This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Measuring Model-Induced Discrimination via Efficient Fairness Approximation

Yijun Bian and Yujie Luo

***Abstract*—Providing various machine learning (ML) applications in the real world, concerns about discrimination hidden in ML models are growing, particularly in high-stakes domains. Existing techniques for assessing the discrimination level of ML models include commonly used group and individual fairness measures. However, these two types of fairness measures are usually hard to be compatible, and even two different group fairness measures might be incompatible as well. To address this issue, we investigate and evaluate the discrimination level of classifiers from a manifold perspective and propose a fairness measure named "*harmonic fairness via manifolds* (*HFM*)" based on distances between sets. Yet the direct calculation of distances might be too expensive to afford, reducing its practical applicability. Therefore, we devise an approximation algorithm named "*approximation of distance between sets (ApproxDist)*" to facilitate accurate estimation of distances, and we further demonstrate its algorithmic effectiveness under certain reasonable assumptions. Empirical results indicate that the proposed fairness measure *HFM* reflects bias from both individual and group fairness aspects and that the proposed *ApproxDist* is effective and efficient.**

***Index Terms*—Approximation, distance between sets, fairness measure, machine learning (ML), metric space.**

NOMENCLATURE

x	Scalars, denoted by italic lowercase letters.
$\boldsymbol{x}$	Vectors, denoted by bold lowercase letters.
X	Matrices/sets, denoted by italic uppercase letters.
X	Random variables, denoted by serif uppercase letters.
$\mathbb{R}$	Real numbers.
$\mathbb{Z}$	Integers.
$\mathbb{Z}_+$	Positive integers.
$\mathbb{P}(\cdot)$	Probability measure of one random variable.
$\mathbb{E}(\cdot)$	Expectation of one random variable.
$\mathbb{V}(\cdot)$	Variance of one random variable.
$\mathcal{F}$	Hypothesis space.
$f(\cdot)$	Models in the space $\mathcal{F}$.
$i \in [n]$	$i \in \{1, 2, \ldots, n\}$ for brevity.

I. INTRODUCTION

Numerous real-world applications nowadays have been advanced to assist humans thanks to the flourishing development of machine learning (ML) techniques, such as healthcare, transportation, recruitment, and jurisdiction. For instance, many scholars have attempted to develop ML systems to alleviate resource constraints and reduce overall healthcare costs, such as the early diagnosis of Alzheimer's disease and automated detection of suspicious mammographic lesions for biopsy [1]. However, deploying ML systems in high-risk scenarios raises concerns regarding unresolved safety issues. These issues include erroneous predictions, lack of confidence in predictions, imperfect generalisability, black box decision-making, and insensitivity to impact and automation complacency [1]. Among them, an area to investigate for trustworthy AI is gradually gaining researchers' attention, for example, fairness in ML.

The reason why people worry about the fairness and reliability of ML systems is that discriminative models may perpetuate or even exacerbate inappropriate human prejudices, which harm not only model performance but also society. Taking the ML systems deployed for clinical use, for example, shifted distributions exist within the collected medical datasets, thus causing prediction failures, as a higher incidence of a disease may be reported in people seeking medical treatment in hospitals, whose characteristics are different from those of a dataset collected for population studies [2]. Besides, potential sources of dataset bias also exist, such as imbalanced proportions for different groups, specific measurement biases captured via imaging devices, systematic biases introduced by human annotators, and distorted research subject to dataset availability. Last but not least, inappropriate evaluation for ML models may miss the desired target, and a strong focus on benchmark performance would lead to diminishing returns, where the performance gains of increasingly great efforts are actually smaller and smaller [3]. It is worthwhile to choose suitable metrics in evaluation, as improvements in traditional metrics such as accuracy do not necessarily translate to practical clinical outcomes. Therefore, it is crucial in the responsible deployment of ML systems to evaluate model performance from both accuracy and fairness.

Unlike precise accuracy, however, the concept of "fairness" is interdisciplinary with broadly varying definitions across disciplines, such as law, politics, social sciences, quantitative fields (mathematics, computer science, statistics, and economics), and philosophy. Even in the field of computer science that we mainly discuss in this article, there exist so many fairness explanations or measures to choose from when applied to the deployment of ML systems, of which two typical categories are group- and individual-level fairness. While individual-level fairness follows the intuition that similar predicted outcomes are expected for similar *individuals* (in other words, similar treatment) given some specific

Received 17 August 2024; revised 18 September 2025, 2 February 2026, and 5 April 2026; accepted 17 June 2026. This research is funded by the European Union (MSCA, FairML, project no. 101106768). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of European Union or the Research Executive Agency. Neither European Union nor the granting authority can be held responsible for them. *(Yijun Bian and Yujie Luo contributed equally to this work.) (Corresponding author: Yijun Bian.)*

Yijun Bian is with the Department of Computer Science, University of Copenhagen, 2100 Copenhagen, Denmark (e-mail: yibi@di.ku.dk).

Yujie Luo is with the Department of Mathematics, National University of Singapore, Singapore 117543 (e-mail: lyj96@nus.edu.sg).

Digital Object Identifier 10.1109/TNNLS.2026.3706648

2162-237X © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

similarity metrics [4], [5], group-level fairness cares more about the *statistical* disparity among groups divided by certain sensitive/unprivileged/protected characteristics, typically including age, race, gender, disability, pregnancy, and sexual orientation [6], [7]. Typical group-level fairness measures include demographic parity (DP) (aka. statistical parity) [4], [8], [9], [10], equalized odds [11], and predictive quality parity (PQP) [12], and they, respectively, correspond to independence, separation, and sufficiency in scenarios for sensitive attributes with binary values [13]. However, it is hard to meet all three criteria (aka. independence, separation, and sufficiency) simultaneously, as imposing any two of them at the same time would overconstrain the space to the point where only degenerate solutions remain [13]. Moreover, satisfying group fairness such as statistical parity may violate individual fairness terribly [4], [5], [14]; the inherent tradeoff between fairness and accuracy also draws researchers' attention [15]. Therefore, it is necessary to deliberate on the incompatibility of these criteria when evaluating ML model performance.

To this end, we investigate the possibility of assessing the discrimination level of ML models from both individual and group fairness aspects. In this article, we propose a novel fairness measure from a manifold perspective, named "*harmonic fairness via manifolds (HFM)*," to fulfill this purpose. Intuitively, by viewing instances as points on data manifolds, *HFM* measures discrimination through the geometric discrepancy between manifolds induced by privileged and unprivileged groups. However, the calculation of *HFM* relies on a core distance between two sets, which might be pretty costly. To speed up the calculation and increase its practical applicability, we propose an approximation algorithm to quickly estimate the distance between sets, named as "*approximation of distance between sets (ApproxDist)*." We also further investigate its algorithmic properties under certain reasonable assumptions, in other words, how effective it is to achieve the approximation goal. Our contribution in this work is fourfold.

1) We propose a fairness measure *HFM* that could reflect the discrimination level of classifiers from both individual and group fairness aspects, which is built on a concept of distances between sets.
2) We propose an approximation algorithm named "*ApproxDist*" to speed up the estimation of distances between sets, in order to mitigate the disadvantage of its costly direct calculation.
3) We further investigate the algorithmic effectiveness of *ApproxDist* under certain assumptions and provide detailed explanations.
4) Comprehensive experiments are conducted to verify the effectiveness of the proposed *HFM* and *ApproxDist*.

II. Related Work

In this section, we first introduce the sources of biases and then summarize existing techniques to measure and enhance fairness in turn.

A. Sources of Biases

Before addressing fairness concerns, it is necessary to understand the origins of biases. Existing literature identifies two primary sources of unfairness: biases from the data and biases from the algorithm [16]. Biased data collected from various sources, like device measurements and historically biased human decisions, directly influence ML algorithms, perpetuating these biases. In addition, missing data, such as instances or values, can introduce disparities between the dataset and the target population, leading to biased outcomes. Even with clean data, learning algorithms might yield unfair results due to proxy attributes for sensitive features or tendentious algorithmic objectives. For example, optimizing aggregated prediction errors can advantage privileged groups over marginalized ones.

B. Mechanisms to Enhance Fairness

Numerous mechanisms have been proposed to enhance fairness and mitigate biases in ML models, typically categorized as preprocessing, in-processing, and postprocessing mechanisms [14], [17], [18]. Preprocessing mechanisms [19], [20], [21], [22] usually take place when the algorithm is allowed to modify the training data, manipulating features or labels before instances are fed into the algorithm, in order to align the distributions of unprivileged and privileged groups to minimize discrimination. For instance, recent scholarship utilizes counterfactual samples [23], [24] or counterfactual-like data augmentation [25] to promote fairness. In contrast, postprocessing mechanisms [11], [26], [27] are normally used when the learned model can only be treated as a black box without any ability to modify the training data or learning algorithm, by manipulating output scores or decoupling predictions for each group. While preprocessing and postprocessing mechanisms offer flexibility across classification tasks, they have drawbacks. Preprocessing mechanisms suffer from high uncertainty in final accuracy, while postprocessing mechanisms often yield inferior results due to the late-stage application in the learning process. In addition, they may not fully achieve individual fairness. In-processing mechanisms [28], [29], [30], [31], [32] incorporate fairness constraints during training by using penalty/regularization terms, while some of them [33], [34] adjust fairness constraints in minimax or multiobjective optimization settings. They usually impose explicit tradeoffs between fairness and accuracy in the objective function, yet the designated ML algorithm itself is tightly coupled with them as well. It is worth noting that these approaches primarily focus on mitigation rather than measurement, and their effectiveness is relevant to the choice of the fairness measure that is applied. It is hard to say which one outperforms the others in all cases; the results depend on the fairness measures, datasets, and even the handling of training-test splits [35].

C. Types of Fairness Measures

Various fairness measures have been proposed to facilitate the design of fair ML models, generally divided into distributive and procedural fairness measures. Procedural fairness concerns decision-making processes and includes feature-a priori fairness, feature-accuracy fairness, and feature-disparity fairness [36]. However, these measures are dependent on features and a set of users who view the corresponding features as fair ones and thus still risk hidden discrimination in the data. Moreover, these user judgments may change over time, resulting in unstable outcomes and unpredictable computational

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

demands as systems are recalibrated repeatedly. In addition, an FAE-based (feature attribution explanation) metric has recently been proposed to assess group procedural fairness, which, however, depends on the specific FAE techniques employed [37]; BENN [38] is also proposed for bias estimation, but requires another unsupervised neural network to be trained, adding some computational burdens.

Distributive fairness refers to decision-making outcomes (predictions) and includes unconscious/unawareness fairness, group fairness, individual fairness, and counterfactual fairness [4], [39], [40], [41], [42], [43]. As the simplest, unawareness fairness avoids explicitly using sensitive attributes but does not address biases stemming from associations between protected and unprotected attributes. Group fairness focuses on statistical equality among groups defined by sensitive attributes, including but not limited to DP, equalized odds, equality of opportunity (EO)/equal opportunity, PQP, and disparate impact [9]. In contrast, individual fairness is based on the principle that "similar individuals should be evaluated or treated similarly," where similarity is measured by some distance between individuals, while the specified distance also matters a lot. Besides, counterfactual fairness aims to explain the sources of discrimination and qualitative equity through causal inference tools. Other distributive fairness definitions include fairness through unawareness [4], [44], calibration [45], multicalibration [46], [47], multiaccuracy [48], differentially fair and intersectionality [49], and strategic minimax fairness [50]. Yet none of these are quantitative measures, except group benefit ratio and worst case min–max ratio [51].

Moreover, the group fairness measures introduced in different types of fairness may hardly be compatible with each other, for example, the occurrence between equalized odds and DP, or that between equalized calibration and equalized odds [11], [39], [52]. Individual and group fairness, such as DP, are also incompatible except in trivial degenerate cases. Meeting all three fairness criteria (i.e., independence, separation, and sufficiency) simultaneously is demonstrated to be challenging, often only achievable in degenerate scenarios [13]. Recently, a fairness measure named "discriminative risk (DR)" has been proposed to evaluate the bias level of classifiers from both individual- and group-fairness aspects [53], and has been incorporated into fairness-oriented preprocessing frameworks such as FairSHAP [25]. Besides, achieving higher fairness can often come at the cost of compromised accuracy [54]. Although some unique scenarios [55], [56] have recently been proposed where fairness and accuracy can be concurrently improved, constrained optimization with additional fairness constraints likely results in reduced accuracy compared to accuracy-focused optimization.

