3%↓)	20.5% (18.3%↓)

"↑" means value increased, "↓" means value decreased.

Regarding discrimination mitigation, FairFictPlay proposes an in-processing method to mitigate subgroup discrimination, while THEMIS did not propose any discrimination mitigation approach. Here, we compare the effect of discrimination mitigation between TestSGD and FairFictPlay. We adopt TestSGD's and FairFictPlay's discrimination mitigation approaches based on the same models trained on six structured datasets. Note that FairFictPlay does not support models trained on textual data. The results are shown in Table 9. In detail, we first show the original accuracy and then show the accuracy changes and discrimination score changes after adopting the discrimination mitigation approach. While both TestSGD and FairFictPlay are effective in mitigating subgroup discrimination, we observe that TestSGD improves subgroup fairness with less accuracy drop. For all models, the accuracy drops after processing by FairFictPlay. In some cases, FairFictPlay sacrifices accuracy significantly. For example, FairFictPlay improves the discrimination score by 2.9% but decreases the accuracy by 29% with the model trained on **German Credit** dataset.

5.4 Discussions

In the following, we discuss some insights coming from our testing results in two aspects:

First, based on the experimental results, we observe that many models indeed suffer from subtle discrimination. In some cases, discrimination may be found on unexpected subgroups. Also, those subgroups that are expected to be discriminated against may be treated as privileged groups instead. For example, different from prior conjecture, the model trained on the **COMPAS** dataset is more likely to predict Hispanic males as low risk of recidivism. These results thus suggest that we should have dedicated methods and tools to identify and mitigate subtle discrimination.

Second, the results of identifying subtle discrimination largely depend on multiple hyperparameters, such as the discrimination score threshold and the threshold for the support. These thresholds are used to filter out discrimination that might not be relevant. For example, setting a small support threshold would result in identifying discrimination against small subgroups. Whether such discrimination is interesting is debatable and likely application-dependent. It calls for regulators to clearly define what fairness and subgroup support are relevant. It also calls for flexible methods and tools that allow us to identify relevant subtle discrimination, respectively.

6 THREATS TO VALIDITY

Despite our efforts to design and conduct this research, there are several potential threats to the validity. In this section, we discuss potential threats to our study design and methodology, as well as other factors that may impact the validity of our evaluation results.

Distributional shift in the data: We mentioned that our error bounds rely on the i.i.d. assumption. In this work, since the prior distribution of the original datasets is not available, we approximate the original dataset's distribution by adding perturbation on one attribute systematically, which is the same approach adopted in References [41, 49] and many other machine learning literature [38]. It is still an open problem on how to guarantee that the i.i.d. assumption is satisfied in practice. To alleviate this threat, one practical method is to split the whole dataset randomly into training set and test set and do sampling from the test set. However, in practice, the size of some datasets might be too small to generate an estimation with a certain level of statistical confidence.

Selection of the datasets: We evaluate our method with six structured datasets and two textual datasets that are identified by systematically searching through relevant publications in recent years. Although they are the most common public benchmarks used in the fairness testing literature, it is not clear whether similar group discrimination can be identified on other datasets. To mitigate this threat, TestSGD is designed to be easily extendable to additional tabular or textual datasets. Furthermore, it is possible to extend our method to image data by further defining the sensitive attributes and frequent rule sets for images.

Limited model structures: We evaluate our method with feed-forward neural networks (for structured data) and CNN with LSTM (for textual data). As our method does not require the inner details of the neural network, TestSGD also supports other deep learning architectures such as RNN, and transformer-based models (for textual data), although its performance should be further evaluated accordingly.

Limited fairness metrics: We evaluate our results according to the group discrimination score by measuring the probabilities of predicting certain favorable label by the given model and measuring the difference. This group discrimination score is similar with the Statistical Parity Difference metric, which is one of the popular fairness metrics in the literature. Our work could also work for other group fairness metrics, e.g., false positive rate differences. Whether TestSGD works as effectively remains to be evaluated in future work.

