Non-Invasive Fairness in Learning through the Lens of Data Drift

Ke Yang
University of Texas at San Antonio, TX, USA
ke.yang@utsa.edu

Alexandra Meliou
University of Massachusetts Amherst, MA, USA
ameli@cs.umass.edu

2024 IEEE 40th International Conference on Data Engineering (ICDE) | 979-8-3503-1715-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDE60146.2024.00172

***Abstract*—Machine Learning models are widely employed to drive many modern data systems. While they are undeniably powerful tools, ML models often demonstrate imbalanced performance and unfair behaviors. The root of this problem often lies in the fact that different subpopulations commonly display divergent trends: as a learning algorithm tries to identify trends in the data, it naturally favors the trends of the majority groups, leading to a model that performs poorly and unfairly for minority populations. Our goal is to improve the fairness and trustworthiness of ML models by applying only *non-invasive* interventions, which don't alter the data or the learning algorithm. We use a simple but key insight: the divergence of trends between different populations, and, consecutively, between a learned model and minority populations, is analogous to data drift, which indicates poor conformance between parts of the data and the trained model.**

We explore two strategies (model-splitting and reweighing) to resolve this drift, aiming to improve the overall *conformance* of models to the underlying data. Both our methods introduce novel ways to employ the recently-proposed data profiling primitive of Conformance Constraints. Our splitting approach is based on a simple data drift strategy: training separate models for different populations. Our DIFFAIR algorithm enhances this simple strategy by employing conformance constraints, learned over the data partitions, to select the appropriate model to use for predictions on each serving tuple. However, the performance of such a multi-model strategy can degrade severely under poor representation of some groups in the data. We thus propose a single-model, reweighing strategy, CONFAIR, to overcome this limitation. CONFAIR employs conformance constraints in a novel way to derive weights for training data, which are then used to build a single model. Our experimental evaluation over 7 real-world datasets shows that both DIFFAIR and CONFAIR improve the fairness of ML models. We demonstrate scenarios where DIFFAIR has an edge, though CONFAIR has the greatest practical impact and outperforms other baselines. Moreover, as a model-agnostic technique, CONFAIR stays robust when used against different models than the ones on which the weights have been learned, which is not the case for other states of the art.

***Index Terms*—data management, fairness, data profiling**

I. INTRODUCTION

While Machine Learning (ML) models are widely employed in many modern data systems for their undeniable predicting power, they often demonstrate imbalanced performance and unfair behaviors (e.g., different model performance across subpopulations). Such fairness issues have been extensively studied within the machine learning and data management communities, among others, in the past decade [1]–[19].

In this paper, we recast these fairness issues as a problem of data drift, and we address it with solutions that directly aim to improve the conformance between data and model. Although fairness issues may be caused by a breadth of factors, they often manifest as data imbalances (e.g., skewed representation in populations or positive labels, or subpopulations exhibiting differing patterns in the distribution of their attributes and labels). Such imbalances can be modeled as a type of internal drift between subpopulations (or groups for brevity), which can cause a model to perform poorly over minority groups in its deployment. Specifically, as a learning algorithm attempts to identify a pattern within a given population, it tends to prioritize the pattern of the majority group[1] due to their prevalence. The produced model thus does not *conform* to the minority group, and, as a result, its predictions for members of that group are unfair and less reliable.

Example 1. *The dataset in Fig.1 contains two groups, which are color-coded in blue and orange. The attributes $X1$ and $X2$ of these groups show dissimilar distributions, as can be observed from the x and y-axis, respectively, indicating a data drift over groups. The positive and negative ground truth labels for a classification model are marked by circles and triangles, respectively. A model trained on this dataset (black line) tends to conform to the majority group (blue points). As a result, a significant number of minority records (orange points) receive incorrect predictions (orange points with red outline).*

We view unfair model behavior (regardless of cause) as drift between groups and we use drift quantification techniques to characterize and resolve it. Our solutions are *non-invasive* i.e., they do not alter the data or the learning algorithm. Instead, we aim to *improve fairness by improving the conformance between minority data and models*. We resolve drift between groups w.r.t. the model's conformity through two strategies:

Strategy 1: Our **model-splitting** approach, **DIFFAIR**, is designed around a simple strategy to address data drift over groups: training separate models for different groups, such that each produced model conforms to its training data better.

Example 2. *For the dataset in Fig.1,* DIFFAIR *produces a separate model (orange dashed line) for the minority group (orange points), which is significantly different from the overall model (black line).* DIFFAIR *also produces a model for the majority group, closely aligned with the black line (not displayed*

[1]We use majority and minority groups to refer to the populations that are over- and under-represented, respectively, in the data or in the preferred labels.

2375-026X/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDE60146.2024.00172

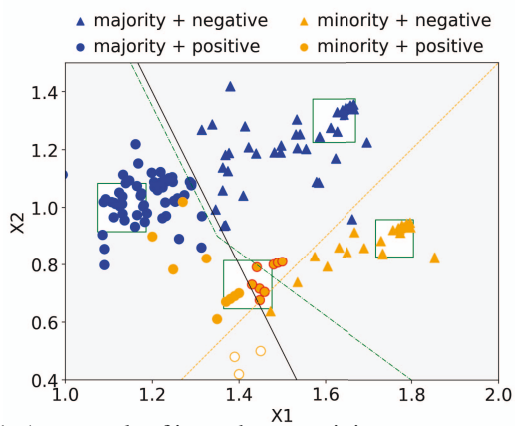


Fig. 1: An example of input data containing two groups: majority and minority color-coded in blue and orange, respectively. The attributes X_1 and X_2 of these groups show dissimilar distributions, indicating a drift over groups. An unfair model (black line) prioritizes the pattern of the majority group (blue points), and predicts poorly (with fewer positive outputs) for minorities (e.g., orange points with red outline). DIFFAIR improves the conformance between data and model by building separate models for different groups (e.g., orange dashed line for minorities). CONFAIR improves the conformance by deriving a single model (green dash-dot line) that emphasizes the densest areas (green squares) of the input data for both groups.

to avoid visual clutter). By building models that conform to different groups, DIFFAIR *reduces the number of incorrect predictions for the minority, leading to a fair outcome, i.e., the ratio of positive model outcomes is similar for both groups.*

Similar strategies of developing and deploying multiple models to address drift in unseen data have also been employed in production settings of ML models [20]–[25]. For example, ensemble learning directly combines the output of several models according to some aggregation rules or explicit fairness objectives to derive predictions. Within the family of model-splitting strategies, DIFFAIR differs from these methods by deriving predictions based on models' conformity with training data. The novelty of our approach lies in the use of Conformance Constraints (CCs) to quantify such conformance. DIFFAIR is unique in the way that it uses CC-based data profiling to assign classifiers, i.e., a point is assigned with the classifier trained on the subset of data it most resembles. DIFFAIR does not aggregate predictions, though one could design a hybrid method that uses CC violations to appropriately weigh different predictions. CCs automatically learn from a given dataset numerical constraints that summarize the distribution of data points in terms of their distance to the densest areas in the input data [26]. We use CCs as an off-the-shelf tool to derive these summary descriptors for each subgroup, and to quantify the degree of drift of a serving tuple from each group.

Example 3. *Profiling the four subsets of the data in Fig.1 (blue circles, blue triangles, orange circles, and orange triangles) using CCs, results in four sets of constraints (depicted as green rectangles). Each set of constraints describes the densest areas in the corresponding subset of the data by some distributive patterns of the attributes. For example, the constraints for the minority positive group (orange circles) specify the rectangular region* $1.38 \leq X1 \leq 1.5 \wedge 0.68 \leq X2 \leq 0.8$. *The distance of a point from this region positively corresponds to the point's violation of these constraints, while points inside the region get zero violations.*

During the deployment of group-dependent models, DIFFAIR selects for each tuple the model that the tuple best conforms to, i.e., the model that results in the minimum CC violation for the serving tuple. DIFFAIR has three advantages over the naive approach of using group membership to separate models: (1) it does not need group membership information, which may be unavailable due to legal and discrimination considerations (e.g., protected attributes such as gender, race, and disability status); (2) it similarly avoids quality issues commonly observed with demographic attributes due to privacy and discrimination considerations, which disproportionately affect minority groups [27]; (3) it assigns the best model to individuals, who may deviate from their own group's pattern and be served better by another group's model. The alternative approach of learning a separate model to determine model selection is prone to mistakes due to drift between groups, while DIFFAIR is robust to data drift by design (i.e., explicitly quantifying drift through CCs).

A general limitation of model-splitting approaches, however, is that their performance can degrade significantly when a group's representation is particularly poor in the data. For example, if the population of a group is very small, or if its labels are severely skewed (e.g., mostly negative labels), the model trained on such data will likely be of low quality. Suppose a minority group has 90% negative and 10% positive labels; a model that always assigns a negative prediction may achieve high accuracy over this data, but it is clearly unreliable. This limitation is hard to address with a model-splitting approach, as small data size and skewed representation offer little opportunity for improving the models. Instead, we propose a single-model reweighing approach that naturally avoids these pitfalls, which we describe next.

Strategy 2: Collecting more data to address skewed representation is often an expensive proposition. Our **reweighing** approach, **CONFAIR**, is a single-model strategy designed to overcome this limitation by appropriately adjusting the weights of existing data. CONFAIR profiles the data using CCs to identify the densest areas of the input, and assigns higher weights to tuples that best conform to the identified CCs. These weights are then used in model training. Models that do not support weights directly can still employ a weighted sampling strategy to preprocess the training data accordingly.

Example 4. *For the dataset in Fig.1,* CONFAIR *assigns high weights to tuples located within the constraint areas (e.g., points inside green squares). The produced model (green dash-dot line) corrects several erroneous predictions of the original*

	DRO [28]	LAH [29]	CAP [18]	KAM [2]	OMN [30]	CONFAIR
non-invasive wrt data	✓	✓	×	✓	✓	✓
non-invasive wrt model	×	×	✓	✓	✓	✓
flexible intervention	×	×	×	×	✓	✓
intra-group variability	✓	✓	×	×	×	✓

Fig. 2: CONFAIR provides non-invasive and flexible interventions; by allowing for variable weights within the same group, CONFAIR can better balance the fairness-accuracy tradeoff.

model (black line), i.e., most of the red-outlined orange points are now correctly classified. The two groups also get similar ratios of positive predictions, indicating a fair outcome.

Reweighing strategies have been used in prior art to improve the fairness of ML models [2], [18], [28]–[30]. The intuition of such strategies is that balancing the weighted representation of groups can amplify the loss of the minority group during training, thus leading to models that better optimize for this loss. Much of the prior work focuses on adjusting the weights during iterative training of a model [28], [29], [31]. Such interventions learn the weights through a black-box training process, which cannot be audited or adjusted. In contrast, CONFAIR supports *flexible intervention*: by allowing users to control the reweighing impact, they can adjust the tradeoff between fairness and accuracy. Moreover, CONFAIR follows a *non-invasive* strategy that does not alter the data or model. Among other non-invasive techniques [2], [30], CONFAIR stands out by allowing variability in the weights assigned to the members of a subpopulation. Instead of assigning identical weights to all tuples within a minority group, CONFAIR only increases the weights of those individuals that conform to the densest part of the group's data. This way, CONFAIR avoids amplifying outliers and noise, which could mislead the training and harm model accuracy. Figure 2 summarizes these points of comparison between CONFAIR and prior art.