III. Methodology

In this section, we formally study the measurement of fairness from a manifold perspective. A list of some of the standard notations we use is summarized in Nomenclature.

We also use $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ to denote a dataset where the instances are independent and identically distributed (i.i.d.), drawn from a feature/input-label/output space $\mathcal{X} \times \mathcal{Y}$ based on an unknown distribution. The feature space $\mathcal{X}$ is arbitrary, and the label space $\mathcal{Y} = \{1, 2, \ldots, n_c\} (n_c \geqslant 2)$ is finite, which could be binary or multiclass classification depending on the number of labels (aka. the value of n_c). Presuming that the considered dataset S is composed of the instances including sensitive attributes, also denoted by $\{(\check{\boldsymbol{x}}_i, \boldsymbol{a}_i, y_i)\}_{i=1}^n$, the features of one instance including protected or sensitive attributes is represented as $\boldsymbol{x} \triangleq (\check{\boldsymbol{x}}, \boldsymbol{a})$, where for clarity $\check{\boldsymbol{x}} = [x_1, \ldots, x_{n_x}]^\mathsf{T}$ is nonsensitive attributes (possibly involving proxy attributes though), $n_x \in \mathbb{Z}_+$ is the number of nonsensitive attributes, $\boldsymbol{a} = [a_1, \ldots, a_{n_a}]^\mathsf{T}$ is the sensitive attributes, $n_a \in \mathbb{Z}_+ (n_a \geqslant 1)$ is the number of sensitive attributes allowing multiple attributes, and $a_i \in \mathbb{Z}_+ (1 \leqslant i \leqslant n_a)$ allows both binary and multiple values. A function $f \in \mathcal{F} : \mathcal{X} \mapsto \mathcal{Y}$ represents a hypothesis in a space of hypotheses $\mathcal{F}$, of which the prediction for one instance $\boldsymbol{x}$ is denoted by $f(\boldsymbol{x})$ or $\hat{y}$ for brevity.

A. Model Fairness Assessment From a Manifold Perspective

In this article, we mainly discuss the fairness measure in scenarios for sensitive attributes with binary values, that is, $a_i \in \mathcal{A} = \{0, 1\}$ and $a_i = 1$ represents the majority or privileged group for $i \in [n_a]$. Note that the observations could be extended to sensitive attributes with multiple values as well. In the case of instances with only one sensitive attribute, that is, $n_a = 1$ and $\boldsymbol{x} = (\check{\boldsymbol{x}}, a_1)$, the original dataset S can be divided into a few disjoint sets according to the value of the sensitive attribute a_1, that is, $S_j = \{(\boldsymbol{x}, y) \in S \mid a_1 = j\} \; \forall j \in \mathcal{A}$. Because $\mathcal{A} = \{0, 1\}$ is considered in this article, we could get S_0 and S_1, while the division also works for sensitive attributes with multiple values, where $\mathcal{A} = \{0, 1, 2, \ldots\}$ is a finite set. In other words, as $a_i = 1$ represents the majority or privileged group, two subsets S_1 and S_0 denote the privileged group and the unprivileged group(s), respectively.

Inspired by the principle of individual fairness—similar treatment for similar individuals, *if we view the instances as data points on manifolds, the manifold representing members from the unprivileged group(s) is supposed to be as close as possible to that representing members from the privileged group* (**motivation**). A "manifold" refers to a topological space that locally resembles Euclidean space in mathematics. Then, given a specific distance metric on instances $\mathbf{d}(\cdot, \cdot)$,[1] we can define the distance between sets (that is, S_0 and S_1 here) by

$$\mathbf{D}(S_0, S_1) \triangleq \max \left\{ \max_{(\boldsymbol{x}, y) \in S_0} \overbrace{\min_{(\boldsymbol{x}', y') \in S_1} \mathbf{d}\left((\check{\boldsymbol{x}}, y), (\check{\boldsymbol{x}}', y')\right)}^{\text{for } (\boldsymbol{x}, y) \text{ in } S_0, \text{ to find the nearest point in } S_1}, \; \max_{(\boldsymbol{x}', y') \in S_1} \underbrace{\min_{(\boldsymbol{x}, y) \in S_0} \mathbf{d}\left((\check{\boldsymbol{x}}, y), (\check{\boldsymbol{x}}', y')\right)}_{\text{for } (\boldsymbol{x}', y') \text{ in } S_1, \text{ to find the nearest point in } S_0} \right\}, \quad (1)$$

and view it as an approximation of the distance between the manifold of unprivileged groups and that of the privileged

[1] Here we use the standard Euclidean metric. In fact, any two metrics $\mathbf{d}_1, \mathbf{d}_2$ derived from norms on the Euclidean space $\mathbb{R}^d$ are equivalent in the sense that there are positive constants c_1, c_2 such that $c_1 \mathbf{d}_1(x, y) \leqslant \mathbf{d}_2(x, y) \leqslant c_2 \mathbf{d}_1(x, y)$ for all $x, y \in \mathbb{R}^d$.

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

group. Notice that the distance defined above satisfies the following basic properties.

1) For any two data sets $S_0, S_1 \in \mathcal{X} \times \mathcal{Y}, \mathbf{D}(S_0, S_1) = 0$ if and only if S_0 equals S_1.
2) For any sets S_0, S_1, and S_2, we have the triangle inequality $\mathbf{D}(S_0, S_2) \leqslant \mathbf{D}(S_0, S_1) + \mathbf{D}(S_1, S_2)$.

Analogously, for a trained classifier $f(\cdot)$, we can calculate

$$\begin{aligned}&\mathbf{D}_f(S_0, S_1)\\ &= \max\left\{\max_{(\boldsymbol{x},y)\in S_0}\min_{(\boldsymbol{x}',y')\in S_1}\mathbf{d}\left((\check{\boldsymbol{x}},\hat{y}),(\check{\boldsymbol{x}}',\hat{y}')\right),\right.\\ &\qquad\left.\max_{(\boldsymbol{x}',y')\in S_1}\min_{(\boldsymbol{x},y)\in S_0}\mathbf{d}\left((\check{\boldsymbol{x}},\hat{y}),(\check{\boldsymbol{x}}',\hat{y}')\right)\right\}. \qquad (2)\end{aligned}$$

We remark that $\mathbf{D}(S_0, S_1)$ reflects the biases from the data, and $\mathbf{D}_f(S_0, S_1)$ reflects the biases from the algorithm. Then, the following value could be used to reflect the fairness degree of this classifier, that is,

$$\mathbf{df}(f) = \frac{\mathbf{D}_f(S_0, S_1)}{\mathbf{D}(S_0, S_1)} - 1. \qquad (3)$$

We name the fairness degree of one classifier by (3) as "*harmonic fairness via manifolds (HFM).*" *HFM* is designed to consider both individual- and group-fairness aspects simultaneously, and the reason why it can achieve this goal is that: 1) *HFM* is built upon the concept of distance between sets, which includes distances between individuals as a basis; 2) the similarity between instances is indicated by the distance between individuals, while the disparity between two subgroups divided by sensitive attributes is indicated by the distance between sets; and 3) by taking them together, *HFM* is able to reflect fairness from both individual- and group-level aspects.

We further give a few remarks on $\mathbf{df}(f)$ defined in (3).

1) Notice a degenerate case that $\mathbf{d}((\check{\boldsymbol{x}}, y_1), (\check{\boldsymbol{x}}, y_2)) = |y_1 - y_2|$. We set $X_i = \{\check{\boldsymbol{x}} \mid (\boldsymbol{x}, y) \in S_i\}$ and $Y_i = \{y \mid (\boldsymbol{x}, y) \in S_i\}$ for $i \in \{0, 1\}$. If $X \triangleq X_0 = X_1$, then $\mathbf{D}(S_0, S_1) \leqslant \max_{\check{\boldsymbol{x}}\in X}\{|y - y'|\}$. Similarly we have $\mathbf{D}_f(S_0, S_1) \leqslant \max_{\check{\boldsymbol{x}}\in X}\{|f(\boldsymbol{x}) - f(\boldsymbol{x}')|\}$. In particular, if $\mathbf{D}(S_0, S_1) > 0$ (resp. $\mathbf{D}_f(S_0, S_1) > 0$), then there exist two data points with exactly the same nonsensitive attributes, while their sensitive attributes and corresponding labels (predictions) are different.
2) $\mathbf{df}(f)$ measures the bias of the classifier concerning the biases from the data. In particular, if $\mathbf{df}(f) = 0$, then $\mathbf{D}_f(S_0, S_1) = \mathbf{D}(S_0, S_1)$, indicating that the classifier does not introduce additional biases beyond those inherent in the data. If $\mathbf{df}(f) > 0$, then $\mathbf{D}_f(S_0, S_1) > \mathbf{D}(S_0, S_1)$, suggesting that the algorithm introduces biases beyond those initially present in the dataset. As $\mathbf{df}(f)$ grows, the relative bias of the classifier with respect to the biases from the dataset increases. Finally, if $\mathbf{df}(f) < 0$, then the classifier somehow reduces the bias arising from the original data.
3) Assume that $\mathbf{D}(S_0, S_1) = 0$, which means that the original dataset is perfectly unbiased. If $\mathbf{D}_f(S_0, S_1) = 0$ as well, then we write $\mathbf{df}(f) = 0/0 - 1 = 0$, indicating that the classifier is also unbiased. On the other hand, if $\mathbf{D}_f(S_0, S_1) > 0$, then $\mathbf{df}(f) = +\infty$, indicating that the classifier introduces biases by providing different outputs for identical data with differing sensitive attributes.
4) As we can see in the definition, $\mathbf{df}(f)$ is achieved by $\mathbf{d}((\check{\boldsymbol{x}}, y), (\check{\boldsymbol{x}}', y'))$ for some $(\boldsymbol{x}, y) \in S_0$ and $(\boldsymbol{x}', y') \in S_1$, and this implies that $\mathbf{df}(f)$ is highly sensitive to the instances used in its computation.

HFM is interpretable in relation to data bias, because it provides a signed, relative notion of model-induced bias with a direct interpretation against the original data manifold discrepancy. Note that, unlike common fairness measures, *HFM* rather isolates model-induced bias relative to the bias already present in the data and assesses the algorithmic behavior. This distinction is important in practice, as it enables practitioners to assess whether a model exacerbates existing disparities beyond what is inherited from the data. The practitioners are therefore advised to accompany other fairness measures, because a model cannot be claimed to be fair only due to $\mathbf{df}(f) = 0$. Also note that n_0 and n_1 are the number of instances in S_0 and S_1, respectively. Then, the computational complexity of directly calculating (1) or (2) would be $\mathcal{O}(n_0 n_1)$, that is, $\mathcal{O}(n_0(n - n_0))$, which is expensive and less practical to be applied on large datasets.

B. Quick Approximation of Distances Between Sets for Euclidean Spaces

Given the high computational complexity of directly calculating (1) and (2), that is, $\mathcal{O}(n^2)$, it would be necessary to speed up the calculation if we intend to use them to measure the discriminative level of classifiers in practice.

Notice that the core operation in (1) and (2) is evaluating the distance between data points inside $\mathcal{X} \times \mathcal{Y}$. To reduce the number of distance evaluation operations involved in (1) and (2), we observe that the distance between similar data points tends to be closer than others after projecting them onto a general 1-D linear subspace. In fact, let $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a projection, then,

$$\left|g(\boldsymbol{x}, y) - g\left(\boldsymbol{x}', y'\right)\right| \leqslant \mathbf{d}\left((\check{\boldsymbol{x}}, y), (\check{\boldsymbol{x}}', y')\right). \qquad (4)$$

To be concrete, one possible candidate for the projection g could be

$$g(\boldsymbol{x}, y; \boldsymbol{w}) = g(\check{\boldsymbol{x}}, \boldsymbol{a}, y; \boldsymbol{w}) = \left[y, x_1, \ldots, x_{n_x}\right]^{\mathsf{T}} \boldsymbol{w}, \qquad (5a)$$

$$g(\boldsymbol{x}, f; \boldsymbol{w}) = g(\check{\boldsymbol{x}}, \boldsymbol{a}, f(\boldsymbol{x}); \boldsymbol{w}) = \left[\hat{y}, x_1, \ldots, x_{n_x}\right]^{\mathsf{T}} \boldsymbol{w}, \qquad (5b)$$

where $\boldsymbol{w} = [w_0, w_1, \ldots, w_{n_x}]^{\mathsf{T}}$ is a random vector that meets $w_i \in [-1, 1]$ for any $i \in \{0, 1, \ldots, n_x\}$ and $\sum_{i=0}^{n_x} |w_i| = 1$.

