Outlier Detection Bias Busted: Understanding Sources of Algorithmic Bias through Data-centric Factors

Xueying Ding
Carnegie Mellon University
Pittsburgh, PA, USA
xding2@cs.cmu.edu

Rui Xi
Carnegie Mellon University
Pittsburgh, PA, USA
rxi2@andrew.cmu.edu

Leman Akoglu
Carnegie Mellon University
Pittsburgh, PA, USA
lakoglu@cs.cmu.edu

ABSTRACT

The astonishing successes of ML have raised growing concern for the fairness of modern methods when deployed in real world settings. However, studies on fairness have mostly focused on supervised ML, while unsupervised outlier detection (OD), with numerous applications in finance, security, etc., have attracted little attention. While a few studies proposed fairness-enhanced OD algorithms, they remain agnostic to the underlying driving mechanisms or *sources of unfairness*. Even within the supervised ML literature,there exists debate on whether unfairness stems solely from algorithmic biases (i.e. design choices) or from the biases encoded in the data on which they are trained.

To close this gap, this work aims to shed light on the possible sources of unfairness in OD by auditing detection models under different data-centric factors.By injecting various known biases into the input data—as pertain to sample size disparity, under-representation, feature measurement noise, and group membership obfuscation—we find that the OD algorithms under the study all exhibit fairness pitfalls, although differing in which types of data bias they are more susceptible to. Most notable of our study is to demonstrate that OD algorithm bias is <u>not</u> merely a data bias problem. A key realization is that the data properties that emerge from bias injection could as well be organic—as pertain to natural group differences w.r.t. sparsity, base rate, variance, and multi-modality. Either natural or biased, such data properties can give rise to unfairness as they interact with certain algorithmic design choices.

Our work provides a deeper understanding of the possible sources of OD unfairness, and serves as a framework for assessing the unfairness of future OD algorithms under specific data-centric factors. It also paves the way for future work on mitigation strategies by underscoring the susceptibility of various design choices.

KEYWORDS

outlier detection, algorithmic fairness, data bias, sources of harm

1 INTRODUCTION

With ML claiming its unprecedented place in the society, there exists growing concern for its responsible use and potential harm to already under-served societal groups that may exacerbate the pre-existing inequities. While much work has been dedicated to measuring and mitigating unfairness of ML algorithms, they remain mostly agnostic to the underlying sources of unfairness—treating the symptom rather than the cause. In fact, the literature is quite short on understanding the underlying *sources of algorithmic unfairness*, i.e. what drives unfairness to emerge in the first place.

Data is widely acknowledged as the key influencer of the fairness of the algorithms that are trained on it [4, 28]. However, what kind of data-centric factors give rise to certain algorithmic behavior is not well understood. Moreover, the community has mainly focused on supervised ML while the fairness of unsupervised algorithms such as for outlier detection (OD) has attracted significantly less attention, despite the numerous applications and punitive decision-making scenarios that OD algorithms are employed in, such as fraud detection, policing and surveillance, to name a few.

In this paper, we aim to contribute to a deeper understanding of the possible sources of algorithmic unfairness for unsupervised OD. Specifically, we empirically investigate the role of various types of data bias as potential contributors to OD unfairness. Our findings highlight the shortcomings of several established OD algorithms in the presence of four different carefully injected data biases in a controlled simulation framework. Data biases that we study are namely, group sample size disparity, target under-representation, feature measurement noise, and group membership obfuscation.

In addition, we identify certain data-centric properties—in particular, those related to group-wise differences in sparsity, base rate, variance, and multi-modality—that emerge as a result of each specific data bias, making it easier to understand the interaction between such data properties and the underlying working assumptions of a given OD algorithm. Perhaps more importantly, we remark that such data properties as pertain to group differences that emerge from data bias could as well be organic, i.e. natural characteristics of the input data. Then our results, when seen through the lens of data-centric factors (and not just from the perspective of data bias), provide evidence toward moving beyond the "algorithmic bias is a data problem" debate [17].

Our work is one of the few that presents a rigorous empirical framework toward understanding the effect of various data-centric factors on algorithmic bias, and the first one specifically focusing on unsupervised OD algorithms. We summarize our main contributions as follows. Fig. 1 provides an overview.

- **Data bias as possible source of harm (§2):** We curate a list of potential data biases in the real world that may lead to OD unfairness; as pertain to group sample size disparity (inducing minority samples), target under-representation (distorting base rates), feature measurement noise (leading to extreme values), and group membership obfuscation (giving rise to within-group subpopulations/multi-modality).
- **Fairness (stress-)testing popular OD models (§3):** In extensive controlled simulation settings whereby we inject known data biases, we (stress-)test several established OD techniques (both fairness-unaware as well as fairness-enhanced) w.r.t. both fairness and performance metrics.
- **Extensive empirical analysis (§4):** Our analysis shows that all OD methods under study are susceptible to data

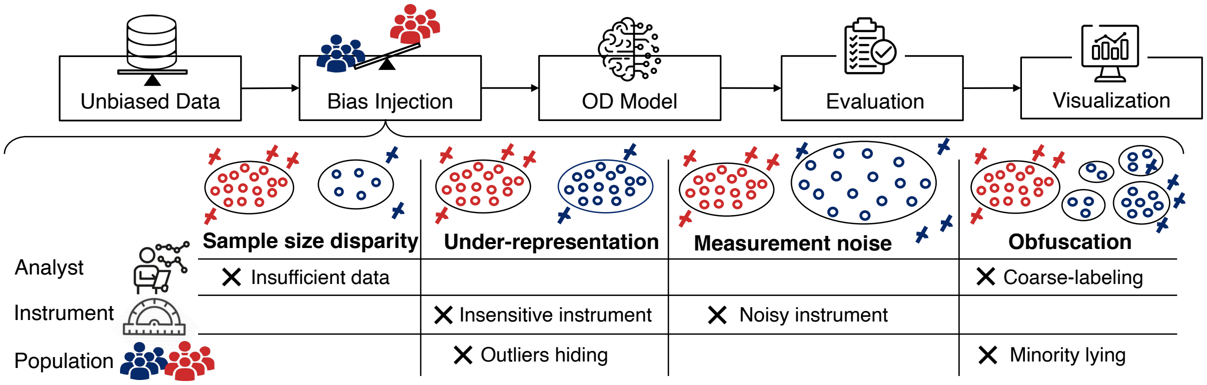

Figure 1: Overview of our study: Starting with simulated unbiased data containing outliers, we inject known types of bias into the data in a controlled setup. We then evaluate the fairness and performance of outlier detection (OD) models under various measures and report any vulnerabilities. We study four different OD models (both shallow&deep, and both fairness-unaware&fairness-enhanced), under four different types of data bias with potential implications on OD: size disparity, target under-representation, measurement error, membership obfuscation (§2.2). These data biases are graphically illustrated when data points (shown with ○) are composed of two protected groups (in color; red & blue). Outliers are shown with cross-marks (✗).

bias, although their robustness vary notably depending on the type of bias. This suggests that without a clear understanding of the type of biases a dataset may exhibit, it would be a challenge to choose an effective OD model to employ. Further, not only data bias may be a source of OD unfairness, but in some cases it also severely harms the overall detection performance.

- **Theoretical analysis (§5):** We provide detailed analyses of how certain data biases interact with the working mechanism of various OD models, leading them to incur disparate impact on different populations in the data.
- **Evidence to move beyond "(OD) algorithm bias is a data problem" (§5):** A key understanding derived from our study is that what drives OD algorithm bias is mainly the misalignment between an algorithm's working assumptions/design choices and certain input data characteristics; e.g. whether the detector respects density variability in the feature space or is susceptible to outlier masking (the notion that outliers get hidden when clustered). Crucially, while such data-centric factors may arise as a result of data bias, they could also simply be organic. This implies that OD algorithm bias can arise due to reasons beyond data bias.

Accessibility and Reproducibility: All code and datasets are open-sourced at https://anonymous.4open.science/r/ODbias.

2 DATA BIAS: TYPES AND SOURCES

2.1 Preliminaries

In general, algorithmic decisions could be punitive (e.g. imprisoning) or assistive (e.g. loan approval). Our work focuses on the former, considering the scenarios where outlier detection (OD) is often applied on population data to flag risky individuals.

For simplicity, we represent a population as composed of two protected groups, associated with a sensitive attribute $G \in \{a, b\}$. We assume group b is the underprivileged. We denote by $\mathbf{X} \in \mathbb{R}^d$ and $Y \in \{0, 1\}$ the sets of input features and the target, respectively. In OD scenarios Y translates to *true risk*, and the task is to effectively flag the individuals with $Y = 1$. Let $S \in \mathbb{R}$ indicate the output outlierness score of an OD algorithm and $O \in \{0, 1\}$ is the decided outlier label based on S given a threshold.

Given the shorthand notation, the *base rate* (also, prevalence) of a group is defined as $br_G = P(Y = 1|G)$. On the other hand, the *flag rate* of a group by a detector is given as $fr_G = P(O = 1|G)$.

2.2 Four Types of Data Bias

Based on a survey of the algorithmic fairness literature that covers possible sources of algorithmic harm [3, 28, 38, 42], we curated a list of data biases most relevant to OD applications in the real world. We provide an overview of four such biases in the context of OD as follows. We also discuss possible sources—the data collector, the measurement instrument or the population—that may drive the presence of each type of bias in the data. Fig. 1 presents a flowchart of our work, along with a graphical illustration of the data biases discussed in this section.

2.2.1 Group sample size bias:

This type of bias reflects the real-world scenarios with data scarcity for the underprivileged subpopulation (group b in our study). In effect, group b with a smaller size constitutes the statistical minority samples in the dataset. It is easy to see that OD would be susceptible to this bias as its goal is identifying outliers, i.e. statistically rare instances in the data.

The presence of sample size bias can be attributed to the data collector whom might have failed to collect enough samples for some group(s). It may also be driven by the population itself, where some groups are in minority by nature.

2.2.2 Target under-representation bias:

This type of bias is similar to the sample size bias, but only impacting the *positively-labeled* (i.e. target) individuals in the underprivileged group b. In particular, the rate of positive instances (i.e. the base rate) appear smaller in group b than that in group a. In effect, base rates are purported and the assumption of equal base rates across groups may no longer hold.

This bias could stem in the real world from the measurement instrument doing a poor job in "sensing" such individuals, as well as from the fact that such individuals may be better at "hiding" from being sampled. Alternately, the base rates may simply be different between the groups by nature.

2.2.3 **Feature measurement or response bias:** This type of bias is reflective of systematically erroneous or noisy measurements in the real world associated with individuals from the underprivileged group. Such errors and noise may inflate the variance of certain features, and thereby increasing the propensity of extreme values. Extreme values appearing as outliers, in turn, makes OD susceptible to this type of data bias.

The measurements for some group(s) may also display systematic under- or over-estimations for certain features. Consider for example the number of (re)arrests when used as a feature for risk assessment in various domains, which could be over-represented for African-Americans due to racial disparities in arrests [35]. Another example is the systematically under-estimated SAT scores for African-American students due to implicit biases in the questions that lead to racial disparities in outcomes [32].

The source of this bias is often the measurement instrument (such as questions in the survey or test, the camera, the lab test, etc.) which, for various reasons, is not well tuned to measuring underprivileged individuals accurately. In other cases, the source could be the population itself due to self-reporting; for example, when surveys are used to collect the measurements, some group(s) may provide more erroneous or noisy responses.

2.2.4 **Group-membership obfuscation bias:** This bias mimics the real world scenarios in which some underprivileged individuals misreport their group membership for various reasons including fear of disclosure or discrimination [42]. In addition to obfuscating race/gender/etc., we expect them to also alter several of their demographic/proxy features that correlate with the sensitive attribute, to camouflage and better align with this obfuscation. For example, individuals who falsify race may also misreport their country of origin, often trying to resemble in some aspects to those individuals in privileged groups. While obfuscation may occur in all groups, we assume it to be more prevalent for the underprivileged.

This bias, stemming from the population forging various feature values, induces "fragments" in the data, i.e. subgroups within a group. In the presence of obfuscation, an underprivileged group would break into smaller subgroups, resembling other groups in some ways but not others. This can be considered as heterogeneity or multi-modal distribution within the group. Interestingly, such heterogeneity, i.e. a group composed of various different subpopulations, could be an organic property (e.g. in a large country like India, not all Indians are alike). In such a case, it can be seen as a labeling misstep attributed to the data collector to file multiple heterogeneous subgroups under a single sensitive attribute value.

Remarks. The four data biases we focus on in this study are not necessarily comprehensive. While there may be other OD-relevant data issues that may trigger unfair OD outcomes, there are also other known data biases that are not applicable in OD settings. For example, the problem of "tainted" (historical) labels does not apply, since unsupervised OD algorithms do not train on Y. Moreover, train/test data distribution mismatch problem is not a concern, as we consider transductive OD wherein outliers are to be detected in a given dataset, with test data being the same as train data. While all possible sources of algorithmic harm are not fully understood, it is our plan to expand our current list in the future.

3 OD UNDER DATA BIAS: SANDBOX SETUP

Our goal is to study the effectiveness of OD algorithms under counterfactually injected bias into the simulated unbiased data. Thus, we restrict our study to simulated data. This choice is intentional, because real-world datasets lack an in-depth understanding of whether they exhibit any data bias or what types of bias(es) they may exhibit if any. In this regard, our work parallels the study by Akpinar et al. [3] which performed similar fairness and fidelity analyses on simulated controlled environments for classification.

In this section, we present the details of our (unbiased) data simulation (§3.1), bias injection steps (§3.2), the OD models under study (§3.3), and the evaluation metrics (§3.4).

3.1 Data Simulation

To ensure we start with an unbiased dataset (into which we will inject specific, known biases), we simulate a population with the input features and the target described as follows.

The individuals are represented in a feature space consisting of three types of variables; $\mathcal{X} = \mathcal{X}_g \cup \mathcal{X}_c \cup \mathcal{X}_o$, where

- $X_g \in \mathcal{X}_g$ denotes the set of *proxy* variables that correlate with Group membership G, but not with the target Y;
- $X_c \in \mathcal{X}_c$ depicts the "Culprit" or *incriminating* variables that are relevant (to the OD task) and hence correlate with or are reflective of the target Y; and
- $X_o \in \mathcal{X}_o$ capturing the *non-incriminating* Occlusion (or irrelevant) variables that neither correlate with G nor Y. The inclusion of such attributes make the OD task more realistic and non-trivial, as outliers often hide within feature *subspaces*, i.e. they stand out only w.r.t. a few relevant (but not all) features (in this case X_c).