Model-splitting vs reweighing. CONFAIR and DIFFAIR are designed to support different scenarios of data drift over groups. In cases of significant drift, DIFFAIR is generally better, as it may not be possible to build a single well-conforming model (see evaluation in Section IV-B).

Example 5. *For the dataset in Fig.1,* CONFAIR *does not resolve all erroneous predictions for the minority group, i.e., red-outlined points still fall on the wrong side of the green dash-dot line. In contrast,* DIFFAIR *can produce a model (orange dashed line) that better conforms to the minority group.*

When drift over groups is less stark, CONFAIR can be more effective than DIFFAIR as it applies an early-stage intervention (focusing on the training data), while avoiding the loss of predicting power in splitting input and developing multiple group-dependent models.

Scope. In this paper, we aim to improve the fairness of ML models by increasing the conformance between the model and data. Our work focuses on *group fairness*, which characterizes if any group, collectively, is discriminated against. In this paper, we focus on group fairness measured by disparate impact [1], [5], [32], but our approach also supports other fairness metrics (e.g., Equalized Odds), discussed in the full version [33]. These groups are often defined by demographic attributes, such as gender, race, disability status, etc., but this is not a requirement for our methods.

In relation to methods in the fairness literature, our approach focuses on data-oriented interventions but requires no invasive changes to the data itself. Compared to those methods that alter the data directly (known as pre-processing interventions) [2], [5], [7], [8], [18], our approach may be less powerful due to the non-invasive setting, while the former allows arbitrary changes to the data such that one can achieve greater flexibility in obtaining desired fairness improvement. However, by being non-invasive, our approach poses a lower risk of introducing unintended drift between the training and serving data. Furthermore, we take into account the distribution of numerical attributes, providing a rich space for fine-tuning the balance between fairness and utility, and enabling our approach to be easily combined with others that operate in the categorical domain. Our approach is also different from the methods that alter the learning algorithms or the outcomes directly [3], [4], [6], [10], [13]–[15], [17], [34]–[36], known as in- or post-processing interventions. These methods often require access to models or learning algorithms to fine-tune (or reassign) the loss for each data point during the development (or deployment) of fair models, making them less interpretable and difficult to audit due to technical complexity. In contrast, our approach is explicit and easy to interpret and audit. Our techniques rebalance fairness for specified minority and majority groups. This process may lead to imbalances in the treatment of other unidentified subpopulations, which is a common effect in fairness repairs (e.g., repairing fairness w.r.t. gender may lead to imbalances w.r.t. race) [37], [38].

In relation to clustering, our approach is designed for supervised learning tasks rather than unsupervised settings such as clustering tasks. Fair clustering tasks may differ based on their definitions of fairness (we refer the reader to a survey [39] for more detail). The employment of conformance constraints in our approach resembles clustering, but the two have different objectives (identifying clusters vs determining dense areas of the input data). Clustering may be repurposed to perform the same task, but it is not an effective alternative to the use of numerical constraints as in CCs. This is because most clustering techniques are sensitive to the separation of clusters in input data, requiring that clusters are well separated from each other. This assumption is not valid in much of our experimental data, where drift over groups (or clusters) exists but the groups are not clearly separated in the input space. Moreover, clustering methods are less useful in scenarios where individuals may deviate from their own cluster and would receive better outcomes if they were assigned to another cluster (e.g., assigned to the model for another group in DIFFAIR). By analyzing the distribution of attributes, CCs are more robust towards data drift than clustering.

Considering other data profiling primitives, CCs offer two important advantages. (1) Numerical attributes provide rich data context and great flexibility in achieving desired

fairness balance, and have not been exploited in deriving fairness interventions. The focus on a continuous domain makes our approach orthogonal to methods that work in categorical domains to derive interventions, thus presenting the potential for hybrid methods. (2) Constraints can be derived efficiently over large datasets (i.e., linear in the number of tuples and cubic in the number of attributes), which makes our methodology practical for real-world data. Ultimately, our approach can integrate with other profiling tools that produce similar quantitative descriptions of input data.

Contributions. We make the following contributions:

- We recast the problem of fairness in ML models as an issue of drift over groups in input data, and, consecutively, as a problem of conformance of the model to its underlying data. (Section II)
- We present DIFFAIR, a model-splitting strategy that improves conformance between model and data by deriving group-dependent models and deploying these models based on the similarity of serving tuples to the training data of each model. Experiments show that DIFFAIR is a better solution to improve the fairness of ML models for scenarios, where a single model is impossible to conform to all groups of input data. (Section III-A)
- We present CONFAIR, a single-model strategy that reweighs the training tuples based on the densest areas of input data, thus producing a single model with balanced predictive accuracy across groups. Experiments show that CONFAIR outperforms existing reweighing techniques, and remains robust when its weights are used by different learning algorithms, in contrast with other prior art. (Section III-B)
- We augment our techniques with density estimation to improve the tightness of derived conformance constraints. (Section III-C)
- We evaluate our methods against 7 real-world datasets and 4 alternative approaches. We demonstrate gains against these baselines and show that our methods improve fairness in ML models, while maintaining utility on par with that before interventions. (Section IV)

II. FRAMING FAIRNESS AS DATA DRIFT

In this section, we formalize our notation and problem, we then provide a high-level description of our model-splitting and reweighing strategies, and, finally, we review some necessary background on Conformance Constraints (CCs), a recently-proposed profiling primitive that we use as an off-the-shelf tool in our methods.

A. Notations and problem statement

We first discuss the notations used in the paper. We denote variables with upper-case letters, e.g., X and Y; values with lower-case letters, e.g., n, m, c, i, and j; sets of variables or values with boldface symbols, e.g., $\mathbf{X}$ or $\mathbf{t}$; and bags of variables with calligraphic symbols, e.g., $\mathcal{D}$.

Data. We assume data $\mathcal{D}$ that consists of $n = |\mathcal{D}|$ tuples. Each tuple is described by a set of attributes $\mathbf{X}$ with cardinality $m = |\mathbf{X}|$ and a target attribute Y with c distinct classes (or labels).

Groups. For ease of exposition, and without loss of generality, we assume that $\mathcal{D}$ can be partitioned into a majority group $\mathbf{W}$ and a minority group $\mathbf{U}$.[2] For the purposes of our work, we use the term minority to refer to a group $\mathbf{U}$ that is under-represented in the data, either with respect to the overall population, i.e., $|\mathbf{U}|$ is small, or with respect to the target attribute Y within $\mathbf{U}$, i.e., there exists $i \in [1, c]$, with $\mathbf{U}_i = \{\mathbf{t} | \mathbf{t} \in \mathbf{U} \wedge \mathbf{t}.Y = i\}$, such that $|\mathbf{U}_i|$ is small. We further assume a user-specified binary mapping function $g : \mathbb{R}^{n \times m} \mapsto [0, 1]$ that takes as input a tuple $\mathbf{t}$ and maps it to $\mathbf{W}$ or $\mathbf{U}$. Typically, g is a simple function over one or more attributes in $\mathbf{X}$. For example, based on the color of the data points in Fig. 1, a tuple can be assigned to the "blue" majority group or the "orange" minority group.

Model. We assume a model $f : \mathbb{R}^{n \times m} \mapsto \mathbb{R}^{n \times c}$, which takes as input a tuple $\mathbf{t} \in \mathcal{D}$ and outputs a prediction as one of the c classes of the target attribute Y. We denote the predictions of f on $\mathcal{D}$ by $\hat{Y}$. We use the following standard process to develop a model f. We partition the input $\mathcal{D}$ into three disjoint sets: training $\mathcal{D}^t$, validation $\mathcal{D}^v$, and deploy $\mathcal{D}^d$. We train f on $\mathcal{D}^t$, optimize for its hyperparameters on $\mathcal{D}^v$, and deploy and evaluate it on $\mathcal{D}^d$. Tuples are assigned into these three sets independently at random (i.i.d.).

Metrics. A fairness metric $\Delta(\mathbf{W}, \mathbf{U})$ quantifies the difference in predictions $\hat{Y}$ between the majority $\mathbf{W}$ and minority $\mathbf{U}$. A lower value of $\Delta(\mathbf{W}, \mathbf{U})$ indicates less bias in the predictions of f. A utility function $\Sigma(Y, \hat{Y})$ quantifies the similarity between the target attribute Y and the output $\hat{Y}$ of f. A higher value of $\Sigma(Y, \hat{Y})$ indicates higher utility for the model f.

Definition 1 (Non-invasive fair learning)**.** *Given a dataset $\mathcal{D}$, a mapping function g, and a learning algorithm f, non-invasive fair learning seeks a learning framework that, without altering the data in $\mathcal{D}$ or the learner f, it trains a model f' using learner f, such that the fairness difference $\Delta(\mathbf{W}, \mathbf{U})$ is minimized, while the utility $\Sigma(Y, \hat{Y})$ is maximized.*

B. Strategy overview: improving conformance

We described how data drift (across groups) leads to unfairness in ML models. As a result, the produced model may not conform to the minority group, whose predictions are, thus, not reliable. To improve the conformance between the model and data, we propose two strategies: a model-splitting approach (DIFFAIR) and a reweighing approach (CONFAIR).

DIFFAIR follows a simple strategy: train separate models for different groups and deploy these group-dependent models collectively to improve the conformance between model and data. A naive version of this strategy, which we will simply refer to as MULTIMODEL, may split the input data based on group membership (e.g., blue and orange points in Fig. 1), train multiple models (one for each group), and choose a model to use for a serving tuple based on its group membership during deployment. In contrast to the naive MULTIMODEL

[2]Our approach can be easily extended to the general case, where the input data contains multiple majority and minority groups.

method, DIFFAIR does not use group membership in assigning models for serving tuples. Instead, it learns constraints to describe each group's training data using CCs. For each serving tuple, DIFFAIR chooses the model that minimizes the tuple's violation score against the CCs of the model's training data.

This strategy has two important advantages compared to simply relying on group membership:

- DIFFAIR affords compliance with legal considerations regarding discrimination when it does not rely on group membership during deployment. Such membership information can be sensitive and protected (e.g., gender, race, disability status, etc.). Additionally, DIFFAIR is robust to erroneous membership during deployment, i.e., individuals with wrong membership information (e.g., auto-filled or misclassified) still receive correct predictions.
- DIFFAIR is flexible at handling individuality. Instead of deploying a model based on group membership, DIFFAIR chooses a model for each serving tuple considering the distribution of its attributes, i.e., assigning a model to which a tuple conforms better, regardless of which group the tuple formally belongs to.

The novelty of DIFFAIR lies in its use of CCs to model drift and separate data based on this drift. DIFFAIR builds a simple mechanism around this intuition: it serves each tuple using the model that results in the minimum CC violation score. One can easily augment this with more sophisticated mechanisms (e.g., ensemble learning), where conformance constraints can be used as an explicit heuristic for aggregating predictions from involved models.