Now, we choose a random projection $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, then we sort all the projected data points on $\mathbb{R}$. According to (4), *it is likely that for the instance* $(\boldsymbol{x}, y)$ *in* S_0*, the desired instance* $\operatorname{argmin}_{(\boldsymbol{x}',y')\in S_1}\mathbf{d}((\check{\boldsymbol{x}}, y), (\check{\boldsymbol{x}}', y'))$ *would be someone near it after the projection, and vice versa.* Thus, by using the projections in (5a), we could accelerate this process in (1) by checking several adjacent instances rather than traversing the whole dataset. Analogously, we could accelerate the process in (2) through the projection in (5b) as well. For simplification, we record $\{y_i\}_{i=1}^n$ and $\{\hat{y}_i\}_{i=1}^n$ as one denotation, that is to say, $\ddot{y}$ could be

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 1 Approximation of Distance Between Sets, aka. `ApproxDist`$(\{(\check{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n; m_1, m_2)$

Input: Dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n = \{(\check{\boldsymbol{x}}_i, \boldsymbol{a}_i, y_i)\}_{i=1}^n$, prediction of S by the classifier $f(\cdot)$ that has been trained, that is, $\{\hat{y}_i\}_{i=1}^n$, and two hyperparameters m_1 and m_2 as the designated numbers for repetition and comparison respectively

Output: Approximation of distance $\mathbf{D}_{\cdot}(S_0, S_1)$ in (6)

1: **for** j from 1 to m_1 **do**
2: Take a random vector $\boldsymbol{w}$ from the space $\mathcal{W} = \left\{\boldsymbol{w} = [w_0, w_1, \ldots, w_{n_x}]^\top \mid \sum_{i=0}^{n_x} |w_i| = 1\right\} \subseteq [-1, 1]^{1+n_x}$
3: $d_{\max}^j = AcceleDist(\{(\check{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n, \boldsymbol{w}; m_2)$
4: **end for**
5: **return** $\min\{d_{\max}^j \mid j \in [m_1]\}$

the true label y or the prediction by the classifier $f(\cdot)$ or $\hat{y}$, then we can rewrite (1), (2), and (5) as

$$\mathbf{D}_{\cdot}(S_0, S_1) = \max\left\{\max_{(\boldsymbol{x},y)\in S_0}\min_{(\boldsymbol{x}',y')\in S_1}\mathbf{d}\left((\check{\boldsymbol{x}}, \ddot{y}), (\check{\boldsymbol{x}}', \ddot{y}')\right), \max_{(\boldsymbol{x}',y')\in S_1}\min_{(\boldsymbol{x},y)\in S_0}\mathbf{d}\left((\check{\boldsymbol{x}}, \ddot{y}), (\check{\boldsymbol{x}}', \ddot{y}')\right)\right\}, \quad (6)$$

$$g(\boldsymbol{x}, \ddot{y}; \boldsymbol{w}) = g(\check{\boldsymbol{x}}, \boldsymbol{a}, \ddot{y}; \boldsymbol{w}) = \left[\ddot{y}, x_1, \ldots, x_{n_x}\right]^\top \boldsymbol{w}, \quad (7)$$

respectively. Note that $\mathbf{D}_{\cdot}(\cdot, \cdot)$ is symmetrical.

Then we could propose an approximation algorithm to estimate the distance between sets, named as "*approximation of distance between sets (ApproxDist)*," shown in Algorithm 1. Note that there exists a sub-route within *ApproxDist* to obtain an approximated distance between sets, which is named as "*acceleration sub-procedure in approximation (AcceleDist)*" and shown in Algorithm 2. The sub-route will be repeatedly executed several times with independently sampled random projection vectors, and this design is intended to reduce the chance of missing true nearest neighbors under an unfavorable projection direction. As the time complexity of sorting in line 2 of Algorithm 2 could reach $\mathcal{O}(n \log n)$, we could get the computational complexity of Algorithm 2 as follows: 1) the complexity of line 1 is $\mathcal{O}(n)$; and 2) the complexity from line 4 to line 10 is $\mathcal{O}(2m_2 + 1)$. Thus, the overall time complexity of Algorithm 2 would be $\mathcal{O}(n(1 + \log n + 2m_2))$ and that of Algorithm 1 be $\mathcal{O}(m_1 n(\log n + m_2))$. As both m_1 and m_2 are the designated constants, the time complexity of computing the distance is reduced to $\mathcal{O}(n \log n)$, which is more desirable than the $\mathcal{O}(n_0 n_1)$ complexity for the direct computation in Section III-A, which is upper bounded by $\mathcal{O}(n^2)$.

It is worth noting that in line 5 of Algorithm 1, we use the minimum instead of their average value. The reason is that in each projection, the exact distance for one instance would not be larger than the calculated distance for it via *AcceleDist*; and the same observation holds for all of the projections in *ApproxDist*. Thus, the calculated distance via *ApproxDist* is always no less than the exact distance, and the minimal operator should be taken finally after multiple projections. It also needs to be restated that neither *ApproxDist* nor *AcceleDist* calculate $\mathbf{D}_{\cdot}(S_0, S_1)$ in the same way as its definition in (6), and they are supposed to provide its approximated value. The idea behind them is that: 1) the distance between similar data points tends to be closer than others after them being projected onto a general 1-D linear subspace, with theoretical foundations presented in Section III-C; and 2) thus, by using the projections, we can accelerate the process in (6) by checking several adjacent instances rather than traversing the whole dataset.

Algorithm 2 Acceleration Sub-Procedure in Approximation, aka. `AcceleDist`$(\{(\check{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^n, \{\ddot{y}_i\}_{i=1}^n, \boldsymbol{w}; m_2)$

Input: Data points $\{(\check{\boldsymbol{x}}_i, \boldsymbol{a}_i)\}_{i=1}^n$, its corresponding value $\{\ddot{y}_i\}_{i=1}^n$, where $\ddot{y}_i$ could be its true label y_i or prediction $\hat{y}_i$ by the classifier $f(\cdot)$, a random vector $\boldsymbol{w}$ for projection, and a hyperparameter m_2 as the designated number for comparison

Output: Approximation of distance $\mathbf{D}_{\cdot}(S_0, S_1)$ in (6)

1: Project data points onto a one-dimensional space based on (7), in order to obtain $\{g(\boldsymbol{x}_i, \ddot{y}_i; \boldsymbol{w})\}_{i=1}^n$
2: Sort original data points based on $\{g(\boldsymbol{x}_i, \ddot{y}_i; \boldsymbol{w})\}_{i=1}^n$ as their corresponding values, in ascending order
3: **for** i from 1 to n **do**
4: Set the anchor data point $(\boldsymbol{x}_i, \ddot{y}_i)$ in this round
5: `// If` $\boldsymbol{a}_i = j \in \{0, 1\}$`, in order to approximate` $\min_{(\boldsymbol{x}',y')\in S\setminus S_j}\mathbf{d}((\check{\boldsymbol{x}}_i, \ddot{y}_i), (\check{\boldsymbol{x}}', \ddot{y}'))$
6: Compute the distances $\mathbf{d}((\check{\boldsymbol{x}}_i, \ddot{y}_i), \cdot)$ for at most m_2 nearby data points that meets $\boldsymbol{a} \neq \boldsymbol{a}_i$ and $g(\boldsymbol{x}, \ddot{y}; \boldsymbol{w}) \leqslant g(\boldsymbol{x}_i, \ddot{y}_i; \boldsymbol{w})$
7: Find the minimum among them, recorded as $d_{\min}^s$
8: Compute the distances $\mathbf{d}((\check{\boldsymbol{x}}_i, \ddot{y}_i), \cdot)$ for at most m_2 nearby data points that meets $\boldsymbol{a} \neq \boldsymbol{a}_i$ and $g(\boldsymbol{x}, \ddot{y}; \boldsymbol{w}) \geqslant g(\boldsymbol{x}_i, \ddot{y}_i; \boldsymbol{w})$
9: Find the minimum among them, recorded as $d_{\min}^r$
10: $d_{\min}^{(i)} = \min\{d_{\min}^s, d_{\min}^r\}$
11: **end for**
12: **return** $\max\{d_{\min}^{(i)} \mid i \in [n]\}$

C. Algorithmic Effectiveness Analysis of ApproxDist

As *ApproxDist* in Algorithm 1 is devised to facilitate the approximation of direct calculation of the distance between sets, in this section, we detail more about its algorithmic effectiveness under some conditions.

Before delving into the main result (see Proposition 1), we first introduce an important lemma that confirms the observation that "the distance between similar data points tends to be closer than others after projecting them onto a general 1-D linear subspace." Note that projecting to a certain fixed line L in the Euclidean space is like choosing a fixed unit vector $\boldsymbol{w}$ (in the direction of L) and consider the inner product $\langle \boldsymbol{w}, \boldsymbol{v} \rangle$ for each $\boldsymbol{v}$, the number $\langle \boldsymbol{w}, \boldsymbol{v} \rangle$ means the length of the projected vector of $\boldsymbol{v}$ on L, for simplicity we denote it as $\mathrm{Proj}_L(\boldsymbol{v})$. We notice that $\langle \boldsymbol{w}, \boldsymbol{v} \rangle$ has a sign, which stands for the direction of the projected vector on the line L. The absolute value of $\langle \boldsymbol{w}, \boldsymbol{v} \rangle$ means the real distance.

Lemma 1: Let $\boldsymbol{v}_1$ (resp. $\boldsymbol{v}_2$) be a vector in the n-dimensional Euclidean space $\mathbb{R}^n$ with length r_1 (resp. r_2) such that $r_1 \leqslant r_2$. Let $\boldsymbol{w} \in \mathbb{R}^n$ be a unit vector. We define $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2)$ as the

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

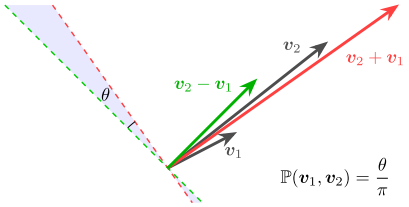


Fig. 1. Illustration of Lemma 1. The dashed lines represent the hyperplanes that are perpendicular to $(\boldsymbol{v}_1 - \boldsymbol{v}_2)$ and $(\boldsymbol{v}_1 + \boldsymbol{v}_2)$, respectively, and θ denotes the acute angle between the hyperplanes.

probability that $|\langle \boldsymbol{w}, \boldsymbol{v}_1\rangle| \geqslant |\langle \boldsymbol{w}, \boldsymbol{v}_2\rangle|$. Then,

$$\frac{\sin\ \phi}{\pi} \cdot \frac{r_1}{r_2} \leqslant \mathbb{P}\left(\boldsymbol{v}_1, \boldsymbol{v}_2\right) \leqslant \left(1 + \frac{r_1^2}{r_2^2}\right)^{-1/2} \cdot \frac{r_1}{r_2} \tag{8}$$

where ϕ represents the angle between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

Proof: Notice that $|\langle \boldsymbol{w}, \boldsymbol{v}_1\rangle| \geqslant |\langle \boldsymbol{w}, \boldsymbol{v}_2\rangle|$ is equivalent to

$$\langle \boldsymbol{v}_2 - \boldsymbol{v}_1, \boldsymbol{w}\rangle\langle \boldsymbol{v}_1 + \boldsymbol{v}_2, \boldsymbol{w}\rangle \leqslant 0. \tag{9}$$

If $\boldsymbol{w}$ satisfies (9), then it lies between two hyperplanes that are perpendicular to $\boldsymbol{v}_1 - \boldsymbol{v}_2$ and $\boldsymbol{v}_1 + \boldsymbol{v}_2$, respectively. Denote by θ the angle between these two hyperplanes (which is equal to the acute angle between $\boldsymbol{v}_2 - \boldsymbol{v}_1$ and $\boldsymbol{v}_1 + \boldsymbol{v}_2$, as shown in Fig. 1), and $0 \leqslant \theta \leqslant \pi/2$, then $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \theta/\pi$.[2] Moreover,

$$\sin^2\ \theta = 1 - \cos^2\ \theta = \frac{4\left\|\boldsymbol{v}_1\right\|^2 \left\|\boldsymbol{v}_2\right\|^2 - 4\left\langle \boldsymbol{v}_1, \boldsymbol{v}_2\right\rangle^2}{\left(\left\|\boldsymbol{v}_1\right\|^2 + \left\|\boldsymbol{v}_2\right\|^2\right)^2 - 4\left\langle \boldsymbol{v}_1, \boldsymbol{v}_2\right\rangle^2}. \tag{10}$$

