Mitigating Subgroup Unfairness in Machine Learning Classifiers: A Data-Driven Approach

Yin Lin
University of Michigan
irenelin@umich.edu

Samika Gupta
University of Michigan
samika@umich.edu

H. V. Jagadish
University of Michigan
jag@umich.edu

2024 IEEE 40th International Conference on Data Engineering (ICDE) | 979-8-3503-1715-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDE60146.2024.00171

***Abstract*—Fairness in machine learning, particularly in classifiers, is receiving increasing attention. However, most studies on this topic focus on fairness metrics for a limited number of predefined groups and do not address fairness across intersectional subgroups. In this paper, we investigate ways to improve subgroup fairness where subgroups are defined by the intersection of protected attributes. Specifically, our paper reveals the correlation between the representation bias of training data and model fairness. We demonstrate that biased sample collection due to historical biases and a lack of control over data collection can lead to unfairness in learned models. We introduce the concept of an "Implicit Biased Set (IBS)", which refers to regions in the intersectional attribute space where positive and negative examples are not proportionately represented. For example, if our training data set has a disproportionate representation of black male recidivists, then criminal risk assessment tools are more likely to discriminate against black males, even if they are innocent. We propose an efficient pre-processing approach that initially identifies IBS and then employs techniques to remedy the data collection within IBS. Our evaluation shows that our method effectively mitigates various subgroup biases regardless of the downstream machine learning models used.**

I. Introduction

Machine Learning (ML) systems have a profound impact on society and are widely used in various applications. Users expect these systems to make fair decisions based on historical data. However, biased or insufficient data collection can lead to disparate outcomes in ML systems, as seen, for example, in the higher false positive rate for black individuals in the recidivism prediction algorithm COMPAS [3] and the lower accuracy for darker-skinned females in commercial face recognition services [6]. Ensuring fairness in classification models is a crucial research area in ML.

There are many definitions of fairness in ML [32], and various techniques have been developed to achieve fairness at different stages of the modeling process[5], i.e., pre-processing [19], [37], in-processing [2], [21], [7], and post-processing [20], [15], [28].

One prevalent fairness objective is *group fairness*, which aims to achieve approximate parity of classifier statistics across demographic groups, focusing on the outcome rather than the process. Group fairness is often specified assuming an *independent* setting, where fairness is addressed by considering only one sensitive attribute (such as gender or race) at a time. (So, if there are enough women and enough African Americans, then group counts are satisfied even if there are few African American women). At best, even if independence is not explicitly assumed, group fairness is defined to require parity of some statistical fairness measures over a small number of pre-defined groups. However, a given algorithm might be independently fair on the sensitive attribute but not on intersectional subgroups as shown in Example 1.

Example 1 (Intersectional Biases in COMPAS)**.** *ProPublica released a dataset for evaluating the COMPAS tool [3], which is used to predict the likelihood of recidivism for criminal defendants based on their criminal history and demographic information. One important fairness measure to consider is "predictive equality", which aims to ensure that the protected and unprotected groups have a similar false positive rate (FPR), where FPR is the probability of a subject in the negative class receiving a positive prediction. In the COMPAS dataset, the overall FPR for the entire dataset is 0.088. If we consider only one sensitive attribute* ***gender****, the FPR for* ***Males*** *and* ***Females*** *are 0.09 and 0.07, respectively, which are similar to the overall FPR. However, if we look into the intersectional subgroups of multiple attributes, unfair subgroups can be found, for example, (****race = African-American, sex = Male****) has an FPR of 0.15.*

Ideally, we would like to require fairness for every intersectional subgroup, a concept that is referred to as *subgroup fairness* [22]. Given a set of protected attributes, subgroup fairness applies a statistical fairness constraint (say, predictive equality) to the arbitrary intersection of these attributes, rather than a fixed number of pre-defined groups. The space of all possible sub-groups is large. Automatic tools, such as DivExplorer [26] and SliceFinder[10] have been proposed to efficiently identify significant unfair subgroups in this extensive space. *Fairness gerrymandering* [21], [22] uses a two-player zero-sum game formulation with a Learner and an Auditor to achieve intersectional fairness. While these methods can be effective, they are all either post-processing (manipulating prediction results) or in-processing (altering the learning process). In this paper, our focus is on the pre-processing analysis of training data quality—a crucial factor contributing to system misbehavior, as we elaborate next.

Data collection can introduce various biases originating from multiple sources. For example, a model used to select job candidates trained on historical employment data that favors Caucasian male applicants may continue to perpetuate

2375-026X/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDE60146.2024.00171

discrimination in its future predictions. Additionally, unintentional biases can arise from errors in data collection, such as a flawed sampling algorithm that only gathers data from a limited portion of the population, leading to unrepresentative data [29]. Regardless of the machine learning model used, if the data is biased, it poses a significant risk of systematic discrimination. Consequently, addressing these data-related issues is a fundamental step towards mitigating model unfairness, without necessitating access to the model training process.

In this paper, we establish a connection between representation bias [30], [31] in subgroups and the potential unfairness in ML predictions. Representation bias occurs and can be defined in various ways [30], [31]. Here, we focus specifically on whether the collected datasets contain skewed subsets, often caused by sampling biases [31]. For instance, in Example 1, the performance gap in FPR between Afr-Am males and the entire dataset can be attributed to the dataset containing an excessive number of positive instances in the region representing Afr-Am males.

We demonstrate that unfair subgroups are often associated with specific data subsets (referred to as *regions*) within these subgroups, exhibiting a divergent class distribution compared to other regions. We call such regions in the dataset as *Implicit Biased Sets (IBS)*. We propose a distance-based notion to gather instances from other regions in the intersectional space (referred to as *neighboring regions*) for this comparison.

With this observation, we are able to mitigate subgroup unfairness by enhancing the data collection within IBS. We formulate and propose effective solutions for two tasks: (i) identifying IBS and (ii) remedying dataset biases to mitigate subgroup unfairness.

In particular, our contribution includes the following:

- *Exploring causes of subgroup unfairness.* We propose the imbalance score as a metric to quantify the data distribution within intersectional regions. Formally defining the notion of IBS based on the imbalance score, we provide both *theoretical* and *empirical* evidence to demonstrate that the biased data collection in IBS can significantly contribute to performance divergence in unfair subgroups.
- *IBS identification.* We develop efficient algorithms to traverse the exponentially large lattice of intersectional regions to identify IBS. We show that the problem of IBS identification has *no polynomial-time solution*. Since the computation of the imbalance scores for neighboring regions can overlap across different regions, we propose an optimized algorithm to support result reuse and minimize the number of neighbors to explore, thus enhancing efficiency and scalability.
- *Mitigating subgroup unfairness.* We demonstrate that addressing biased data collection through dataset preprocessing and achieving an unbiased class distribution in IBS effectively mitigates subgroup unfairness.
- *Evaluation.* We analyze our approach on real datasets, validate the relationship between IBS and unfair subgroups, and assess the trade-off between fairness and accuracy. Furthermore, we compare our method to a range of state-of-the-art subgroup unfairness mitigation baselines [21], [4], [35], [8], [19]. Additionally, we evaluate the efficiency and scalability of our algorithms.

TABLE I: Table of notations.

Symbol	Description
$\mathcal{X}$	A set of protected attributes
$\Delta\gamma_g$	Divergence of g under model statistical measure γ.
$ratio_r$	Imbalance score for region r
$ratio_{r_n}$	Imbalance score for the neighboring region of r
τ_d, τ_c	Discrimination threshold and imbalance threshold
T	Distance threshold of the neighboring region
$\mathcal{I}$	Implicit Biased Set (IBS)
$\mathcal{H}$	Hierarchy

II. Definitions

In this section, we first review fairness measures and formally define biased data collection as a critical factor contributing to subgroup fairness. For convenience, we summarize the core symbols in Table I.

A. Fairness Measures and Unfair Subgroups

We study fairness of binary classifiers. Consider a dataset $\mathcal{D}$ with a set of training features $\mathcal{A} = \{a_1, \cdots, a_m\}$, where the domain of attribute a_i is represented by $dom(a_i)$. The input data for prediction is represented by $x = (x_1, \cdots, x_m) \in dom(a_1) \times \cdots \times dom(a_m)$, and the class label of x is $y_x \in \{0, 1\}$. For a given model $h : \mathcal{X}_d \rightarrow \mathcal{Y}_d$, trained on the dataset $\mathcal{D} = \{(x^1, y^1), \cdots, (x^k, y^k)\}$, the prediction of the data x is $h(x) \in \{0, 1\}$.

We consider the fairness of overlapping subgroups defined by the intersection of protected attributes $\mathcal{X} = \{a_{i1}, \ldots, a_{ij}\} \subseteq \mathcal{A}$. Each attribute $a_{ik} \in \mathcal{X}$ takes a categorical (or discretized) value x_{ik} from a finite data domain $dom(a_{ik})$, as is common in (sub)group fairness definitions [32], [26]. A subgroup g_i is the set of instances that match a pattern p_i given by a conjunction of attribute-value assignment $p_i = (a_{i1} = x_{i1} \wedge \cdots \wedge a_{ij} = x_{ij})$, where values can be deterministic $x_{ik} \in Dom(a_{ik})$ or non-deterministic with '$a_{ik} = X$' meaning we do not care about the value assignment of a_{ik}. We use d to represent the number of deterministic elements in p.