CONFAIR aims to achieve better conformance between the model and data through a reweighing strategy. It assigns weights to tuples in the training data, and an ML model then takes the new weighted data as input. CONFAIR determines these weights based on the conformance constraints that are learned over each group's data. It increases the weights of the tuples that best conform to the produced constraints (e.g., points located inside the green rectangles in Fig. 1). Training a single model makes CONFAIR more robust against the poor representation of groups, whereas model-splitting approaches, such as MULTIMODEL and DIFFAIR, are limited by the need for adequate group representation to train a reasonable model.

C. Background on Conformance Constraints

We proceed with a brief overview of conformance constraints [26]. We generally follow the formalism and notations of the original paper, but we omit or simplify some details in the summary we provide here; we refer the reader to Fariha et al. [26] for more detail.

A conformance constraint is a constraint over arithmetic relationships involving multiple numerical attributes. More formally, a constraint ϕ is an expression of the form $\phi := \epsilon^{lb} \leq F(\mathbf{X}) \leq \epsilon^{ub}$, where ϵ^{lb} and ϵ^{ub} are the lower and upper bounds of the projection $F(\mathbf{X})$. $F(\mathbf{X})$ is a linear combination of numerical attributes $\mathbf{X}$ in data $\mathcal{D}$. We use $\mathbf{\Phi}$ to denote a set of conjunctive constraints. For a tuple $\mathbf{t}$, $\mathbf{\Phi}(\mathbf{t})$ is computed as follows: $\mathbf{\Phi}(\mathbf{t}) := \phi_1(\mathbf{t}) \wedge \phi_2(\mathbf{t}) \cdots \wedge \phi_r(\mathbf{t})$, and $\phi_i(\mathbf{t}) := \epsilon_i^{lb} \leq F_i(\mathbf{t}) \leq \epsilon_i^{ub}, \forall i \in \{1, 2, \ldots, r\}$. $F_i(\mathbf{t})$ is simplified from $F_i(\mathbf{t}.\mathbf{X})$ for readability. In this Boolean semantics, a tuple $\mathbf{t}$ satisfies the constraints $\mathbf{\Phi}$ when $\mathbf{\Phi}(\mathbf{t}) = 1$. Otherwise, $\mathbf{t}$ violates the constraints $\mathbf{\Phi}$.

Fariha et al. [26] also propose quantitative semantics to measure the violation of a tuple $\mathbf{t}$ for constraints $\mathbf{\Phi}$, denoted as $[\![\mathbf{\Phi}]\!](\mathbf{t})$. We compute the violation $[\![\mathbf{\Phi}]\!](\mathbf{t})$ as follows:

$$
\begin{aligned}
[\![\mathbf{\Phi}]\!](\mathbf{t}) &= \sum_{i=1}^{r} q_i \cdot [\![\phi_i]\!](\mathbf{t}) \\
[\![\phi_i]\!](\mathbf{t}) &= \eta(\frac{dist(F_i, \mathbf{t})}{\sigma(F_i(\mathbf{t}))}), \forall i \in \{1, 2, \ldots, r\} \\
dist(F_i, \mathbf{t}) &= max(0, F_i(\mathbf{t}) - \epsilon_i^{lb}, \epsilon_i^{ub} - F_i(\mathbf{t})) \\
\eta(x) &= 1 - e^{-x}
\end{aligned}
$$

Where $q_i \in \mathbb{R}^+, \forall i \in \{1, 2, \ldots, r\}$ is the coefficient of the expression $\phi_i \in \mathbf{\Phi}$ and $\sum_{i=1}^{r} q_i = 1$. This factor represents the importance of the expression ϕ_i and is computed as $q_i = 1 - \frac{\sigma(F_i)}{max(\sigma(\mathbf{F})) - min(\sigma(\mathbf{F}))}$, where $\mathbf{F} = \{F_1, \ldots, F_r\}$ consists of all the projections involved in expressions $\mathbf{\Phi}$. It is saying that the lower the standard deviation $\sigma(F_i)$ of the projection F_i is, the more important the expression ϕ_i is in computing the violation of the tuple $\mathbf{t}$. In other words, the set of constraints, whose projections have low standard deviations, is more effective at characterizing tuples in $\mathcal{D}$.

In this quantitative semantic, a tuple $\mathbf{t}$ satisfies the constraints $\mathbf{\Phi}$ (i.e., $\mathbf{\Phi}(\mathbf{t}) = 1$) when the violation $[\![\mathbf{\Phi}]\!](\mathbf{t}) = 0$. Otherwise, the lower the violation $[\![\mathbf{\Phi}]\!](\mathbf{t})$ is, the more $\mathbf{t}$ conforms to $\mathbf{\Phi}$. We employ these quantitative semantics in our approach to profile groups' data. In this paper, we use $\mathbf{\Phi}^w$ and $\mathbf{\Phi}^u$ to denote the sets of constraints derived over the majority and minority groups $\mathbf{W}$ and $\mathbf{U}$, respectively.

III. FAIRNESS THROUGH CONFORMANCE

In this section, we present two methods that aim to improve fairness in learning, by improving the conformance of models to underlying data. We first describe DIFFAIR, which enhances the naive method MULTIMODEL by using conformance constraints to deploy the appropriate model for serving tuples (Section III-A). Next, we introduce CONFAIR, which uses conformance constraints to assign weights to the training data and then build a single model over weighted data (Section III-B). Finally, we present an optimization that improves the effectiveness of the derived CCs: we use density estimation to preprocess the input and filter high-variance data, leading to tighter constraints.

A. DIFFAIR

Our model-splitting approach is designed around a simple strategy: train separate models for different groups, such that each produced model better conforms to its underlying data. DIFFAIR augments this simple strategy with conformance constraints: Roughly, DIFFAIR derives CCs from the training data of each model, and calculates the violation of each serving tuple against each set of constraints; it then selects the model that corresponds to the lowest violation to serve the tuple.

Algorithm 1 DIFFAIR

Require: Dataset $\mathcal{D}$ with attributes $\mathbf{X}$, a target attribute Y, and a mapping function g.
Ensure: A fair model f' over $\mathcal{D}$

1: $Partition(\mathcal{D}) \rightarrow \{\mathcal{D}^t, \mathcal{D}^v, \mathcal{D}^d\}$
▷*Identify majority and minority groups* $\mathbf{W}$ *and* $\mathbf{U}$ *in* $\mathcal{D}^t$ *and* $\mathcal{D}^v$
2: $\mathbf{W}^t = \{\mathbf{t} | g(\mathbf{t}) = 0, \mathbf{t} \in \mathcal{D}^t\}$, $\mathbf{U}^t = \{\mathbf{t} | g(\mathbf{t}) = 1, \mathbf{t} \in \mathcal{D}^t\}$
3: $\mathbf{W}^v = \{\mathbf{t} | g(\mathbf{t}) = 0, \mathbf{t} \in \mathcal{D}^v\}$, $\mathbf{U}^v = \{\mathbf{t} | g(\mathbf{t}) = 1, \mathbf{t} \in \mathcal{D}^v\}$
4: $\mathbf{C}^w = \emptyset$, $\mathbf{C}^u = \emptyset$
5: **for** $i \leftarrow 1, \dots c$ **do**
6: $\quad \mathbf{W}_i^t = \{\mathbf{t} | \mathbf{t}.Y = i, \mathbf{t} \in \mathbf{W}^t\}$; $\mathbf{U}_i^t = \{\mathbf{t} | \mathbf{t}.Y = i, \mathbf{t} \in \mathbf{U}^t\}$
7: $\quad \mathbf{\Phi}_i^w = GetCCs(\mathbf{W}_i^t)$; $\mathbf{\Phi}_i^u = GetCCs(\mathbf{U}_i^t)$
8: $\quad \mathbf{C}^w \leftarrow \mathbf{C}^w \cup \mathbf{\Phi}_i^w$, $\mathbf{C}^u \leftarrow \mathbf{C}^u \cup \mathbf{\Phi}_i^u$
9: Train f^w on $\mathbf{W}^t$; Train f^u on $\mathbf{U}^t$;
10: Validate f^w on $\mathbf{W}^v$; Validate f^u on $\mathbf{U}^v$;
11: **for** $\mathbf{t} \in \mathcal{D}^d$ **do**
▷*Produce predictions for all serving tuples*
12: $\quad$ PREDICT(t, $\mathbf{C}^w$, $\mathbf{C}^u$)
13: **return** $f' \leftarrow (f^w, f^u, \mathbf{C}^w, \mathbf{C}^u)$

14: **procedure** PREDICT(t, $\mathbf{C}^w$, $\mathbf{C}^u$)
15: $\quad v^w(\mathbf{t}) = min_{\mathbf{\Phi}^w \in \mathbf{C}^w} [\![\mathbf{\Phi}^w]\!](\mathbf{t})$
16: $\quad v^u(\mathbf{t}) = min_{\mathbf{\Phi}^u \in \mathbf{C}^u} [\![\mathbf{\Phi}^u]\!](\mathbf{t})$
17: $\quad$ **if** $v^w(\mathbf{t}) < v^u(\mathbf{t})$ **then**
18: $\quad\quad$ **return** $f^w(t)$
19: $\quad$ **else**
20: $\quad\quad$ **return** $f^u(t)$

Algorithm 1 presents this strategy in more detail. The algorithm takes as input a dataset $\mathcal{D}$ and a mapping function g to define groups. While the pseudo-code assumes a binary function g, it can easily generalize to more than two groups. The algorithm first splits $\mathcal{D}$ into three disjoint sets: training $\mathcal{D}^t$, validation $\mathcal{D}^v$, and deployment $\mathcal{D}^d$ (line 1). Then it proceeds to identify groups within the first two sets using the mapping function g (lines 2–3). The algorithm proceeds to derive constraints for groups within the training set $\mathcal{D}^t$, and DIFFAIR does so within each set of labels (lines 4–8). This is because, in practice, individuals with positive and negative labels may display distinct patterns in their attributes (e.g., triangles and circles in Fig. 1). DIFFAIR, therefore, leads to a tighter and higher-quality set of constraints.

DIFFAIR proceeds to train two group-dependent models f^w and f^u for the majority and minority, respectively, and optimizes their parameters over the corresponding validation sets (lines 9–10). The PREDICT procedure (lines 14–20) outputs predictions for each serving tuple $\mathbf{t}$ in $\mathcal{D}^d$ solely by the constraints $\mathbf{C}^w$ and $\mathbf{C}^u$ without referring to the mapping function g. The goal of PREDICT is to identify the best model to deploy. First, we determine the label group within the majority and the label group within the minority that tuple t is closest to (has minimal violation (lines 15–16). Then, comparing the majority and minority violation scores, we select the appropriate model (lines 17–20). Note that DIFFAIR picks the model with the best conformance, even if a tuple does not actually belong to that group. By prioritizing conformance, DIFFAIR achieves better accuracy, especially for members of the minority group, leading to improved model fairness.

Algorithm 2 CONFAIR

Require: Dataset $\mathcal{D}$ with attributes $\mathbf{X}$, a target attribute Y, a mapping function g, and intervention factors α^w and α^u for the majority and minority groups, respectively.
Ensure: Dataset $\mathcal{D}$, augmented with a weight attribute.