Note that the culprit variables associate with true risk (i.e. the target) by design, while the proxy variables do not. This implies that we assume *equal base rates* across groups in the unbiased dataset. As we will discuss later, this may not always be the case in real world OD settings, where unequal base rates across groups (e.g. Internet crime propensity by ethnicity) may exist by nature.

Definition 1 (Fair Outlier Detection). *In terms of X_g, X_c and X_o, the output of an OD algorithm is considered fair as long as*

$$P(O = 1|X_c, X_g, X_o) \ = \ P(O = 1|X_c) \ , \tag{1}$$

that is, when the assigned outlier labels are independent of group membership/proxy variables as well as irrelevant features, given the incriminating variables.

To stress-test the fairness of OD algorithms in a controlled setting, we simulate an equal number of (1000) samples per group, and also set equal base rates, $br_a = br_b$ at 0.05 or 0.1, since outliers are rare. We use Gaussians to simulate the inliers, with group-wise means respectively at 5 and 20 for $\mathcal{X}_g$, and zero-mean for $\mathcal{X}_c$, both with unit standard deviation. $\mathcal{X}_o$ is drawn from a standard Gaussian uniformly at random.

We create two separate datasets by injecting **clustered** (repetitive or collusive) and **scattered** outliers, respectively. Clustered outliers have the same distribution for $\mathcal{X}_g$ and $\mathcal{X}_o$ as with the inliers, but in $\mathcal{X}_c$ they are drawn from a Gaussian with a higher mean of 3. Scattered outliers are created by randomly sampling a subset of the dimensions in $\mathcal{X}_c$, and inflating the variance (originally, 1) by a factor of $\{3, 6, 9, 12, 15\}$ chosen uniformly at random per outlier.

Finally, we set an equal number of dimensions (5) for $\mathcal{X}_g$, $\mathcal{X}_c$, and $\mathcal{X}_o$. While it remains a valuable direction to investigate the severity of unfairness that data bias may drive as a function of the relative number of the proxy and irrelevant variables to the number of incriminating variables, we focus on unfairness for the case when they all have equal proportions in the feature space.

3.2 Data Bias Injection

The crux of our sandbox is the study of OD algorithms under different types of known data biases, as discussed in §2. We describe the steps for injecting our unbiased data with each bias type as follows. For brevity, additional details are given in Apdx. §A.

1. **Group sample size bias:** With probability β_s, we independently exclude samples from the dataset where $G = b$. This leads to sample size disparity between groups.
   We inject varying degrees of sample size disparity using $\beta_s \in \{0.01, 0.05, 0.10, 0.2, 0.4, 0.6, 0.8\}$.
2. **Target under-representation bias:** With probability β_u, we independently exclude samples where $G = b$ *and* $Y = 1$. This is sample size bias impacting only the positively-labeled (i.e. target) individuals in the underprivileged group. We study bias $\beta_u \in \{0.01, 0.05, 0.10, 0.2, 0.4, 0.6, 0.8\}$.
3. **Feature measurement or response bias:** To reflect measurement noise, we inject variance-shift; where we inflate (i.e. multiply) the variance of the distributions of $\mathcal{X}_g$ and $\mathcal{X}_c$ for $G = b$ by a factor of β_v. To ensure that the supports of the group-wise distributions remain separate after variance inflation, we set the means of $\mathcal{X}_g$ to 5 and 20 for group a and b, respectively. Similarly, to be able to distinguish outlier-vs-inlier distributions upon inflation, we set the means of $\mathcal{X}_c$ to 0 and 10 for inliers and outliers, respectively. We vary $\beta_v \in \{0, 0.05, 0.1, 0.2, 0.5, 1, 2, 4, 6\}$.
   To mimic systematic over-estimation of task-relevant variables, we also experiment with mean-shift bias, where the mean of $\mathcal{X}_c$ is shifted additively for $G = b$ by $\beta_m \in \{0, 2, 4, 6, 8\}$.
4. **Group-membership obfuscation bias:** With probability β_g, we flip/swap the group membership of individuals with $G = b$ to $G = a$. Note that following un-awareness, OD models do not use the group membership indicator/variable G for detection. Thus, for individuals in $G = b$ whose membership has been flipped, we also swap/draw a random subset of their feature values in $\mathcal{X}_g$ from the distributions of $G = a$, which alters the input data.
   This bias mimics the real world scenarios in which some underprivileged individuals misreport their group membership for various reasons including fear of disclosure or discrimination [42], in addition to altering several of their demographic/proxy features in $\mathcal{X}_g$ to better align with this obfuscation. While obfuscation may occur in both groups, we assume more prevalent obfuscation in the underprivileged group; setting the rate for $G = a$ to be zero for simplicity, and performing our study as β_g varies for $G = b$, in $\{0.05, 0.1, 0.15, 0.2, 0.3, 0.4\}$.

3.3 Outlier Detection Models

In this work, we aim to stress-test the performance and fairness of OD algorithms in the face of data bias. There is a long list of algorithms for OD [2], including the modern deep neural network models [30]. The vast majority of these do not consider fairness, while only recently a handful of fairness-enhanced OD algorithms are proposed (see §6).

To be representative of the literature while keeping the computational effort of our study feasible, we consider four OD models comprising both "shallow"&deep, and both fairness-unaware&fairness-enhanced detectors: Local Outlier Factor (**LOF**) [7], Isolation Forest (**iForest**) [24], **DeepAE** [47], and **FairOD** [36]. LOF and iForest are two popular shallow techniques that have been shown to be most effective on benchmark evaluations [14]. Those two are also *mechanistic*, i.e. directly model/define what an outlier is[1]. On the other hand, DeepAE and FairOD are *learning-based* and leverage deep neural networks with end-to-end learnable parameters. Moreover, FairOD is *fairness-enhanced* while others are standard detectors. We present more details as follows. (Further details on LOF and iForest can be found in Apdx. §D.2, based on which we theoretically analyze these mechanistic OD models later in §5 and Apdx. §D.3.)

3.3.1 Local Outlier Factor.

LOF outlier score is based on the reachability distance of a point to its k nearest neighbors (NN) relative to those distances for its NNs. As its name suggests, LOF evaluates a point w.r.t. the *local density* in the vicinity it resides.

3.3.2 Isolation Forest.

iForest makes random threshold cuts sequentially on features chosen at random, thus building an ensemble of extremely randomized trees, and considers the average number of steps required to *isolate* a point from others as its outlier score.

3.3.3 DeepAE.

Based on a deep autoencoder architecture, DeepAE employs *compression* followed by decompression of the data points, where the reconstruction error is taken as the outlier score. The working assumption is that the majority of the data (i.e. inliers) exhibit patterns which can be compressed well while minimizing the total (reconstruction) loss, while outliers that do not obey such patterns receive poor reconstruction.

3.3.4 FairOD.

Finally, the fairness-enhanced FairOD also uses a deep autoencoder as a base model but enhances its loss objective with two additional terms for fairness regularization; one enforcing statistical parity and another toward achieving a heuristic approximation of the equality of opportunity (a.k.a. recall or TPR parity).

Hyperparameter (HP) tuning. OD algorithms come with HPs; e.g. number of nearest neighbors k for LOF and many others for deep models including architectural (depth and width), regularization (e.g. dropout rate) and optimization (e.g. learning rate) HPs. Critically, OD model performance is quite sensitive to HP choices [12, 26], which is nontrivial to set without any labeled data.

[1] For LOF, a point with larger reachability distance than its neighbors is an outlier. For iForest, an outlier is a point that can be isolated with few randomized axis-splits.

Table 1: Qualitative summary of results for fairness stress-testing of various OD algorithms (columns) when different type data bias (rows) is applied on group *b* with clustered outliers. Shades of colors red and blue depict the degree of *unfairness*, when disproportionately inflicted on group *a* or group *b*, respectively; blank implying no notable difference. Arrow counts reflect relative change in overall detection *performance*; no arrows implying no notable change.

Bias type / OD model	LOF	iForest	DeepAE	FairOD
Size disparity (Fig. 2)	🟥 ↓↓	🟦 ↓↓↓	🟦	🟦
Under-repres. (§C.1.2)	🟦	🟦	🟦	🟦
Meas. noise (§C.1.3)		🟦	🟦	
Obfuscation (§C.1.4)	🟦 ↓↓	🟦	🟦 ↓	🟦

Table 2: Qualitative summary of results for fairness stress-testing of various OD algorithms (columns) when different type data bias (rows) is applied on group *b* with scattered outliers. Shades of red and blue depict the degree of *unfairness*, if disproportionately inflicted on group *a* or group *b*, respectively; blank implying no notable difference. Arrows reflect the relative change in detection *performance*; no arrows implying no notable change.

Bias type / OD model	LOF	iForest	DeepAE	FairOD
Size disparity (§C.2.1)	🟥 ↓	🟦 ↓↓	🟦	🟦
Under-repres. (§C.2.2)	🟦	🟦	🟦	🟦
Meas. noise (§C.2.3)	🟦 ↓	🟦 ↓↓	🟦 ↓↓	🟦 ↓↓
Obfuscation (§C.2.4)	🟦 ↓	🟦 ↓	🟦 ↓	🟦 ↓

Notably, we search for the *best HP configuration* on each testbed, to prevent the situation where poor HP setting becomes a possible confounding source of unfairness. In other words, we carefully tune the HP(s) on the overall input population (without considering group memberships), to be able to attribute observed unfairness solely to (injected) data bias, rather than to poor HP tuning.

For standard algorithms LOF, iForest and DeepAE, we set HPs that yield the highest overall performance, and for fairness-aware FairOD we pick the HPs that yield the best performance-fairness trade-off. Details on HP configurations for each model is given in Apdx. §B. Interestingly, even under optimal HPs, we observe negative implications of data bias not only on fairness but also on detection performance, as we present results shortly in §4.

3.4 Evaluation Metrics

We evaluate detection performance by AUROC; area under the ROC curve as well as F1; the harmonic mean of Precision and Recall. AUROC quantifies the overall ranking, while F1 requires a threshold on the outlier scores. We set a threshold on S to obtain as many flagged outliers with $O = 1$ as number of true outliers with $Y = 1$.

We evaluate fairness based on *group-wise ratios* w.r.t. (*i*) positive or flag rates (FR), (*ii*) true positive rates (TPR), (*iii*) false positive rates (FPR) and (*iv*) positive predictive values (PPV, a.k.a. Precision).

When FR ratio $\frac{fr_a}{fr_b}$ is equal to 1, demographic or statistical parity is satisfied. One caveat is when group base rates differ, i.e. $br_a \neq br_b$. Then, it is suitable to measure bias amplification [22] as the ratio between FR ratio and the ground truth base rate ratio. For all other ratios, the ideal/unbiased value is 1. Unequal rates imply over/under-policing in monitoring for punitive OD settings.

For shallow and deep models, we report results averaged over 10 and 5 independent simulation runs, respectively, as the latter take longer to train and tune (more HPs).

4 OD ON BIASED DATA: EMPIRICAL FINDINGS

Starting with the unbiased datasets containing an equal number of samples in each group as well as equal group base rates, we create different biased datasets by varying β_s, β_u, β_v (or β_m), and β_g, depending on the type of bias being injected. Note that the bias is always injected on group b. Each model's hyperparameters are tuned as described in §3.3 for each resulting dataset separately.

Tables 1 and 2 (best in color) provide a qualitative summary of our results across OD models and bias types for datasets with *clustered* and *scattered* outliers, respectively. We discuss several of our notable findings in this section, while the detailed results are given in Apdx. §C for brevity.

Overall, as depicted by the varying shades of blue, we see that the bias-injected group b is impacted disproportionately across models and bias types (with a couple of exceptions), although the severity of unfairness against b varies. Moreover, data bias impacts not only fairness but also detection performance of OD models; in fact, there are cases when performance drops considerably, as depicted by downward arrows.

It is interesting to note the stark difference between the detectors in regard to susceptibility to different biases. For example, LOF is most susceptible to Obfuscation in the clustered-outliers setting and to Measurement bias for scattered outliers. iForest is susceptible to Sample size disparity, while DeepAE is most sensitive to Obfuscation in both settings. Fairness-enhanced FairOD is no exception; as it remains comparably brittle under Obfuscation. These suggest the lack of a "winner" detector.

In addition, we find that the models behave quite differently against a certain bias depending on the dataset characteristics. Notable is the Measurement bias, where models are fairly robust when outliers are clustered, which however lead to considerable unfairness as well as performance drop on datasets with scattered outliers.

We present a detailed analysis of the findings in the following subsections, respectively for each type of data bias.

4.1 Group sample size bias

When samples from group b are dropped at random to inject size disparity, the inliers and outliers both sparsify relative to group a. We find that OD models react to density differences in the feature space differently. The stark contrast can be explained based on the modeling assumptions of these models, as they interact with density variation (see §3.3).