For example, consider the intersection of two attributes $\mathcal{X}$ = $\{$*Age, Race*$\}$, the subgroup of all African Americans can be represented by pattern p = *(Age = X, Race = Afr-Am)* with $d = 1$. In later sections, intersectional subgroups are simply referred to as "subgroups". Non-deterministic elements will be omitted from the patterns when clear in context.

In subgroup fairness notions, prior works [26], [21], [24] have explored various statistical measures to ensure similar prediction behavior across different subgroups. We will be concerned about two common statistical measures: false-positive rates (considered in the *equalized opportunity* [15], [23] fairness constraint) and false-negative rates (considered in the *equalized odds* [15] fairness constraint).

The computation of the false-positive rate (*FPR*), expressed as $Pr[h(x) = 1 | y = 0]$, and the false-negative rate (*FNR*), indicated by $Pr[h(x) = 0|y = 1]$, is applicable to both subgroups and the entire dataset. We use γ to represent the selected model statistic, with γ_d representing the statistic across the entire dataset, and γ_g representing the statistic for subgroup g. We focus on $\gamma = \{FPR, FNR\}$ for algorithm demonstration and evaluation, but we also discuss the availability of other fairness metrics, such as zero-one loss, error rate, and statistical parity, in Section VI.

We adopt the notion of *divergence* as proposed in [26] supporting various γ, expressing the behavioral distinction between a specific subgroup and the entire dataset. The divergence of a model statistic γ for a given g relative to the overall data is defined as follows:

$$\Delta\gamma_g = |\gamma_g - \gamma_d|$$

Therefore, we can formally define the subgroup fairness as:

Definition 1 (Subgroup Fairness)**.** *Given a discrimination threshold τ_d and a subgroup g, g is said to be τ_d-fair under model statistic γ, when $\Delta\gamma_g = |\gamma_g - \gamma_d| \leq \tau_d$.*

Example 2. *Consider a decision tree model trained on the ProPublica dataset. The overall FPR is 0.276. Consider two subgroups, g_1: (Age = '25-45', #prior= '>3', Race = Afr-Am) and g_2: (Race = Afr-Am), where #prior is the number of previous offenses of the defendants. The FPR for g_1 is 1, and for g_2, it is 0.369. Suppose we set the discrimination threshold τ_d to be 0.1. The divergence of g_1 with respect to the FPR statistical measure is $\Delta\gamma_{g_1} = |1-0.276| = 0.724$, which is greater than the discrimination threshold τ_d. This result suggests that g_1 is not 0.1-fair under the FPR statistical measure. The divergence of g_2 is $\Delta\gamma_{g_2} = |0.369-0.276| = 0.093$, which is less than the discrimination threshold τ_d, and indicates that g_2 is 0.1-fair under the FPR statistical measure.*

B. Exploring Causes of Subgroup Unfairness

a) Subgroup Features: We first introduce the concept of *dominance relationship*, which illustrates that instances represented by a specific pattern may constitute a subset of a more general subgroup g delineated by a broader data pattern. We refer to this set of instances as a *region* that is dominated by the subgroup g. We observe that unfair subgroups arise from regions having biased data representation within a subgroup, as demonstrated later in this section.

Definition 2 (Dominance Relationship)**.** *A region r_i is dominated by subgroup g_j if pattern p_j can be obtained by replacing any deterministic elements $A_{i_k} = x_{i_k}$ in p_i with the non-deterministic elements $A_{i_k} = X$ while keeping the other deterministic elements unchanged. We denote this dominance relationship as $r_i \preceq g_j$.*

Example 3. *The region (Age = '25-45', #prior = '>3', Race = Afr-Am) is dominated by subgroup (Age = '25-45', #prior = '>3') as the subgroup pattern can be obtained by replacing the deterministic element "Race = Afr-Am" with "Race = X".*

b) Definition of IBS: We introduce a definition to capture biased data representation. To begin, we define the *imbalance score* based on the ratio of positive and negative instances for evaluating the class distribution within regions.

Definition 3 (Imbalance Score)**.** *Given a region r, we use $|r|$ to represent the number of instances that belong to r, i.e. $|\{x|x \in r\}|$, and $|r^+|$ represents the number of positive instances in r, i.e. $|\{x|x \in r \wedge y_x = 1\}|$. Similarly, $|r^-|$ represents the number of negative instances in r, i.e. $|\{x|x \in r \wedge y_x = 0\}|$. The imbalance score of region r is $ratio_r = |r^+|/|r^-|$. When $|r^-| = 0$, we set $ratio_r = -1$.*

Example 4. *In the ProPublica dataset, there are 1,279 instances in the region (Age = '25-45', #prior = '>3'). Among these, 882 instances are positive, and 397 are negative. Therefore, the imbalance score of this region is $\frac{882}{397} = 2.22$.*

In order to reduce subgroup performance divergence, we compare the imbalance score within each given region r to other regions in the intersectional space. If a region's imbalance score significantly deviates from others, we consider it a biased region. We specifically employ a distance metric [27], [39] to formally define the set of nearby instances within the intersectional space of protected attributes. We refer to the union of these regions as the *neighboring region* of r, defined using the Euclidean distance metric.

Definition 4 (Neighboring Region)**.** *Given a region r defined by the pattern $(a_1 = x_1 \wedge \cdots \wedge a_m = x_m)$, where $a_i \in \mathcal{A}$, the Euclidean distance between r and a region r_j is $d(r, r_j) = ||\mathbf{x} - \mathbf{x_j}|| = \sqrt{(x_1 - x_{j1})^2 + \cdots (x_m - x_{jm})^2}$. We consider region r_j to be in the neighboring region of region r if, for a given distance threshold T, the Euclidean distance $d(r, r_j)$ is less than or equal to T. The neighboring region of r is the union of all regions with a distance $\leq T$.*

In the basic setting, we consider all values of the attribute to be one unit distance apart. This approach is generally suitable for most categorical attributes like gender or race. However, in cases where there is a meaningful structure within the attribute value domain, such as a natural numeric ordering for age groups or educational degrees, it is reasonable and straightforward to refine the attribute distance accordingly.

When examining the neighboring region, we exclusively consider regions with identical deterministic attributes. For example, two regions (*Age=25-45'*) and (*#prior=>3'*) are not regarded as neighboring regions for any value of T, as they exist in different dimensions and the instances in these regions are not directly comparable.

In our definition, a default choice for T is 1, indicating that we only consider the union of regions in close proximity as the neighboring region. We demonstrate in the evaluation that this is effective in most cases. However, we can also set T to a larger value, e.g. $T = |\mathcal{X}|$, where we consider the union of all intersectional regions of protected attributes as the neighboring region. We explore this scenario further in Section V-B3.

Example 5. *In the ProPublica dataset, suppose the data domain of Age and #prior attributes are $dom(Age)$ ={'>45', '25-45', '<25'}, $dom(\#prior)$ ={'0', '[1-3]', '>3'}* [1]. *The neighboring region for (Age = '25-45', #prior = '>3') with $T = 1$ is the union of all instances that satisfy patterns (Age = '25-45', #prior = '0'), (Age = '25-45', #prior ='[1-3]'), (Age ='<25', #prior ='>3'), and (Age =>45' , #prior='>3').*

Lastly, we define the *Implicit Biased Set (IBS)* as regions whose imbalance score is significantly different from their neighboring regions. The imbalance score of the neighboring regions can be computed in a manner similar to Definition 3.

Definition 5 (Implicit Biased Set)**.** *For a region r within the intersectional space of protected attributes $\mathcal{X}$, and its neighboring region r_n, with an imbalance threshold τ_c, r is in the implicit biased set if $|ratio_r - ratio_{r_n}| > \tau_c$. We refer to r as a biased region.*

Example 6. *Continuing with the running example, let us consider a region (Age = '25-45', #prior = '>3'). We have computed its imbalance score in Example 4 as $ratio_r = 2.2$. The neighboring region of this region (given in Example 5) has an imbalance score $ratio_{r_n} = 0.64$. Therefore, with an imbalance threshold $\tau_c = 0.3$, we can compare it with the neighboring region using $|ratio_r - ratio_{r_n}| = |2.2 - 0.64| > 0.3$. Thus, this region is in IBS.*

c) Connection between Subgroup Unfairness and IBS: In this subsection, we establish a formal connection between the set of unfair subgroups and the *Implicit Biased Set (IBS)*. Additionally, we provide both theoretical insight and a preliminary case study to support our hypothesis.