$\mathbf{t}.S = 0, \forall \mathbf{t} \in \mathcal{D}$ ▷*add weight attribute* S *with initial value 1*
for all c **do** ▷*partition* $\mathcal{D}$ *according to target* Y *and function* g
$\quad \mathbf{W}_c = \{\mathbf{t} | g(\mathbf{t}) = 0, \mathbf{t}.Y = c, \mathbf{t} \in \mathcal{D}\}$
$\quad \mathbf{U}_c = \{\mathbf{t} | g(\mathbf{t}) = 1, \mathbf{t}.Y = c, \mathbf{t} \in \mathcal{D}\}$
$\quad \mathbf{\Phi}_c^w = GetCCs(\mathbf{W}_c)$; $\mathbf{\Phi}_c^u = GetCCs(\mathbf{U}_c)$ ▷*get constraints*
▷*Update weights for population and label skew*
$\quad \mathbf{t}.S \leftarrow \mathbf{t}.S + \frac{|\{\mathbf{t} | \mathbf{t}.Y = c, \mathbf{t} \in \mathcal{D}\}|}{|\mathcal{D}|} * \left(\frac{g(\mathbf{t}) * |\mathbf{U}|}{|\mathbf{U}_c|} + \frac{(1 - g(\mathbf{t})) * |\mathbf{W}|}{|\mathbf{W}_c|}\right)$
▷*Find conforming tuples*
$\quad \mathbf{T}_c^w = \{\mathbf{t} | \mathbf{t} \in \mathbf{W}_c, [\![\mathbf{\Phi}_c^w]\!](\mathbf{t}) == 0\}$
$\quad \mathbf{T}_c^u = \{\mathbf{t} | \mathbf{t} \in \mathbf{U}_c, [\![\mathbf{\Phi}_c^u]\!](\mathbf{t}) == 0\}$
▷*Increase the weight of tuples conforming with minority positive labels*
for $\mathbf{t} \in \mathbf{T}_1^u$ **do**
$\quad \mathbf{t}.S \leftarrow \mathbf{t}.S + \alpha^u$
▷*Increase the weight of tuples conforming with majority negative labels*
for $\mathbf{t} \in \mathbf{T}_0^w$ **do**
$\quad \mathbf{t}.S \leftarrow \mathbf{t}.S + \alpha^w$
return $\mathcal{D}$

The run-time complexity of Algorithm 1 is bounded by the derivation of conformance constraints—$O(q^3)$ with q numerical attributes for computing projections and $O(nm^2)$ with n tuples and m attributes for producing the constraints [26]. The training of models takes $O(nm)$ with n tuples and m attributes using classification algorithms such as Logistic Regression.

B. CONFAIR

Multi-model approaches are more susceptible to groups' poor representation in the input. Splitting the dataset to produce multiple models weakens the predictive power over the input with small and often skewed group representation. We explore a single-model strategy that boosts conformance between the model and data (especially of minority groups), without diluting learning power as in multi-model approaches.

In this section, we present CONFAIR in Algorithm 2, a single-model approach that uses CCs in a novel way to derive weights for the training data. For ease of exposition, the pseudo-code assumes binary labels (i.e., $c = 2$). It further assumes that the positive labels are over-represented in the majority group, while the opposite holds for the minority. These assumptions are simply for ease of presentation, and not true restrictions of the framework. The intervention degrees α^w and α^u are adjustable weight parameters to control the level of intervention that users may wish to apply to the majority and minority groups, respectively. By default, CONFAIR optimizes for disparate impact by applying these weights to appropriate labels; our technical report discusses support for other fairness metrics through adjusting α^w and α^u [33]. CONFAIR only augments the weights of tuples that conform to the identified CCs, resulting in a monotonic behavior of improvement in fairness with respect to the intervention degree; this facilitates tuning the parameter to each application's fairness requirements. In contrast, prior art [30] augments the weights of all tuples in a group; as data is inevitably noisy, this lead to a non-

monotonic relationship between the level of intervention and the achieved fairness (see details in our technical report [33]).

CONFAIR adds a weight attribute S to input $\mathbf{D}$ initialized in line 1. It then partitions $\mathbf{D}$ based on a mapping function g and target attribute Y. For example, the dataset in Fig. 1 would be separated into four parts: the majority group with positive labels (blue circles), the majority group with negative labels (blue triangles), the minority group with positive labels (orange circles), and the minority group with negative labels (orange triangles). CONFAIR proceeds to derive constraints on each part (line 5). It then balances the weights of tuples in each part according to the skew in the groups' population and labels, i.e., increase weights for the minority and decrease values for the majority (line 6). Next, CONFAIR focuses on the tuples that conform to each part (lines 7 and 8). Based on the intervention factors α^w and α^u, CONFAIR adjusts the weights of these conforming tuples (lines 9–12). Recall that the presented pseudo-code makes the assumption that the majority part of the data skews toward positive labels and that the minority part of the data skews toward negative labels. Thus, to achieve balanced predictive accuracy across groups, CONFAIR increases the weights of majority-negative-conforming tuples by α^w, and the weights of minority-positive-conforming tuples by α^u. Note that this assumption is made here for readability. The approach can easily generalize to the labels with multiple classes. And the skew of groups toward the labels can be easily estimated from the data, which can guide the tuning of the intervention factors (e.g., increase weights for the minority group with positive labels or vice-versa). Finally, CONFAIR returns the weight-augmented data to build an ML model.

Similar to Algorithm 1, the run-time complexity of Algorithm 2 is also bounded by the derivation of conformance constraints, which takes $O(q^3)$ with q numerical attributes for computing projections and $O(nm^2)$ with n tuples and m attributes for deriving the constraints [26].

C. Optimizing the derivation of CCs

The effectiveness of conformance constraints is affected by the variance of attributes in the input. A set of constraints learned from data with high variance has low discriminative power: most tuples will have high conformance with broad, permissive constraints. Such weak constraints can critically impact the effectiveness of our methods. In this section, we propose a pre-processing optimization step that filters the input data $\mathcal{D}$ using density estimation, leading to stronger sets of conformance constraints, and by implication, increased effectiveness for DIFFAIR and CONFAIR.

We present our optimization in Algorithm 3. The algorithm processes each target class separately (lines 2–4), and uses density estimation on the majority and minority sets within the target class (lines 5–6). In our implementation, we employ a state-of-art, tree-based, non-parametric kernel density estimator [40], implemented in the scikit-learn library [41].Other kernel density estimators can also work for this step [42]–[44]. Algorithm 3 proceeds to sort the sets $\mathbf{W}_i$ and $\mathbf{U}_i$ based on density and selects the first k tuples to add to $\mathcal{D}'$ (lines 7–8).

Algorithm 3 Optimization for stronger CCs

Require: Dataset $\mathcal{D}$ with attributes $\mathbf{X}$, a target attribute Y, a mapping function g, and a density threshold k.
Ensure: Dataset $\mathcal{D}' \subset \mathcal{D}$
1: $\mathcal{D}' = \emptyset$
2: **for** $i \leftarrow 1, \dots c$ **do** ▷*process each class in target attribute Y.*
3: $\quad \mathbf{W}_i = \{\mathbf{t} | \mathbf{t}.Y = i, g(\mathbf{t}) = 0, \mathbf{t} \in \mathbf{D}\}$
4: $\quad \mathbf{U}_i = \{\mathbf{t} | \mathbf{t}.Y = i, g(\mathbf{t}) = 1, \mathbf{t} \in \mathbf{D}\}$
5: $\quad d^w \leftarrow EstimateDensity(\mathbf{W}_i)$
6: $\quad d^u \leftarrow EstimateDensity(\mathbf{U}_i)$
7: $\quad$ Sort $\mathbf{W}_i$, $\mathbf{U}_i$ in descending order of d^w, d^u, respectively
8: $\quad \mathcal{D}' \leftarrow \mathcal{D}' \cup \{$ first k tuples in $\mathbf{W}_i\} \cup \{$ first k tuples in $\mathbf{U}_i\}$
return $\mathcal{D}'$

The run-time complexity of Algorithm 3 is bounded by the density estimation, which takes $O(mn^2)$ with n tuples and m attributes. This run-time can be improved to $O(m \log(n))$ using optimized data structures such as KD-Tree [40] or Ball Tree [45] for input data in higher dimensions (e.g., $m > 20$).

IV. EXPERIMENTAL EVALUATION

We evaluate DIFFAIR and CONFAIR against a breadth of datasets and methods. Our experiments demonstrate that: (1) CONFAIR outperforms prior art in improving the fairness of models, while maintaining high accuracy (Section IV-A); (2) DIFFAIR can be a better solution compared to CONFAIR for scenarios, where it is difficult to derive a single conforming model (Section IV-B); (3) our optimization for deriving stronger conformance constraints is essential (Section IV-C); (4) our approach shows a reasonable run-time compared to prior art (tech report [33]). We describe each experimental component below.

Datasets. We experiment with 7 real-world datasets used frequently in fairness literature. They include people's demographics and information collected from various domains such as financial and health-related services. We provide a summary description of the major aspects and statistics of each dataset in Figure 3. For more detail, we refer the reader to Bellamy et al. [46] for the *MEPS* and *LSAC* datasets, Kaggle [47] for the *Credit* dataset, and Ding et al. [48] for the American Community Survey (ACS) datasets. We employ four predictive tasks using the ACS datasets, which pertain to people's health insurance (*ACSP* and *ACSH*), employment (*ACSE*), and income (*ACSI*). We choose to skip other frequently-used datasets such as Adult Income [49] and COMPAS [50] because these datasets include very few numerical attributes (e.g., no more than 2) for deriving conformance constraints.

Models. As our methods do not depend on nor intervene with the learners, one could apply them in combination with any learning algorithm. We experiment with two types of learners: Logistic Regression (LR) and XGBoost tree (XGB) from the scikit-learn library [41].

Methods. We briefly describe each approach in our evaluation.
▷ NO-INTERVENTION. This baseline trains a model over the input without applying any fairness intervention. The goal of all

dataset	*MEPS*	*LSAC*	*Credit*	*ACSP*	*ACSH*	*ACSE*	*ACSI*
size	15,675	24,479	120,269	86, 600	250, 847	250, 847	250, 847
# of attributes numerical / categorical	6 / 34	6 / 4	6 / 0	4 / 14	4 / 21	4 / 11	6 / 13
minority group U	non-White	African-American	age $<$ 35	African-American	African-American	African-American	African-American
population of U	61.6%	7.7%	13.7%	9.2%	7.3%	7.3%	7.3%
% positive labels in U	11.4%	56.6%	10.7%	48.3%	9.3%	39.3%	40.2%
% positive labels in W	25.3%	82.5%	6.4%	64.3%	15.1%	45.9%	31.4%
predictive task	high hospital utilization	passing bar exam	serious delay in 2 years	covered by private insurance companies	having health insurance	employment	income poverty rate $<$ 250

Fig. 3: Summary statistics and main aspects of the 7 real-world datasets used in our experiments.

other methods is to achieve improvement in the fairness metrics against this baseline, while maintaining comparable utility.
▷ MULTIMODEL is a simple model-splitting baseline. It partitions the input data into groups by a mapping function g, builds separate models for different groups, and deploys these group-dependent models based on the function g.
▷ DIFFAIR (Section III-A) augments MULTIMODEL, by using conformance constraints in the model deployment, rather than rely on group membership (or the mapping function g), and is thus more flexible and robust to inaccuracies in membership.
▷ CONFAIR (Section III-B) is a single-model reweighing strategy, which derives weights for training tuples, based on their conformance to the CCs in each subgroup.
▷ KAM-CAL (KAM) [2] is a reweighing method (like CONFAIR). It assigns weights to achieve statistical independence between the demographic attributes (defining the groups) and labels. Relying on groups' statistics in the input, KAM does not support adjusting the level of interventions.
▷ OMNIFAIR (OMN) [30] is another reweighing method that aims to achieve fairness for a given metric. We use a variant of OMN that optimizes for Disparate Impact, as this is the metric that CONFAIR inherently optimizes as well (discussed in more detail in the evaluation metrics below).
▷ CAPUCHIN (CAP) [18] is an *invasive* fairness intervention that modifies the input data to ensure that certain constraints hold over its outputs. Since CAP is designed for categorical data, we evaluate this method using the XGB models, which are a better fit for categorical input.