As shown in Fig. 2, LOF's flag rate for group b *drops*, which in turn disadvantages group a with increased FPR and decreased Precision (top row). In contrast, iForest behaves in the *opposite* fashion, disadvantaging group b with significantly higher flag rate, higher FPR and lower Precision (2nd row). This contrast is due to these models treating density locally or globally. LOF evaluates outlierness *locally*, by comparing points to their neighbors. When group b inliers sparsify, the clustered b-outliers "hide" better due to masking [18]. Isolation-based iForest, on the other hand, can more quickly isolate now-globally-sparser points in group b, overly

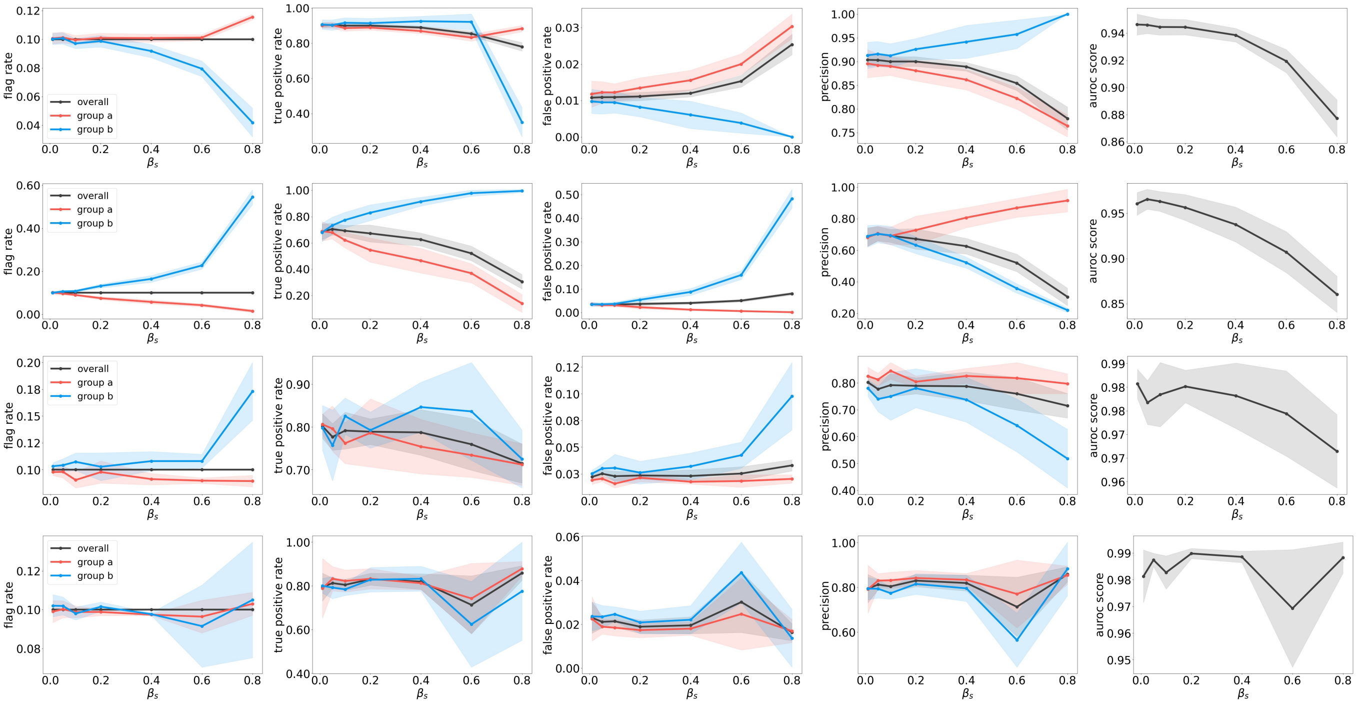

Figure 2: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under sample size bias on *clustered* outliers.

flagging them, hence maintaining high TPR, but larger FPR and lower Precision. iForest's brittleness is also evident from its overall performance falling more drastically with increasing bias.

Among the deep models, DeepAE (3rd row) behaves similar to iForest in terms of group fairness, although through a different mechanism. Since subsampling group b renders them rare *minority* samples, their impact on the total loss of the compression-based DeepAE diminishes. As a result, their poor reconstruction reflects as higher flag rate, larger FPR and lower Precision. As these group-wise differences are less extreme than for iForest, DeepAE overall detection performance remains relatively stable. Finally, we find FairOD (bottom row) to be robust against size disparity, where it achieves statistical as well as TPR parity through explicit optimization. The other quantities (FPR and Precision) follow suit, with negligible impact on performance.

These findings on clustered outliers continue to hold on scattered outliers as given in Apdx. §C.2.1 Fig. 11.

4.2 Target under-representation bias

The algorithm behaviors change considerably when exposed to the under-representation bias, where we drop only the target (positive) samples from group b. Results are given for all models in Apdx. §C.1.2 Fig. 6 and Apdx. §C.2.2 Fig. 12 in the clustered and scattered outliers settings, respectively.

Note that dropping outlier samples from b renders group a's observed base rate higher. With a higher base rate, group a's clustered outliers are slightly masked for LOF, which reduces group a's flag rate and TPR, while increasing those for b. FPR increases for both groups, proportionate to the overall increase. The masking effect goes away when outliers are scattered, with no notable TPR and FPR difference between groups. However, Precision is specifically lower for group b in both cases, mainly because b's flag rate is larger than its base rate. iForest, on the other hand, is robust to masking as it can find cuts in the feature space that isolate the outlier clusters at once, with higher TPR but lower Precision for group b.

In fact, lower Precision (due to higher flag rate than base rate) for group b is a common trend across all models, including the compression-based DeepAE and FairOD. Group b outliers continue to stand out for both models as dropping few outlier samples does not change the bulk of the data and hence the compression quality of group b's inliers. However, because group a's base rate and hence frequency is higher, its flag rate is relatively lower. This translates to a higher flag rate for b at the cost of lower Precision. These results showcase the vulnerability of all OD models in our study in the face of unequal base rates.

We see that the overall detector performances remain quite robust against this type of data bias, in both settings with clustered and scattered outliers, likely because the overall data distribution is not impacted much, especially compared to the other types of data biases we studied.

4.3 Feature measurement bias

4.3.1 **Variance shift for measurement noise.** In the clustered setting (see Apdx. §C.1.3 Fig. 7), feature variance loosens the clusters among both inliers and outliers in group b. As a result, LOF ranks group a outliers strictly above those from group b as they showcase a starker contrast to their relative inliers, while still retaining the ability to correctly flag almost all true outliers. iForest is similarly robust with only slightly higher FPR and slightly lower TPR and Precision for group b than for a, while retaining high overall performance. Moreover, both DeepAE and FairOD maintain high performance as the smaller outlier clusters do not compress as well as the inliers despite higher variance. As such, deep models are similarly robust with no notable disparity between groups.

While we find all detectors to be quite robust against feature variance disparity on clustered outliers, the results differ considerably when outliers are scattered (see Apdx. §C.2.3 Fig. 13). We show DeepAE as a representative case in Fig. 3 on clustered versus scattered outliers for comparison.

In the scattered outliers setting, feature variance loosens the distribution of group b leading to extreme-valued inliers and outliers that are both the at the "skirts" of the distribution. As a result, as group b data sparsifies, LOF tends to blur the inliers and outliers at the periphery, with higher FPR, and lower TPR and Precision for group b. Performance also drops accordingly with increasing bias levels. Results are similar for iForest, which overly flags the now-sparser group b samples. Both deep models are also sensitive in this case, yielding unfair outcomes for group b along with reduced performance. While clustered outliers, despite larger variance, form a separate modality in the data that is hard to compress (hence easier to detect), scattered outliers are harder to detect as they are more similar to the inliers at the outskirts of the inlier distribution.

The different behavior of models between the two settings demonstrates that unfairness may arise from the interaction between the data distribution, the type of bias, as well as the model assumptions.

4.3.2 **Mean shift for over-estimation.** We find that the OD models are *not* impacted by mean-shifting group b, provided that the groups are sufficiently apart along the proxy variables. Specifically, the shift does not impact the locality of points for LOF, the axis-cuts of iForest that isolate the outliers simply shift accordingly once the groups are cut separate along the proxy variable axes, and the compression-based deep models remain intact when one of the modalities shifts in the feature space. Results are shown for LOF and iForest for brevity in Apdx. §C.1.3 Fig. 8 and Apdx. §C.2.3 Fig. 14 in the clustered and scattered outliers settings, respectively.

4.4 Membership obfuscation bias

Different from earlier bias scenarios in which the data collector or their instrument may be inflicting the bias, here the individuals in the population themselves bias the data distribution. They do so by inducing mixed-up subpopulations, which resemble in some parts to group b and in other parts disguising as group a. As the obfuscation variables are randomized, this leads to *multi-modality* within group b, comprising many small subpopulations. Results can be found in Apdx. §C.1.4 Fig. 9 and Apdx. §C.2.4 Fig. 15 in the clustered and scattered outliers settings, respectively.

As shown in Fig. 4 (top), LOF is extremely sensitive to this type of bias as it mistakenly flags more and more of those small micro-clusters in group b. In fact, it seizes to flag any group a instances beyond a certain bias level as group b breaks down into more subpopulations. This leads to reduced TPR for both groups, and high FPR and low Precision for group b. LOF's overall performance also drops with increasing levels of bias. iForest, on the other hand, is relatively more robust because it is *subspace*-based; that is, it is able to isolate outliers as long as it makes axis-cuts on the incriminating variables. Put differently, as long as it does not solely pick proxy variables, it does not wrongly flag group b micro-clusters. The probability of this event is low, hence the slight unfairness against group b, with high overall performance retained.

Similar to LOF, DeepAE is sensitive to the presence of small micro-clusters as more variability in data patterns imply harder compression. Thus its overall performance drops, with higher flag rate, higher FPR and lower Precision for group b. Higher FPR and lower Precision for group b are common across models including FairOD, as shown in Fig. 4 (bottom), which better balances flag rate and TPR rates between groups thanks to explicit regularization.

5 ALGORITHMIC BIAS IS NOT MERELY A DATA (BIAS) PROBLEM

When data that is comprised of multiple protected groups is inflicted with data bias on some but not all group(s), it often leads to differences in the observed data distributions between the groups. While this can be interpreted as a data bias problem, it is important to note that differences in data distributions between groups may as well be *natural.* In fact, assuming the data is not stripped off of all the proxy variables (which is unlikely), it is perhaps easier to imagine that different groups would follow different data distributions organically, at least with respect to the proxy variables.

In the following subsections, we aim to make connections between specific data properties that emerge as a result of bias and how such properties may also appear in the real world organically. These connections support the argument toward moving beyond the "algorithmic bias is a data (bias) problem" debate [17]. That is, algorithmic bias can arise solely from the interaction between modeling assumptions and certain properties the input data exhibits, without the data being necessarily inflicted with any bias.

In a nutshell, we argue that group-wise differences in certain data properties (as pertain to sparsity, prevalence or base rate, variance, and multi-modality) — *either induced by data bias or exhibited naturally* — could result in unfair outcomes. Whichever the source may be, it is useful to understand which OD algorithms are more prone to unfairness in the face of such group-wise differences.

Theoretical Analysis. We mathematically show the (in)sensitivity of the mechanistic (i.e. non-learning based) models, LOF and iForest, in the presence of sparsity difference and multi-modality in the clustered outliers setting (Propositions 1–4). Analysis can be repeated for scattered outliers, e.g. Proposition 5. We refer to Apdx. D for preliminaries, notation, and the detailed proofs.

5.1 Group Sample Size Bias mimics Sparsity Difference between Groups

In injecting group sample size bias, the process of downsampling data from one group creates a population that is sparsely sampled from its underlying manifold. This induces density variation in the feature space where one group's distribution is relatively sparser (see Fig. 1). As presented in §4.1, we observed that this property makes isolation-based iForest quite brittle, as it tends to more easily (yet mistakenly) isolate the samples in sparser regions.

However, it may as well be natural that different groups exhibit different distributions with unequal sparsity. This would be especially realistic for proxy variables. For example hair length, a proxy for gender, could follow distributions with varying density between

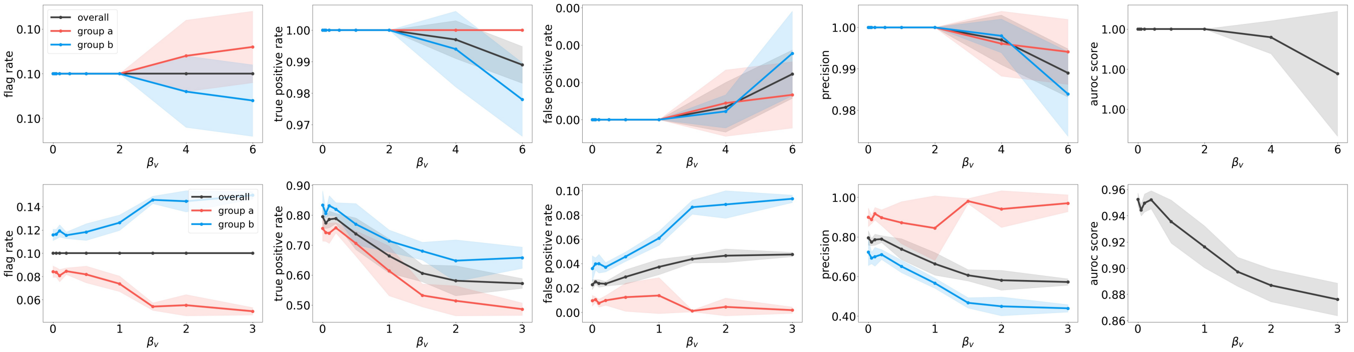

Figure 3: (best in color) Group-wise fairness metrics and AUROC for DeepAE under feature measurement bias (variance shift) on (top) *clustered* and (bottom) *scattered* outliers.

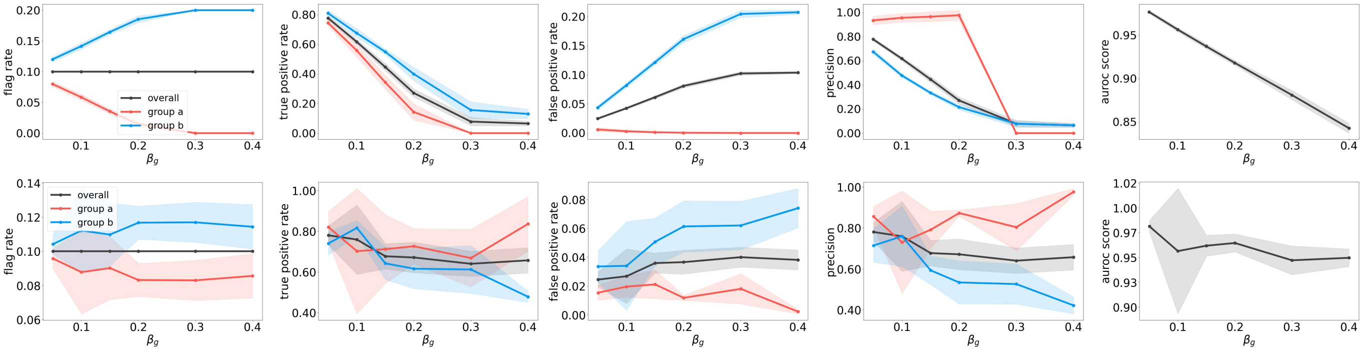

Figure 4: (best in color) Group-wise fairness metrics and AUROC for (top) LOF and (bottom) FairOD under membership obfuscation bias on *clustered* outliers.

groups. A similar argument can be made for income, a potential proxy for race, with different group-wise density distributions.

Note that in group size bias injection, down- or sparsely-sampling group b's distribution, which is the *same* as group a's, induced not only size disparity but also sparsity disparity. Here we argue that OD models could remain prone to producing unfair outcomes when group a and b exhibit *different* density distributions even in the *absence* of any group size disparity.

Proposition 1*: In the clustered outliers setting, let groups a and b have equal size $n_a = n_b$ and equal base rate $P(Y = 1|a) = P(Y = 1|b)$. Let d and D denote intra-group distance between inlier pairs (same for outlier pairs) for group a and b, respectively, where $D > d$ as group b is sparser. Also denote by Δ_a and Δ_b the average distance between outliers and inliers in each group (See Fig. 17). Then, assuming LOF hyperparameter k is set s.t. $k > n_a \cdot P(Y = 1|a)$ and $\Delta_a = \Delta_b$, LOF tends to assign higher scores to group a-outliers, increasing flag rate $P(O = 1|a)$. Further, when $\Delta_b \approx D$, LOF score of b-outliers is ≈ 1, i.e. close to inlier scores, leading to low TPR for group b due to masking.*

In plain words, in the presence of group-wise sparsity variation, the proposition states that the local reachability distance (i.e. LOF score) of a-outliers tends to be larger since their inlier neighbors' reachability distance is smaller when/because group a is denser. Further, as group b continues to sparsify such that the gap between inlier and outlier clusters of group b shrinks, b-outliers get harder for LOF to distinguish from inliers, decreasing TPR. Proposition 5 in Apdx. D.3.7 shows a similar result for LOF for scatter outliers.