Hypothesis 1. *In dataset $\mathcal{D}$, let $\mathcal{X}$ be a set of protected attributes. Let $\mathcal{G}$ be the set of unfair subgroups in the prediction result of any machine learning classifier, and let $\mathcal{I}$ be the set of IBS. We propose that the biased class distribution in $\mathcal{I}$ can contribute to subgroup unfairness in $\mathcal{G}$. Specifically, subgroups that have biased class distribution or dominate significant regions in $\mathcal{I}$ are more likely to suffer from unfairness.*

Theoretical Insight for Hypothesis 1 Consider a set of protected attributes denoted as $\mathcal{X}$, and let C represent the set of all combinations of the protected attribute values, where each $c_i \in C$ represents a region/subgroup in the intersectional space. We begin with the simplest case where there is only one protected attribute, i.e. $|\mathcal{X}|$=1. Assuming for any $c_i \in C$, it contains more positive records than its neighboring region $C_i^n = C_{\{c_j \in C | c_j \neq c_i\}}$. In a machine learning classifier optimized for accuracy, it tends to favor the majority class (say, the positive class) in c_i, resulting in a higher probability of misclassifying a negative example as positive than in C_i^n, thus leading to performance divergence in the false positive rate on the protected attribute. Extending this analysis to an arbitrary number of protected attributes, when considering the largest available value of the distance threshold T (where $T = |\mathcal{X}|$), any intersectional region and its neighboring region (the rest of the subsets in C) are equivalent to the c_i and C_i^n in the single protected attribute situation. For smaller values of T, we are effectively examining a subset of C_i^n. Local performance divergence is likely to occur in this case, potentially leading to subgroup unfairness, even though constructing comprehensive theoretical support remains a challenge. ■

To further substantiate Hypothesis 1, we proceed with a case study on the running example, delving into the cause of the unfairness of a specific subgroup. Furthermore, we provide a comprehensive discussion on the correlation between all unfair subgroups and the set of IBS in Section V-B1.

Case 1 (IBS and Subgroup Unfairness.)**.** *Let us analyze the decision tree model trained on the ProPublica dataset, where the overall FPR is 0.276. We observe an unfair subgroup g : (Age ='25-45', #prior = '>3') with an FPR of 0.965, which is significantly higher than the overall FPR. Individuals in this subgroup are more likely to be wrongly classified as having a high risk of reoffending compared to the overall dataset. The imbalance score of g is* 2.22*, significantly higher than its neighboring region's score of* 0.64*. This discrepancy indicates a biased data collection in the class distribution within g, where there are excessive positive records. Consequently, the decision tree classifier is more likely to predict instances in g as positive, thus resulting in a higher FPR of g.*

C. Problem Definition

To tackle the underlying causes of unfairness in intersectional subgroups, our objective is to identify and remedy all regions with biased data representation in the intersectional space of the protected attributes. We may ignore regions with a small number of instances as they may have minimal impact on classification results and model fairness [21], [26]. We only consider significant regions in IBS with a size greater than k. Here we use the rule-of-thumb from the central limit theorem and set k to a default value of 30.

We now define the IBS identification problem as follows:

Problem 1 (Implicit Biased Set Identification)**.** *Given a dataset $\mathcal{D}$, and an imbalance threshold τ_c, find all biased regions $\mathcal{I}$ within the protected attribute set $\mathcal{X}$ with a size greater than k.*

Theorem 1. *The IBS identification problem has no polynomial-time solution.*

The IBS identification problem involves an analogous task to finding frequent patterns, a well-established task in data mining. In our case, we seek to identify regions represented by their patterns with a size greater than k and an imbalance ratio difference greater than τ_c. While efficient heuristic algorithms exist for frequent itemset mining, the problem lacks a polynomial time solution [14].

In the subsequent sections, we present algorithms for IBS identification in Section III, and discuss strategies for mitigating representation bias within IBS through data remedy approaches in Section IV.

[1] For simplicity, each attribute value is assumed to be one unit distance apart with no numeric ordering.

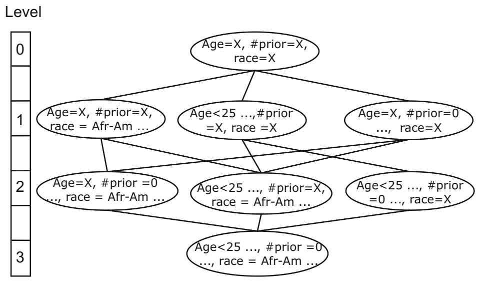

Level
0
1
2
3
Age=X, #prior=X, race=X
Age=X, #prior=X, race = Afr-Am ...
Age<25 ...,#prior =X, race =X
Age=X, #prior=0 ..., race=X
Age=X, #prior =0 ..., race = Afr-Am ...
Age<25 ..., #prior=X, race = Afr-Am ...
Age<25 ..., #prior =0 ..., race=X
Age<25 ..., #prior =0 ..., race = Afr-Am ...

Fig. 1: Hierarchy of regions on $\mathcal{X}$ = {Age, #prior, Race}.

III. IBS identification

We first introduce a data structure, called hierarchy, to facilitate the traversal of all regions within the intersectional space defined by protected attributes $\mathcal{X}$.

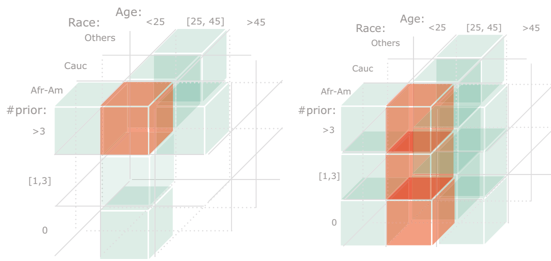
The hierarchy is constructed based on the dominance relationship discussed in Definition 2. In the hierarchy, we represent all regions/subgroups by their patterns and group the patterns having the same deterministic attributes into a node. As seen in Figure 1, considering the intersectional space of three protected attributes {*Age, #prior, Race*}, each node in the hierarchy contains the set of patterns that have the same deterministic attributes but with different value assignments. For instance, *(Race = Hispanic)* and *(Race = Afr-Am)* are both in the first node at level 1. The levels are determined by the number of deterministic elements in their patterns. Leaf-level subgroups have no non-deterministic elements, while the group at level 0 is the entire dataset. Lines indicate the parent/child relationship between the nodes: for each region r_c in the child node, there exists a subgroup g_p in the parent node, such that $r_c \preceq g_p$.

To identify IBS, we traverse the hierarchy from the leaf level to level 1, where the number of nodes is exponential in the number of protected attributes. We first present a naïve algorithm to illustrate the main idea and then introduce an optimized algorithm with a lower time overhead.

A. Naïve Algorithm

The naïve algorithm to identify IBS involves the following steps. Firstly, we traverse the hierarchy in a bottom-up manner. For each region with a size greater than k, to determine whether the region is in IBS, we compute the imbalance score $ratio_r$ for it, and $ratio_{r_n}$ for its neighboring region.

The $ratio_r$ can be easily obtained by computing the number of positive and negative instances in r. The computation of $ratio_{r_n}$ is illustrated in Figure 2. Consider a set of protected attributes $\mathcal{X}$ = {*Age, #prior, Race*} with attribute domains shown on the axes. For example, in Figure 2a, the region *(Age='[25,45]', #prior='>3', Race=Afr-Am)* is marked in red, and the green cubes represent its neighboring region with $T = 1$. The algorithm calculates the $|r^+|$ and $|r^-|$ within the red cube and computes its imbalance score as $ratio_r = |r^+|/|r^-|$. Next, it computes the ratio for the neighboring region by calculating $|r^+_{ni}|$ and $|r^-_{ni}|$ for each of the green cubes r_{ni}, and thus $ratio_{r_n} = \frac{\sum_i |r^+_{ni}|}{\sum_i |r^-_{ni}|}$.



(a) ('[25,45]', '>3', 'Afr-Am') (b) ('[25,45]', X, 'Afr-Am')

Fig. 2: The neighboring region, $\mathcal{X}$={Age, #prior, Race}.

While a top-down traversal could be an alternative, the non-monotonicity of the biased region definition prevents effective pruning or optimization in this approach. Therefore, we focus on the bottom-up search in the IBS identification.

However, the naïve algorithm calculates $ratio_{r_n}$ by summing the counts of positive and negative instances from r's neighbors. With c being the average cardinality of protected attributes $\mathcal{X}$, this leads to exploring $(c-1) \cdot d \cdot T$ neighbors (d is the number of deterministic elements in p, T is a constant in the neighboring region definition). For instance, with $T = 1$, to compute $ratio_{r_n}$ for the region represented by the red cube in Figure 2a, the naïve algorithm computes the counts of positive and negative instances of $(3-1) \times 3 = 6$ green cubes. Similarly, in Figure 2b, it explores the counts of $(3-1) \times 2 = 4$ green cuboids to compute $ratio_{r_n}$ for the region represented by the red cuboid.

B. Optimized Algorithm

The naïve algorithm counts the neighbors of r separately in the computation of $|r^+_n|$ and $|r^-_n|$ which is inefficient and contains repeated operations. A more efficient approach is to group the neighbors into more general regions and use their counts to calculate $ratio_{r_n}$. To create these general regions, we consider a set of regions that dominate r and are T levels up in the hierarchy. We denote this set as R_d.

To obtain R_d, we start with the pattern p in r and remove one deterministic element at a time. We can then calculate the imbalance score of the neighboring region using the formula: $ratio_{r_n} = \frac{\sum_{r_k \in R_d} |r^+_k| - |R_d| \times |r^+|}{\sum_{r_k \in R_d} |r^-_k| - |R_d| \times |r^-|}$, where $|R_d|$ is the size of R_d and also the over-counting factor for $|r^+|$ and $|r^-|$ in the neighboring regions.