Metrics. We evaluate all methods concerning the fairness and utility of the produced models. We use balanced accuracy (*BalAcc*) as the utility metric, which has been used extensively in the literature to evaluate fairness interventions [46]. It is computed as $\frac{TPR+TNR}{2}$, where TPR and TNR are the True Positive Rate (or sensitivity) and True Negative Rate (or specificity), respectively. *BalAcc* is similar to the Area Under the ROC curve (AUC); both utility metrics are sensitive to the poor representation of minority groups. *BalAcc* is in the range of $[0, 1]$, with higher values corresponding to greater utility.

We report two fairness metrics that are frequently used in the literature: Disparate Impact (*DI*) [5] and Average Odds Difference (*AOD*) [6]. In our implementation, CONFAIR targets *DI*, but our results show that it performs well for both metrics. *DI* is computed as $\frac{SR_{\mathbf{U}}}{SR_{\mathbf{W}}}$ where $SR_{\mathbf{U}} = \frac{|\{\mathbf{t}|\mathbf{t}.\hat{Y}=1,\mathbf{t}\in\mathbf{U}\}|}{|\mathbf{U}|}$ and $SR_{\mathbf{W}} = \frac{|\{\mathbf{t}|\mathbf{t}.\hat{Y}=1,\mathbf{t}\in\mathbf{W}\}|}{|\mathbf{W}|}$ are the selection rates for the minority $\mathbf{U}$ and majority $\mathbf{W}$, respectively. *DI* takes values from 0 to ∞ with 1 being the optimum, i.e., the two groups have the same rate in receiving a positive prediction. Values greater than 1 indicate bias favoring the minority group, which may in fact be reasonable or even desirable in some applications where minorities have suffered from historical disadvantages. CONFAIR implicitly optimizes *DI*, as it increases the weights of tuples with positive labels within the minority group, thus leading to an improvement of the selection rate for the minority.

AOD is a generalized version of Equalized Odds [6], which is computed as $\frac{(FPR_{\mathbf{U}}-FPR_{\mathbf{W}})+(TPR_{\mathbf{U}}-TPR_{\mathbf{W}})}{2}$ where $FPR_{\mathbf{U}}$ and $FPR_{\mathbf{W}}$ are the False Positive Rates for the minority $\mathbf{U}$ and majority $\mathbf{W}$, respectively, and $TPR_{\mathbf{U}}$ and $TPR_{\mathbf{W}}$ are the TPRs for these two groups. *AOD* ranges from 0 to 1, with 0 indicating an optimal case where there is no difference in how a model makes positive predictions for the two groups. *AOD* captures a different aspect of model behavior compared to *DI*; even though CONFAIR is not designed to optimize *AOD*, our experiments will show that its fairness remains robust under this metric.

For ease of interpretation, we report simple transformations of these metrics, so that higher values correspond to better outcomes. Specifically, we report $DI^* = \min(DI, \frac{1}{DI})$, where unfairness ($DI \to 0$ or $DI \to \infty$) is mapped to a low value of DI^*. We report $AOD^* = 1 - abs(AOD)$ such that higher values of *AOD* represent improved fairness. For the remainder of this section, we simply use *DI* and *AOD* to refer to DI^* and AOD^*, respectively.

Experimental steps. We first prepare our data for training. For the datasets *MEPS* and *LSAC*, we use the same preprocessing steps as in the IBM AI Fairness 360 toolkit [46]. Similarly, we preprocess the other data by removing null values, normalizing numerical attributes, and one-hot encoding categorical attributes. We split the processed data into training (70%), validation (15%), and test (15%) sets. For both multi-model and single-model settings, we tune hyperparameters on the validation set and evaluate model performance on the test set. To eliminate the effect of randomness, we repeat the process 20 times and report the average results in our evaluation.

Algorithm parameters. We automatically search for the optimal value of α^u (i.e., the intervention degree for the minority group) over the validation set of each real-world dataset and set $\alpha^w = \alpha^u/2$ (i.e., the intervention degree for the majority group). The tuning of intervention degrees in CONFAIR implicitly optimizes *DI* (i.e., brings it closer to 1). We set the density threshold $k = 0.2 * n$ for all the datasets.

Implementation. We implemented DIFFAIR and CONFAIR in Python 3.7.0, and ran experiments on a computing cluster with 9 nodes (2.40 GHz processor and 256 GB RAM). We open-source our code at https://github.com/DataProfilor/ConFair.

A. Evaluation of CONFAIR

We start with the evaluation of CONFAIR, which is our primary fairness-improvement strategy. (As we will see in Section IV-B, DIFFAIR is strong in cases of significant drift, but loses to CONFAIR in most practical settings.) We compare CONFAIR against three state-of-the-art methods: reweighing strategies, KAM and OMN, and a data-invasive intervention, CAP. We do not compare against in-processing methods [28], [29], [31], [51], which alter the learners, or post-processing methods [3], [6], [34], which alter predictions. A broad evaluation against the existing extensive landscape of fairness interventions is in itself an independent research contribution [52].

CONFAIR vs KAM. As we noted, prior methods have employed weighing strategies as a fairness intervention, but the novelty of CONFAIR lies in the use of CCs to identify and increase the weights of tuples that conform to the densest areas of the input. In contrast, prior art like KAM increases the weights of all tuples within a group, which may end up amplifying outliers and noise.

Figure 4 demonstrates a comparison between CONFAIR and KAM across 7 datasets and two learning strategies. The white bars in the graphs show the performance of the LR (Figures 4a, 4b, and 4c) and XGB models (Figures 4d, 4e, and 4f) before any fairness interventions are applied (NO-INTERVENTION). We note that many of these results indicate significant bias (low fairness measures). CONFAIR and KAM both succeed at improving the fairness of predictions, without notable drops in accuracy (Figures 4c and 4f).

We note that CONFAIR robustly outperforms KAM for the *DI* metric over the *MEPS*, *LSAC*, *ACSP*, and *ACSE* datasets, in the case of XGB models (Fig. 4d); its gains are clear, yet more modest, in the case of LR models (Fig. 4a). Even though CONFAIR does not directly target the *AOD* metric, it still achieves significant improvements over NO-INTERVENTION, and comparable performance to KAM (Figures 4b and 4e). It is important also to highlight that, while the *AOD* values are similar between CONFAIR and KAM, CONFAIR more reliably favors the minority group (striped bars ▨).

For the datasets *Credit* and *ACSI*, CONFAIR performs better against the LR models, concerning both *DI* and *AOD*, and is closer to KAM against the XGB models. For the *ACSH* dataset, CONFAIR outperforms KAM in improving the fairness (for *DI*) of the LR models but behaves worse than KAM with the XGB models. This is because the *ACSH* dataset is very sensitive to the intervention factors of CONFAIR, which are used to determine the increment of tuples' weights. We found that the fine-tuning of these factors for this dataset specifies that $\alpha^u = 0.028$, while the optimum value of α^u for this dataset is 0.03. Overall, CONFAIR gains an edge over KAM as it can fine-tune weights for tuples within groups, which is not possible in KAM.

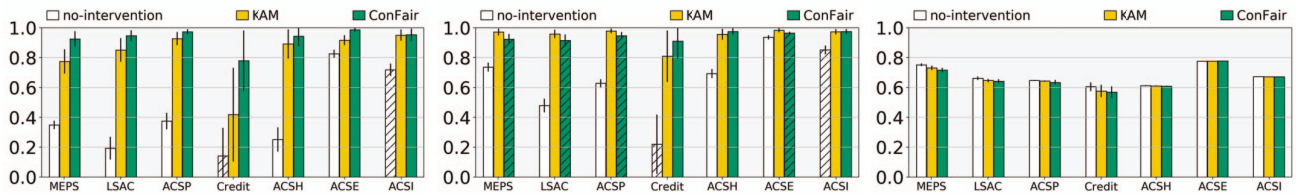
CONFAIR vs OMN. Figure 5 repeats the experiment against an alternative state-of-the-art. Similar to CONFAIR, OMN reweighs input data with an adjustable degree of intervention, but its performance is much less reliable across different datasets. We observe that CONFAIR significantly outperforms OMN in terms of *DI* across all datasets using the XGB models (Fig. 5d). Note that OMN is unable to return a model for the *Credit* and *ACSI* datasets in this setting (the corresponding bars are missing in Figures 5d–5f as XGB fails to converge with the weights produced by OMN). Under the LR models, OMN does particularly poorly for the *Credit*, *ACSH*, *ACSE*, and *ACSI* datasets (Figures 5a and 5b), while CONFAIR demonstrates significant improvement over the fairness metrics.

Another important observation is that OMN interventions result in significant loss of accuracy, while CONFAIR produces models with utility that remain on par with that before interventions (Figures 5c and 5f). In particular, even cases with apparent improvements in fairness for OMN, come at a problematic utility loss. For example, in the case of LR models over the *LSAC* and *ACSP* datasets, OMN shows high improvements in *DI* and *AOD* (Figures 5a and 5b); however, the resulting models in these cases only predict one class (*BalAcc* =0.5 with TPR=1 or TNR=1), which renders the models useless. We indicate these cases with crisscross bars (▩) in the utility graphs (e.g., Figure 5c).

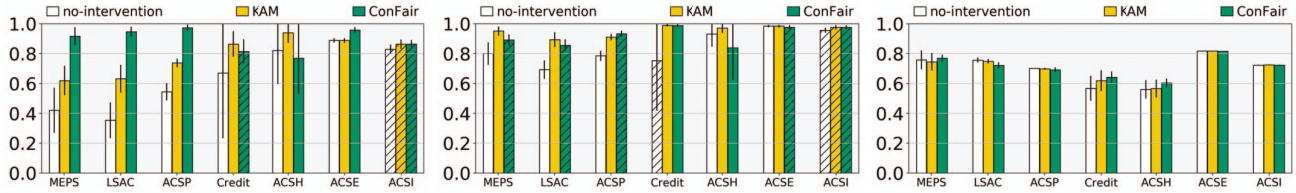
We highlight that, despite both being reweighing strategies, CONFAIR and OMN demonstrate vastly different performances. This is due to their distinct weighing methodologies: OMN adjusts its weights according to the model output, while CONFAIR uses tuple conformance to fine-tune the weights, thus achieving a much more robust fairness-utility balance.