Here $\|\boldsymbol{v}_i\|^2 = \langle \boldsymbol{v}_i, \boldsymbol{v}_i\rangle = r_i^2$ $(i = 1, 2)$ is the square length of the vector. Recall that $\langle \boldsymbol{v}_1, \boldsymbol{v}_2\rangle = \|\boldsymbol{v}_1\|\|\boldsymbol{v}_2\| \cos\ \phi$. By (10), we have

$$\frac{\left\|\boldsymbol{v}_1\right\|^2}{\left\|\boldsymbol{v}_2\right\|^2} \sin^2\ \phi \leqslant \sin^2\ \theta \leqslant \frac{4\left\|\boldsymbol{v}_1\right\|^2 \left\|\boldsymbol{v}_2\right\|^2}{\left(\left\|\boldsymbol{v}_1\right\|^2 + \left\|\boldsymbol{v}_2\right\|^2\right)^2}. \tag{11}$$

Combining (11) with the fact that $2\theta/\pi \leqslant \sin\ \theta \leqslant \theta$, we conclude that the probability $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \theta/\pi$ satisfies the desired inequalities. Note that by (8), when the ratio r_1/r_2 goes to zero, the probability $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2)$ goes to zero. Also, when $r_1 = r_2$ and $\boldsymbol{v}_1 \neq \pm\boldsymbol{v}_2$, it is easy to calculate that $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v}_2) = 1/2$, while the cases $\boldsymbol{v}_1 = \pm\boldsymbol{v}_2$ are degenerate. □

To understand Lemma 1, suppose that we have two vectors $\boldsymbol{v}_i$ with length r_i. We can choose a random line L_{random} of the Euclidean space, and we consider the projection to L_{random}. We know that for some L_{random}, it may happen that after projection, the length of $\boldsymbol{v}_1$ is larger than the length of $\boldsymbol{v}_2$, and for the rest of the L_{random}, the opposite relation holds. Now we suppose that $r_1 < r_2$ (actually in application, we may assume that $r_1 \ll r_2$ in most cases), and we randomly choose a line L (to do the projection), what is the probability that, after projecting onto L, we still have length $\text{Proj}_L(\boldsymbol{v}_1) < \text{Proj}_L(\boldsymbol{v}_2)$? *The lemma says that the probability of reversing the original size relationship of the lengths of vectors is at most of order r_1/r_2, and hence the probability of preserving the original size relationship is at least $1 - \mathcal{O}(r_1/r_2)$.* In particular, if r_1/r_2 goes to zero (the length of $\boldsymbol{v}_2$ is much larger than the length of $\boldsymbol{v}_1$), then the probability that $\text{Proj}_L(\boldsymbol{v}_1) < \text{Proj}_L(\boldsymbol{v}_2)$ goes to one. Now we take three data points $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$, and we may consider the distance vector $\boldsymbol{v}_1 = \boldsymbol{p}_2 - \boldsymbol{p}_1$ and $\boldsymbol{v}_2 = \boldsymbol{p}_3 - \boldsymbol{p}_1$, the length of these vectors corresponds to the distance of corresponding data points. We may apply the previous lemma to $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ to estimate the probability that after taking a projection to a random line, the projection keeps the size relationship of the distances between the given points (if the distance between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ is less than the distance of $\boldsymbol{p}_1$ and $\boldsymbol{p}_3$, then with the probability estimated above and after projecting to the line, the three projected points $\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3$ satisfy that the distance of $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ is less than the distance of $\boldsymbol{q}_1$ and $\boldsymbol{q}_3$). This explains why "the distance between similar data points tends to be closer than others after projecting them onto a general 1-D linear subspace" in our observation. Furthermore, computing the distance between two sets of points usually involves a lot of computation of distances of points and comparing the size relationship of lengths. To avoid the huge computation, we can project the data points to random lines and sort the projected points on this line. As the previous lemma suggests that, for a fixed comparison of two distances from the same anchor point, a random projection is unlikely to reverse the order when the shorter distance is much smaller than the longer one, and since such distance comparisons are likely to be preserved under a random projection in the above sense, we mainly need to evaluate the distances of data points that are nearby after projection, which will save a huge amount of computation.

Proposition 1: Let $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ be a $(k+1)$-D (instances have $k+1$ features) dataset, evenly distributed dataset with a size of n that is a random draw of the training instances. For any two subsets of S with distance d [ref. (6)], suppose further that the scaled density

$$\sup_r \left\{\inf_{\mathbf{B}\ \text{ball of radius r}} \frac{1}{\text{Vol}\left(\mathbf{B}\right)} \#\left(\mathbf{B} \cap S\right)\right\} = \frac{\mu}{\text{Vol}\left(\mathbf{B}\left(d\right)\right)} \tag{12}$$

for some positive real number μ (here # denotes the number of points of a finite set and $\mathbf{B}(d)$ denotes a ball of radius d). Then, with probability at least

$$1 - \left(\frac{\pi\mu}{m_2 \text{Vol}\left(\mathbf{B}\left(1\right)\right)} \left(\left(1 + \frac{n}{\mu}\right)^{\frac{1}{k+1}} - \alpha\right)\right)^{m_1}, \tag{13}$$

ApproxDist could reach an approximate solution that is at most α times the distance between these two subsets.

Proof: Let S_0 and S_1 be two sub-datasets of S. We fix the instance $\boldsymbol{v}_0 \in S_0$ such that $d \triangleq \mathbf{D}(S_0, S_1) = \mathbf{d}(\boldsymbol{v}_0, \boldsymbol{v}_1)$ for some $\boldsymbol{v}_1 \in S_1$. For simplicity, we may set $\boldsymbol{v}_0$ as the origin. The probability that an instance $\boldsymbol{v} \in S_1$ has a shorter length than $\boldsymbol{v}_1$ after projection to a line [see (5)] is denoted as $\mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v})$. By assumption, we only need to consider those instances whose length is greater than αd (outside the ball $\mathbf{B}(\alpha d)$ centered at the origin), that is, $\{\boldsymbol{v} \mid \boldsymbol{v} \notin \mathbf{B}(\alpha d)\}$. Then, the probability that *AcceleDist* fails to include such a candidate is bounded above by $1/(m_2)\sum_{\boldsymbol{v}\notin\mathbf{B}(\alpha d)} \mathbb{P}(\boldsymbol{v}_1, \boldsymbol{v})$; the probability that we could not find an approximate solution by *ApproxDist*

[2]Since the event only depends on the two inner products with $\boldsymbol{v}_2 - \boldsymbol{v}_1$ and $\boldsymbol{v}_1 + \boldsymbol{v}_2$, by rotational invariance, it suffices to consider the 2-D plane spanned by these two vectors. In this plane, the set of directions satisfying the above inequality has total angle 2θ out of 2π.

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
DATASET STATISTICS. THE COLUMN "#INST" REPRESENTS THE NUMBER OF INSTANCES

Datasets	#inst	#feature		#member in the privileged group	
		raw	binarized	1st priv	2nd priv
ricci	118	5	6	68 in race	—
credit	1000	21	59	690 in sex	851 in age
income	30162	14	99	25933 in race	20380 in sex
ppr	6167	11	402	4994 in sex	2100 in race
ppvr	4010	11	328	3173 in sex	1452 in race

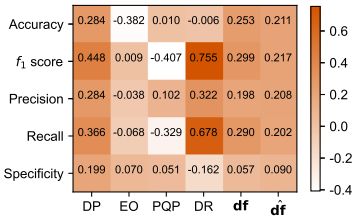


Fig. 2. Comparison of baseline fairness measures and the proposed *HFM*, evaluated on test data: Correlation heatmap between the evaluation metric and fairness. Note that **df** is calculated directly by (3), and $\hat{\mathbf{df}}$ is calculated approximately with distances between sets computed using *ApproxDist*.

is $\left(1/(m_2)\sum_{v\notin\mathbf{B}(\alpha d)}\mathbb{P}(\boldsymbol{v}_1,\boldsymbol{v})\right)^{m_1}$. Hence, the desired probability is bounded from below by

$$1-\left(\frac{1}{m_2}\sum_{\boldsymbol{v}\notin\mathbf{B}(\alpha d)}\mathbb{P}\left(\boldsymbol{v}_1,\boldsymbol{v}\right)\right)^{m_1}. \tag{14}$$

However, (14) is based on the extreme assumption that all instances lie on the same 2-D plane. In our case, the instances are evenly distributed. Hence, we may adjust the probability by multiplying

$$\frac{\mathrm{Vol}\left(S^1\left(\frac{\|\boldsymbol{v}\|}{d}\right)\right)}{\mathrm{Vol}\left(S^k\left(\frac{\|\boldsymbol{v}\|}{d}\right)\right)}=\frac{\Gamma\left(\frac{k+1}{2}\right)}{\pi^{\frac{k-1}{2}}}\cdot\left(\frac{d}{\|\boldsymbol{v}\|}\right)^{k-1}$$

where $\Gamma(\cdot)$ denotes the Gamma function and $\mathrm{Vol}(S^i(r))$ denotes the area of the ith dimensional sphere of radius r. Note that $\mathrm{Vol}(S^i(r))=(2\pi^{(i+1)/2}r^i)/(\Gamma((i+1)/2))$. Hence, by Lemma 1, the desired probability is lower bounded by

$$1-\left(\frac{1}{m_2}\sum_{\boldsymbol{v}\notin\mathbf{B}(\alpha d)}\left(1+\frac{d^2}{\|\boldsymbol{v}\|^2}\right)^{-\frac{1}{2}}\frac{\Gamma\left(\frac{k+1}{2}\right)}{\pi^{\frac{k-1}{2}}}\cdot\left(\frac{d}{\|\boldsymbol{v}\|}\right)^{k}\right)^{m_1}. \tag{15}$$

Under our assumption, (15) attains the lowest value when the data are evenly distributed inside a hollow ball $\mathbf{B}_0\setminus\mathbf{B}(d)$ centered at $\boldsymbol{v}_0$. The radius of $\mathbf{B}_0$, denoted as r_0, satisfies

$$n-1=\mu\frac{\mathrm{Vol}\left(\mathbf{B}_0\setminus\mathbf{B}\left(d\right)\right)}{\mathrm{Vol}\left(\mathbf{B}\left(d\right)\right)}=\mu\left(\left(\frac{r_0}{d}\right)^{k+1}-1\right). \tag{16}$$

Note that $\mathrm{Vol}(\mathbf{B}(d))$ denotes the volume of a ball of radius d, and, for a ith dimensional ball, $\mathrm{Vol}(\mathbf{B}^i(d))=(\pi^{i/2}d^i)/(\Gamma(i/2+1))$. Also note their relationship is $\mathrm{Vol}(S^k(r))=d/(dr)\mathrm{Vol}(\mathbf{B}^{i+1}(r))$.