Example 7. *In Figure 2a, the set of dominating regions of r = (Age='[25,45]', #prior='>3', Race=Afr-Am) with $T = 1$ is R_d = {(Age='[25,45]', Race=Afr-Am), (Age='[25,45]', #prior= '>3'), (#prior= '>3', Race=Afr-Am)}. These regions are represented by the three cuboids covering the red cube. If we sum up the counts of positive and negative instances in these three cuboids to obtain the counts of the neighboring region, we will overcount $|r^+|$ and $|r^-|$ for the red cube (representing r) threefold as it does not belong to r_n. Therefore, to calculate $|r^+_n|$, we need to sum the positive instances of dominating regions and subtract $3 \times |r^+|$. The same approach applies to computing $|r^-_n|$.*

To avoid redundant computation, the optimized algorithm also maintains counts of dominating regions. Consider three regions *(Age=‘[25, 45]’, Race=Afr-Am)*, *(Age=‘<25’, Race=Afr-Am)*, and *(Age=‘>45’, Race=Afr-Am)*, which have the same deterministic element on the *Race* attribute. To assess whether each of these regions is an IBS, we consistently require the count of the dominating region *(Race=Afr-Am)* with the *Age* attribute's element removed. Therefore, we propose maintaining the counts of R_d's for regions within the same node in the hierarchy. This is because regions in the same node share overlapping sets of dominating regions. This strategy allows the optimized algorithm to reduce the number of counts that need to be computed and stored.

Algorithm 1: Implicit Biased Set Identification

```
input : Dataset $\mathcal{D}$, imbalance threshold $\tau_c$, and a set
        of protected attributes $\mathcal{X}$, size threshold $k$.
output: Implicit Biased Set $\mathcal{I}$.
Initialize the set of IBS as $\mathcal{I} = \{\}$.
1  $\mathcal{H}_o \leftarrow$ CONSTRUCTHIERARCHY ($\mathcal{X}$).
2  $\mathcal{H} \leftarrow$ FILTERREGIONSBYSIZE ($\mathcal{H}_o$, $k$).
3  foreach node $v$ in $\mathcal{H}$ do
4      $V_p \leftarrow$ GETPARENTNODE ($v$).
5      $R \leftarrow$ GETREGIONS ($V_p$).
6      Compute and store the counts of regions in $R$
7      foreach region $r \in v$ do
8          $raio_r \leftarrow |r^+|/|r^-|$
9          Obtain the set of dominating regions $R_d \subseteq R$
             that dominate $r$.
10         $ratio_{r_n} \leftarrow \frac{\sum_{r_{ni} \in R_d} |r^+_{ni}| - |R_d| \times |r^+|}{\sum_{r_{ni} \in R_d} |r^-_{ni}| - |R_d| \times |r^-|}$.
11         if $|ratio_r - ratio_{r_n}| > \tau_c$ then
12             $\mathcal{I}.add(r)$
13 return $\mathcal{I}$
```

Algorithm 1 presents the optimized algorithm for identifying IBS. First, the algorithm creates the hierarchy $\mathcal{H}$ based on the set of protected attributes (line 1) and then filters the regions with a size greater than k in the hierarchy (line 2). For each node v in $\mathcal{H}$, the algorithm obtains the set of its parent nodes V_p and stores the counts for all regions R in V_p (lines 3-6). Next, it enumerates the regions in v (line 7), for each region r in v, calculates the imbalance score $ratio_r$ for r (line 8), and determines the set of its dominating regions R_d (line 9). The algorithm employs the precomputed counts for each dominating region in R_d to calculate $ratio_{r_n}$ (line 10). If the difference between the imbalance scores for r and its neighboring region r_n exceeds the specified imbalance threshold, r is added to the set of IBS (lines 11-12).

Complexity Analysis. In Theorem 1, we establish that the IBS identification problem does not have a polynomial-time solution. Unlike frequent pattern mining, IBS identification not only considers regions with a size exceeding k but also requires an imbalanced score surpassing τ_c. Consequently, existing pruning-based algorithms [14] for frequent pattern mining are ineffective in optimizing IBS identification. In the worst case, the hierarchy contains $c^{|\mathcal{X}|}$ regions to explore, where c denotes the average protected attribute cardinality, and $|\mathcal{X}|$ is the number of protected attributes. However, in exploring each region, compared with the naive algorithm, Algorithm 1 reduces the neighbors to explore from $(c-1) \times d \cdot T$, to $d \times T$ for each region. This optimization, in practice, results in a substantial reduction in time overhead given the exponential number of regions to explore, as we will show in Section V-B5.

For example, in Figure 2a, with $T = 1$, to compute $ratio_{r_n}$, the algorithm only explores $d = 3$ cuboids for the region represented by the red cube, while for the region represented by the red cuboid in Figure 2b, it only needs to explore $d = 2$ cuboids, representing *(Age=‘[25,45]’)* and *(Race=Afr-Am)*.

IV. Dataset remedy

To address representation bias in IBS, we aim to adjust the class distribution $|r^+|/|r^-|$, computed as the imbalance score, within each r in IBS so that $|ratio_r - ratio_{r_n}| < \tau_c$. To achieve this, we employ pre-processing sampling techniques to transform the class distribution within each region.

Definition 6 (Number of Instances to Update)**.** *Given a region r with $|ratio_r - ratio_{r_n}| > \tau_c$. Let p_r denote the number of positive instances to be updated in r, and let n_r denote the number of negative instances to be updated in r. These updates are chosen such that the updated imbalance score for r is equal to $ratio_{r_n}$. The values of p_r and n_r can be computed using the following equation:*

$$\frac{|r^+| + p_r}{|r^-| + n_r} = ratio_{r_n} \tag{1}$$

Here, $|r^+|$ and $|r^-|$ represent the number of positive and negative instances in r, respectively.

The values of p_r and n_r vary with different pre-processing techniques, as demonstrated later. If the values of p_r and n_r are not integers, they will be rounded to the nearest integer. We next formalize the data remedy problem:

Problem 2 (Dataset Remedy)**.** *Given a dataset $\mathcal{D}$ and the Implicit Biased Set $\mathcal{I}$, compute p_r and n_r for each $r \in \mathcal{I}$ and mitigate the biased data representation in r by updating p_r positive instances and n_r negative instances.*

Algorithm 2 outlines the process for remedying biased data representation in $\mathcal{I}$. This data remedy process requires iterative IBS identification at each node since adjusting the class distribution for specific regions will impact the imbalance score of all regions that either dominate or are dominated by them. The algorithm begins by constructing the hierarchy $\mathcal{H}$ from the original dataset and protected attributes (line 1). For each node v, the algorithm uses the same process described in Algorithm 1 to identify the set of biased regions $\mathcal{I}_v$ belonging to v (lines 2-3). Next, for each region, $r \in \mathcal{I}_v$, the number of positive instances to update p_r and the number of negative instances to update n_r are then computed based on the chosen pre-processing technique, and the dataset is updated

Algorithm 2: Dataset Remedy

```
input : Dataset $\mathcal{D}$, a set of protected attributes $\mathcal{X}$, and
        a pre-processing technique $alg$.
output: Dataset after remedy $\mathcal{D}_r$.
1 $\mathcal{H} \leftarrow$ CONSTRUCTHIERARCHY ($\mathcal{X}$).
2 foreach node $v$ in $\mathcal{H}$ do
3     $\mathcal{I}_v \leftarrow$ GETBIASEDREGIONS ($v$).
4     foreach region $r \in \mathcal{I}_v$ do
5         $p_r$, $n_r$ = COMPUTEUPDATES ($alg$, $r$)
6         $\mathcal{D}_r \leftarrow$ UPDATEDATASET ($\mathcal{D}, p_r, n_r, alg$)
7 return $\mathcal{D}_r$
```

accordingly (lines 4-6). Next, we talk about the pre-processing techniques used to transform the class distribution.