Selection of the baseline: We have limited baselines to compare with. Our approach focuses on explaining under what conditions the discrimination exists in a human-understandable way. Our goal is thus different from that of the baselines such as **THEMIS [16]** and **FairFictPlay [24]**. To alleviate this threat, we extract their results in an interpretable way. In particular, **THEMIS [16]** is not designed to test against subtle discrimination and, thus, we extract the discrimination conditions of **THEMIS** based on the testing results; as for **FairFictPlay [24]**, we omit the linear threshold functions it generates, since they are too complicated to interpret in general. To comprehensively evaluate the performance of our method, we also compare with baselines from different aspects, including detection effectiveness, efficiency, and discrimination mitigation effect.

7 RELATED WORK

Many existing works attempted to test discrimination according to different fairness definitions and measurements [12, 14]. In Reference [15], Feldman et al. provide a fairness definition that is measured according to demographic parity of model predictions. It measures how well the sensitive class can be predicted based on classification accuracy. In Reference [20], Hardt et al. present an alternative definition of fairness based on demographic parity. It requires a decision to be independent of the sensitive attribute. In Reference [27], Kusner et al. define counterfactual discrimination, which focuses on single decisions towards an individual. A prediction is counterfactual fair if it is the same in the actual group and a different demographic group. In Reference [16], Galhotra et al. propose causal discrimination to measure the fraction of inputs for which model causally discriminates. This definition is similar to counterfactual fairness, but it takes instances of discrimination into account. In Reference [24], Kearns et al. proposed an in-processing algorithm aiming to improve the fairness of given subgroups, where subgroups are defined as conjunctions of attributes, linear threshold functions, or bounded degree polynomial threshold functions over multiple protected features. Most existing works [7, 16, 25] use positive classification rate as fairness measurement.

Subsequently, many works focus on individual discrimination to generate individual discriminatory instances [3, 22, 49, 50]. They tried to generated instances that are classified differently after changing sensitive attributes. In Reference [3], Agarwal et al. present an automated testing approach to generate test inputs to find individual discrimination. In Reference [36], Ruoss et al. propose a fairness representation framework to generalize individual fairness to multiple notions. It learns a mapping from similar individuals to latent representations. However, the testing on individual discrimination cannot provide a statistical measurement of fairness.

Some other existing works attempted to test model discrimination with discrimination score measurements. In Reference [40], Tramer et al. propose an unwarranted associations framework to detect unfair, discriminatory, or offensive user treatment in data-driven applications. It identifies discrimination according to multiple metrics including the CV score, related ratio, and associations between outputs and sensitive attributes. In Reference [25], Kleinberg et al. also test multiple discrimination scores and compare different fairness metrics. In Reference [16], Galhotra et al. propose a tool called THEMIS to measure software discrimination. It tests discrimination with two fairness definitions, i.e., group discrimination score and causal discrimination score. In Reference [2], Adebayo et al. try to determine the relative significance of a model's inputs in determining the outcomes and use it to assess the discriminatory extent of the model.

Some prior work has been done on fairness for text classification tasks as well. In Reference [8], Blodgett et al. discuss the impact of unfair natural language in NLP and show how statistical discrimination arises in processing applications. In Reference [10], Bolukbasi et al. show gender bias in the word embedding and provide a methodology for modifying an embedding to remove gender bias. In Reference [13], Dixon et al. measure discrimination using a set of common demographic identity terms and propose a method to mitigate the unintended bias by balancing the training data.

Compared with all the above-mentioned existing works, we provide further fairness testing. Instead of measuring the overall discrimination, our approach systematically identifies and measures subtle discrimination. That is, we not only measure statistical discrimination with a confidence guarantee but also offer interpretable rule sets to represent subtle discrimination.

This work is remotely related to works on applying rule-based models for model explanation. In Reference [46], Yang et al. present an algorithm for building probabilistic rule lists with logical IF-THEN structure. In Reference [28], Lakkaraju et al. propose interpretable decision sets to interpret model predictions with high accuracy and high interpretation. Our work leverages such rule-based interpretable structure to present subtle discrimination in models.