Proposition 2: *In the clustered outliers setting, let sp denote an iTree's split, and $P(x_a < sp < x_b)$ denote the probability for a split to occur between points x_a and x_b. Let groups a and b have equal base rate $P(Y = 1|a) = P(Y = 1|b)$ and equal average distance between outliers and inliers $\Delta_a = \Delta_b$. As group b is sparser, $d < D$ denotes the intra-group distances between inlier and outlier pairs for groups a and b, respectively (See Fig. 20). Let $\{\mathcal{O}_a, \mathcal{I}_a\}$, $\{\mathcal{O}_b, \mathcal{I}_b\}$ represent the outliers, inliers sets for groups a and b. Then, for $o_b, p_b \in \mathcal{O}_b$, and $o_a, p_a \in \mathcal{O}_a$, we have $P(p_b < sp < o_b) > P(p_a < sp < o_a)$, indicating an iTree is more likely to split among group-b outliers. The difference between $P(p_b < sp < o_b)$ and $P(p_a < sp < o_a)$ becomes larger when iTree is built in higher dimensions. In addition, for $q_b \in \mathcal{I}_b$, $q_a \in \mathcal{I}_a$, $P(o_b < sp < q_b) = P(o_a < sp < q_a)$, i.e. an iTree is equally likely to split between the inlier and outlier clusters.*

iForest assigns higher outlier scores to points that can be easily isolated with few axis splits. The proposition shows that an iTree is more likely to split among the sparser b-outliers than a-outliers, yielding higher flag rate and FPR for group b. The discrepancy is more notable in higher dimensional splits. iTree is equally likely to split between the inliers and outliers for both groups. Due to this special split that isolates the clustered outliers at once, iForest can flag some fraction of a-outliers, retaining TPR for group a.

Empirically, variance shift experiments could serve as a similar scenario where samples of group b are more sparsely scattered than those of a due to inflated variance, which we observed is subject to higher FPR and lower TPR and Precision, consistent across all models (see Apdx. §C.2.3 Fig. 13). This shows one possible scenario, wherein group b organically exhibits greater sparsity, and group-wise sparsity difference leads to disparate OD outcomes.

5.2 Target Under-Representation Bias mimics Base-Rate Difference between Groups

In injecting target under-representation bias, the process of down-sampling group b outliers induces its population to have a lower observed base rate than group a. Here we argue that the protected groups may as well exhibit unequal base rates naturally, where $P(Y = 1|G = a) \neq P(Y = 1|G = b)$.

For example, it could be the case that some group(s) are more inclined to criminal activities (such as Internet crime, human trafficking, money laundering, click farming, etc.) than others by nature. In fact, this is arguably one of the key differences between punitive OD settings and assistive ML settings. While it may be more ethical to provide equal assistance (hiring, loan approval, etc.) by ML algorithms across various protected groups where there is no reason or evidence to suggest one group should necessarily be more deserving than the other, it could be appropriate to penalize various protected groups at different rates when there is evidence or reason to believe that the group base rates indeed differ.

Then, in the presence of (natural) *unequal* base rates, unfair OD outcomes observed in §4.2 continue to hold. To demonstrate this phenomenon, we repeat our experiments by varying the base rates between groups while keeping the total number of outliers (and group-wise inliers) the same. Results are shown in Fig. 5 (top) for LOF for brevity, which remain qualitatively similar to those under target under-representation bias as shown in Fig. 5 (bottom). Results for all models are in Apdx. §C.1.5 Fig. 10 and Apdx. §C.2.5 Fig. 16 in the clustered and scattered outliers settings, respectively.

5.3 Feature Measurement Bias mimics Variance Difference between Groups

Variance-shift reflecting measurement noise may change the propensity of extreme values and low-frequency observations. As studied in §4.3, this may induce unfair OD outcomes for the high-variance group especially when the outliers are scattered.

In the real world, certain features could also exhibit natural variation between groups. For example, biomarker features when used for screening may exhibit natural genetic variation by age, gender, or race. Another, classical example is the Nymwars controversy[2] in 2011, where Google, Facebook, and other tech companies aimed to block users who used uncommon (hence, presumably fake) names. While the policy has been criticized for various reasons, one could see the potential discrimination it could cause given the inherent variability in African first names.

We argue that variance difference between groups mimics the scenario when such difference is induced by additive measurement noise inflicted on some group(s) but not others. This is true for Gaussians; Gaussian noise added to a Gaussian distribution yields another Gaussian with larger variance. In fact, for any distribution from the stable distribution family, a linear combination of two distributions remains in the same distribution, only with different location and scale parameters [27]. This implies that the unfair OD outcomes observed in §4.3 would continue to hold when one group exhibits naturally higher-variance distribution for certain features.

5.4 Membership Obfuscation Bias mimics Multi-modality within Groups

Membership obfuscation – where members of a group disguising as those from another group – creates (sub)groups within groups, where each subgroup disguises somewhat differently within the proxy feature space. In effect, obfuscation fragments a population, inducing multi-modality within the group.

Arguably, members of a group may exhibit within-group heterogeneity naturally. That is, some group(s) may be composed of many more natural clusters than others. In such cases, it is perhaps a negligence of the analyst to assign them to a single broad sensitive attribute. Consider the Asian or Hispanic ethnicity; those are very coarse classifications of large populations of individuals that in fact break down to many natural, more coherent subpopulations.

When a protected group comprises multiple smaller subgroups, the risk increases that each may stand out as a minority micro-cluster. As we presented in §4.4, OD models are typically tuned to flagging micro-clusters as (clustered) outliers, leading to many false positives at the expense of low recall for the multi-modal group. In the following, we mathematically show that LOF is quite vulnerable in the presence of this data characteristic, supporting the empirical observations in Fig. 4 (top). (See all proofs in Apdx. D.3).

***Proposition 3**: In the clustered outliers setting, let groups a and b have equal size $n_a = n_b$ and equal base rate $P(Y = 1|a) = P(Y = 1|b)$. Assume group b-inliers form f smaller populations, $\mathcal{I}_b^i$, for $i \in [[1, f]]$, and group b-outliers form g smaller populations, O_b^j for $j \in [[1, g]]$. Let d and D denote intra-group distance between inlier pairs (same for outlier pairs) for group a and each sub-population in b, respectively, where $D = d$. Denote Δ_a as the average distance between outliers and inliers in group a, and Δ_b as the average distance between each subpopulation in group b for both inliers and outliers (See Fig. 19). Then, assuming LOF hyperparameter k is set s.t. $k > n_a \cdot P(Y = 1|a)$, and $\Delta_a = \Delta_b$ while $\Delta_a > d$ and $\Delta_b > D$, LOF tends to assign higher scores to group b-outliers, increasing flag rate $P(O = 1|b)$. Further, when $|\mathcal{I}_b^j| < |O_b^i| < k$, LOF score of b-inliers is larger than of b-outliers, leading to both high FPR and low TPR for group b.*

We also show via Proposition 4 in Appx. D.3.6 that iForest is relatively more robust than LOF against within-group multi-modality since, unlike LOF that relies on nearest neighbor distances in the *full* feature space, it seeks outliers in *sub*spaces. As protected groups form clusters only w.r.t. proxy features, iForest is prone to false positives when axis-splits are limited to proxy features, yet, making axis-splits along the incriminating features allows iForest to isolate true outliers and hence maintain recall.

6 RELATED WORK

A vast body of algorithmic fairness literature is on the measures and mitigation of unfairness, with less attention to the driving mechanisms and sources of unfairness. Our work focuses on the latter. Further, while most work considers supervised ML/classification, we study unfairness in unsupervised ML/outlier detection (OD).

Measuring and mitigating unfairness: Various work studied possible definitions and measures of ML fairness [5, 16, 20, 39, 41],

[2] https://en.wikipedia.org/wiki/Nymwars

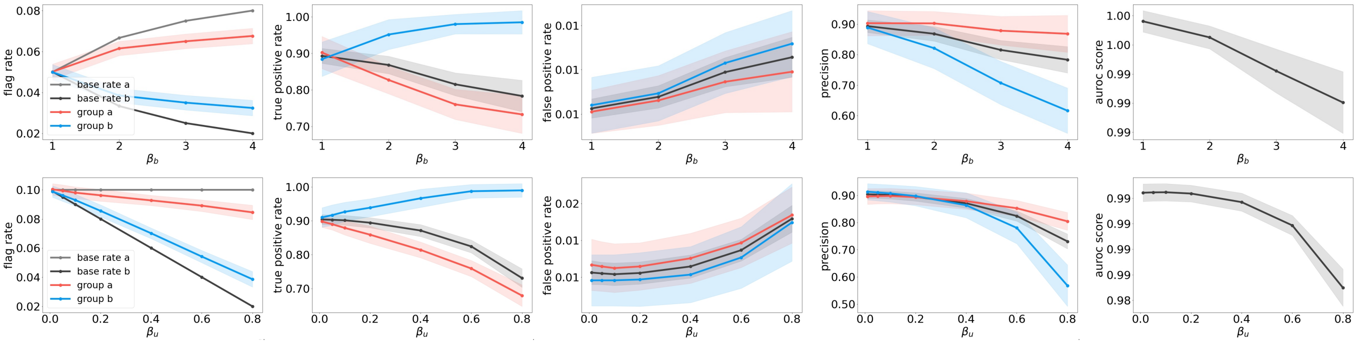

Figure 5: (best in color) Group-wise fairness metrics and AUROC for LOF under (top) unequal base rates and (bottom) target under-representation on *clustered* outliers. The results are qualitatively similar.

fairness auditing [15, 21, 34], with most emphasis on mitigation in ML [28, 33], through optimization [43], regularization [19], constraints [44], adversarial learning [11], and representation learning [45], to name a few. In contrast, only a handful of work studied auditing [10] and mitigation [1, 25, 36, 37, 46] for unsupervised OD.

Sources of unfairness: On developing a deeper understanding of what drives discrimination by ML, various work studied the impact of data collection [8], algorithmic factors [6], as well as model and data interactions [31], while others aimed to identify and list possible sources of harm [28, 38]. Our work is inspired by the testbed by Akpinar *et al.* [3] that studied the explicit impact of counterfactually injected biases on *supervised* ML models. In similar vein, our study is the first to demonstrate that popular *unsupervised* OD models are susceptible to certain forms of data bias.

7 CONCLUSION AND DISCUSSIONS

Summary. We presented a descriptive measurement study that stress-tested the fairness and performance of various OD models when exposed to certain data biases. Our analyses have been expository, unearthing the pitfalls of various algorithmic design choices as they interact with certain data characteristics—such as group-wise differences in sparsity, prevalence, variance and multi-modality. Among the findings, those that stand out are as follows.

All models (shallow/deep, fairness un/aware) are prone to yield disparate impact that is stacked against the underprivileged group. The impact varies depending on the relation between the type of data bias (and the data characteristics it induces) and the model assumptions. In our study, the shallow detectors directly model/define what an outlier is, and yield unfair results when these definitions do not align with the data. On the other hand, the deep models aim to identify what is normal/compressible and flag outliers indirectly as the samples that do not obey the normal patterns. When more than one normal is present and the normal pattern(s) are more dominant/coherent for some protected group(s) than others, these detectors also tend to produce unfair outcomes.

While our study initially looked at OD unfairness through the lens of data bias, we realized that what is really at play is certain data characteristics that do not "play well" with certain algorithmic assumptions and design choices. Therefore, it is worthy to note that while data bias may induce such data characteristics, they could also be natural. This implies that OD models can naturally fall into the same pitfalls we underscored, even when the analyst, measurement instrument and the population are not at fault of inflicting bias.

Limitations. Our paper presents a controlled measurement study, limited to simulated data. This is motivated by the fact that it is not known which type(s) of data bias, if any, real-world datasets exhibit. We studied datasets injected with one (known) type of data bias at a time. Our simulations also included simple data and outlier distributions. While simulations are an excellent way to understand the basics, the design of our simulations may not align too closely with real-world datasets, which limits the external validity of our observations. Nevertheless, even in these simple scenarios, we are able to showcase key shortcomings of various modeling choices that we argued may occur naturally in the real world.

We did not perform experiments on real-world datasets intentionally. Our aim has been to study which data bias gives rise to unfair outcomes in a controlled setup. Explaining unfairness observed in the wild by pinpointing the root-causes would be a reverse-engineering effort, which is not our intended scope. Our work lays out possible sources for investigation, although our bias list is not necessarily comprehensive. Further limitations pertain to our theoretical analyses, as discussed in Apdx. D.4.

Future directions. Our work paves the way to identifying other types of data biases that may adversely impact fair OD outcomes, and developing remediation strategies toward mitigating the potential pitfalls we underscored. We briefly discuss possible directions here, while a deeper investigation is left as future work.

Fairness interventions are often grouped as pre-, post-, and in-processing, which respectively, modify the input data, modify the output scores or decisions, and account for fairness during model training. As discussed in §5, OD unfairness can stem from algorithmic bias alone when group-wise differences are natural. Then, pre-processing strategies become voided as it is not clear how to modify organic, unbiased data. Post-processing could select different thresholds for each group separately [9, 29], where the group-specific thresholds could either be selected in a data-driven fashion, or optimized for demographic parity if it is a desired fairness metric (e.g. assuming equal base rates). Note that metrics based on true labels cannot be optimized (at least directly) due to lack of ground truth in training unsupervised OD. In-processing techniques are also limited to only enforcing demographic parity, which as we showed, remain susceptible to unfairness. A strategy that has not been applied to OD is decoupling [13, 40], where a different detector is trained for each group, while optimizing a joint loss.

We remark that post-processing and decoupling exhibit treatment disparity; they both assume it is ethical and legal to use the sensitive attribute at decision time – in particular, to select which threshold or detector to employ on a new sample. When there are differences *between* groups, coming to terms with disparate treatment might be the only way to get around disparate impact, as argued in [23]. These, however, do not address unfairness against multi-modal subpopulations *within* groups, i.e. within-group discrimination. Here, one direction is to explore clustering-based OD. Alternative could be establishing a more nuanced or granular sensitive attribute, assigning a different value for each subpopulation.