A. Pre-processing Techniques

In this section, we discuss four pre-processing techniques: *oversampling, undersampling, preferential sampling*, and *data massaging*. These techniques are incorporated into Algorithm 2 in the UPDATEDATASET procedure.

a) Oversampling: The objective of oversampling [25] is to adjust the biased class distribution by duplicating instances from the minority class in each subgroup. If a region has $ratio_r > ratio_{r_n}$, meaning that more negative instances are needed, p_r is set to 0, and n_r can be computed using Equation (1) as $\frac{|r^+|}{|r^-|+n_r} = ratio_{r_n}$. Similarly, if $ratio_r < ratio_{r_n}$, n_r is set to 0, and p_r is computed using $\frac{|r^+|+p_r}{|r^-|} = ratio_{r_n}$. For each biased region, after determining p_r and n_r, instances in r^+ or r^- are randomly selected for duplication. Oversampling is a simple method that doesn't require changing any existing instances but may result in an increase in storage overhead or model overfitting.

b) Undersampling: Undersampling [25] aims at reducing the data collection of the majority class in biased regions. If a region exhibits $ratio_r > ratio_{r_n}$, indicating a need to remove positive instances, n_r is set to 0, and p_r is calculated using $\frac{|r^+|+p_r}{|r^-|} = ratio_{r_n}$, with $p_r < 0$ indicating the removal of instances from r^+. Similarly, if $ratio_r < ratio_{r_n}$, p_r is set to 0, and n_r is computed as $\frac{|r^+|}{|r^-|+n_r} = ratio_{r_n}$, with $n_r < 0$. In undersampling, instances from the majority class are selected and skipped uniformly. It is preferred when the dataset is large, but can lead to the loss of information and affect model accuracy when the dataset is small.

c) Preferential Sampling: Preferential sampling [19] is a combination of the previous two methods which assigns a higher priority to borderline instances for being duplicated or skipped. It uses a ranker, such as a Naïve Bayes model, to identify the borderline instances, which have a higher probability of belonging to another class. If a region has $ratio_r > ratio_{r_n}$, we duplicate the top-k instances from the negative class and remove the top-k instances from the positive class, where $|p_r| = |n_r| = k$. In Equation (1), the values of p_r and n_r can be computed using $\frac{|r^+|+p_r}{|r^-|+n_r} = ratio_{r_n}$, where $p_r < 0$ and $n_r > 0$ indicate that positive instances are removed and negative instances are added. Similarly, if $ratio_r < ratio_{r_n}$, positive instances are duplicated, and negative instances are removed. Preferential sampling provides a more refined approach to remedying the dataset by taking into account the instance's risks, although it might have a higher time overhead of ranking the instances.

d) Data Massaging: Data massaging [18] aims to select the set of best candidates to relabel. As in preferential sampling, a ranker is used in the massaging technique to select the borderline instances to relabel. In the data massaging, we flip the label of the top-k majority class to reduce the number of majority instances and increase the number of minority instances. If a region has $ratio_r > ratio_{r_n}$, we relabel p_r positive instances as negative, and p_r can be computed by $\frac{|r^+|-p_r}{|r^-|+p_r} = ratio_{r_n}$. If a group has $ratio_r < ratio_{r_n}$, we relabel n_r negative instances as positive, ensuring $\frac{|r^+|+n_r}{|r^-|-n_r} = ratio_{r_n}$. Data massaging has been shown to be effective in removing biases in prediction results [19], and it doesn't change the size of the dataset. However, the data massaging algorithm can be intrusive as it changes labels, which may compromise the validity of the results.

Example 8. *Continuing with Example 6, consider the region (Age = '25-45', #prior = '>3'), with* $ratio_r = 2.2$ *and* $ratio_{r_n} = 0.64$. *The region contains 882 positive instances and 397 negative instances. To address the biased class distribution and adhere to Equation* (1)*: (1) Oversampling: add 984 negative instances uniformly to r, s.t.* $ratio_r = \frac{882}{397+984} = 0.64$. *(2) Undersampling: remove 629 positive instances uniformly from r, s.t.* $ratio_r = \frac{882-659}{397} = 0.64$. *(3) Preferential sampling: remove 384 borderline positives and add 384 borderline negatives in r, s.t.* $ratio_r = \frac{882-384}{397+384} = 0.64$. *(4) Data massaging: flip the label of 384 borderline positives to negative in r, s.t.* $ratio_r = \frac{882-384}{397+384} = 0.64$.

V. Experimental Study

The first question to examine is whether representation bias in the *Implicit Biased Set (IBS)* is the key cause of subgroup unfairness. We examined this question by comparing the set of IBS to unfair subgroups under different statistical measures (γ = FPR, FNR) and machine learning classifiers. We also explored the fairness-accuracy trade-off of our approach and discussed the impact of different parameters. Moreover, we conducted a comparative analysis against state-of-the-art baselines in mitigating subgroup unfairness. Lastly, we evaluated the time performance of our algorithms under varying numbers of protected attributes and data sizes.

A. Experimental Setup

We implemented all algorithms in Python 3.7 and conducted experiments on a Linux machine with a 3.8 GHz Intel Xeon processor and 64GB memory. Code is available [2].

[2] https://github.com/niceIrene/remedy

TABLE II: Dataset characteristics.

	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{X}\rvert$	Protected attributes	Data size
Adult	13	6	age, race, gender, marital-status, relationship, country	45,222
ProPublica	6	3	age,race,sex	6,172
Law School	12	4	age, gender, race, family-income	4,590

a) Data Sets: We used three real-world datasets commonly utilized in fairness literature [5], [22], [39]. For each dataset, we randomly split the data into 70% as training and 30% as testing. The test set is used exclusively for evaluation and no data remedy is applied to it. We performed standard pre-processing of the datasets, which includes removing any missing values and bucketizing continuous values for protected attributes. For each dataset, we adhered to the guidelines in Equality Act 2010 [1] to define the set of protected attributes. We present a summary of the dataset characteristics in Table II.

- *AdultCensus [12].* Contains 45,222 records about individuals' annual income based on census data. The protected attributes include age, race, gender, and so on.
- *ProPublica [3].* Contains 6,172 records about the demographic information and criminal history of defendants. We included age, race, and gender as protected attributes.
- *Law School [33].* Contains information on over 4,000 law students, including details on demographics and school performance. Since the original dataset was extremely imbalanced with respect to the prediction label, we conducted uniform sampling, resulting in an equal number of positive and negative records. Additionally, to prevent discrimination against students from economically disadvantaged backgrounds, we incorporated family income alongside age, race, and gender as protected attributes.

b) Classification Models and Methods: Our proposed approach to mitigate subgroup unfairness is model agnostic and can be applied to any machine learning (ML) classifiers. To evaluate our approach, we considered four downstream classifiers: *decision tree (DT), random forest (RF), logistic regression (LG)*, and *neural network (NN)*. For each classifier, we used grid search to obtain the optimal hyperparameters.

To identify IBS, our algorithms traverse the lattice space of the hierarchy. We compared our approach, ***Lattice***, against two methods: one concentrating exclusively on intersectional regions at the leaf level (***Leaf***) and another specifically addressing biases at the highest hierarchical level (***Top***). This comparison aims to underscore that a comprehensive approach is necessary, as solely focusing on groups defined by the protected attribute or their intersections, as demonstrated by *Top* and *Leaf*, are insufficient for achieving subgroup fairness. Additionally, we examined the time efficiency of the IBS identification algorithms by comparing the runtime of the ***Naïve algorithm*** in Section III-A to the ***Optimized algorithm*** in Section III-B.

To address representation bias in IBS, we evaluated the runtime and effectiveness of the remedy algorithm with pre-processing techniques: ***Oversampling, Undersampling, Preferential Sampling***, and ***Data Massaging*** in Section IV-A.

c) Baselines: We compared our approach to the following baselines aimed at addressing subgroup unfairness, including four pre-processing algorithms and one in-processing algorithm.

- ***Coverage*** [4] is a pre-processing technique that identifies subgroups lacking sufficient representation in the dataset and addresses both the identification and the enhancement of lack of data coverage. For additional tuples required by [4] to augment the coverage of a subgroup g, we randomly sampled additional tuples from that subgroup.
- ***Reweighing*** method [19] generates weights for training instances for each (subgroup, label) combination to achieve equivalent class distribution across all subgroups.
- ***FairBalance*** [35] also proposes a reweighing algorithm to ensure not only equal but also balanced (1:1) class distribution in all subgroups to achieve equalized odds.
- ***Fair-SMOTE*** [8] serves a similar purpose to the previous baseline by oversampling training data with synthetic data points from the minority class in each subgroup.
- ***GerryFair*** [21] is an in-processing algorithm that trains fair classifiers and audits classifier predictions for subgroup fairness violations.

d) Metrics: While *divergence* can be used to measure the unfairness of a specific subgroup, there is a lack of a measure to evaluate subgroup unfairness for the entire dataset. Previous studies [21] have suggested focusing solely on the most significant unfair subgroup, but this approach may not be sufficient as it overlooks other unfair subgroups. To assess the effectiveness of our algorithms in mitigating unfairness across the dataset, we introduce a *Fairness Index* to quantify overall subgroup unfairness. The index is calculated as the sum of the divergences for each unfair subgroup with a support (as a fraction of the dataset size) over 0.1 and a statistically significant divergence (as determined by the t-test). The fairness index represents the weighted sum of the divergence for all significant unfair subgroups. Lower values indicate higher levels of fairness.

We utilized DivExplorer [26], a highly efficient automated tool to identify all unfair subgroups in the dataset. Given a statistical measure γ, DivExplorer provides a set of unfair subgroups with their support and performance divergence, and ranks them based on the performance divergence.

B. Performance Analysis

1) Validation: Connection between Representation Bias in IBS and Unfair Subgroups: We first investigated the correlation between unfair subgroups and regions having biased data collection in IBS, utilizing the *ProPublica* dataset.

Specifically, we examined the cause of subgroup unfairness under both statistical measures—FPR and FNR—for all four machine learning models: DT, RF, LG, and NN. Employing our method with $\tau_c = 0.1$ and $T = 1$, we identified the set of IBS and compared it with the unfair subgroups.