In this situation, we may write the summation part of (15) as an integration. To be more specific, (15) is lower bounded by

$$1-\left(\frac{1}{m_2}\int_{\alpha d}^{r_0}A\left(x\right)\mu\,\mathrm{Vol}\left(S^k\left(x\right)\right)dx\right)^{m_1} \tag{17}$$

where $A(x)=(1+(d^2)/(x^2))^{-1/2}(\Gamma((k+1)/2))/(\pi^{(k-1)/2})\cdot(d/x)^k$. Moreover, (17) can be simplified as

$$1-\left(\frac{1}{m_2\mathrm{Vol}\left(\mathbf{B}\left(1\right)\right)}\int_{\alpha d}^{r_0}\frac{\pi\mu}{d}\cdot\frac{x}{\sqrt{x^2+d^2}}dx\right)^{m_1}. \tag{18}$$

Combining (16) and (18), we conclude that the desired probability is lower bounded by (19), that is,

$$1-\left(\frac{\pi\mu}{m_2\mathrm{Vol}\left(\mathbf{B}\left(1\right)\right)}\left(\left(\left(1+\frac{n}{\mu}\right)^{2/(k+1)}+1\right)^{\frac{1}{2}}-\left(\alpha^2+1\right)^{\frac{1}{2}}\right)\right)^{m_1}. \tag{19}$$

And then the proposition follows from (19). □

The analysis in Proposition 1 relies on a uniform ball assumption, which is admittedly strong and not directly verifiable for real-world data. This assumption is introduced to enable a tractable analysis and to derive a nontrivial lower bound on the probability that *ApproxDist* includes true nearest neighbors in the candidate set. Intuitively, the uniform ball condition ensures that each local neighborhood has sufficient mass in all directions, allowing random projections to capture nearby points with bounded probability, while the scaled density parameter μ abstracts the worst case local sparsity relative to the global scale. When these assumptions are violated (for example, in the presence of dense clusters, anisotropic structures, or highly uneven data densities), the resulting bound may become loose or even vacuous, as the underlying geometric argument no longer applies. Importantly, this does not imply that *ApproxDist* fails in practice. From an algorithmic perspective, violations of the uniform ball assumption primarily affect the tightness of the probabilistic bound, rather than the correctness of the approximation procedure itself, since *ApproxDist* always produces an upper bound of the true set distance by construction. In fact, under clustered or nonuniform distributions, random projections often preserve local neighbor relations well, making the approximate candidate sets likely to contain true nearest neighbors with high empirical probability. Since clustered structures increase local point density, the probability that true nearest neighbors appear among nearby projected candidates is typically higher than the worst case scenario captured by the uniform analysis. Moreover, our experiments on nonuniform and clustered datasets (see Sections IV-D and IV-E) confirm that the approximation quality remains high beyond the scope of the current theoretical guarantees. Extending the analysis to relax these assumptions using data-dependent geometric measures, such as intrinsic dimension or doubling dimension, is an interesting direction for future work.

Moreover, note that by (13), the efficiency of *ApproxDist* decreases as the scaled density $\mu/(\mathrm{Vol}(\mathbf{B}(d)))$ of the original dataset increases. Meanwhile, when dealing with a large-scale dataset, the more insensitive attributes we have, the more efficient *ApproxDist* is. In general, the efficiency of *ApproxDist* depends on the shape of these two subsets of S. Roughly speaking, the more these two sets are separated from each other, the more efficient *ApproxDist* is.

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

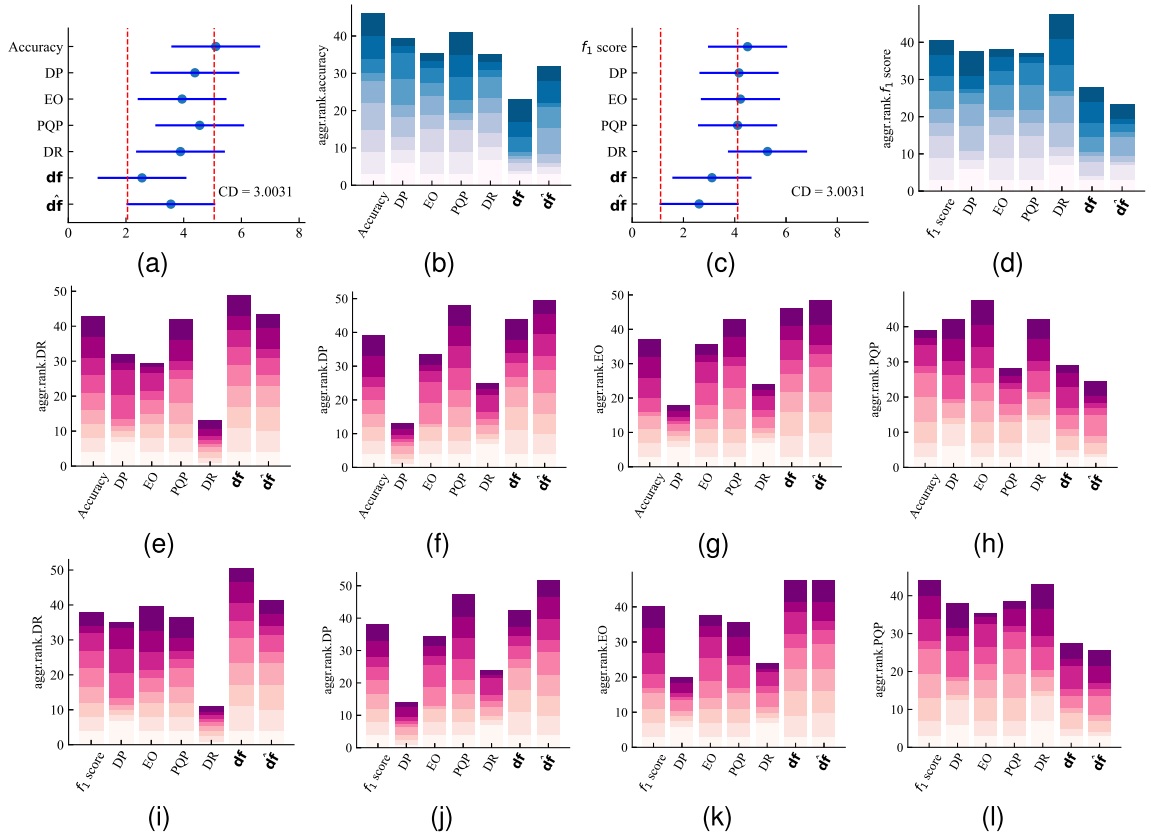


Fig. 3. Comparison of baseline fairness measures and the proposed *HFM*. (a) Friedman test chart (nonoverlapping means significant difference) on the test accuracy, which rejects the null hypothesis that "all fairness-relevant rules have the same evaluation performance" at the significance level of 5%, and where CD means the critical difference of average rank difference, calculated by Nemenyi posthoc test [58]; (b) aggregated rank of each fairness-relevant rule (the smaller the better) [59] on the test accuracy. (c) and (d) Friedman test chart and the aggregated rank of each fairness-relevant rule on the f_1 score, evaluated on test data. (e)–(h) Aggregated rank of each fairness-relevant rule on the DR and three group fairness measures, respectively. (i)–(l) Aggregated rank of each fairness-relevant rule on the DR, DP, EO, and PQP, respectively. Note that the ranking rule in (a) and (b) and (e)–(h) is 0.05 $\times$ error rate + 0.95 $\times$ fairness, and that in (c) and (d) and (i)–(l) is 0.1 $\times$ (1 − f_1 score) + 0.9 $\times$ fairness. Also note that in all of them except (a) and (c), the smaller (in the y-axis), the better.

Now we discuss our choice of m_1 and m_2 according to (13). In fact, (13) can be approximately written as $1 - c \cdot n^{(m_1)/(k+1)}/m_2^{m_1}$. We can calculate the order of magnitude of $n^{(m_1)/(k+1)}/m_2^{m_1}$ by taking the logarithm

$$-\lambda \triangleq \lg\left(n^{\frac{m_1}{k+1}}/m_2^{m_1}\right) = m_1\left(\frac{\lg\ n}{k+1} - \lg\ m_2\right). \tag{20}$$

The *ApproxDist* could reach an approximate solution with probability at least $1 - c \cdot 10^{-\lambda}$. In practice, we choose positive integers m_2 and m_1 such that λ is reasonably large, ensuring that the algorithm will reach an approximate solution with high probability. That is to say, a fine-grained tuning of hyperparameters (m_1 or m_2) is not required, while they may grow moderately with the dataset size.

IV. Empirical Results

In this section, we elaborate on our experiments to evaluate the effectiveness of the proposed *HFM* in (3) and *ApproxDist* in Algorithm 1. These experiments are conducted to explore the following research questions.

RQ1: Compared with the prevailing baseline fairness measures, does the proposed *HFM* capture the discriminative degree of one classifier effectively, and can it capture the discrimination level from both individual and group fairness aspects?

RQ2: Can *ApproxDist* approximate the direct computation of distances in (6) precisely, and how efficient is *ApproxDist* compared with the direct computation of distances?

RQ3: Will the choice of hyperparameters (that is, m_1 and m_2 in *ApproxDist*) affect the approximation results, and if the answer is yes, how?

A. Experimental Setups

In this section, we present the experimental setting, including datasets, evaluation metrics, baseline fairness measures, and implementation details.

1) Datasets: Five public datasets were adopted in the experiments: Ricci,[3] Credit,[4] Income (aka. Adult),[5] COMPAS PPR, and COMPAS PPVR.[6] Each of them has two sensitive attributes except Ricci, with more statistical details provided in Table I, and is commonly used in the literature [57, Table 2]. Among these datasets, Ricci was used in a court case dealing with racial discrimination; COMPAS PPR and PPVR were relevant to racial discrimination in criminal risk assessments.

[3]https://rdrr.io/cran/Stat2Data/man/Ricci.html
[4]https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data
[5]https://archive.ics.uci.edu/dataset/2/adult
[6]https://github.com/propublica/compas-analysis/, that is, Propublica - Recidivism and Propublica-Violent-Recidivism datasets

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

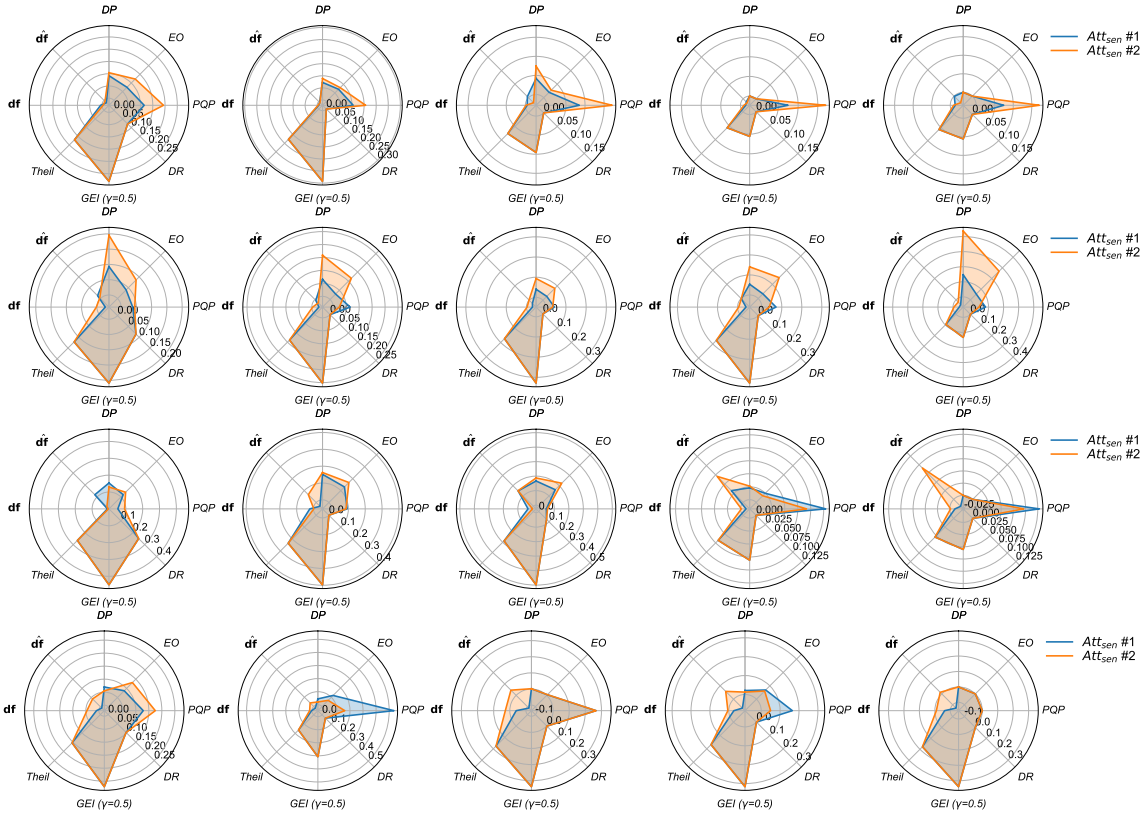


Fig. 4. Comparison of baseline fairness measures and the proposed *HFM*, where "Att_{sen} #i" indicates the unfairness degree of the corresponding algorithm regarding a_i, that is, the ith sensitive attribute in the dataset. Four rows correspond to results on the Credit, Income, PPR, and PPVR datasets, respectively; Five columns correspond to results using bagging, AdaBoost, LightGBM, AdaFair (mitigating bias from the first sensitive attribute), and AdaFair (mitigating bias from the second sensitive attribute), respectively. Also note that $\gamma = 0.5$ in the generalized entropy indices (GEI) computation.