CONFAIR and OMN are both *practically* model-agnostic, in the sense that the weights they derive can be used by any learning algorithm. However, both methods assume a particular model to calibrate their weights. In the experiment of Fig. 5, this calibration was done using the same learner (LR or XGB) as the corresponding experiment. Figure 6 repeats the experiment, but this time each method calibrates its weights assuming a different model than the one eventually trained. For example, in Figures 6a–6c, we calibrate the weights of CONFAIR and OMN over each dataset assuming an XGB model, but we subsequently use the weights to train a LR model. Conversely, in Figures 6d–6f we assume a LR model for weight calibration, but then train XGB models.

CONFAIR's performance naturally drops compared to the previous experiments, however, it still maintains robust improvements in fairness across most datasets, while maintaining high utility. In contrast, OMN becomes less reliable, with inconsistent performance across datasets, and more severe loss of accuracy. For example, OMN is not able to improve the fairness at all for the XGB models (Fig. 6e) over the *Credit*, *ACSH*, *ACSE*, and *ACSI* datasets (corresponding yellow bars have zero height, indicating the maximum difference between groups). This is because the model is not well trained under the input weights (e.g., only outputting one type of prediction), resulting in *BalAcc* lower than 0.5

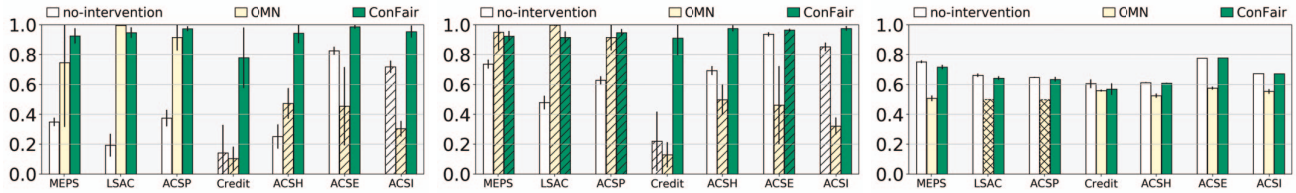


(a) Disparate Impact (*DI*), LR models (b) Average Odds Difference (*AOD*), LR models (c) Balanced Accuracy (*BalAcc*), LR models

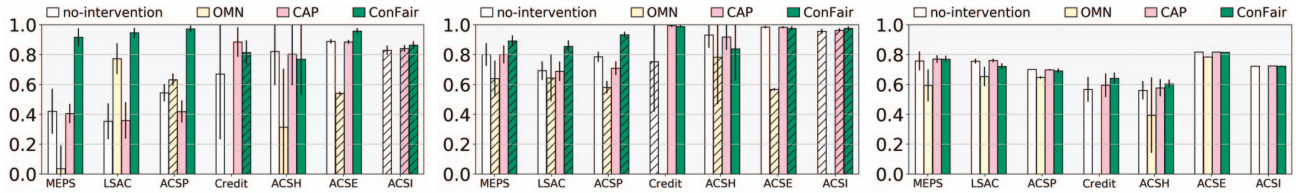


(d) Disparate Impact (*DI*), XGB models (e) Average Odds Difference (*AOD*), XGB models (f) Balanced Accuracy (*BalAcc*), XGB models

Fig. 4: Comparing ConFair to KAM over fairness (measured by *DI* and *AOD*) and utility (measured by *BalAcc*). Striped bars (▨) represent bias favoring minorities.

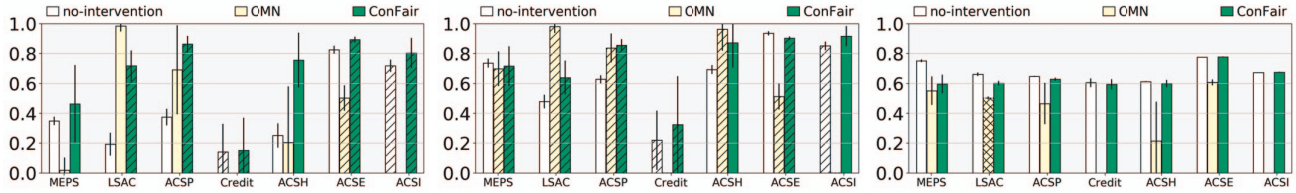


(a) Disparate Impact (*DI*), LR models (b) Average Odds Difference (*AOD*), LR models (c) Balanced Accuracy (*BalAcc*), LR models

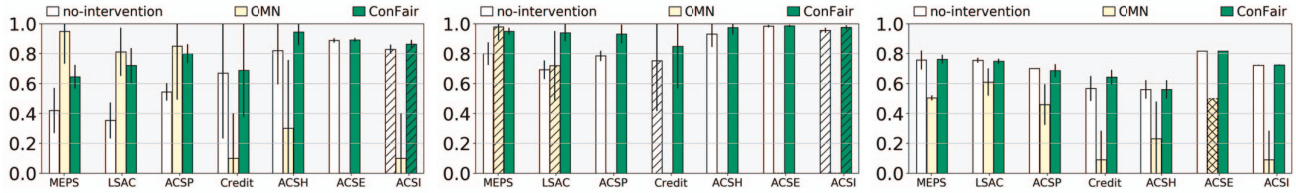


(d) Disparate Impact (*DI*), XGB models (e) Average Odds Difference (*AOD*), XGB models (f) Balanced Accuracy (*BalAcc*), XGB models

Fig. 5: Comparing ConFair to OMN and CAP over fairness (measured by *DI* and *AOD*) and utility (measured by *BalAcc*). Crisscross bars (▩) indicate models that have devolved to useless predictions (e.g., predicting only one class).



(a) Disparate Impact (*DI*), LR models (b) Average Odds Difference (*AOD*), LR models (c) Balanced Accuracy (*BalAcc*), LR models



(d) Disparate Impact (*DI*), XGB models (e) Average Odds Difference (*AOD*), XGB models (f) Balanced Accuracy (*BalAcc*), XGB models

Fig. 6: Comparing ConFair to OMN when models are derived using weights that are not tuned for them. In Fig. 6a, 6b, and 6c, both methods train an LR model using weights tuned for an XGB model. In Fig. 6d, 6e, and 6f, the setting is reversed.

(Fig. 6f). Notably, cases that appear to demonstrate fairness gains for OMN (e.g., the *LSAC* dataset under LR models) come with unacceptable utility loss, producing a model that outputs only one class of predictions (*BalAcc* =0.5 in Fig. 6c).

In summary, despite its model-agnostic premise, OMN demonstrates high model dependence, as its ability to improve fairness is severely hindered when its weights are not calibrated using the "right" model. In contrast, CONFAIR demonstrates robustness, as its weights are primarily driven by conformance over the training data and not the model output.

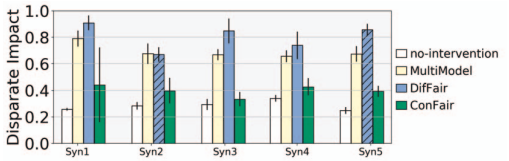
CONFAIR vs CAP. CAP is an invasive intervention that alters the input data to improve the fairness of ML models. Figures 5d–5f demonstrate the performance of CAP across all 7 datasets over XGB models. We see that CONFAIR significantly outperforms CAP in improving *DI* over the *MEPS*, *LSAC*, and *ACSP* datasets. These gains remain present but are more modest concerning *AOD*. The two methods have similar performance across the rest of the datasets, and maintain similar high utility. We need to highlight that CONFAIR achieves performance on par with and often better than CAP, *while remaining non-invasive.* This distinction is significant, as invasive methods are naturally poised to achieve greater fairness improvements, simply due to the flexibility that data changes can afford them. Nevertheless, CONFAIR outperforms CAP, while also avoiding the potential issues of invasive approaches, such as introducing unintended drift in the data.

Key takeaways: CONFAIR outperforms prior art in improving the fairness of ML models, while maintaining high utility. It shows clear and consistent gains compared to other reweighing methods, and achieves on-par or better performance compared to invasive alternatives. It further stays robust when using learners different from those used to calibrate its weights, thus being effectively model-agnostic.

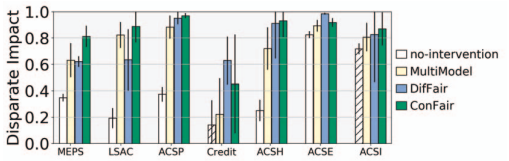
B. Evaluation of DIFFAIR

In this section, we contrast DIFFAIR and CONFAIR, highlighting scenarios where DIFFAIR is the preferable strategy. Intuitively, as a single-model approach, CONFAIR is more generally applicable, and, as we showed in Section IV-A, performs well across our real-world datasets. By design, it can effectively address inter-group drift that may not be obvious in the data. In contrast, DIFFAIR can more effectively address cases of significant drift across groups, where a single-model strategy is unlikely to be able to derive an effective single model. We simulate these scenarios with synthetic data to highlight this strength of DIFFAIR. We proceed to describe the synthetic data generation next.

We generate synthetic data with $N = 11,000$, with $8,000$ majority and $3,000$ minority elements, and 50% positive and 50% negative labels within each group. We generate five synthetic datasets using the *make_classification* function from the scikit-learn library [41]. In the synthetic datasets, the majority and minority groups are distributed over similar areas of the space, with their positive and negative labels following dissimilar distributions, making the generation of a single model extremely challenging. Our technical report describes the synthetic data generation in further detail [33].



(a) synthetic data, LR models



(b) real-world data, LR models

Fig. 7: DIFFAIR can produce stronger fairness outcomes compared to CONFAIR in cases of significant drift (a), and is comparable to CONFAIR in most real-world settings (b).

Figure 7 presents results over the synthetic data (Figure 7a) and the real-world data (Figure 7b) over LR models.[3] While CONFAIR is generally the better choice over the real-world datasets, DIFFAIR results in stronger fairness outcomes over the synthetic data. These improvements have an impact on accuracy, which can be unavoidable in some cases, but the models remain reasonable. For comparison, we also display the results of MULTIMODEL, which uses the group mapping function g to select which model to deploy for each tuple. We observe that the use of CCs in DIFFAIR results in starkly different behavior among the split-model strategies.

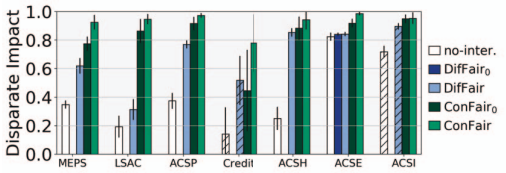
In the real-world datasets (Figure 7b), DIFFAIR performs comparably to CONFAIR in most cases, but the latter is a better choice for two out of the five datasets. The results considering *AOD* as the fairness metric display similar trends, as do the XGB models over the real-world data (see details in our technical report [33]).

Key takeaways: DIFFAIR can be a better approach to improving fairness in learning, in scenarios where there is significant drift across groups and it is difficult to derive a single conforming model.