In Figure 3, we depicted all unfair subgroups in the prediction outcome under γ = FPR of all ML models. We marked each subgroup in grey if the corresponding region (represented

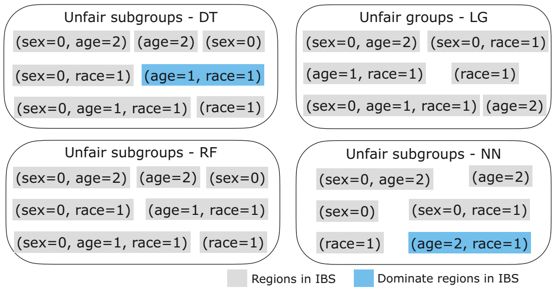


Fig. 3: Unfair subgroups in the prediction outcomes of DT, RF, LG, and NN, within IBS or dominate regions within IBS.

by the same data pattern) exhibits representation bias, i.e., belonging to IBS. We marked each unfair subgroup in blue if it dominates significant biased regions.

As illustrated in Figure 3, nearly all unfair subgroups exhibit representation bias, displaying significantly divergent class distributions compared to their neighboring regions. The remaining two subgroups marked in blue (age = 1, race=1) and (age = 2, race=1) also dominate significant regions in IBS, specifically: (age = 1, race=1, sex =1) $\preceq$ (age = 1, race=1) and (age = 2, race=1, sex =0) $\preceq$ (age = 2, race=1). Furthermore, we observed that regions in IBS with $ratio_r > ratio_{r_n}$ are consistently associated with unfair subgroups exhibiting a higher FPR, while unfair subgroups under γ = FNR tend to have $ratio_r < ratio_{r_n}$ or dominate such regions. This is because, in the biased regions, the majority class is more likely to be preferred in the classification results. Therefore, subgroups with a higher percentage of positive samples are more likely to have high FPR, and vice versa.

2) The Fairness-accuracy Trade-off: We next evaluated the trade-off between accuracy and fairness for all datasets. For identifying IBS, we compared our method of exploring the *Lattice* space to baselines that only identify IBS on the *Top* or *Leaf* level in the hierarchy. For addressing biased data collection in IBS, we compared the results of all pre-processing techniques mentioned in Section IV-A. Our experiments show that the *Lattice* and *Preferential Sampling (PS)* methods yield the best fairness and accuracy. Thus, we employed the *Lattice* method for IBS identification when comparing different pre-processing techniques, and used the *PS* method for pre-processing when comparing different IBS identification methods. For parameters, we selected $T = 1$ and $\tau_c = 0.1$ for the *ProPublica* and *Law School* datasets, and $\tau_c = 0.5$ for the *Adult* dataset for optimal performance, as elaborated later in Section V-B3.

The experimental results for the three datasets are shown in Figure 4, 5, and 6. We started by comparing different IBS identification algorithms. We reported the fairness index under statistical measures γ = FPR (Figure 4a, 5a and 6a) and FNR (Figure 4b, 5b and 6b), as well as the model accuracy (Figure 4c, 5c and 6c).

For the *Lattice* algorithm, it demonstrates a significant enhancement in the fairness index. Additionally, we observed that it can simultaneously mitigate subgroup unfairness for both statistical measures FPR and FNR. This is because, by addressing biased class distribution in regions having $ratio_r > ratio_{r_n}$ and $ratio_r < ratio_{r_n}$, we are able to effectively improve both types of unfairness concurrently. For instance, our method reduces the fairness index of the *Adult* dataset from as high as 0.6 to less than 0.05, as illustrated in Figure 4a. It also at the same time mitigates subgroup unfairness under FNR by reducing the fairness index from over 1.5 to less than 0.4, as depicted in Figure 4b.

This improvement was consistently observed across different ML models, statistical measures, and datasets. The model accuracy for *Lattice* also decreases by less than 0.1 across all ML algorithms and datasets. This decline in accuracy is attributed to the remedy of the biased class distributions in IBS, resulting in discrepancies between the distribution of the training and testing sets. Similar accuracy reductions are commonly observed in fairness mitigation approaches [17].

For the *Leaf* baseline, as it updated a smaller fraction of the dataset, its accuracy performance is better than *Lattice* while having a poorer fairness performance (does not significantly improve the fairness index). The *Top* baseline performs coarse-level modifications to the datasets and exhibits less effectiveness in fairness improvement.

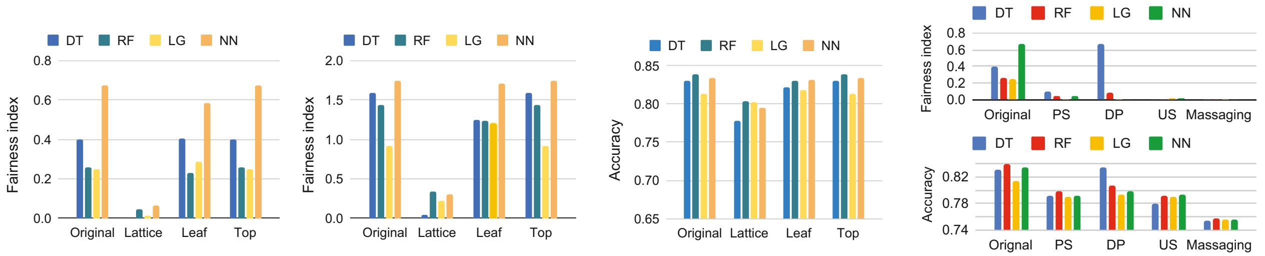
In Figures 4d, 5d, and 6d, we compared different pre-processing techniques used in the data remedy algorithm (PS = preferential sampling, US = undersampling, DP = oversampling, Massaging). We presented the fairness index under the statistical measure $\gamma =$ FPR and the model fairness of different ML models.

For large datasets like *Adult*, both *preferential sampling* and *undersampling* have good fairness and accuracy performance, with a fairness index below 0.05 and an accuracy decrease of less than 0.1. However, the *oversampling* method exhibits a substantial increase in memory consumption, as it introduces a large number of records to the dataset. we show this effect in our scalability experiments. On the other hand, the *massaging* method, being a more intrusive approach that alters data labels, shows comparatively poorer accuracy performance.

Similar trends are observed across other datasets and also indicate that *preferential sampling* tends to yield slightly better fairness performance (lower fairness index) compared to *undersampling* when the dataset sizes are relatively smaller.

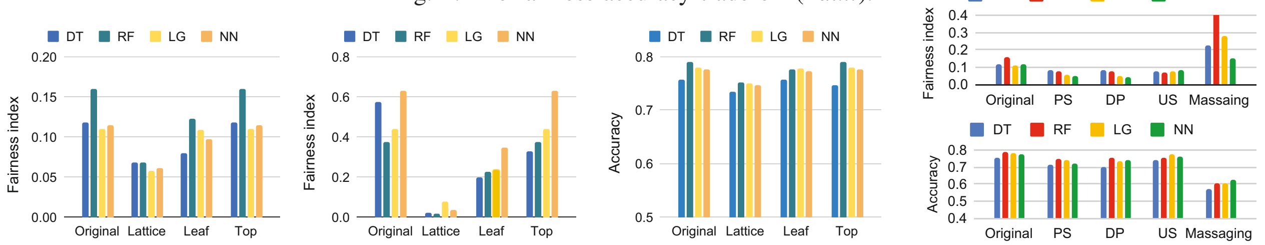
3) Effects of Parameters: We discussed the impact of parameters in our proposed method, which include two tunable parameters: the imbalanced threshold τ_c and the distance threshold T of the neighboring region. We used the *ProPublica* and *Adult* datasets with *decision tree* in this discussion.

The imbalanced threshold, as in Definition 5, determines how much disparity in class distribution is considered "biased". A smaller value of τ_c typically identifies more regions as biased, resulting in more instance updates to adjust class distribution within these regions. In Figure 7, with $T = 1$, we varied τ_c from 0.1 to 0.9, reporting the fairness index (γ = FPR) and accuracy. Figure 7a demonstrates that lower τ_c values, leading to more instance updates, generally result in greater fairness improvement but lower model accuracy,



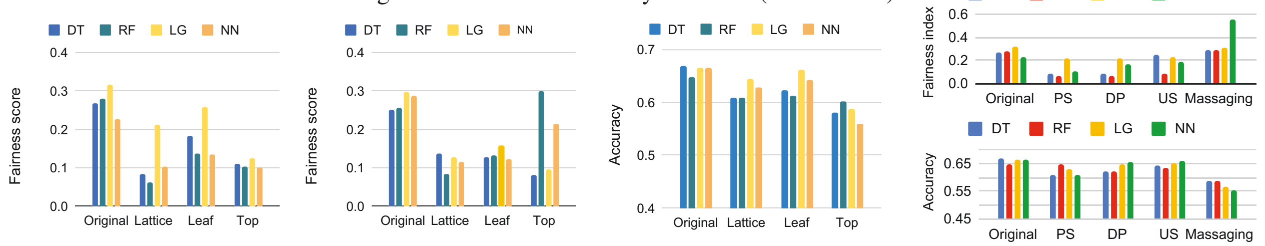
(a) Preferential sampling, FPR (b) Preferential sampling, FNR (c) Preferential sampling, acc (d) Pre-processing methods comparison

Fig. 4: The fairness-accuracy trade-off (*Adult*).