ACKNOWLEDGMENTS

This work is sponsored by the National Science Foundation IIS-2310481. Any conclusions expressed in this material are those of the authors and do not necessarily reflect the views, expressed or implied, of the funding parties.

REFERENCES

[1] Savitha Sam Abraham et al. 2021. Fairlof: fairness in outlier detection. *Data Science and Engineering* 6, 4 (2021), 485–499.

[2] Charu C Aggarwal. 2016. Outlier analysis second edition.

[3] Nil-Jana Akpinar, Manish Nagireddy, Logan Stapleton, Hao-Fei Cheng, Haiyi Zhu, Steven Wu, and Hoda Heidari. 2022. A sandbox tool to bias (Stress)-test fairness algorithms. *arXiv preprint arXiv:2204.10233* (2022).

[4] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *California law review* (2016), 671–732.

[5] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Allison Woodruff, Christine Luu, Pierre Kreitmann, Jonathan Bischof, and Ed H Chi. 2019. Putting fairness principles into practice: Challenges, metrics, and improvements. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society.* 453–459.

[6] William Blanzeisky and Pádraig Cunningham. 2021. Algorithmic factors influencing bias in machine learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer, 559–574.

[7] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data.* 93–104.

[8] Irene Chen, Fredrik D Johansson, and David Sontag. 2018. Why is my classifier discriminatory? *Advances in neural information processing systems* 31 (2018).

[9] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *ACM SIGKDD.* 797–806.

[10] Ian Davidson and Selvan Suntiha Ravi. 2020. A framework for determining the fairness of outlier detection. In *ECAI 2020.* IOS Press, 2465–2472.

[11] Pieter Delobelle, Paul Temple, Gilles Perrouin, Benoît Frénay, Patrick Heymans, and Bettina Berendt. 2021. Ethical adversaries: Towards mitigating unfairness with adversarial machine learning. *ACM SIGKDD Explorations Newsl.* 23, 1 (2021).

[12] Xueying Ding, Lingxiao Zhao, and Leman Akoglu. 2022. Hyperparameter sensitivity in deep outlier detection: Analysis and a scalable hyper-ensemble solution. *NeurIPS* 35 (2022), 9603–9616.

[13] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. 2018. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on fairness, accountability and transparency.* PMLR, 119–133.

[14] Andrew F Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. 2013. Systematic construction of anomaly detection benchmarks from real data. In *ACM SIGKDD workshop on outlier detection and description.*

[15] Gemma Galdon Clavell, Mariano Martín Zamorano, Carlos Castillo, Oliver Smith, and Aleksandar Matic. 2020. Auditing algorithms: On lessons learned and the risks of data minimization. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society.* 265–271.

[16] Pratyush Garg, John Villasenor, and Virginia Foggo. 2020. Fairness metrics: A comparative analysis. In *IEEE Big Data.* 3662–3666.

[17] Sara Hooker. 2021. Moving beyond "algorithmic bias is a data problem". *Patterns* 2, 4 (2021).

[18] Shuli Jiang, Robson LF Cordeiro, and Leman Akoglu. 2022. D. MCA: Outlier Detection with Explicit Micro-Cluster Assignments. In *IEEE ICDM.* IEEE.

[19] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *2011 IEEE 11th international conference on data mining workshops.* IEEE, 643–650.

[20] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *Advances in neural information processing systems* 30 (2017).

[21] Erwan Le Merrer, Ronan Pons, and Gilles Trédan. 2023. Algorithmic audits of algorithms, and the law. *AI and Ethics* (2023), 1–11.

[22] Nianyun Li, Naman Goel, and Elliott Ash. 2022. Data-centric factors in algorithmic fairness. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society.* 396–410.

[23] Zachary Lipton, Julian McAuley, and Alexandra Chouldechova. 2018. Does mitigating ML's impact disparity require treatment disparity? *Advances in neural information processing systems* 31 (2018).

[24] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining.* IEEE, 413–422.

[25] Haoyu Liu, Fenglong Ma, Shibo He, Jiming Chen, and Jing Gao. 2021. Fairness-aware outlier ensemble. *arXiv preprint arXiv:2103.09419* (2021).

[26] Martin Q Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. 2023. The need for unsupervised outlier model selection: A review and evaluation of internal evaluation strategies. *ACM SIGKDD Expl. Newsl.* 25, 1 (2023).

[27] Francesco Mainardi. 2007. LECTURE NOTES ON MATHEMATICAL PHYSICS: Lévy Stable Distributions in the Theory of Probability. Preliminary Version. Available online at www.fracalmo.org.

[28] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)* 54, 6 (2021), 1–35.

[29] Aditya Krishna Menon and Robert C Williamson. 2018. The cost of fairness in binary classification. In *Conference on Fairness, accountability and transparency.* PMLR, 107–118.

[30] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)* 54, 2 (2021), 1–38.

[31] José Pombal, André F Cruz, João Bravo, Pedro Saleiro, Mário AT Figueiredo, and Pedro Bizarro. 2022. Understanding Unfairness in Fraud Detection through Model and Data Bias Interactions. *arXiv preprint arXiv:2207.06273* (2022).

[32] Sarwat Amin Rattani. 2016. SAT: does racial bias exist? *Creative Education* 7, 15 (2016), 2151–2162.

[33] Charan Reddy, Deepak Sharma, Soroush Mehri, Adriana Romero Soriano, Samira Shabanian, and Sina Honari. 2021. Benchmarking Bias Mitigation Algorithms in Representation Learning through Fairness Metrics. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks.*

[34] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. 2018. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577* (2018).

[35] Cydney Schleiden, Kristy L Soloski, Kaitlyn Milstead, and Abby Rhynehart. 2020. Racial disparities in arrests: a race specific model explaining arrest rates across black and white young adults. *Child and adolescent social work journal* 37 (2020).

[36] Shubhranshu Shekhar, Neil Shah, and Leman Akoglu. 2021. Fairod: Fairness-aware outlier detection. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society.* 210–220.

[37] Hanyu Song, Peizhao Li, and Hongfu Liu. 2021. Deep clustering based fair outlier detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 1481–1489.

[38] Harini Suresh and John Guttag. 2021. A framework for understanding sources of harm throughout the machine learning life cycle. In *Equity and access in algorithms, mechanisms, and optimization.* 1–9.

[39] Zeyu Tang, Jiji Zhang, and Kun Zhang. 2023. What-is and how-to for fairness in machine learning: A survey, reflection, and perspective. *Comput. Surveys* 55, 13s (2023), 1–37.

[40] Berk Ustun, Yang Liu, and David Parkes. 2019. Fairness without harm: Decoupled classifiers with preference guarantees. In *ICML.* PMLR, 6373–6382.

[41] Sahil Verma and Julia Rubin. 2018. Fairness definitions explained. In *Proceedings of the international workshop on software fairness.* 1–7.

[42] Serena Wang, Wenshuo Guo, Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Michael Jordan. 2020. Robust optimization for fairness with noisy protected groups. *Advances in neural information processing systems* 33 (2020), 5190–5203.

[43] Violet Xinying Chen and John N Hooker. 2023. A guide to formulating fairness in an optimization model. *Annals of Operations Research* 326, 1 (2023), 581–619.

[44] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics.* PMLR, 962–970.

[45] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *ICML.* PMLR, 325–333.

[46] Hongjing Zhang and Ian Davidson. 2021. Towards fair deep anomaly detection. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency.* 138–148.

[47] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *ACM SIGKDD.* 665–674.

APPENDIX

A DATA BIAS INJECTION DETAILS

A.1 Datasets with Clustered Outliers

A.1.1 *Group sample size bias.*

Number of samples in group a: 1000
Number of samples in group b: 1000
$br_a = br_b = 0.1$
$\beta_s \in [0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8]$
Outlier counts in groups (a, b) respectively for varying β_s:
$[(100, 100), (100, 97), (100, 87), (100, 80), (100, 56), (100, 33), (100, 24)]$
Each algorithm selected hyperparameters that yielded the best performance metrics.
LOF selected hyperparameter k respectively for varying β_s:
$[240, 237, 227, 220, 196, 173, 154]$
DeepAE selected hyperparameters respectively for varying β_s:

β	HP
0.01	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.05	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.1	{num_layer: 4, input_decay: 2.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.2	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.4	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.6	{num_layer: 2, input_decay: 2.5, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.8	{num_layer: 4, input_decay: 1.0, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0.2, threshold: 1}

FairOD selected hyperparameters α and γ respectively for varying β_s:

β	HP
0.01	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1 α: 0.01, β: 0.8}
0.05	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.05, β: 0.5}
0.1	{num_layer: 4, input_decay: 2.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1 α: 0.5, β: 0.8}
0.2	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.5}
0.4	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.8}
0.6	{num_layer: 2, input_decay: 2.5, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.05, β: 0.5}
0.8	{num_layer: 4, input_decay: 1.0, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0.2, threshold: 1, α: 0.01, β: 0.8}

A.1.2 *Target under-representation bias.*

Number of samples in group a: 1000
Number of samples in group b: 1000
$br_a = br_b = 0.1$
$\beta_u \in [0.01, 0.05, 0.10, 0.2, 0.4, 0.6, 0.8]$
Outlier counts in groups (a, b) respectively for varying β_u:
$[(100, 99), (100, 95), (100, 90), (100, 80), (100, 60), (100, 40), (100, 20)]$
Each algorithm selected hyperparameters that yielded the best performance metrics.
LOF selected hyperparameter k respectively for varying β_u:
$[239, 235, 230, 220, 200, 180, 160]$
DeepAE selected hyperparameters respectively for varying β_u:

β	HP
0.01	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.05	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.1	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 0, dropout: 0, threshold: 1}
0.2	{num_layer: 4, input_decay: 2, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.4	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.6	{num_layer: 4, input_decay: 1.0, epochs: 100, lr: 0.001, weight_decay: 1e-05, dropout: 0.2, threshold: 1}
0.8	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}

FairOD selected hyperparameters α and γ respectively for varying β_u:

β	HP
0.01	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1 α: 0.01, β: 0.01}
0.05	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.2}
0.1	{num_layer: 4, input_decay: 2.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1 α: 0.2, β: 0.01}
0.2	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.8, β: 0.8}
0.4	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.5, β: 0.01}
0.6	{num_layer: 2, input_decay: 2.5, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.5, β: 0.01}
0.8	{num_layer: 4, input_decay: 1.0, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0.2, threshold: 1, α: 0.05, β: 0.01}

A.1.3 *Feature measurement bias.*

A.1.4 *Mean-shift.*

Number of samples in group a: 1000
Number of samples in group b: 1000
$br_a = br_b = 0.1$
$\beta_m \in [0, 2, 4, 6, 8]$
Each algorithm selected hyperparameters that yielded the best performance metrics.
LOF selected hyperparameter k respectively for varying β_m:
$[240, 240, 240, 240, 240]$

A.1.5 *Variance-shift.*

Number of samples in group a: 1000
Number of samples in group b: 1000
$br_a = br_b = 0.1$
$\beta_v \in [0, 0.05, 0.1, 0.2, 0.5, 1, 2, 4, 6]$
Each algorithm selected hyperparameters that yielded the best performance metrics.
LOF selected hyperparameter k respectively for varying β_v:
$[120, 120, 120, 120, 120, 130, 130, 150, 200]$
DeepAE selected hyperparameters respectively for varying β_v:

β	HP
0	{num_layer: 4, input_decay: 2.5, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.05	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.1	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.2	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.5	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
1	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
2	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
4	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}
6	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}

FairOD selected hyperparameters α and γ respectively for varying β_v:

β	HP
0	{num_layer: 4, input_decay: 2.5, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.05	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.1	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.2	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.5	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
1	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
2	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
4	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.05, β: 0.8}
6	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.8, β: 0.2}

A.1.6 *Membership obfuscation bias.*

Number of samples in group a: 1000
Number of samples in group b: 1000
$br_a = br_b = 0.1$
$\beta_o \in [0.05, 0.1, 0.15, 0.2, 0.3, 0.4]$
Each algorithm selected hyperparameters that yielded the best performance metrics.
LOF selected hyperparameter k respectively for varying β_o:
$[240, 240, 240, 240, 240, 240]$
DeepAE selected hyperparameters respectively for varying β_o:

β	HP
0	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.05	{num_layer: 4, input_decay: 2.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.1	{num_layer: 4, input_decay: 1.5, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.2	{num_layer: 4, input_decay: 1.5, epochs: 250, lr: 0.0001, weight_decay: 0, dropout: 0, threshold: 1}
0.3	{num_layer: 2, input_decay: 2, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.4	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1}

FairOD selected hyperparameters α and γ respectively for varying β_o:

β	HP
0	{num_layer: 4, input_decay: 2, epochs: 100, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.05, β: 0.01}
0.05	{num_layer: 4, input_decay: 2.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.05, β: 0.01}

β	HP
0.1	{num_layer: 4, input_decay: 1.5, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.2, β: 0.01}
0.2	{num_layer: 4, input_decay: 1.5, epochs: 250, lr: 0.0001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.05, β: 0.01}
0.3	{num_layer: 2, input_decay: 2, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.2, β: 0.01}
0.4	{num_layer: 4, input_decay: 1.0, epochs: 250, lr: 0.0001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.05, β: 0.2}

A.2 Datasets with Scattered Outliers

A.2.1 Group sample size bias.

Number of samples in group a: 1000
Number of samples in group b: 1000
$br_a = br_b = 0.1$
$\beta_s \in [0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8]$
Outlier counts in groups (a, b) respectively for varying β_s:
$[(100, 100), (100, 97), (100, 87), (100, 80), (100, 56), (100, 33), (100, 24)]$
Each algorithm selected hyperparameters that yielded the best performance metrics.
LOF selected hyperparameter k respectively for varying β_s:
$[240, 237, 227, 220, 196, 173, 154]$.
DeepAE selected hyperparameters respectively for varying β_s:

β	HP
0.01	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.05	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.2	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.4	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.6	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.8	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}

FairOD selected hyperparameters α and γ respectively for varying β_s:

β	HP
0.01	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.5, β: 0.01}
0.05	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.01}

β	HP
0.2	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.4	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.2, β: 0.01}
0.6	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.2, β: 0.01}
0.8	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.05, β: 0.01}

A.2.2 Target under-representation bias.