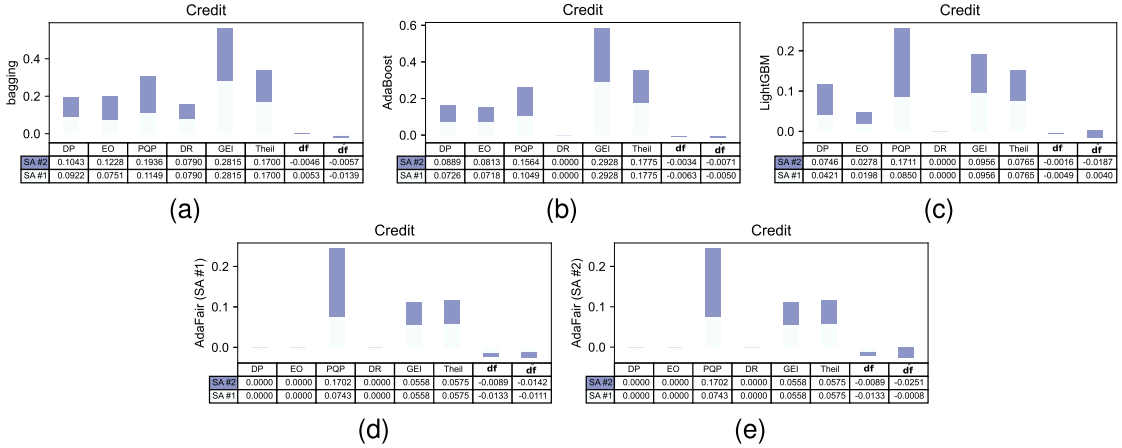


Fig. 5. Another expression of sub-figures in the first row of Fig. 3, where (a)–(e) correspond to five columns (i.e., learning algorithms) respectively.

2) Evaluation Metrics: As data imbalance usually exists within unfair datasets, we consider several criteria to evaluate the prediction performance, including accuracy, precision, recall (aka. sensitivity), f_1 score, and specificity. For efficiency metrics, we directly compare the time cost of different methods.

3) Baseline Methods: To evaluate the validity of *Approx-Dist* in efficiently estimating the exact values of distances, we compare it with EarlyBreak [60], an efficient algorithm for calculating the exact distances. To evaluate the validity of *HFM* in capturing the discriminative degree of classifiers, we compare it with three commonly used group fairness measures (that is, DP [9], [44], EO [11], and PQP [16], [45]),[7] two individual fairness measures (that is, generalized

[7]Three commonly used group fairness measures of one classifier $f(\cdot)$ are evaluated as

$$\text{DP}(f) = \left|\mathbb{P}_{\mathcal{D}}\left[f(\boldsymbol{x}) = 1 | \boldsymbol{a} = 1\right] - \mathbb{P}_{\mathcal{D}}\left[f(\boldsymbol{x}) = 1 | \boldsymbol{a} = 0\right]\right| \tag{21a}$$

$$\text{EO}(f) = \left|\mathbb{P}_{\mathcal{D}}\left[f(\boldsymbol{x}) = 1 | \boldsymbol{a} = 1, y = 1\right] - \mathbb{P}_{\mathcal{D}}\left[f(\boldsymbol{x}) = 1 | \boldsymbol{a} = 0, y = 1\right]\right| \tag{21b}$$

$$\text{PQP}(f) = \left|\mathbb{P}_{\mathcal{D}}\left[y = 1 | \boldsymbol{a} = 1, f(\boldsymbol{x}) = 1\right] - \mathbb{P}_{\mathcal{D}}\left[y = 1 | \boldsymbol{a} = 0, f(\boldsymbol{x}) = 1\right]\right| \tag{21c}$$

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

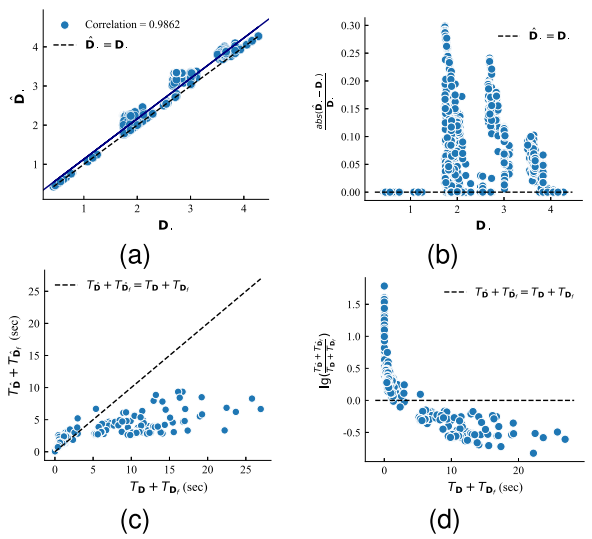


Fig. 6. Comparison of approximation distances between sets with precise distances that are calculated directly by definition, evaluated on test data. (a) Scatter plot showing approximated values and precise values of distances between sets. (b) Relative difference comparison of *ApproxDist* with direct computation concerning distance values. (c) and (d) Comparison of time cost (second) between *ApproxDist* and direct computation based on (6).

entropy indices (GEI) [61] and the Theil index (Theil) [62]),[8] and DR[9] [53]. Note that only DR can take both individual and group fairness aspects into account at the same time. Also note that BENN [38] and the FAE-based metric [37] are not included because they require extra trained models for unfairness evaluation, and therefore, they would break the same conditions in a comparison.

4) Implementation Details: We mainly use bagging, AdaBoost, LightGBM [63], FairGBM [64], and AdaFair [65] as learning algorithms, where FairGBM and AdaFair are two fairness-aware ensemble-based methods. Plus, certain kinds of classifiers are used in Section IV-B—including decision trees (DTs), naive Bayesian (NB) classifiers, k-nearest neighbors (KNN) classifiers, logistic regression (LR), support vector machines (SVMs), linear SVMs (linSVM), and multilayer perceptrons (MLPs)—so that we have a larger learner pool to choose from based on different fairness-relevant rules. Standard fivefold cross-validation is used in these experiments; in other words, in each iteration, the entire dataset is divided into two parts, with 80% as the training set and 20% as the test set. Also, features of datasets are scaled in preprocessing to lie between 0 and 1. Except for the experiments for **RQ3**, we set the hyperparameters $m_1 = 25$ and $m_2 = \lceil 2 \lg n \rceil$ in other experiments, where n is the size of the corresponding dataset. The value of 25 for m_1 is randomly chosen, which is neither too large to incur high computational costs, nor too small to lead to worse approximation effectiveness.

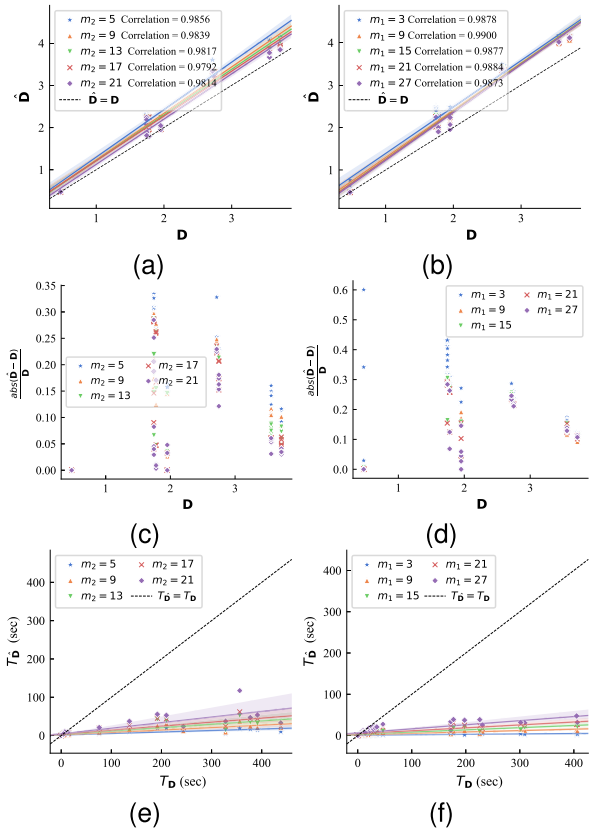


Fig. 7. Effects of hyperparameters m_1 and m_2 in *ApproxDist*. (a) and (c) Effect of the hyperparameter m_2 on the distance value; (e) effect of the hyperparameter m_2 on the time cost, where m_1 is set to 20. (b) and (d) Effect of the hyperparameter m_1 on the distance value; (f) effect of the hyperparameter m_1 on the time cost, where m_2 is set to $\lceil 2 \lg n \rceil$ in terms of n—the size of the corresponding dataset.

B. Comparison Between HFM and Group Fairness Measures

In this experiment, we evaluate the performance of the proposed *HFM* compared with baseline group fairness measures.

As ground-truth discriminative levels of classifiers remain unknown, and it is hard to directly compare different methods from that perspective, we compare the Pearson correlation coefficient between the performance and different fairness measures. The empirical results are reported in Fig. 2. Note that the reported *HFM* includes both **df** and $\hat{\mathbf{df}}$, where $\hat{\mathbf{df}}$ is the approximated value of the precise **df** via *ApproxDist*. While *HFM* does not demonstrate a strong correlation with evaluation performance, its correlation is better than EO and PQP, as shown in Fig. 2. The reason might be that *HFM* is

respectively, where $\boldsymbol{x} = (\check{\boldsymbol{x}}, \boldsymbol{a})$, y, and $f(\boldsymbol{x})$ are, respectively, the features, the true label, and the prediction of this classifier for one instance. Note that $\boldsymbol{a} = 1$ and 0, respectively, mean that the instance $\boldsymbol{x}$ belongs to the privileged and the marginalized groups.

[8]For a constant $\gamma \notin \{0, 1\}$, the GEI for a problem with n instances are defined, to quantify algorithmic unfairness, as

$$\mathrm{GEI}^{\gamma} = \frac{1}{n\gamma(\gamma-1)} \sum_{i=1}^{n} \left(\left(\frac{b_i}{\mu} \right)^{\gamma} - 1 \right) \tag{22}$$

where benefits $b_i = f(\check{\boldsymbol{x}}_i, a_{1i}) - y_i + 1$ and $\mu = \left(\sum_i b_i\right)/n$. The Theil index is a special case for $\gamma = 1$, that is,

$$\mathrm{Theil} = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i}{\mu} \log\left(\frac{b_i}{\mu} \right). \tag{23}$$

They are used additionally to group fairness measures to compare different algorithms and determine which one is considered the fairest from an individual perspective.

[10]The DR of this classifier is evaluated as

$$\mathrm{DR}(f) = \mathbb{E}_{\mathcal{D}} \left[\mathbb{I}\left(f(\check{\boldsymbol{x}}, \boldsymbol{a}) \neq f(\check{\boldsymbol{x}}, \tilde{\boldsymbol{a}}) \right) \right] \tag{24}$$

where $\tilde{\boldsymbol{a}}$ represents the disturbed sensivtive attributes. DR reflects the bias degree of one classifier from both individual and group fairness aspects.

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

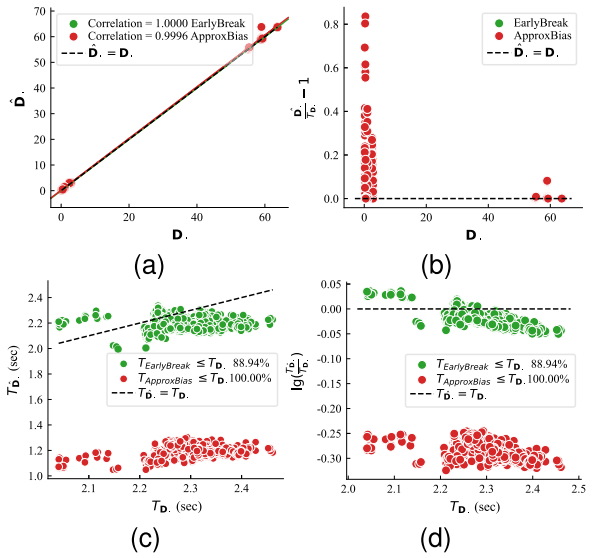


Fig. 8. Comparison between *ApproxDist* and EarlyBreak [60], on the Income dataset. (a) and (b) Estimated values, generated by *ApproxDist* or EarlyBreak, are accurate; (c) and (d) time cost comparison, showing that *ApproxDist* has higher computational efficiency than EarlyBreak.

only used to measure the extra bias introduced by classifiers or in the learning process, unlike other fairness measures that do not distinguish extra bias from overall discrimination within.