(a) Preferential sampling, FPR (b) Preferential sampling, FNR (c) Preferential sampling, acc (d) Pre-processing methods comparison

Fig. 5: The fairness-accuracy trade-off (*Law School*).



(a) Preferential sampling, FPR (b) Preferential sampling, FNR (c) Preferential sampling, acc (d) Pre-processing methods comparison

Fig. 6: The fairness-accuracy trade-off (*ProPublica*).

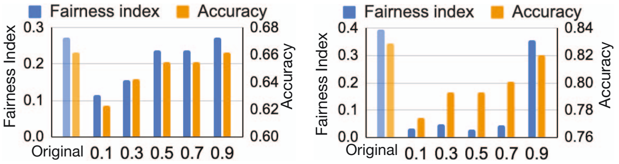
especially evident in the *ProPublica* dataset. However, the *Adult* dataset, with more protected attributes (6 compared to 3 in *ProPublica*), could exhibit robust fairness performance even at higher τ_c values. This is because a larger number of protected attributes may lead to a larger set of IBS, ensuring sufficient updates of the datasets.

The distance threshold T determines the neighboring region for each r. When $T = |\mathcal{X}|$, we compare the class distribution of r to the class distribution of all regions in the intersectional space of $\mathcal{X}$. Conversely, when $T = 1$, we compare r's class distribution to the class distribution of only regions in close proximity. In Figure 7, we compared the effects of $T = |\mathcal{X}|$ and $T = 1$ for the two datasets, reporting the fairness index under $\gamma =$ FPR and FNR, as well as model accuracy. We observed that both T values mitigate subgroup unfairness in all cases. For datasets with a smaller number of protected attributes, e.g., $|\mathcal{X}| = 3$ in the *ProPublica* dataset, $T = |\mathcal{X}|$ outperforms $T = 1$. However, for datasets with a larger number of protected attributes, such as *Adult*, $T = 1$ is more likely to achieve optimal fairness performance. This means as the number of protected attributes grows, ensuring equivalent class distribution in all subgroups becomes less effective.

4) Comparisons with Subgroup Unfairness Mitigation Baselines: The in-processing method *GerryFair* utilizes a distinct subgroup fairness metric based on *fairness violation*, defined as the subgroup with the greatest performance divergence multiplied by its violated group size. For a fair comparison, we used this fairness violation as the evaluation metric for the pre-processing methods in this discussion. We considered the *logistic regression* as the ML classification model for all pre-processing baselines because it is a linear model as in the *GerryFair* classifier. We focused on the *Adult* dataset and considered two protected attributes {*Race, Gender*} as in [35], since certain baselines provide poorer support for a larger set as we demonstrated later. For our *Remedy* approach, we set $\tau_c = 0.1$ and $T = 1$. Optimal parameter settings were also utilized for all other baselines.

In Table III, we presented the fairness violation, model accuracy, and execution time for all approaches. We observed fairness improvements in all baselines except for *Coverage*, indicating that the under-representation identified and enhanced in *Coverage* is not a significant factor contributing to subgroup behavioral divergence. However, enhancing *Coverage* improves overall prediction accuracy.

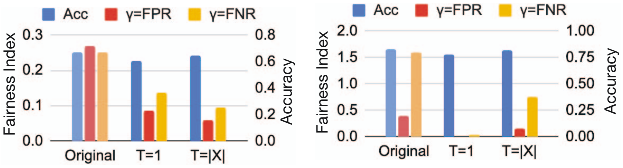
Among pre-processing methods, *Reweighing* achieves optimal performance, reducing fairness violation to 0 by ensuring equivalent class distribution across intersectional subgroups. However, in our analysis with a larger set of protected at-



(a) *ProPublica* dataset.

(b) *Adult* dataset.

Fig. 7: Fairness index and model accuracy, varying τ_c.



(a) *ProPublica* dataset.

(b) *Adult* dataset.

Fig. 8: Fairness index and model accuracy under different T.

tributes as in Table II, *Remedy* outperforms *Reweighing* by achieving class distribution equivalence only in neighboring regions. In addition, *Reweighing* requires the learner to accept sample weights and provide no evidence of the causes of subgroup biases, reducing its flexibility and reliability.

Both *FairBalance* and *Fair-SMOTE* improve fairness by ensuring both equal and balanced class distribution among subgroups. However, in real-world datasets, which are rarely balanced, such approaches are likely to incur low accuracy as observed in Table III due to the significant distribution difference between the new training set and the test set. *GerryFair* is also prone to lower accuracy performance when confronted with imbalanced datasets [21].

For time efficiency, *Fair-SMOTE* incurs a long execution time as it utilizes the k-nearest-neighbors to generate the new synthetic data, which is impractical for large datasets and numerous protected attributes. Similarly, the in-processing algorithm *GerryFair* exhibits lower efficiency compared to other methods, with training times escalating significantly as dataset sizes and subgroup numbers increase. Other pre-processing approaches complete within a reasonable timeframe ($< 3s$), while our *Remedy* method requires slightly more time to train the ranker for preferential sampling.

5) Scalability: We reported results on the *Adult* dataset, to maximize variation in protected attributes and data sizes. To assess algorithm efficiency regarding the number of protected attributes, we expanded the set of protected attributes with two additional categorical attributes: *education* and *occupation*, despite them not being protected characteristics in the dataset. We reported the runtime for IBS identification using the *Naïve* algorithm and the *Optimized* algorithm under varying numbers of protected attributes and data sizes. Additionally, we recorded the runtime for remedying biased data collection in IBS using different pre-processing techniques.

Figure 9a depicts the runtime for IBS identification under varying numbers of protected attributes. The runtime experiences an exponential increase with the growing number of protected attributes, attributed to the exponential expansion of regions to explore in the hierarchy. Our *Optimized* algorithm showcases better efficiency compared to the *Naïve* algorithm, consistently remaining up to 5 times more efficient than the *Naïve* approach, which reaches as high as 600 seconds.

TABLE III: Fairness violation, model accuracy, and execution time, compared with baselines, $\mathcal{X} = \{$*Race, Gender*$\}$ (*Adult*)

Approach	Fairness violation	Accuracy	Time
Original	0.0210	0.813	-
Remedy	0.0055	0.793	2.62
Coverage [4]	0.0218	0.815	0.73
FairBalance [35]	0.0040	0.735	1.16
Fair-SMOTE [8]	0.0120	0.726	1065.6
Reweighing [19]	0	0.802	1.02
GerryFair [21]	0.0032	0.789	593.7

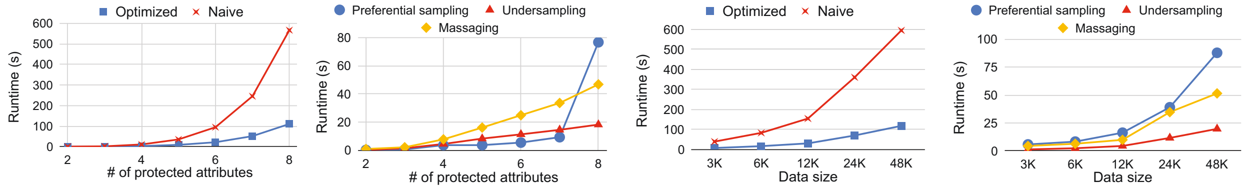
In contrast to the IBS identification time, we observed that the remedy algorithm, regardless of the pre-processing technique employed, can be completed within a shorter time frame (as shown in Figure 9b). The runtime is predominantly influenced by the number of biased regions in IBS, thereby increasing as the number of protected attributes grows. The *oversampling* method exceeded the memory resource limit by introducing an excessive number of instances to the dataset and hence is excluded from this analysis.

We then set the size of the protected attributes to 8 (maximal) and examined the impact of data sizes. In Figure 9c, although the time complexity of the algorithms is not directly related to data sizes, the growth of data sizes increases the number of candidate regions (regions with a size greater than $k = 30$) to explore, thereby escalating the IBS identification runtime. For the data remedy, in Figure 9d, the execution time for all pre-processing techniques is within an acceptable range and only relates to the number of biased regions in IBS. However, the *massaging* and *preferential sampling* techniques have a comparatively longer execution time (up to 50s) as they require training a ranker to obtain borderline instances.

VI. Discussion

Fairness metrics. In this paper, we focus on fairness metrics based on the model statistics FPR and FNR. Although there are other statistical measures for classification models such as *zero-one loss* ($\sum_{i=1}^{n} \mathbb{I}(h(x_i) \neq y_i)$), *error rate* ($P(h(x) \neq y)$), or *accuracy* ($P(h(x) = y)$) [9], we do not consider these measures in our evaluation. This is because these measures are based on prediction accuracy, which can be affected when there is a difference in the data distribution between the training set and the test set. Without any pre-processing on the test set, the accuracy of the model as well as these fairness measures can be affected by such distribution differences. Furthermore, our evaluation is based on the assumption that the data distribution of the test set is also biased (they are usually drawn from the same distribution), which can impact the accuracy-related fairness measures. Therefore, we limit our evaluation in this paper to FPR and FNR, which are more robust to these distributional differences. However, it is worth noting that representation bias is also closely related to unfair subgroups under accuracy-related statistical measures.