Number of samples in group a: 1000
Number of samples in group b: 1000
$br_a = br_b = 0.1$
$\beta_u \in [0.01, 0.05, 0.10, 0.2, 0.4, 0.6, 0.8]$
Outlier counts in groups (a, b) respectively for varying β_u:
$[(100, 99), (100, 95), (100, 90), (100, 80), (100, 60), (100, 40), (100, 20)]$
Each algorithm selected hyperparameters that yielded the best performance metrics.
LOF selected hyperparameter k respectively for varying β_u:
$[149, 145, 140, 130, 110, 90, 70]$
DeepAE selected hyperparameters respectively for varying β_u:

β	HP
0.01	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.05	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.2	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.4	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.6	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.8	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}

FairOD selected hyperparameters α and γ respectively for varying β_u:

β	HP
0.01	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.5, β: 0.01}
0.05	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.05, β: 0.01}
0.1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.5}

β	HP
0.2	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.05, β: 0.01}
0.4	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.5, β: 0.5}
0.6	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.8	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1 α: 0.8, β: 0.8}

A.2.3 Feature measurement bias.

A.2.4 Mean-shift. Number of samples in group a: 1000

Number of samples in group b: 1000

$br_a = br_b = 0.1$

$\beta_m \in [0, 2, 4, 6, 8]$

Each algorithm selected hyperparameters that yielded the best performance metrics.

LOF selected hyperparameter k respectively for varying β_m: $[20, 10, 40, 30, 10]$

A.2.5 Variance-shift. Number of samples in group a: 1000

Number of samples in group b: 1000

$br_a = br_b = 0.1$

$\beta_v \in [0, 0.05, 0.1, 0.2, 0.5, 1, 1.5, 2, 3]$

Each algorithm selected hyperparameters that yielded the best performance metrics.

LOF selected hyperparameter k respectively for varying β_v: $[10, 10, 30, 30, 70, 30, 90, 70, 140]$

DeepAE selected hyperparameters respectively for varying β_v:

β	HP
0	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.05	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.2	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
0.5	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
1.5	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}
2	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
3	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1}

FairOD selected hyperparameters α and γ respectively for varying β_v:

β	HP
0	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.8, β: 0.01}
0.05	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.2	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.5}
0.5	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.05, β: 0.01}
1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
1.5	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.5, β: 0.01}
2	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
3	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 0, dropout: 0, threshold: 1, α: 0.01, β: 0.01}

A.2.6 Membership obfuscation bias.

Number of samples in group a: 1000

Number of samples in group b: 1000

$br_a = br_b = 0.1$

$\beta_o \in [0.05, 0.1, 0.15, 0.2, 0.3, 0.4]$

Each algorithm selected hyperparameters that yielded the best performance metrics.

LOF selected hyperparameter k respectively for varying β_o: $[10, 20, 10, 10, 20, 10]$

DeepAE selected hyperparameters respectively for varying β_o:

β	HP
0.05	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.15	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.2	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.3	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}
0.4	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1}

FairOD selected hyperparameters α and γ respectively for varying β_o:

β	HP
0.05	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.1	{num_layer: 2, input_decay: 1.5, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.01, β: 0.01}
0.15	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.5, β: 0.01}
0.2	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.8, β: 0.01}
0.3	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.8, β: 0.01}
0.4	{num_layer: 2, input_decay: 1.0, epochs: 250, lr: 0.001, weight_decay: 1e-05, dropout: 0, threshold: 1, α: 0.8, β: 0.01}

B HYPERPARAMETER CONFIGURATIONS

LOF: We tune the number of nearest neighbors, k, over the candidate set $k \in \{10, 20, \ldots, 250\}$ and pick the best w.r.t. overall AUROC detection performance.

Isolation Forest: As an ensemble model of extremely randomized trees, iForest has two hyperparameters (HPs); number of trees and number of samples to construct each tree. As with most ensemble models, it has been found to be robust under a wide range of choices for these HPs. Therefore, we use the default values of 100 trees and 256 samples per tree.

DeepAE: We tune the 6 hyperparameters, picking the best among 128 candidate configurations as shown in Table 3 w.r.t. overall AUROC detection performance.

Table 3: Hyperparameter search space for DeepAE

Name	Count	Values
num_layers	2	[2, 4]
weight_decay	2	[0, 1e-5]
learning_rate	2	[1e-3, 1e-4]
num_epochs	2	[100, 250]
input_decay	4	[1.0, 1.5, 2, 2.5]
droupout_rate	2	[0, 0.2]

FairOD: Utilizing a deep autoencoder as its base model, FairOD additionally optimizes a fairness-enhanced loss; specifically with two additional terms besides the reconstruction term. Each additional term has a scalar hyperparameter associated with it, α and γ respectively, to trade-off performance/fit vs. two fairness notions (statistical parity and a heuristic approximation of equality of opportunity). On any given dataset, we use the same best DeepAE configuration for FairOD and only tune these two trade-off HPs in the loss. Specifically, we tune over $\alpha \in \{0.01, 0.05, 0.2, 0.5, 0.8\}$ and $\gamma \in \{0.01, 0.2, 0.5, 0.8\}$, and pick the configuration that yields the best (AUROC, statistical parity, equality of opportunity) outcome/triplet, with shortest distance to the ideal values of (1, 1, 1) for these three quantities.

C ADDITIONAL EMPIRICAL RESULTS

C.1 Datasets with Clustered Outliers

C.1.1 Group sample size bias. The results are presented in the main text in Fig. 2.

C.1.2 Target under-representation bias. See Fig 6.

C.1.3 Feature measurement bias. See Fig. 7 for variance shift and Fig. 8 for mean shift results (LOF and iForest only for brevity).

C.1.4 Membership obfuscation bias. See Fig. 9.

C.1.5 Unequal base rates. See Fig. 10.

C.2 Datasets with Scattered Outliers

C.2.1 Group sample size bias. See Fig. 11.

C.2.2 Target under-representation bias. See Fig. 12.

C.2.3 Feature measurement bias. See Fig. 13 for variance shift and Fig. 14 for mean shift results (LOF and iForest only for brevity).

C.2.4 Membership obfuscation bias. See Fig. 15.

C.2.5 Unequal base rates. See Fig. 16.

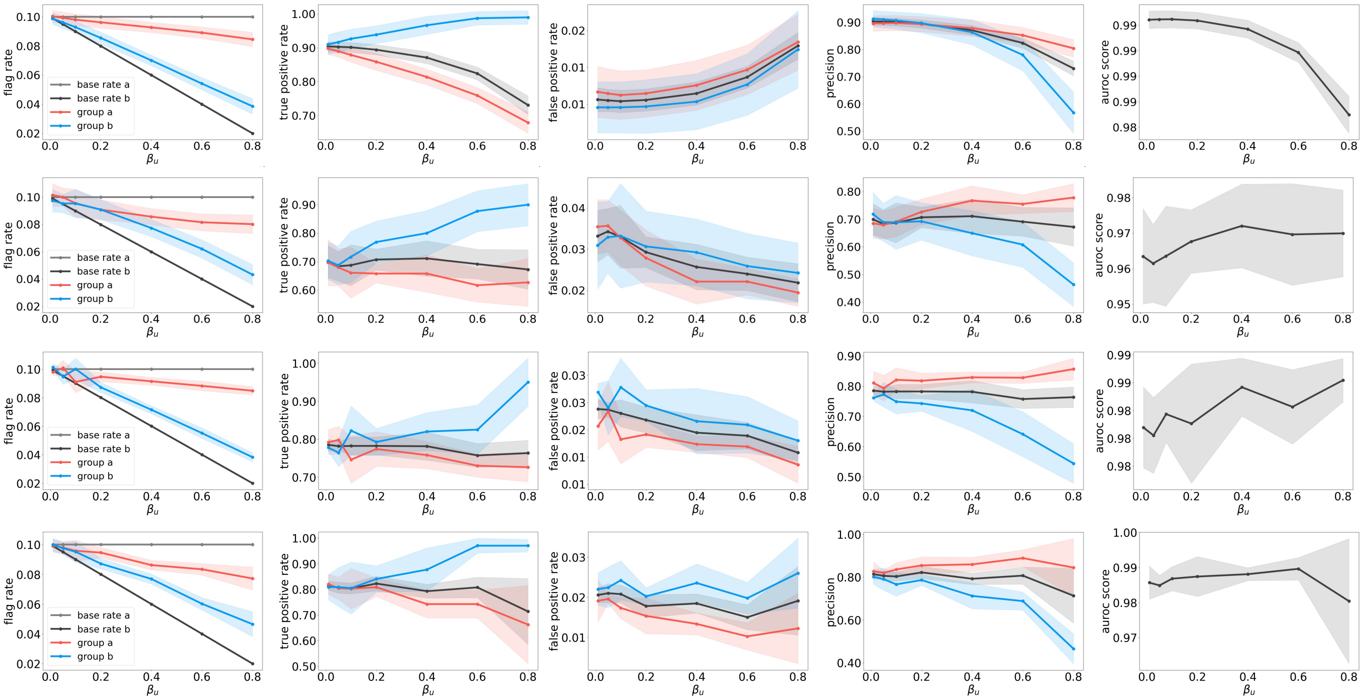

Figure 6: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under target under-representation on *clustered* outliers.

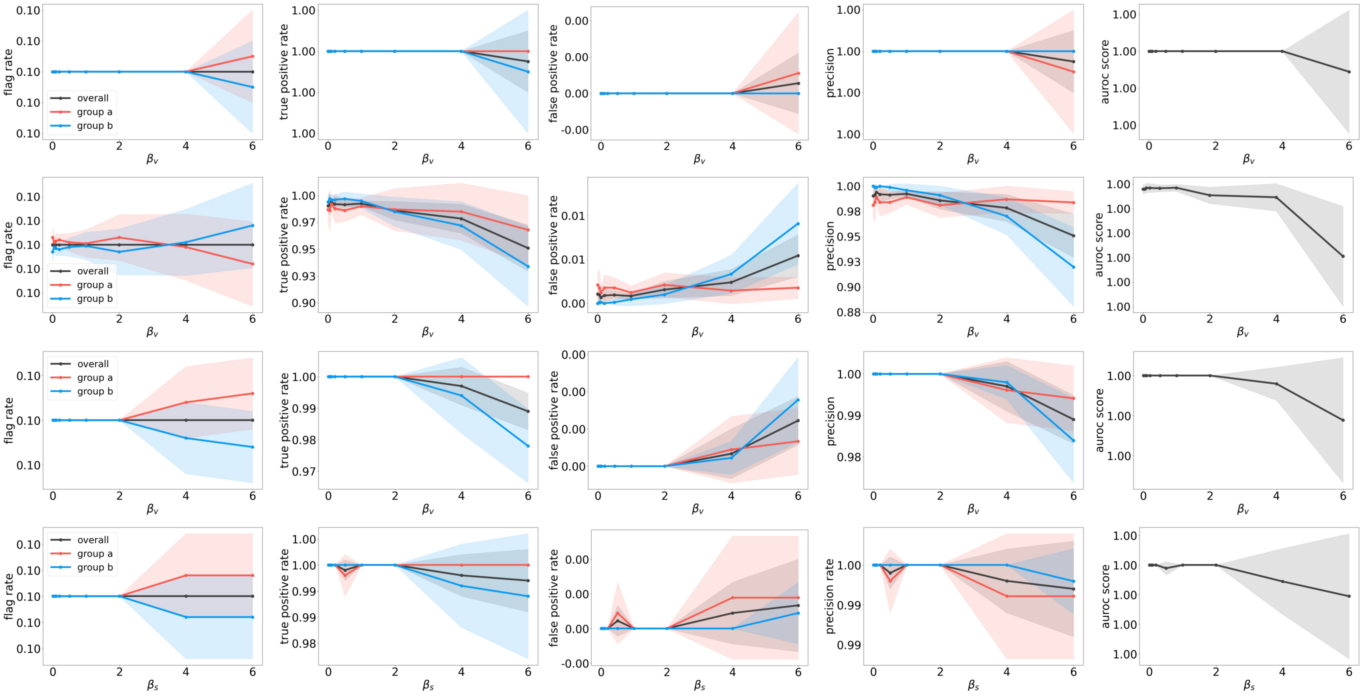

Figure 7: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under feature measurement bias (variance shift) on *clustered* outliers.

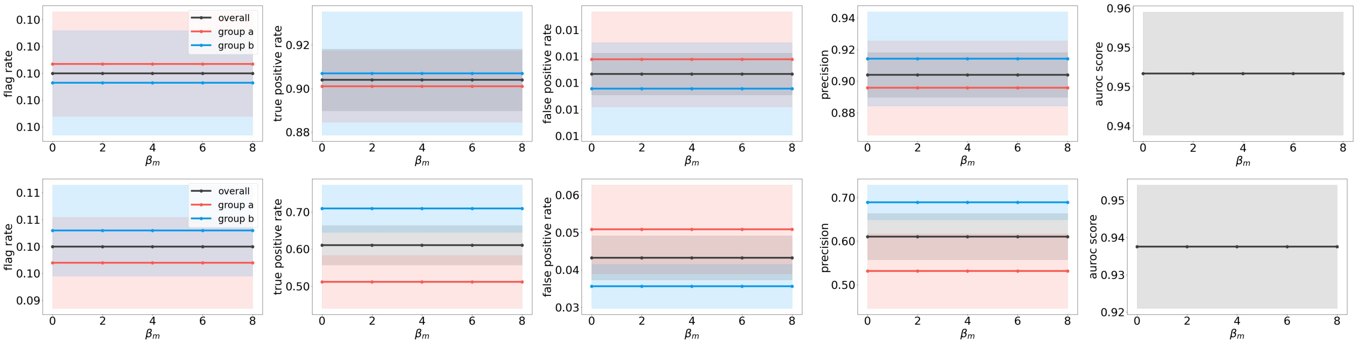

Figure 8: (best in color) Group-wise fairness metrics and AUROC for (top) LOF and (bottom) iForest under feature measurement bias (mean shift) on *clustered* outliers.