8 CONCLUSION

In this work, we focus on testing neural network models against subtle group discrimination and propose a framework to systematically identify interpretable subtle group discrimination based on group fairness measurement with a certain confidence. Our extensive evaluation demonstrates that subtle group discrimination in neural networks is common to a surprising level. We also show that it is possible to mitigate such discrimination by utilizing our testing results to generate more data for retraining.

ACKNOWLEDGEMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard et al. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*. 265–283.

[2] Julius Adebayo and Lalana Kagal. 2016. Iterative orthogonal feature projection for diagnosing bias in black-box models. *arXiv preprint arXiv:1611.04967* (2016).

[3] Aniya Agarwal, Pranay Lohia, Seema Nagar, Kuntal Dey, and Diptikalyan Saha. 2018. Automated test generation to detect individual discrimination in AI models. *arXiv preprint arXiv:1809.03260* (2018).

[4] Rico Angell, Brittany Johnson, Yuriy Brun, and Alexandra Meliou. 2018. THEMIS: Automatically testing software for discrimination. In *Proceedings of the 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 871–875. DOI: https://doi.org/10.1145/3236024.3264590

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks. *ProPublica*. Retrieved from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[6] Lisa C. Anthony and Mei Liu. 2003. Analysis of differential prediction of law school performance by racial/ethnic subgroups based on the 1996–1998 entering law school classes. LSAC research report series. (2003).

[7] 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. DOI: https://doi.org/10.1145/3368089.3409704

[8] Su Lin Blodgett and Brendan O'Connor. 2017. Racial disparity in natural language processing: A case study of social media African-American English. *arXiv preprint arXiv:1707.00061* (2017).

[9] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).

[10] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. (2016).

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

[12] Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining Knowl. Discov.* 21, 2 (2010), 277–292. DOI: https://doi.org/10.1007/s10618-010-0190-x

[13] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 67–73. DOI: https://doi.org/10.1145/3278721.3278729

[14] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. 214–226. DOI: https://doi.org/10.1145/2090236.2090255

[15] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 259–268. DOI: https://doi.org/10.1145/2783258.2783311

[16] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. 2017. Fairness testing: Testing software for discrimination. In *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering*. 498–510. DOI: https://doi.org/10.1145/3106237.3106277

[17] Bishwamittra Ghosh, Debabrota Basu, and Kuldeep S. Meel. 2020. Justicia: A stochastic SAT approach to formally verify fairness. *arXiv preprint arXiv:2009.06516* (2020).

[18] Milena A. Gianfrancesco, Suzanne Tamang, Jinoos Yazdany, and Gabriela Schmajuk. 2018. Potential biases in machine learning algorithms using electronic health record data. *JAMA Intern. Med.* 178, 11 (2018), 1544–1547. DOI: https://doi.org/10.1001/jamainternmed.2018.3763

[19] Ian Goodfellow. 2019. A research agenda: Dynamic models to defend against correlated attacks. *arXiv preprint arXiv:1903.06293* (2019).

[20] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Adv. Neural Inf. Process. Syst.* 29 (2016).

[21] Hans Hofmann. 1994. German credit dataset. Retrieved from https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

[22] Marianne Huchard, Christian Kästner, and Gordon Fraser. 2018. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE'18).* ACM Press.

[23] Matthew Joseph, Michael Kearns, Jamie H. Morgenstern, and Aaron Roth. 2016. Fairness in learning: Classic and contextual bandits. *Adv. Neural Inf. Process. Syst.* 29 (2016).

[24] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2564–2572.

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

[26] Igor Kononenko. 2001. Machine learning for medical diagnosis: History, state of the art and perspective. *Artif. Intell. Med.* 23, 1 (2001), 89–109. DOI:https://doi.org/10.1016/S0933-3657(01)00077-X

[27] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *arXiv preprint arXiv:1703.06856* (2017).