Another fairness metric, statistical parity [13], compares the predicted outcomes for protected groups and aims to



(a) Identification, # of attributes (b) Remedy, # of attributes (c) Identification, data sizes (d) Remedy, data sizes

Fig. 9: Runtime for IBS identification and remedy, varying # of protected attributes, data sizes (*Adult*).

ensure equal outcomes across groups ($P(h(x) = 1|A = a) = P(h(x) = 1|A = \bar{a})$). Unlike other fairness metrics, statistical parity only considers predicted outcomes and does not take into account actual outcomes (y_x). Our mitigation approach can also be applied to mitigate statistical parity. For example, in a hiring model that considers race and gender as protected attributes, the acceptance rate for green females and purple males is 50%, while it is 0% for green males and purple females. Analyzing each attribute independently would suggest fairness, but our method could detect representation bias in each subgroup and help mitigate such biases.

Limitations. Our method mitigates subgroup unfairness by establishing a correlation between representation bias and performance divergence within subgroups. However, this correlation is mainly applicable to classifiers optimized for accuracy, as suggested by the theoretical insight. For cost-sensitive classifiers [36] optimized for misclassification cost, this correlation may not remain valid. In addition, the remedy algorithm does not guarantee achieving an optimal dataset where the difference between the imbalance score and that of the neighboring region is zero for all regions, as adjustments in one region may impact others. Nevertheless, our evaluation shows minimal impact on effectiveness, affirming the validity of our approach.

VII. Related Work

Most existing research in algorithmic group fairness has focused on the simplest scenario, where groups are defined based on a single protected attribute that is independent of the prediction outcome [13]. Subgroup fairness (aka. intersectional fairness) [22] instead demands statistical notions of fairness across the intersections of protected attributes. A subgroup is considered unfair when its behavior under certain statistical measures deviates significantly from the overall performance [26]. Automated tools like DivExplorer [26], SliceFinder [10], and DENOUNCER [24] have been developed to identify subgroups in which a model performs poorly, based on statistics computed from classifiers. These tools help to identify and analyze unfair subgroups more efficiently.

In this paper, we discuss approaches to mitigate subgroup unfairness. Bias mitigation techniques can be categorized into three types: pre-processing interventions [19], [37], [38], [39], [35], [8], in-processing interventions [2], [21], [7], [38], and post-processing interventions [20], [15], [28]. A comprehensive comparison of these approaches can be found in [17]. For mitigating subgroup unfairness, various in-processing techniques have been discussed in the literature [16], [34], [21], [22], which aim to ensure that the subgroup statistics are approximately equal to that of the entire population. Fairness Gerrymandering [21] aims to audit binary classifiers by solving for the equilibrium in a two-player zero-sum game between a learner and the auditor. Hebert-Johnson et al. [16] propose a notion called multicalibration that is similar to subgroup fairness and proposes an iterative algorithm to ensure the calibration constraints using the online learning framework. Yang et al. [34] characterize population optimal predictions by which they propose a weighted empirical risk minimization (ERM) approach for fair classification. While in-processing methods demonstrate effectiveness, our paper focuses on pre-processing techniques, as they offer more flexibility and can be applied to any model without the need to access prediction results or modify classifiers.

Existing pre-processing methods for mitigating bias involve techniques such as data sampling, reweighing, and data massaging, aiming to eliminate group-based discrimination [19]. iFlipper [39] introduces label flipping to enhance individual fairness in predictions, while [11] proposes a probabilistic framework for discrimination prevention. However, these approaches often necessitate predefined protected groups or a similarity measure for individuals, aspects that are orthogonal to the subgroup fairness measures discussed in our paper. Pre-processing techniques for subgroup fairness have received comparatively less attention. FairBalance [35] calculates and assigns weights to training data to improve equalized odds for subgroup fairness. Fair-SMOTE [8] oversamples training data from minority groups with synthetic data points to achieve balanced class distributions within subgroups. In comparison, our focus lies on illustrating the impact of biased class distribution within subgroups [31] on subgroup unfairness.

VIII. Conclusion

This paper explores the causes of subgroup unfairness and proposes a pre-processing method to mitigate this issue. We studied representation bias as the discrepancy in class distribution among subgroups and explored its role in subgroup unfairness. Then, we solved the problem of identifying and remedying the *Implicit Biased Set* (IBS) to address such representation bias. Extensive experiments were conducted to validate the effectiveness and efficiency of our proposed methods in mitigating subgroup unfairness.

Acknowledgements

This research was supported in part by NSF under grants 1934565, 2106176, and 2312931.

REFERENCES

[1] Protected characteristics. https://www.equalityhumanrights.com/equality/equality-act-2010/protected-characteristics, 2010.

[2] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pages 60–69. PMLR, 2018.

[3] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. In *Ethics of data and analytics*, pages 254–264. Auerbach Publications, 2016.

[4] A. Asudeh, Z. M. Jin, and H. V. Jagadish. Assessing and remedying coverage for a given dataset. *ICDE*, pages 554–565, 2018.

[5] R. K. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[6] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[7] L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 319–328, 2019.

[8] J. Chakraborty, S. Majumder, and T. Menzies. Bias in machine learning software: why? how? what to do? In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE '21. ACM, Aug. 2021.

[9] I. Chen, F. D. Johansson, and D. Sontag. Why is my classifier discriminatory? *Advances in neural information processing systems*, 31, 2018.

[10] Y. Chung, T. Kraska, N. Polyzotis, K. H. Tae, and S. E. Whang. Slice finder: Automated data slicing for model validation. In *International Conference on Data Engineering*, pages 1550–1553. IEEE, 2019.

[11] F. du Pin Calmon, D. Wei, K. N. Ramamurthy, and K. R. Varshney. Optimized data pre-processing for discrimination prevention. *CoRR, abs/1704.03354*, 2017.

[12] D. Dua and C. Graff. UCI machine learning repository, 2017.

[13] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[14] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12, 2000.

[15] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[16] U. Hébert-Johnson, M. Kim, O. Reingold, and G. Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

[17] M. T. Islam, A. Fariha, A. Meliou, and B. Salimi. Through the data management lens: Experimental analysis and evaluation of fair classification. In *International Conference on Management of Data*, pages 232–246, 2022.

[18] F. Kamiran and T. Calders. Classifying without discriminating. In *International conference on computer, control and communication*, pages 1–6. IEEE, 2009.

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

[20] F. Kamiran, A. Karim, and X. Zhang. Decision theory for discrimination-aware classification. In *International conference on data mining*, pages 924–929. IEEE, 2012.

[21] M. Kearns, S. Neel, A. Roth, and Z. S. Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*, pages 2564–2572. PMLR, 2018.

[22] M. Kearns, S. Neel, A. Roth, and Z. S. Wu. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 100–109, 2019.

[23] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

[24] J. Li, Y. Moskovitch, and H. Jagadish. Denouncer: detection of unfairness in classifiers. *Proceedings of the VLDB Endowment*, 14(12), 2021.

[25] R. Mohammed, J. Rawashdeh, and M. Abdullah. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *International conference on information and communication systems*, pages 243–248. IEEE, 2020.

[26] E. Pastor, L. De Alfaro, and E. Baralis. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *International Conference on Management of Data*, pages 1400–1412, 2021.

[27] F. Petersen, D. Mukherjee, Y. Sun, and M. Yurochkin. Post-processing for individual fairness. *Advances in Neural Information Processing Systems*, 34:25944–25955, 2021.

[28] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

[29] N. Shahbazi, Y. Lin, A. Asudeh, and H. Jagadish. A survey on techniques for identifying and resolving representation bias in data. *arXiv preprint arXiv:2203.11852*, 2022.

[30] N. Shahbazi, Y. Lin, A. Asudeh, and H. V. Jagadish. Representation bias in data: A survey on identification and resolution techniques. *ACM Computing Surveys*, 55:1 – 39, 2022.

[31] H. Suresh and J. V. Guttag. A framework for understanding sources of harm throughout the machine learning life cycle. *Equity and Access in Algorithms, Mechanisms, and Optimization*, 2019.

[32] S. Verma and J. Rubin. Fairness definitions explained. In *Proceedings of the international workshop on software fairness*, pages 1–7, 2018.

[33] L. F. Wightman. Lsac national longitudinal bar passage study. lsac research report series. 1998.

[34] F. Yang, M. Cisse, and S. Koyejo. Fairness with overlapping groups; a probabilistic perspective. *Advances in neural information processing systems*, 33:4067–4078, 2020.

[35] Z. Yu, J. Chakraborty, and T. Menzies. Fairbalance: How to achieve equalized odds with data pre-processing, 2021.

[36] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. *Third IEEE International Conference on Data Mining*, pages 435–442, 2003.

[37] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

[38] H. Zhang, X. Chu, A. Asudeh, and S. B. Navathe. Omnifair: A declarative system for model-agnostic group fairness in machine learning. In *Proceedings of the 2021 international conference on management of data*, pages 2076–2088, 2021.

[39] H. Zhang, K. H. Tae, J. Park, X. Chu, and S. E. Whang. iflipper: Label flipping for individual fairness. *Proceedings of the ACM on Management of Data*, 1(1):1–26, 2023.