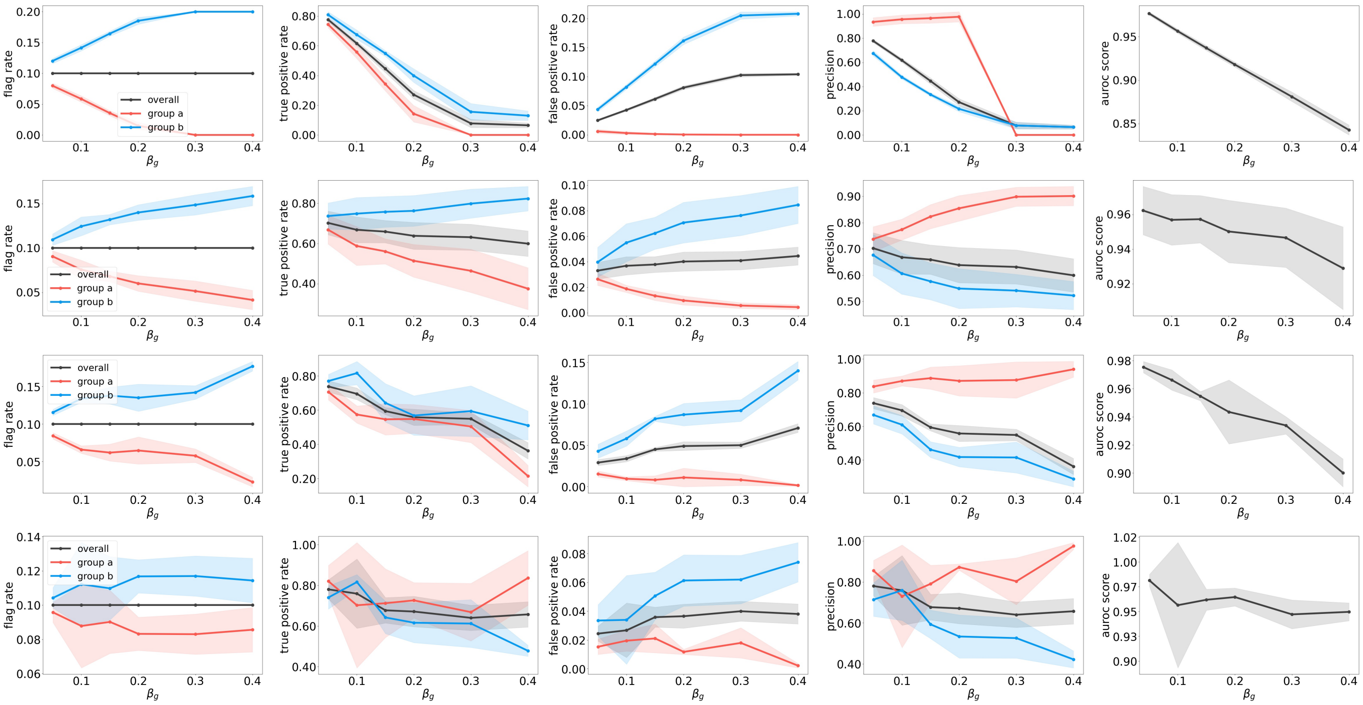

Figure 9: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under membership obfuscation bias on *clustered* outliers.

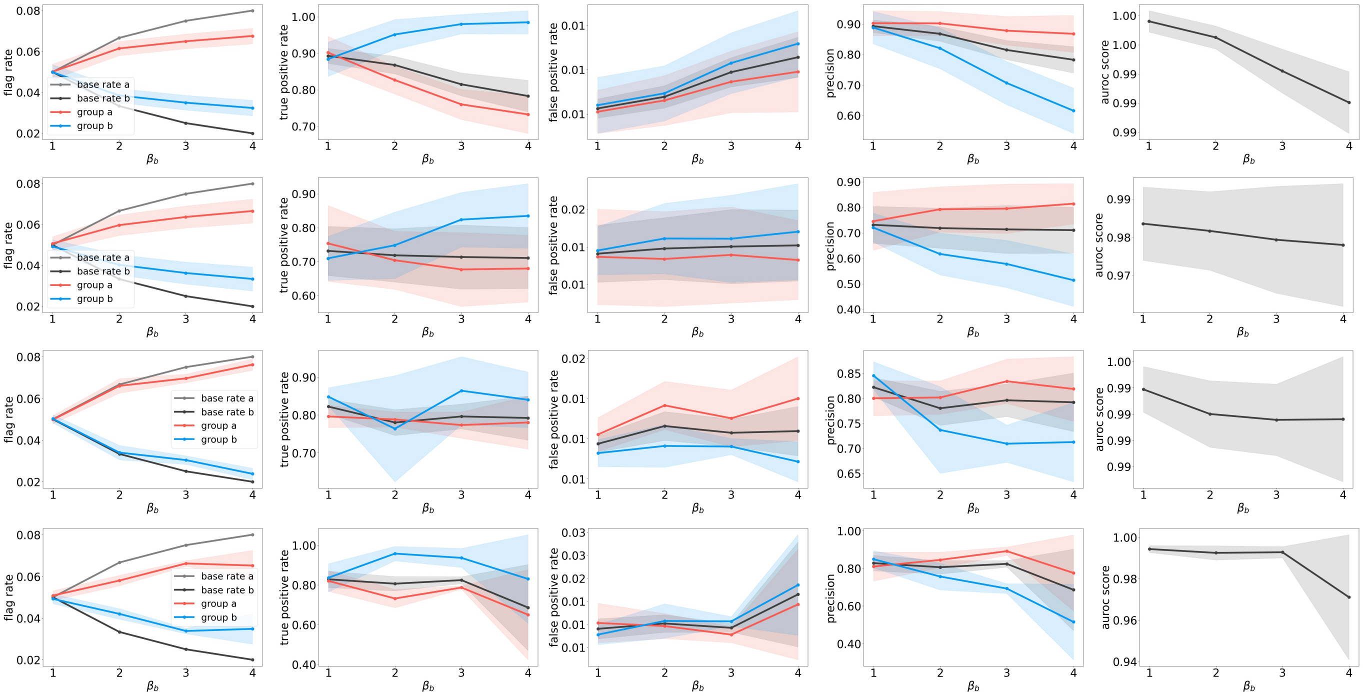

Figure 10: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under unequal base rates on *clustered* outliers.

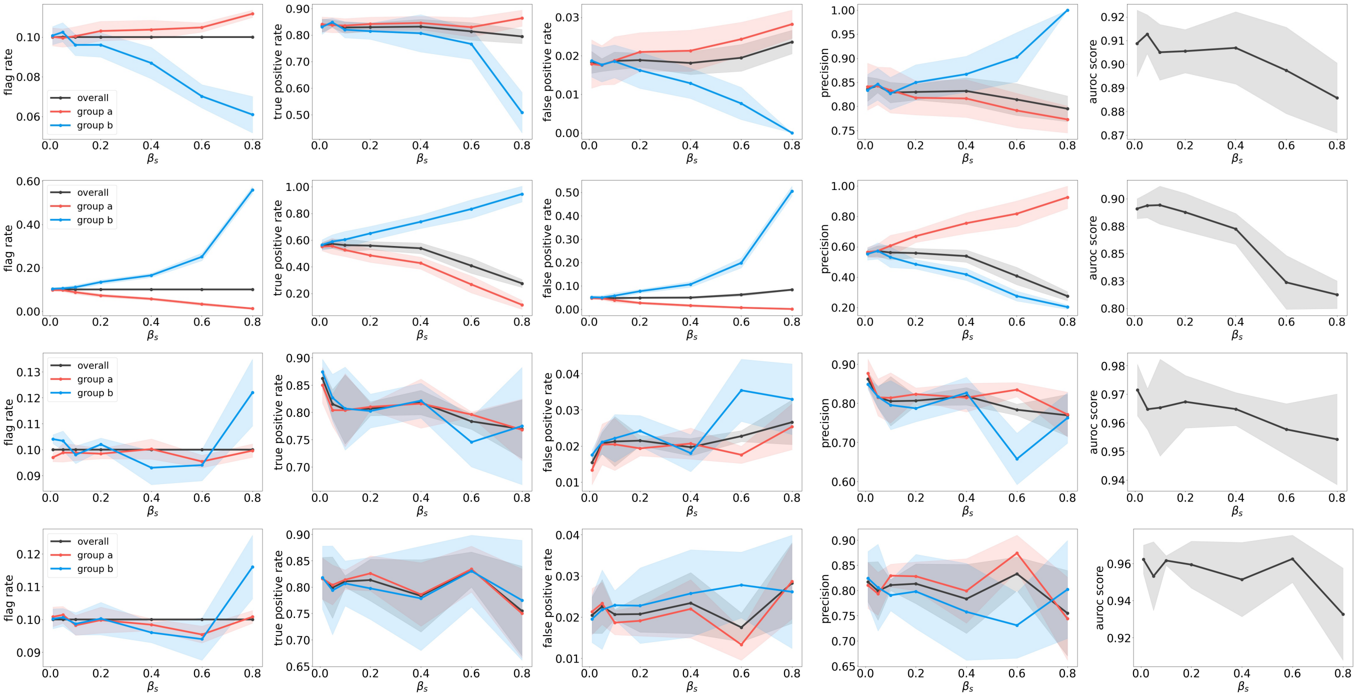

Figure 11: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under sample size bias on *scattered* outliers.

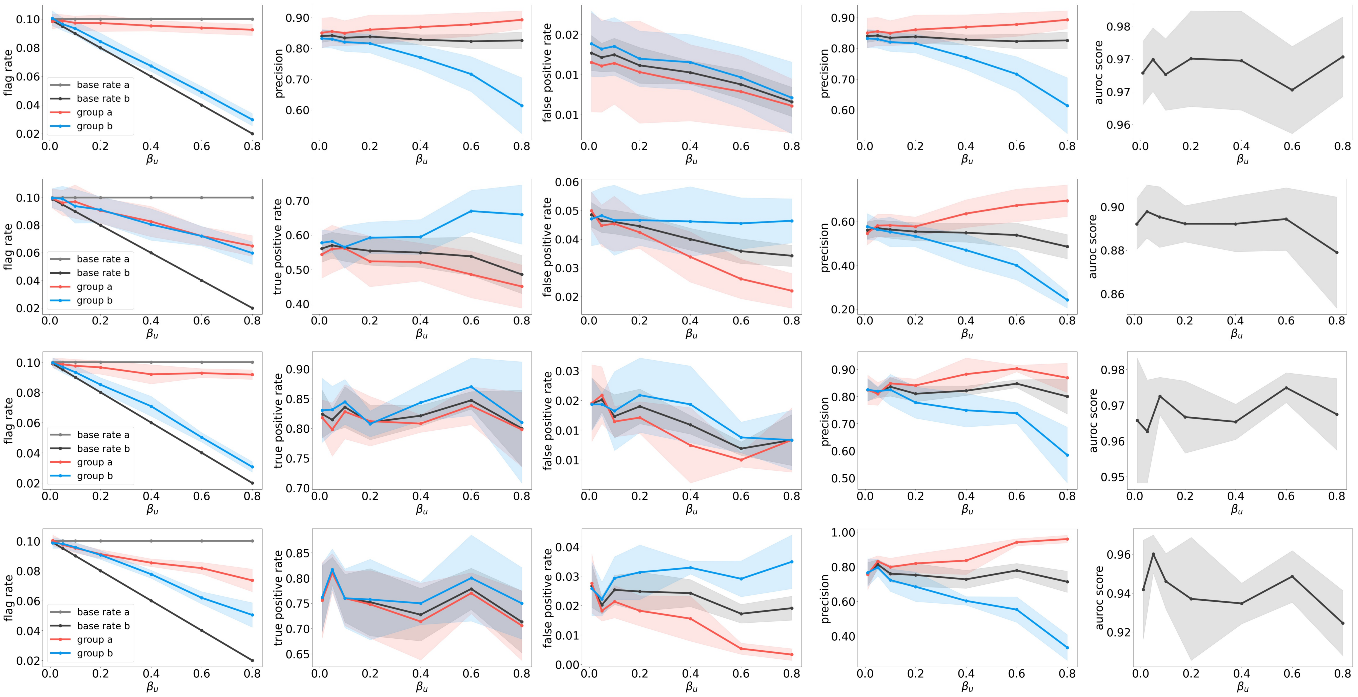

Figure 12: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under target under-representation on *scattered* outliers.

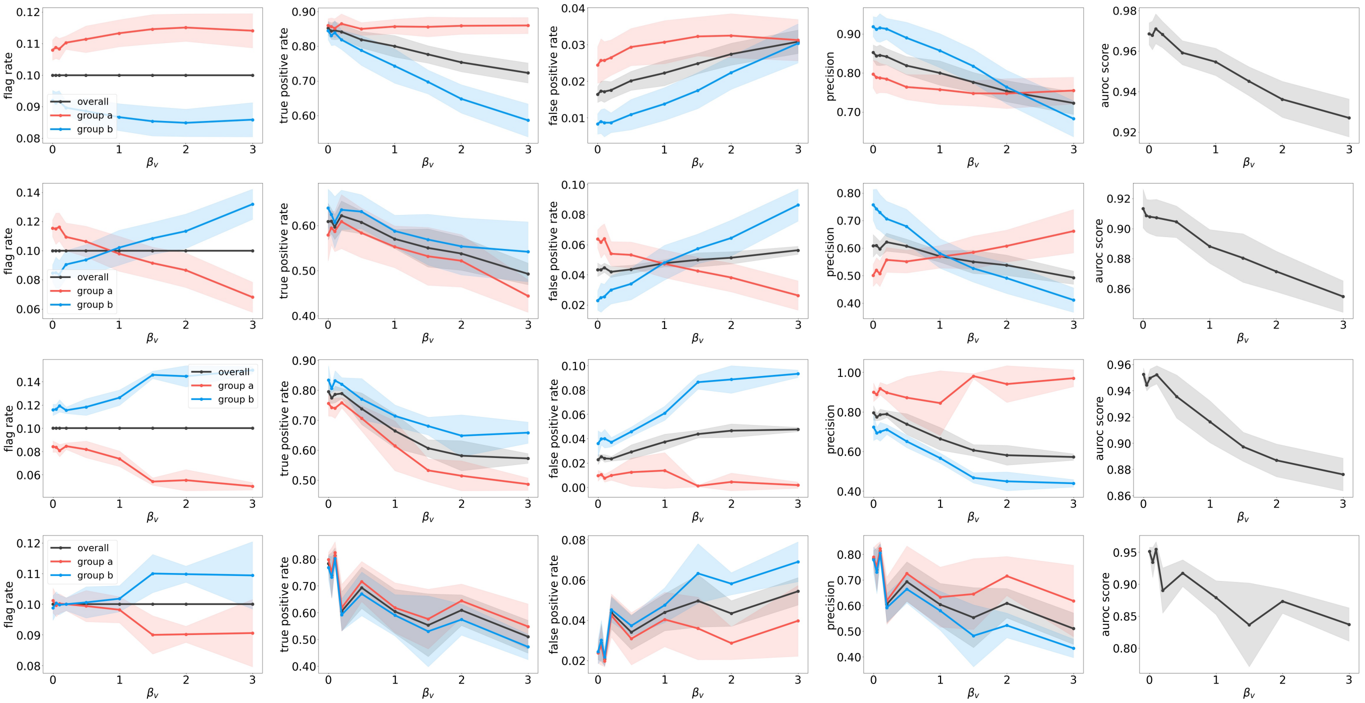

Figure 13: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under feature measurement bias (variance shift) on *scattered* outliers.