[28] Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 1675–1684. DOI:https://doi.org/10.1145/2939672.2939874

[29] Pingchuan Ma, Shuai Wang, and Jin Liu. 2020. Metamorphic testing and certified mitigation of fairness violations in NLP models. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20).* 458–465. DOI:https://doi.org/10.24963/ijcai.2020/64

[30] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics, 142–150. Retrieved from http://www.aclweb.org/anthology/P11-1015.

[31] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decis. Supp. Syst.* 62 (2014), 22–31. DOI:https://doi.org/10.1016/j.dss.2014.03.001

[32] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. DeepXplore: Automated whitebox testing of deep learning systems. In *Proceedings of the 26th Symposium on Operating Systems Principles.* 1–18. DOI:https://doi.org/10.1145/3132747.3132785

[33] Michael Redmond. 2009. Communities and Crime dataset. Retrieved from http://archive.ics.uci.edu/ml//datasets/Communities+and+Crime).

[34] Ronald L. Rivest. 1987. Learning decision lists. *Mach. Learn.* 2, 3 (1987), 229–246. DOI:https://doi.org/10.1023/A:1022607331053

[35] Barry Becker and Ronny Kohavi. 1996. Data mining and visualization. Retrieved from https://archive.ics.uci.edu/ml/datasets/adult.

[36] Anian Ruoss, Mislav Balunović, Marc Fischer, and Martin Vechev. 2020. Learning certified individually fair representations. *arXiv preprint arXiv:2002.10312* (2020).

[37] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the International Conference on Management of Data.* 793–810. DOI:https://doi.org/10.1145/3299869.3319901

[38] Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. (2018).

[39] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. DeepTest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering.* 303–314. DOI:https://doi.org/10.48550/arXiv.1708.08559

[40] 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 *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P'17).* IEEE, 401–416.

[41] Sakshi Udeshi, Pryanshu Arora, and Sudipta Chattopadhyay. 2018. Automated directed fairness testing. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering.* 98–108. DOI:https://doi.org/10.1145/3238147.3238165

[42] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *Proceedings of the IEEE/ACM International Workshop on Software Fairness (Fairware)*. IEEE, 1–7. DOI: https://doi.org/10.1145/3194770.3194776

[43] Abraham Wald. 1945. Sequential tests of statistical hypotheses. *Ann. Math. Statist.* 16, 2 (1945), 117–186. DOI: https://doi.org/10.1007/978-1-4612-0919-5_18

[44] Abraham Wald and Jacob Wolfowitz. 1948. Optimum character of the sequential probability ratio test. *Ann. Math. Statist.* (1948), 326–339. DOI: https://doi.org/10.1214/aoms/1177730197

[45] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*. 1391–1399. DOI: https://doi.org/10.48550/arXiv.1610.08914

[46] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. 2017. Scalable Bayesian rule lists. In *Proceedings of the International Conference on Machine Learning*. PMLR, 3921–3930.

[47] 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.

[48] Mengdi Zhang. 2022. GitHub repository for the subtle discrimination testing project. Retrieved from https://github.com/zhangmengling/subtle_discrimination_testing.git.

[49] Peixin Zhang, Jingyi Wang, Jun Sun, Guoliang Dong, Xinyu Wang, Xingen Wang, Jin Song Dong, and Ting Dai. 2020. White-box fairness testing through adversarial sampling. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE'20)*. DOI: https://doi.org/10.48550/arXiv.2107.08176

[50] Peixin Zhang, Jingyi Wang, Jun Sun, Xinyu Wang, Guoliang Dong, Xingen Wang, Ting Dai, and Jin Song Dong. 2021. Automatic fairness testing of neural classifiers through adversarial sampling. *IEEE Trans. Softw. Eng.* (2021). DOI: https://doi.org/10.1109/TSE.2021.3101478

Received 29 August 2022; revised 28 February 2023; accepted 9 March 2023