We also compare the test performance of the classifier that is chosen on training data based on fairness-relevant rules. In other words, we use a rule of "$\beta \cdot (1 - \text{performance}) + (1 - \beta) \cdot \text{fairness}$" to select the best classifier and then report the corresponding test performance of these selected classifiers. The empirical results are reported in Fig. 3. We observe from Fig. 3(a) and (b) that using *HFM* (i.e., $\mathbf{df}$ and $\hat{\mathbf{df}}$) achieves the first and second best of the test accuracy over using three group fairness measures and DR, and using DR also achieves relatively good performance, which means it is indeed useful to consider the discriminative level of classifiers from both individual- and group-fairness aspects, while DP, EO, and PQP only consider group-fairness aspects. It also demonstrates that *HFM* could measure the discriminative level of classifiers somehow from both individual- and group-fairness perspectives, just like DR does. Using *HFM* also demonstrates superior performance in Fig. 3(c) and (d). As for Fig. 3(e) and (i), using *HFM* (i.e., $\hat{\mathbf{df}}$) presents relatively weaker fairness performance, especially when compared with DR. This is understandable because, unlike all other baseline fairness measures, *HFM* only measures the newly introduced bias by classifiers in the learning, without reflecting the bias within the data. In other words, *HFM* is unlikely to recognize the primitive bias hidden in the data if no extra bias is introduced by the learner/classifier. Thus, *HFM* probably reflects partial bias information in the learning compared with other fairness measures, leading to relatively inferior performance. However, it is worth noting that using *HFM* shows pretty good results when evaluated from the perspective of PQP in Fig. 3(h) and (l). Except for them, similar observations could also be found in other sub-figures of Fig. 3.

C. Comparison Between HFM and Individual Fairness Metrics

In this experiment, we evaluate the performance of *HFM* in comparison with both group and individual fairness, with empirical results presented in Figs. 4 and 5. As we can see from the results on PPR and PPVR datasets (third to fourth rows) in Fig. 4, AdaFair (fourth to fifth columns) presents good performance concerning group fairness (especially, DP and EO) yet still exhibits bias from the perspective of individual fairness (GEI and Theil), and this phenomenon is indicated in *HFM*, suggesting that discrimination is not fully eliminated through AdaFair. Meanwhile, both *HFM* and group fairness indicate the existence of slight bias in AdaBoost and LightGBM (second and third columns). Considering that *HFM* can only capture the extra bias introduced in the learning procedure, we believe *HFM* is already reflecting both group- and individual-fairness aspects in its own way as best as it can. Furthermore, we particularly notice that, in Fig. 5(d) and (e), both DP and EO equal zero, and $\mathbf{df}(f) < 0$, indicating that AdaFair mitigates bias in learning, yet bias is still indicated concerning PQP, GEI, and Theil. This is an interesting observation, suggesting that there is still room for AdaFair to improve and further mitigate bias hidden in data.

D. Validity of ApproxDist to Approximate Distances Between Sets for Euclidean Spaces

In this experiment, we evaluate the performance of the proposed *ApproxDist* compared with the precise distance that is computed by definition directly. To verify whether *ApproxDist* could achieve the true distance between sets accurately and in a timely manner, we employ scatter plots to compare their values and time cost, presented in Figs. 6 and 7 (and Fig. 9 of Appendix). Particularly, we compare *ApproxDist* with EarlyBreak [60], an efficient algorithm for calculating the exact distances, in Figs. 8 and 10 of Appendix. From them, we observe that the estimated values produced by *ApproxDist* are close to the corresponding exact values of distances, and that they consume much less time than those produced by EarlyBreak. This suggests that *ApproxDist* has higher computational efficiency.

Besides, as we can see from Fig. 6(a), the approximated values of distance using *ApproxDist* are highly correlated with their corresponding precise values. Besides, their linear fit line and the identity line (that is, $f(x) = x$) are near and almost parallel. Both observations mean that the approximation values of distance via *ApproxDist* are valid and acceptable. Fig. 6(b) also shows that the relative difference between the approximation values and direct computation of distances is comparatively small and acceptable. It also shows that the relative difference between them becomes smaller as the distance $\mathbf{D}$. increases. As for the time cost of *ApproxDist* shown in Fig. 6(c), all approximated values of distance cost less time than their direct computation of precise distance, and we find in practice that the time saving is usually more satisfactory when the used dataset is pretty large, such as on the Income dataset. Similar observations could also be found in Figs. 6(d), 7(e) and (f), and 9(e) and (g), where in most cases

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

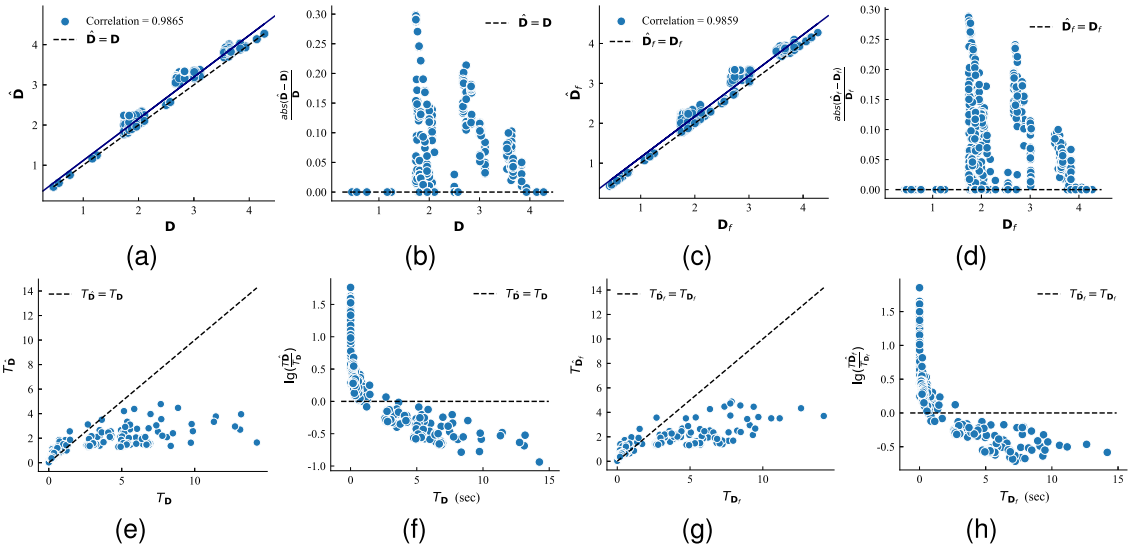


Fig. 9. Continuation of Fig. 6: comparison of approximation distances using *ApproxDist* with precise distances between sets that are calculated directly by definition, evaluated on test data. (a) Scatter plot showing approximated value $\hat{\mathbf{D}}$ and precise value $\mathbf{D}$ of distance between sets in (1). (b) Relative difference comparison between $\hat{\mathbf{D}}$ and $\mathbf{D}$. (c) Scatter plot showing approximated value $\hat{\mathbf{D}}_f$ and precise value $\mathbf{D}_f$ of distance between sets in (2). (d) Relative difference comparison between $\hat{\mathbf{D}}_f$ and $\mathbf{D}_f$. (e) and (f) Comparison of time cost between getting $\hat{\mathbf{D}}$ and $\mathbf{D}$. (g) and (h) Comparison of time cost between getting $\hat{\mathbf{D}}_f$ and $\mathbf{D}_f$. Note that Fig. 6(a) is divided into (a) and (c), Fig. 6(b) is divided into (b) and (d), and Fig. 6(c) and (d) is divided analogously.

of large datasets that need a longer time to proceed with the direct computation of distances, the time cost of *ApproxDist* would not be worse than that of direct computation; yet in smaller dataset cases, the time cost of direct computation is less and completely acceptable, and using *ApproxDist* would be less necessary.

As for Fig. 9, it is a more detailed version of Fig. 6 by separating $\mathbf{D}_f$ and $\mathbf{D}$ from $\mathbf{D}$. in (6). Briefly speaking, Fig. 9(a) and (c) exhibits the same observation as in Fig. 6(a); Fig. 9(b) and (d) exhibits the same observation at in Fig. 6(b); Fig. 9(e) and (g) exhibits the same observation as in Fig. 6(c); Fig. 9(f) and (h) exhibits the same observation as in Fig. 6(d). To summarize, all these figures show that the approximated distances produced by *ApproxDist* closely match the exact distances from direct computation, despite the use of random projections. This empirical evidence also suggests that *ApproxDist* is robust to the randomness introduced by projection vectors and does not exhibit unstable behavior across different runs.

E. Effect of Hyperparameters m_1 and m_2

In this section, we investigate whether different choices of hyperparameters (that is, m_1 and m_2) would affect the performance of *ApproxDist* or not. Different m_2 values are tested when m_1 is set, and vice versa, with empirical results presented in Fig. 7. As we can see from Fig. 7(e) and (f), obtaining approximated values of distance distinctly costs less time than that of precise values, while increasing m_2 (or m_1) would cost more time but still be less than that of precise values. As for the approximation performance shown in Fig. 7(a) and (b), all approximated values are highly correlated to the precise values of distance no matter how small m_1 or m_2 is, which means the effect of less proper choices of hyperparameters is relatively unapparent; As m_1 (or m_2) increases, the approximated values would become closer to the precise values of distance. This means that the approximation quality remains stable over a reasonably wide range of hyperparameter values, and that the risk associated with suboptimal hyperparameter selection is low in practice.

V. Conclusion

In this article, we investigate how to measure the discrimination level of classifiers from both individual and group fairness aspects and present a novel *HFM* based on distances between sets. To facilitate the calculation of distances and reduce its time cost from $O(n^2)$ to $O(n \log\ n)$, we propose an approximation algorithm *ApproxDist* and further present its algorithmic effectiveness analysis. The empirical results have demonstrated that the proposed *HFM* and *ApproxDist* are valid and effective. However, the limitation of *ApproxDist* is that it may not converge and thus can only reach approximated values of the distance. *HFM* also has a limitation that it can only evaluate the extra bias introduced in the learning procedure. In other words, if the learning algorithm is purely "fair" and no extra bias is introduced, it may not be able to detect whether discrimination exists in the data already. It would therefore be beneficial if some other fairness measures (such as GEI, Theil, and DR) are also considered in model evaluation, accompanying *HFM*. In practice, we indeed observed many zeros and even negative values, which leaves an open question about how to interpret them in such cases. In the future, we plan to explore the possibility of a fuller bias evaluation in this direction. To exploit it for improving learning algorithms could be interesting as well.

Appendix
Additional Empirical Results

In this section, we list additional empirical results of Section IV-D that are left out in the main paper to save space, that is, Figs. 9 and 10.

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

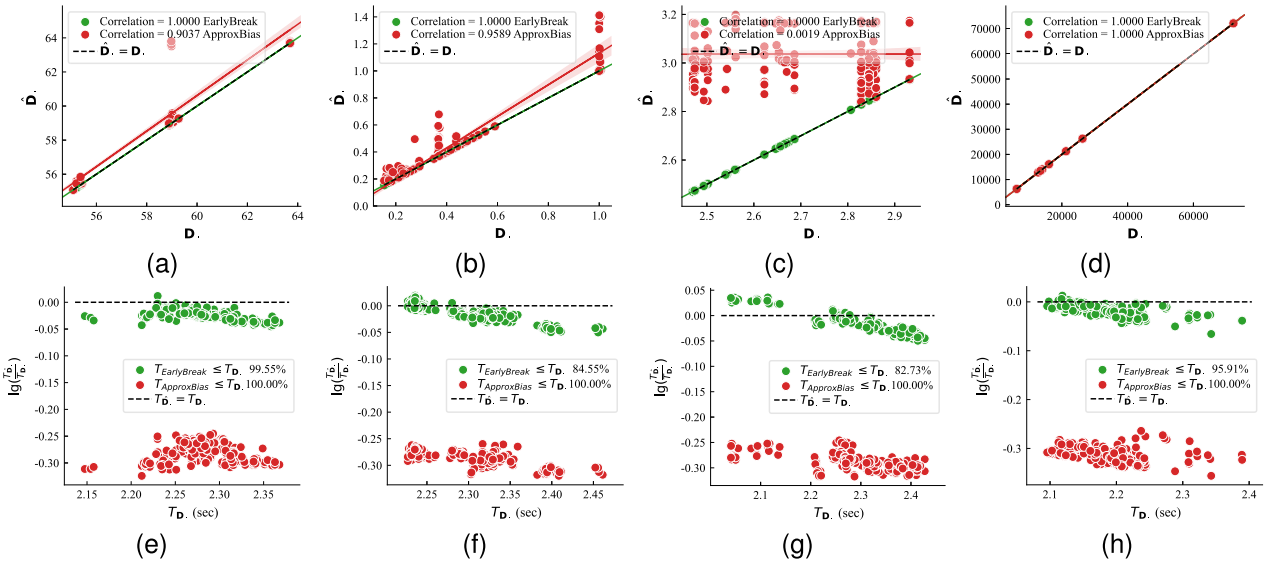


Fig. 10. Continuation of Fig. 8: four columns correspond to different preprocessing methods (standardization, normalization, min–max scaling, and no preprocessing, respectively). (a)–(d) Comparison between the estimated values and exact values of $\mathbf{D}$.; (e)–(h) computational efficiency comparison.