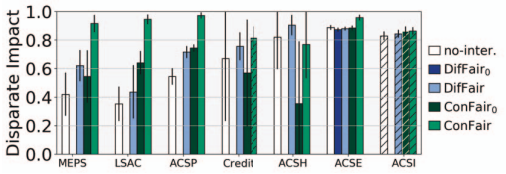
C. Evaluation of CC optimization

Next, we examine the impact of the optimization in Algorithm 3 on the performance of DIFFAIR and CONFAIR. Figure 8 compares DIFFAIR andCONFAIR with variants DIFFAIR$_0$ and CONFAIR$_0$, which do not incorporate the density-based optimization, over the real-world datasets. In both cases of LR (Figure 8a) and XGB models (Figure 8b), the density-based optimization of CCs leads to significant gains in the *DI* metric. DIFFAIR$_0$, in particular, fails in most datasets, so this optimization is critical for DIFFAIR. The reason this optimization is so effective is that it significantly increases the discriminative power of the derived conformance constraints,

[3]XGB models are not a good fit for the synthetic data due to the low separability of the minority group.



(a) real-world data, LR models



(b) real-world data, XGB models

Fig. 8: The density-based optimization is essential in the performance of DIFFAIR and CONFAIR. Variants DIFFAIR_0 and CONFAIR_0 that don't optimize CCs have lower effectiveness.

thus rendering DIFFAIR and CONFAIR more effective. The results follow similar trends for the *AOD* metric, and the utility of the models is not affected by the optimization. Due to space limitations, we omit those plots, as well as additional experiments on the impact of the intervention degree and runtime evaluation; we refer the interested reader to our technical report for additional details and results [33].

V. RELATED WORK

Fair ML. Algorithmic fairness has been studied extensively by the machine learning and data management communities, among others. Several technical reviews survey the topic from different perspectives in recent years [19], [52]–[57]. Our method is similar to approaches that target group fairness or statistical parity to reduce ML bias, which requires equal decision rates across groups [1]–[18]. Another line of work focuses on individual fairness, motivated by the work of Dwork et al. [32], which requires that models assign predictions consistently for similar individuals. Methods are classified as pre-processing, where fairness interventions are applied on the training data [2], [5], [7], [8], [18], in-processing, where interventions are applied on the models [4], [10], [13]–[15], [17], [35], [36], and post-processing, where interventions are applied on predictions [3], [6], [34]. CONFAIR is a pre-processing method, as it operates before deriving ML models. DIFFAIR would also be classified as a pre-processing technique, as it is based on splitting the data before model training. While its primary insight lies in choosing which model to deploy based on conformance, it does not alter model outputs and thus does not fit in the post-processing category as most ensemble learning methods [20]–[25]. Data acquisition [58]–[62] focuses on estimating the cost, benefit, and optimal strategies for collecting additional data; this setting is orthogonal to our case, where we only focus on the data that is available.

The fairness literature has explored reweighing strategies as a method to improve model fairness [2], [28]–[31]. CONFAIR and others [2], [30] adjust the weights of tuples without modifying a learner, whereas many other approaches estimate the weights during the training of a model [28], [29], [31], [51]. Li and Liu [63] estimate the weights for tuples considering how they contribute to multiple fairness metrics and model loss by solving linear programs. CONFAIR adjust weights based on the conformance of tuples to groups' data rather than merely on groups' representation [2] and on model outputs [30]. This also allows CONFAIR to work with user-specified weights for designing fairness interventions tailored to specific tasks.

Data Drift. ML researchers study data drift [64] to identify the drift between different datasets such as training and serving data [25] or identifying drift tuples such as out-of-distribution or misclassified records inside the input data [65]–[67]. Examples of data drift include label [22], [68], [69] and covariate shift [20], [70], [71]. Lahoti [72] et al. focus on data shift between development and deployment stages by identifying different erroneous cases of a produced model on deploying data and advising actions (e.g., collecting more training data) to mitigate the corresponding erroneous cases. Instead of drift between multiple datasets, we focus on the drift over groups within an input, assuming that both label and covariate shifts might appear between groups.

Data Profiling. Many data profiling techniques have been developed in data management to formalize different constraints that characterize the input data [73], such as functional dependencies [74], their variants [75]–[77], [77]–[82], and the more general denial constraints [83]–[86]. Compared to these, conformance constraints [26] describe a dataset with arithmetic expressions of the relations among numerical attributes. Our methods can support other profiling techniques that provide quantitative measures of violations for the profiling constraints.

VI. SUMMARY AND FUTURE WORK

In this paper, we recast the problem of fairness in ML models as a problem of data drift, and, consecutively, as an issue of conformance between data and models. We proposed two intervention strategies that employ conformance constraints (CCs) in novel ways to achieve these conformance goals. Our model-splitting strategy, DIFFAIR, trains separate models for different groups and uses CCs to determine the proper model to derive predictions. Our reweighing strategy, CONFAIR, introduces a novel use of CCs to adjust the weights of tuples in the training data before feeding into an ML model. Our evaluation over seven real-world datasets showed that CONFAIR outperforms prior art and is effectively model-agnostic, and DIFFAIR can be a better option in cases of significant drift, where a single conforming model is unlikely. Future work can explore the integration of other data profiling techniques, which could potentially leverage a combination of attribute types to derive fairness interventions. An overarching goal is to automate fairness repairs and contribute to an end-to-end drift-driven repair system using techniques that detect internal drift and identify the relevant impacted subpopulations.

Acknowledgments. This material is based upon work supported by the NSF under grants 1763423 and 2211918, and by a Google Data Acquisition, Processing, and Analysis award.

REFERENCES

[1] T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," *Data Min. Knowl. Discov.*, vol. 21, no. 2, pp. 277–292, 2010. [Online]. Available: https://doi.org/10.1007/s10618-010-0190-x

[2] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowl. Inf. Syst.*, vol. 33, no. 1, pp. 1–33, 2011. [Online]. Available: https://doi.org/10.1007/s10115-011-0463-8

[3] F. Kamiran, A. Karim, and X. Zhang, "Decision theory for discrimination-aware classification," in *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, M. J. Zaki, A. Siebes, J. X. Yu, B. Goethals, G. I. Webb, and X. Wu, Eds. IEEE Computer Society, 2012, pp. 924–929. [Online]. Available: https://doi.org/10.1109/ICDM.2012.45

[4] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II*, ser. Lecture Notes in Computer Science, P. A. Flach, T. D. Bie, and N. Cristianini, Eds., vol. 7524. Springer, 2012, pp. 35–50. [Online]. Available: https://doi.org/10.1007/978-3-642-33486-3_3

[5] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 259–268.

[6] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *NIPS*, 2016, pp. 3315–3323.

[7] F. P. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 3992–4001. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/9a49a25d845a483fae4be7e341368e36-Abstract.html

[8] L. Zhang, Y. Wu, and X. Wu, "Achieving non-discrimination in data release," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 2017, pp. 1335–1344. [Online]. Available: https://doi.org/10.1145/3097983.3098167

[9] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4066–4076. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html

[10] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02-03, 2018*, J. Furman, G. E. Marchant, H. Price, and F. Rossi, Eds. ACM, 2018, pp. 335–340. [Online]. Available: https://doi.org/10.1145/3278721.3278779

[11] S. Corbett-Davies and S. Goel, "The measure and mismeasure of fairness: A critical review of fair machine learning," *CoRR*, vol. abs/1808.00023, 2018. [Online]. Available: http://arxiv.org/abs/1808.00023

[12] Z. C. Lipton, J. J. McAuley, and A. Chouldechova, "Does mitigating ml's impact disparity require treatment disparity?" in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 8136–8146. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/8e0384779e58ce2af40eb365b318cc32-Abstract.html

[13] M. J. Kearns, S. Neel, A. Roth, and Z. S. Wu, "Preventing fairness gerrymandering: Auditing and learning for subgroup fairness," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 2569–2577. [Online]. Available: http://proceedings.mlr.press/v80/kearns18a.html

[14] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. M. Wallach, "A reductions approach to fair classification," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 60–69. [Online]. Available: http://proceedings.mlr.press/v80/agarwal18a.html

[15] A. Agarwal, M. Dudík, and Z. S. Wu, "Fair regression: Quantitative definitions and reduction-based algorithms," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 120–129. [Online]. Available: http://proceedings.mlr.press/v97/agarwal19d.html

[16] J. E. Johndrow and K. Lum, "An algorithm for removing sensitive information: application to race-independent recidivism prediction," *The Annals of Applied Statistics*, vol. 13, no. 1, pp. 189–220, 2019.

[17] 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, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, danah boyd and J. H. Morgenstern, Eds. ACM, 2019, pp. 319–328. [Online]. Available: https://doi.org/10.1145/3287560.3287586

[18] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu, "Interventional fairness: Causal database repair for algorithmic fairness," in *Proceedings of the 2019 International Conference on Management of Data*, ser. SIGMOD '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 793—-810. [Online]. Available: https://doi.org/10.1145/3299869.3319901

[19] S. Mitchell, E. Potash, S. Barocas, A. D'Amour, and K. Lum, "Algorithmic fairness: Choices, assumptions, and definitions," *Annual Review of Statistics and Its Application*, vol. 8, no. 1, pp. 141–163, 2021. [Online]. Available: https://doi.org/10.1146/annurev-statistics-042720-125902

[20] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning under covariate shift," *J. Mach. Learn. Res.*, vol. 10, pp. 2137–2155, 2009. [Online]. Available: https://dl.acm.org/doi/10.5555/1577069.1755858

[21] A. Kumar, R. McCann, J. F. Naughton, and J. M. Patel, "Model selection management systems: The next frontier of advanced analytics," *SIGMOD Rec.*, vol. 44, no. 4, pp. 17–22, 2015. [Online]. Available: https://doi.org/10.1145/2935694.2935698

[22] Z. C. Lipton, Y. Wang, and A. J. Smola, "Detecting and correcting for label shift with black box predictors," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 3128–3136. [Online]. Available: http://proceedings.mlr.press/v80/lipton18a.html

[23] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data lifecycle challenges in production machine learning: A survey," *SIGMOD Rec.*, vol. 47, no. 2, pp. 17–28, 2018. [Online]. Available: https://doi.org/10.1145/3299887.3299891

[24] C. Dwork, N. Immorlica, A. T. Kalai, and M. Leiserson, "Decoupled classifiers for group-fair and efficient machine learning," in *Conference on fairness, accountability and transparency*. PMLR, 2018, pp. 119–133.

[25] S. Schelter, T. Rukat, and F. Bießmann, "Learning to validate the predictions of black box classifiers on unseen data," in *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, D. Maier, R. Pottinger, A. Doan, W. Tan, A. Alawini, and H. Q. Ngo, Eds. ACM, 2020, pp. 1289–1299. [Online]. Available: https://doi.org/10.1145/3318464.3380604

[26] A. Fariha, A. Tiwari, A. Radhakrishna, S. Gulwani, and A. Meliou, "Conformance constraint discovery: Measuring trust in data-driven systems," in *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*, G. Li, Z. Li, S. Idreos, and D. Srivastava, Eds. ACM, 2021, pp. 499–512. [Online]. Available: https://doi.org/10.1145/3448016.3452795

[27] J. Kappelhof, "Survey research and the quality of survey data among ethnic minorities," *Total survey error in practice*, pp. 235–252, 2017.

[28] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang, "Fairness without demographics in repeated loss minimization," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1929–1938.

[29] P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, and E. Chi, "Fairness without demographics through adversarially reweighted learning," *Advances in neural information processing systems*, vol. 33, pp. 728–740, 2020.

[30] 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*, 2021, pp. 2076–2088.

[31] H. Jiang and O. Nachum, "Identifying and correcting label bias in machine learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 702–712.