REFERENCES

[1] M. Chua et al., "Tackling prediction uncertainty in machine learning for healthcare," *Nature Biomed. Eng.*, vol. 7, no. 6, pp. 711–718, Dec. 2022.

[2] S. Azizi et al., "Robust and data-efficient generalization of self-supervised machine learning for diagnostic imaging," *Nat. Biomed. Eng.*, vol. 7, no. 6, pp. 756–779, Jun. 2023.

[3] G. Varoquaux and V. Cheplygina, "Machine learning for medical imaging: Methodological failures and recommendations for the future," *npj Digit. Med.*, vol. 5, no. 1, p. 48, Apr. 2022.

[4] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proc. 3rd Innov. Theor. Comput. Sci. Conf.*, Jan. 2012, pp. 214–226.

[5] W. Fleisher, "What's fair about individual fairness?," in *Proc. AIES*, 2021, pp. 480–490.

[6] Centers for Disease Control and Prevention. (2022). *Types of Discrimination*. Accessed: Aug. 29, 2022. [Online]. Available: https://www.cdc.gov/eeo/faqs/discrimination.htm

[7] Commission. (2022). *Discrimination By Type*. Accessed: Aug. 29, 2022. [Online]. Available: https://www.eeoc.gov/discrimination-type

[8] T. Calders, F. Kamiran, and M. Pechenizkiy, "Building classifiers with independency constraints," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2009, pp. 13–18.

[9] M. B. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *Proc. SIGKDD*, 2014, pp. 259–268.

[10] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *Proc. ICML*, pp. 325–333.

[11] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 3323–3331.

[12] W. Dieterich, C. Mendoza, and T. Brennan, "Demonstrating accuracy equity and predictive parity performance of the compas risk scales in broward county," Dept. Res., Northpointe Inc., MI, USA, Tech. Rep., 2016. [Online]. Available: https://opencasebook.org/casebooks/6382-ethics-governance-of-artificial-intelligence-egai/resources/3.3-compas-risk-scales-demonstrating-accuracy-equity-and-predictive-parity-by-northpointe-inc-northpointe-2016-read-executive-summary-introduction-and-conclusion/

[13] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*. Cambridge, MA, USA: MIT Press, 2023.

[14] Z. Tang, J. Zhang, and K. Zhang, "What-is and how-to for fairness in machine learning: A survey, reflection, and perspective," *ACM Comput. Surveys*, vol. 55, no. 13s, pp. 1–37, Dec. 2023.

[15] H. Zhao and G. J. Gordon, "Inherent tradeoffs in learning fair representations," *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 2527–2552, 2022.

[16] S. Verma and J. Rubin, "Fairness definitions explained," in *Proc. FairWare*, 2018, pp. 1–7.

[17] D. Pessach and E. Shmueli, "Algorithmic fairness," in *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*. Cham, Switzerland: Springer, 2023, pp. 867–886.

[18] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–35, Jul. 2022.

[19] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, and T. Wagner, "Scalable fair clustering," in *Proc. ICML*, vol. 97, 2019, pp. 405–413.

[20] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," in *Proc. NIPS*, vol. 30, 2018, pp. 5029–5037.

[21] F. P. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," in *Proc. NIPS*, vol. 30, 2017, pp. 3992–4001.

[22] H. Tian, B. Liu, T. Zhu, W. Zhou, and P. S. Yu, "MultiFair: Model fairness with multiple sensitive attributes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, pp. 5654–5667, Mar. 2025.

[23] Q. Yang, Y. Deng, J. Huang, L. Lu, P. Zhou, and G. Min, "CIRCUS: A causal intervention-based framework for enhancing counterfactual fairness in trained classifiers," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/11435129

[24] Y. Xia, X. Zhang, and Z. Li, "Learning counterfactual fair representation under covariate shift via reflux," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 37, no. 2, pp. 878–892, Feb. 2026.

[25] L. Zhu, Y. Bian, and L. You, "FairSHAP: Preprocessing for fairness through attribution-based data augmentation," in *Proc. AISTATS*, Tangier, Morocco, May 2026.

[26] C. Dwork, N. Immorlica, A. T. Kalai, and M. Leiserson, "Decoupled classifiers for group-fair and efficient machine learning," in *Proc. FAT*, vol. 81, 2018, pp. 119–133.

[27] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic decision making and the cost of fairness," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 797–806.

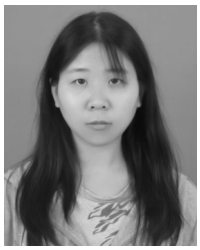
[28] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 1171–1180.

[29] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *Proc. AISTATS*, 2015, pp. 962–970.

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.

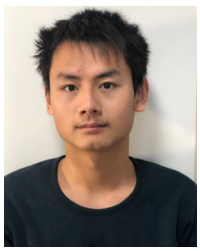
This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[30] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro, "Learning non-discriminatory predictors," in *Proc. COLT*, 2017, pp. 1920–1953.
[31] N. Quadrianto and V. Sharmanska, "Recycling privileged learning and distribution matching for fairness," in *Proc. NIPS*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html
[32] R. Du, D. Muthirayan, P. P. Khargonekar, and Y. Shen, "Long-term fairness for real-time decision making: A constrained online optimization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 7, pp. 13149–13161, Jul. 2025.
[33] A. Agarwal, M. Dudík, and Z. S. Wu, "Fair regression: Quantitative definitions and reduction-based algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 120–129.
[34] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach, "A reductions approach to fair classification," in *Proc. ICML*, vol. 80, 2018, pp. 60–69.
[35] 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 *Proc. FAT*, 2019, pp. 329–338.
[36] N. Grgić-Hlača, M. B. Zafar, K. P. Gummadi, and A. Weller, "Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning," in *Proc. AAAI*, 2018, vol. 32, no. 1, pp. 51–60.
[37] Z. Wang, C. Huang, K. Tang, and X. Yao, "Procedural fairness in machine learning," 2024, *arXiv:2404.01877*.
[38] A. Giloni et al., "BENN: Bias estimation using a deep neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 117–131, Jan. 2024.
[39] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, "Fairness in criminal justice risk assessments: The state of the art," *Sociol. Methods Res.*, vol. 50, no. 1, pp. 3–44, Feb. 2021.
[40] I. Žliobaitė, "Measuring discrimination in algorithmic decision making," *Data Mining Knowl. Discovery*, vol. 31, no. 4, pp. 1060–1089, Jul. 2017.
[41] M. Joseph, M. Kearns, J. H. Morgenstern, and A. Roth, "Fairness in learning: Classic and contextual bandits," in *Proc. NIPS*, vol. 29, 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/hash/eb163727917cbba1eea208541a643e74-Abstract.html
[42] H. Nilforoshan, J. D. Gaebler, R. Shroff, and S. Goel, "Causal conceptions of fairness and their consequences," in *Proc. ICML*, vol. 162, 2022, pp. 16848–16887.
[43] M. J. Kusner, J. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *Proc. NIPS*, vol. 30, 2017, pp. 4069–4079.
[44] P. Gajane and M. Pechenizkiy, "On formalizing fairness in prediction with machine learning," 2017, *arXiv:1710.03184*.
[45] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big Data*, vol. 5, no. 2, pp. 153–163, Jun. 2017.
[46] U. Hébert-Johnson, M. Kim, O. Reingold, and G. Rothblum, "Multicalibration: Calibration for the (computationally-identifiable) masses," in *Proc. ICML*, 2018, pp. 1939–1948.
[47] U. Gohar and L. Cheng, "A survey on intersectional fairness in machine learning: Notions, mitigation, and challenges," in *Proc. IJCAI*, 2023, pp. 6619–6627.
[48] M. P. Kim, A. Ghorbani, and J. Zou, "Multiaccuracy: Black-box post-processing for fairness in classification," in *Proc. AIES*, 2019, pp. 247–254.
[49] J. R. Foulds, R. Islam, K. N. Keya, and S. Pan, "An intersectional definition of fairness," in *Proc. ICDE*, Apr. 2020, pp. 1918–1921.
[50] E. Diana, S. Sharifi-Malvajerdi, and A. Vakilian, "Minimax group fairness in strategic classification," in *Proc. SatML*, 2024, pp. 753–772.
[51] A. Ghosh, L. Genuit, and M. Reagan, "Characterizing intersectional group fairness with worst-case comparisons," in *Proc. Workshop AIDBEI*, vol. 142, pp. 22–34.
[52] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," in *Proc. NIPS*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/b8b9c74ac526fffbeb2d39ab038d1cd7-Abstract.html
[53] Y. Bian, "Improving fairness with ensemble combination: Margin-dependent bounds," in *Proc. FAccT*, Jun. 2026, pp. 352–381.
[54] A. K. Menon and R. C. Williamson, "The cost of fairness in binary classification," in *Proc. FAT*, vol. 81, 2018, pp. 107–118.
[55] D. Pessach and E. Shmueli, "Improving fairness of artificial intelligence algorithms in privileged-group selection bias data settings," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115667.
[56] M. Wick, S. Panda, and J.-B. Tristan, "Unlocking fairness: A trade-off revisited," in *Proc. NeurIPS*, vol. 32, 2019, pp. 8783–8792.
[57] D. Pessach and E. Shmueli, "A review on fairness in machine learning," *ACM Comput. Surv.*, vol. 55, no. 3, pp. 1–44, 2022.
[58] Z.-H. Zhou, *Machine Learning*. Cham, Switzerland: Springer, 2021.
[59] C. Qian, Y. Yu, and Z.-H. Zhou, "Pareto ensemble pruning," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 2935–2941.
[60] A. A. Taha and A. Hanbury, "An efficient algorithm for calculating the exact Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2153–2163, Nov. 2015.
[61] T. Speicher et al., "A unified approach to quantifying algorithmic unfairness: Measuring individual & group unfairness via inequality indices," in *Proc. SIGKDD*, 2018, pp. 2239–2248.
[62] C. Haas, "The price of fairness—A framework to explore trade-offs in algorithmic fairness," in *Proc. ICIS*, 2019. [Online]. Available: https://aisel.aisnet.org/icis2019/data_science/data_science/19/
[63] G. Ke, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. NIPS*, vol. 30, 2017, pp. 3146–3154.
[64] A. F. Cruz, C. Belém, J. Bravo, P. Saleiro, and P. Bizarro, "FairGBM: Gradient boosting with fairness constraints," 2023, *arXiv:2209.07850*.
[65] V. Iosifidis and E. Ntoutsi, "AdaFair: Cumulative fairness adaptive boosting," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 781–790.



Yijun Bian received the B.Sc. degree in computational mathematics from Northwest A&F University, Yangling, China, in 2014, and the Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2020.

She is currently a Marie Skłodowska-Curie Postdoctoral Fellow with the AI Section, Department of Computer Science, University of Copenhagen, Copenhagen, Denmark. Her research interests include machine learning, ensemble methods, and fairness in machine learning.



Yujie Luo received the B.Sc. degree in mathematics from the University of Science and Technology of China, Hefei, China, in 2018, and the Ph.D. degree in mathematics from Johns Hopkins University, Baltimore, MD, USA, in 2023.

He is currently a Peng Tsu Ann Assistant Professor (Visiting Fellow) with the Department of Mathematics, National University of Singapore, Singapore. His research interests include algebraic geometry.

Authorized licensed use limited to: POLITECHNIKI WARSZAWSKIEJ. Downloaded on October 08,2026 at 10:12:41 UTC from IEEE Xplore. Restrictions apply.