[32] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, "Fairness through awareness," in *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, 2012, pp. 214–226. [Online]. Available: https://doi.org/10.1145/2090236.2090255

[33] K. Yang and A. Meliou, "Non-invasive fairness in learning through the lens of data drift," *arXiv*, 2023. [Online]. Available: http://arxiv.org/abs/2303.17566

[34] G. Pleiss, M. Raghavan, F. Wu, J. M. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5680–5689. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/b8b9c74ac526fffbeb2d39ab038d1cd7-Abstract.html

[35] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, ser. Proceedings of Machine Learning Research, A. Singh and X. J. Zhu, Eds., vol. 54. PMLR, 2017, pp. 962–970. [Online]. Available: http://proceedings.mlr.press/v54/zafar17a.html

[36] P. S. Thomas, B. Castro da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill, "Preventing undesirable behavior of intelligent machines," *Science*, vol. 366, no. 6468, pp. 999–1004, 2019.

[37] N. Martinez, M. Bertran, and G. Sapiro, "Minimax pareto fairness: A multi objective perspective," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6755–6764.

[38] A. Krishnaswamy, Z. Jiang, K. Wang, Y. Cheng, and K. Munagala, "Fair for all: Best-effort fairness guarantees for classification," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 3259–3267.

[39] A. Chhabra, K. Masalkovaitė, and P. Mohapatra, "An overview of fairness in clustering," *IEEE Access*, vol. 9, pp. 130 698–130 720, 2021.

[40] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975. [Online]. Available: http://doi.acm.org/10.1145/361002.361007

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[42] L. Wasserman, *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[43] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[44] P. Müller and F. A. Quintana, "Nonparametric bayesian data analysis," *Statistical science*, vol. 19, no. 1, pp. 95–110, 2004.

[45] S. M. Omohundro, *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

[46] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," https://arxiv.org/abs/1810.01943, Oct. 2018.

[47] Kaggle, "Kaggle competition: Give me some credit," https://www.kaggle.com/competitions/GiveMeSomeCredit/overview, 2012.

[48] F. Ding, M. Hardt, J. Miller, and L. Schmidt, "Retiring adult: New datasets for fair machine learning," *Advances in neural information processing systems*, vol. 34, pp. 6478–6490, 2021.

[49] R. Kohavi and B. Becker, "UCI machine learning repository," 1994. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Adult

[50] S. Barocas and A. D. Selbst, "Big data's disparate impact," *Calif. L. Rev.*, vol. 104, p. 671, 2016.

[51] E. Krasanakis, E. Spyromitros-Xioufis, S. Papadopoulos, and Y. Kompatsiaris, "Adaptive sensitive reweighting to mitigate bias in fairness-aware classification," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 853–862.

[52] M. T. Islam, A. Fariha, A. Meliou, and B. Salimi, "Through the Data Management Lens: Experimental Analysis and Evaluation of Fair Classification," in *SIGMOD '22: International Conference on Management of Data*. ACM, 2022, pp. 232–246. [Online]. Available: https://doi.org/10.1145/3514221.3517841

[53] S. Barocas, M. Hardt, and A. Narayanan, "Fairness in machine learning," *Nips tutorial*, vol. 1, p. 2, 2017.

[54] S. Verma and J. Rubin, "Fairness definitions explained," in *Proceedings of the International Workshop on Software Fairness, FairWare@ICSE 2018, Gothenburg, Sweden, May 29, 2018*, Y. Brun, B. Johnson, and A. Meliou, Eds. ACM, 2018, pp. 1–7. [Online]. Available: https://doi.org/10.1145/3194770.3194776

[55] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth, "A comparative study of fairness-enhancing interventions in machine learning," in *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, danah boyd and J. H. Morgenstern, Eds. ACM, 2019, pp. 329–338. [Online]. Available: https://doi.org/10.1145/3287560.3287589

[56] A. Chouldechova and A. Roth, "A snapshot of the frontiers of fairness in machine learning," *Commun. ACM*, vol. 63, no. 5, pp. 82–89, 2020. [Online]. Available: https://doi.org/10.1145/3376898

[57] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, "Fairness in criminal justice risk assessments: The state of the art," *Sociological Methods & Research*, vol. 50, no. 1, pp. 3–44, 2021.

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

[59] A. Asudeh, Z. Jin, and H. Jagadish, "Assessing and remedying coverage for a given dataset," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 554–565.

[60] K. H. Tae and S. E. Whang, "Slice tuner: A selective data acquisition framework for accurate and fair machine learning models," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1771–1783.

[61] F. Nargesian, A. Asudeh, and H. Jagadish, "Tailoring data source distributions for fairness-aware data integration," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 2519–2532, 2021.

[62] C. Chai, J. Liu, N. Tang, G. Li, and Y. Luo, "Selective data acquisition in the wild for model charging," *Proceedings of the VLDB Endowment*, vol. 15, no. 7, pp. 1466–1478, 2022.

[63] P. Li and H. Liu, "Achieving fairness at no utility cost via data reweighing with influence," in *International Conference on Machine Learning*. PMLR, 2022, pp. 12 917–12 930.

[64] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. Mit Press, 2008.

[65] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=Hkg4TI9xl

[66] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, "Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance," *CoRR*, vol. abs/1812.02765, 2018. [Online]. Available: http://arxiv.org/abs/1812.02765

[67] H. Jiang, B. Kim, M. Y. Guan, and M. R. Gupta, "To trust or not to trust A classifier," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 5546–5557. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/7180cffd6a8e829dacfc2a31b3f72ece-Abstract.html

[68] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, B. Schölkopf, J. C. Platt, and T. Hofmann, Eds. MIT Press, 2006, pp. 601–608. [Online]. Available: https://proceedings.neurips.cc/paper/2006/hash/a2186aa7c086b46ad4e8bf81e2a3a19b-Abstract.html

[69] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, ser. JMLR Workshop and Conference Proceedings, vol. 28. JMLR.org, 2013, pp. 819–827. [Online]. Available: http://proceedings.mlr.press/v28/zhang13d.html

[70] P. Von Bünau, F. C. Meinecke, F. C. Király, and K.-R. Müller, "Finding stationary subspaces in multivariate time series," *Physical review letters*, vol. 103, no. 21, p. 214101, 2009.

[71] M. Sugiyama and M. Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

[72] P. Lahoti, K. P. Gummadi, and G. Weikum, "Detecting and mitigating test-time failure risks via model-agnostic uncertainty learning," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1174–1179.

[73] Z. Abedjan, L. Golab, and F. Naumann, "Profiling relational data: a survey," *VLDB J.*, vol. 24, no. 4, pp. 557–581, 2015. [Online]. Available: https://doi.org/10.1007/s00778-015-0389-y

[74] T. Papenbrock, J. Ehrlich, J. Marten, T. Neubert, J. Rudolph, M. Schönberg, J. Zwiener, and F. Naumann, "Functional dependency discovery: An experimental evaluation of seven algorithms," *Proc. VLDB Endow.*, vol. 8, no. 10, pp. 1082–1093, 2015. [Online]. Available: http://www.vldb.org/pvldb/vol8/p1082-papenbrock.pdf

[75] I. F. Ilyas, V. Markl, P. J. Haas, P. Brown, and A. Aboulnaga, "CORDS: automatic discovery of correlations and soft functional dependencies," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, G. Weikum, A. C. König, and S. Deßloch, Eds. ACM, 2004, pp. 647–658. [Online]. Available: https://doi.org/10.1145/1007568.1007641

[76] N. Koudas, A. Saha, D. Srivastava, and S. Venkatasubramanian, "Metric functional dependencies," in *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*, Y. E. Ioannidis, D. L. Lee, and R. T. Ng, Eds. IEEE Computer Society, 2009, pp. 1275–1278. [Online]. Available: https://doi.org/10.1109/ICDE.2009.219

[77] W. Fan, F. Geerts, L. V. S. Lakshmanan, and M. Xiong, "Discovering conditional functional dependencies," in *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*, Y. E. Ioannidis, D. L. Lee, and R. T. Ng, Eds. IEEE Computer Society, 2009, pp. 1231–1234. [Online]. Available: https://doi.org/10.1109/ICDE.2009.208

[78] L. Caruccio, V. Deufemia, and G. Polese, "On the discovery of relaxed functional dependencies," in *Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS 2016, Montreal, QC, Canada, July 11-13, 2016*, E. Desai, B. C. Desai, M. Toyama, and J. Bernardino, Eds. ACM, 2016, pp. 53–61. [Online]. Available: https://doi.org/10.1145/2938503.2938519

[79] S. Kruse and F. Naumann, "Efficient discovery of approximate dependencies," *Proc. VLDB Endow.*, vol. 11, no. 7, pp. 759–772, 2018. [Online]. Available: http://www.vldb.org/pvldb/vol11/p759-kruse.pdf

[80] A. A. Qahtan, N. Tang, M. Ouzzani, Y. Cao, and M. Stonebraker, "Pattern functional dependencies for data cleaning," *Proc. VLDB Endow.*, vol. 13, no. 5, pp. 684–697, 2020. [Online]. Available: http://www.vldb.org/pvldb/vol13/p684-qahtan.pdf

[81] Y. Zhang, Z. Guo, and T. Rekatsinas, "A statistical perspective on discovering functional dependencies in noisy data," in *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, D. Maier, R. Pottinger, A. Doan, W. Tan, A. Alawini, and H. Q. Ngo, Eds. ACM, 2020, pp. 861–876. [Online]. Available: https://doi.org/10.1145/3318464.3389749

[82] R. Karegar, P. Godfrey, L. Golab, M. Kargar, D. Srivastava, and J. Szlichta, "Efficient discovery of approximate order dependencies," in *Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021, Nicosia, Cyprus, March 23 - 26, 2021*, Y. Velegrakis, D. Zeinalipour-Yazti, P. K. Chrysanthis, and F. Guerra, Eds. OpenProceedings.org, 2021, pp. 427–432. [Online]. Available: https://doi.org/10.5441/002/edbt.2021.46

[83] X. Chu, I. F. Ilyas, and P. Papotti, "Discovering denial constraints," *Proc. VLDB Endow.*, vol. 6, no. 13, pp. 1498–1509, 2013. [Online]. Available: http://www.vldb.org/pvldb/vol6/p1498-papotti.pdf

[84] T. Bleifuß, S. Kruse, and F. Naumann, "Efficient denial constraint discovery with hydra," *Proc. VLDB Endow.*, vol. 11, no. 3, pp. 311–323, 2017. [Online]. Available: http://www.vldb.org/pvldb/vol11/p311-bleifub.pdf

[85] E. H. M. Pena, E. C. de Almeida, and F. Naumann, "Discovery of approximate (and exact) denial constraints," *Proc. VLDB Endow.*, vol. 13, no. 3, pp. 266–278, 2019. [Online]. Available: http://www.vldb.org/pvldb/vol13/p266-pena.pdf

[86] E. Livshits, A. Heidari, I. F. Ilyas, and B. Kimelfeld, "Approximate denial constraints," *Proc. VLDB Endow.*, vol. 13, no. 10, pp. 1682–1695, 2020. [Online]. Available: http://www.vldb.org/pvldb/vol13/p1682-livshits.pdf