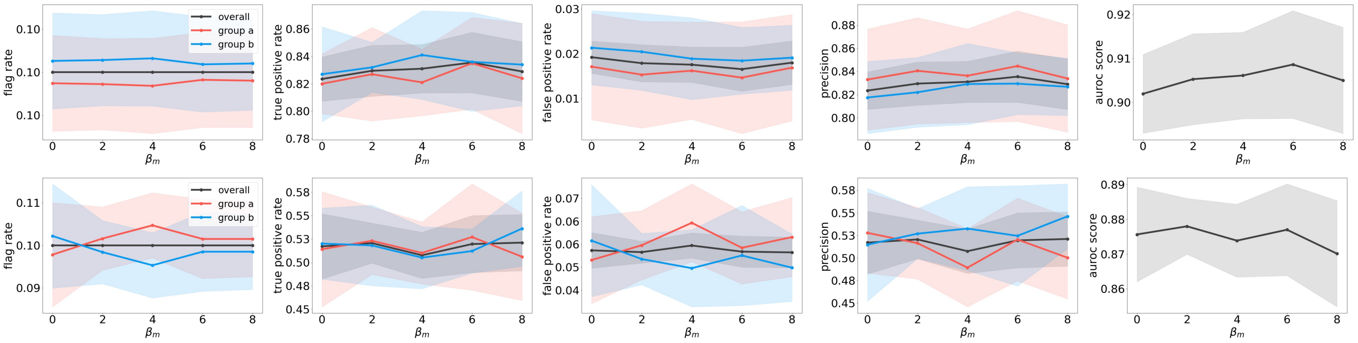

Figure 14: (best in color) Group-wise fairness metrics and AUROC for (top) LOF and (bottom) iForest under feature measurement bias (mean shift) on *scattered* outliers.

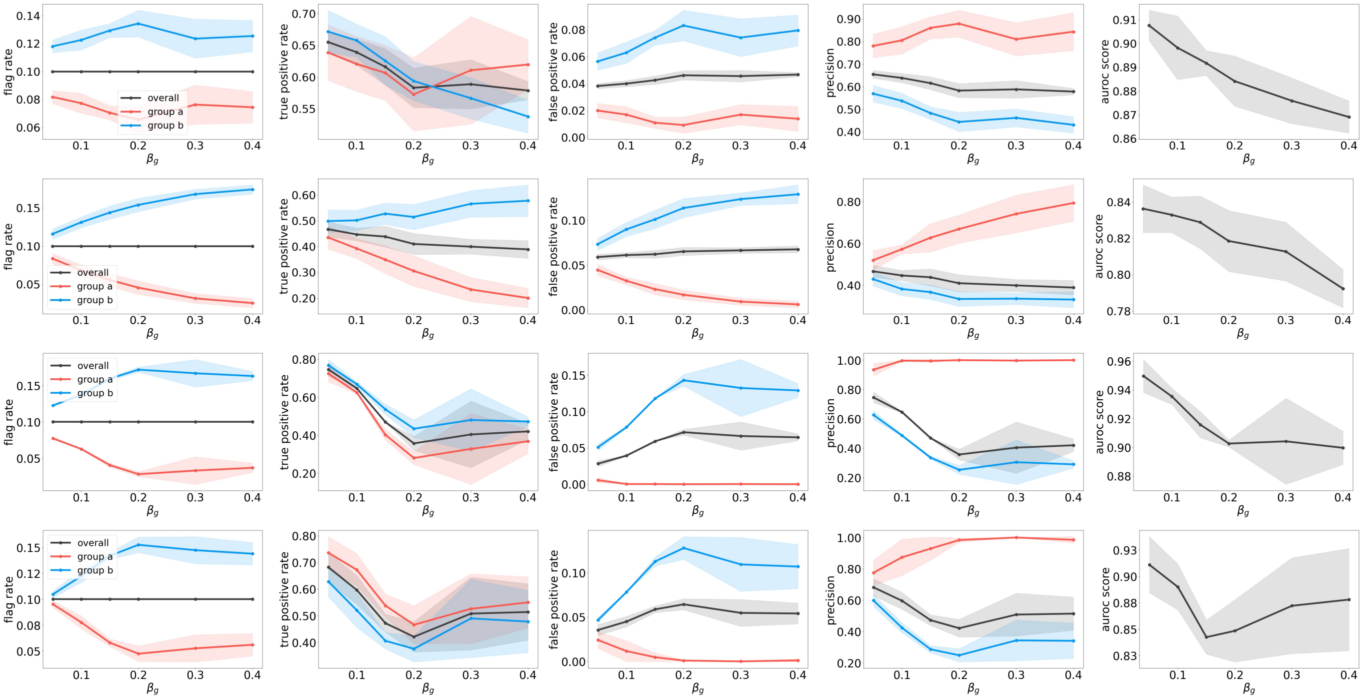

Figure 15: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under membership obfuscation bias on *scattered* outliers.

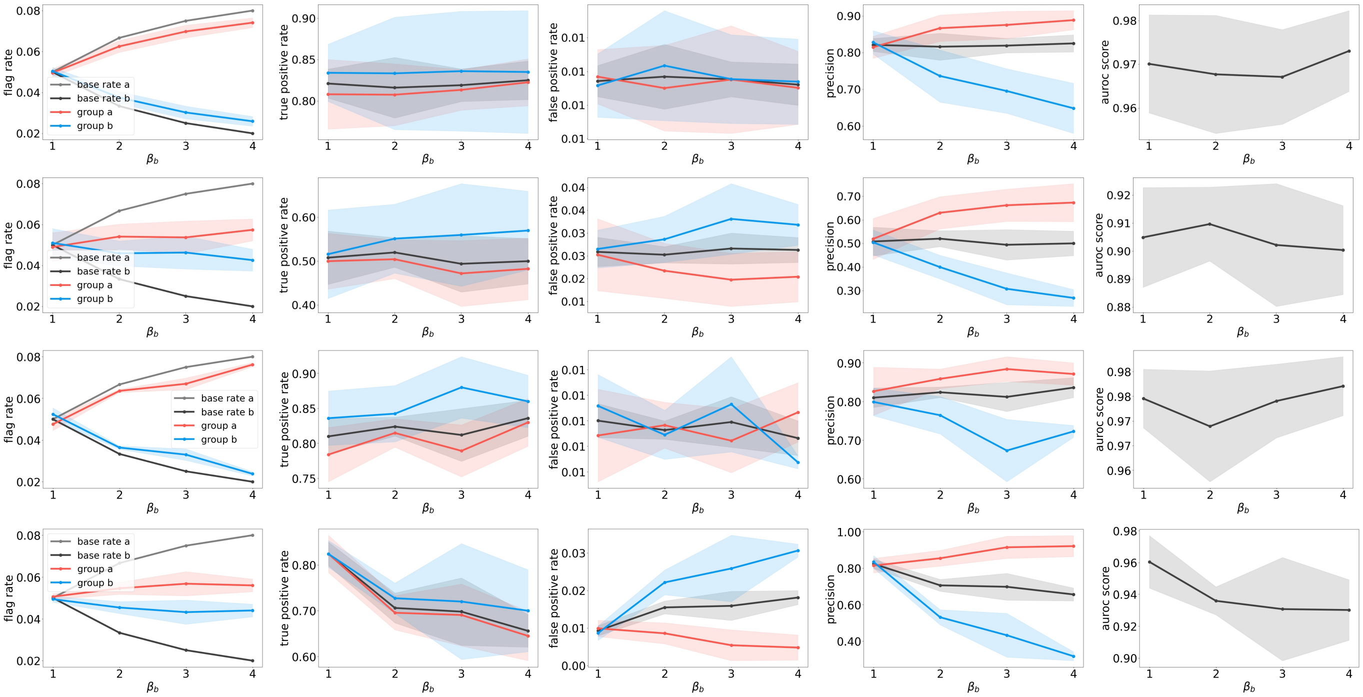

Figure 16: (best in color) Group-wise fairness metrics and AUROC for (top to bottom) LOF, iForest, DeepAE and FairOD under unequal base rates on *scattered* outliers.

D THEORETICAL ANALYSIS

In this section, we derive theoretical justification of why LOF and iForest can adapt well (or poorly) to different injected biases. We first introduce the mathematical notations in Apdx. D.1, the working mechanism and the underlying quantities for these two OD models in Apdx. D.2, and then provide theoretical results in Apdx. D.3.

D.1 Notations

Name	Variable
Size of population	n
Intra-group distance	c
Intra-group dist. of Group a	d
Intra-group dist. of Group b	D
Avg. dist. between inliers and outliers	Δ
Avg. dist. between inliers and outliers of Group a	Δ_a
Avg. dist. between inliers and outliers of Group b	Δ_b
Avg. dist. between subpopulations in $\mathcal{X}_g$	Δ_g
LOF number of nearest neighbors	k
Sensitive attribute of two protected groups	G
Input features	$\mathbf{X}$
Target variable	Y
Outlier score	S
Outlier label of OD model	O
Base rate of a group G	br_G
Flag rate of a group G	fr_G
(Proxy) Variables correlated w/ group membership	$\mathbf{X}_g$
(Incriminating) Variables correlated w/ target	$\mathbf{X}_c$
Non-incriminating occlusion variables	$\mathbf{X}_o$
Set of k-nearest neighbors around a point p	$N_k(p)$
Set of outliers	$\mathcal{O}$
Set of inliers	$\mathcal{I}$
Set of local neighborhoods in scattered outliers	$\mathcal{M}$
An iTree's split point	sp

D.2 Preliminaries

D.2.1 **Probability and Outlier Score.** In the main text, we compare flag rates $P(O = 1|G)$ to measure the fairness of an OD algorithm. However, our outlier detection algorithms rarely give a direct estimation of the true probability scores. For example, LOF algorithm uses local outlier factor(LOF) as the outlier score, iForest measures the outlierness by average path lengths of iTrees, and DeepAE outputs the reconstruction score of decoder.

Given any outlier detection algorithm, the probability of flagging a point as outlier $P(O = 1)$ can be written as

$$\begin{aligned} P(O=1) = \Big[&P(O=1|G=a, Y=1)P(Y=1|a) \\ &+ P(O=1|G=a, Y=0)P(Y=0|a)\Big]P(G=a) \\ &+ \Big[P(O=1|G=b, Y=1)P(Y=1|b) \\ &+ P(O=1|G=b, Y=0)P(Y=0|b)\Big]P(G=b) \qquad (2) \end{aligned}$$

During our analysis, we assume that protected groups a and b have equal size n_a, n_b and equal base rate $P(Y = 1|a) = P(Y = 1|b)$. We also assume that the outlier detection algorithm flags a constant number of outliers (equal to the true count of outliers in the data), making $P(O = 1)$ remain constant throughout our analysis.

Then, we can relate outlier scores to probability. If we assign higher outlier scores S to the points (either inlier or outlier points) from either protected group a or b, it implies that the probability of flagging the points $P(O = 1|G, Y)$ increases for that group, with possible decrease in probabilities of flagging the data points (either inlier or outlier points) from the other group.

D.2.2 **Mechanism of LOF.** LOF computes the local density deviation of a given data point with respect to its neighborhood points, and using the ratio of density deviation as the outlier score S. A score S larger than 1 indicates that a point p is more likely an outlier with higher $P(O = 1|p)$.

To compute the local density deviation, one needs to compute the reachability distance. The detailed calculations are outlined by the definitions below.

Definition 2 (k-distance). *Assume the distance between two data points is $d(p, o)$. Then, given a dataset $\mathcal{D}$, a data $p \in \mathcal{D}$ and a positive integer k, where o_k is the k-th nearest neighbor point of p, the k-distance of p is defined as*

$$k\text{-}distance(p) = dist(p, o_k) \ .$$

Definition 3 (Reachability Distance). *Reachability distance (RD) is the distance to travel from a point to its neighbor. For any two data points $p, o \in \mathcal{D}$, the reachability distance is given as*

$$RD_k(p, o) = \max(k\text{-}distance(o), dist(p, o)) \ .$$

Definition 4 (Local Reachability Density). *For any point p, denote the set of k-nearest neighbor points by $N_k(p) \subseteq \mathcal{D}$. then, the local reachability density (LRD) of p is:*

$$LRD_k(p) = \frac{1}{\sum_{o \in N_k(p)} RD_k(p, o)/k} \ .$$

Definition 5 (LOF Score). *The core of the LOF algorithm is the calculation of local outlier factor (LOF), which measures density deviation between a point and its neighbors. Formally, LOF of p is calculated as*

$$S(p) = LOF_k(p) = \frac{\sum_{o \in N_k(p)} LRD_k(o)}{k \cdot LRD_k(p)} \ .$$

An LOF score that is notably larger than 1 associates with relatively smaller local density and hence larger outlierness, while the factor being close to 1 means that the local density is similar to that of the neighbors', suggesting an inlier.

D.2.3 **Mechanism of iForest.** iForest considers the average number of steps (i.e. randomized splits) required to isolate a point from the others. It starts with building iTrees, which makes random threshold splits sequentially on randomly chosen features. A iTree is complete when all points are the same in a leaf node, or a leaf node contains only a single point.

The outlier score of a point p is the average path length across an ensemble of iTrees to reach p's leaf from the root.

Definition 6 (Path Length). *Path length denoted as $h(p)$ of a point $p \in \mathcal{D}$ is measured by the number of edges p traverses an iTree from the root node until the traversal is terminated at a leaf node.*

Definition 7 (iForest Score). *For a point $p \in \mathcal{D}$ where $|\mathcal{D}| = n$, the score is calculated as*

$$S(p, n) = 2^{-\frac{\mathbb{E}[h(p)]}{c(n)}},$$

where $c(n) = 2H(n-1) - (2(n-1)/n)$ and $H(i)$ is the harmonic number.

Note that as $c(n)$ is fixed by the size of dataset, p stands out as outlier the smaller is the $\mathbb{E}[h(p)]$ (average path length across different iTrees), which indicates that p is easily separated (i.e. isolated) from rest of the points via random threshold-splits.

It is difficult to directly estimate the average path length over an ensemble of iTrees, given that each iTree is built using a subsample of data based on consecutively splitting on random features and random threshold. Therefore, we analyze *how likely* it is for an iTree to split on a feature (or multiple features) and isolate a point as outlier, where we associate a higher likelihood with a shorter expected path length (hence higher outlier score S).

D.3 Theoretical Analysis

Propositions for LOF stand on the following two lemmas we first prove, which relate to the value of the LOF hyperparameter k.

D.3.1 **Lemma 1.** *For the clustered outliers setting, let $\mathcal{I}$ and $\mathcal{O}$ denote the set of inliers and outliers, where $|\mathcal{O}| < |\mathcal{I}|$. Within $\mathcal{I}$ and $\mathcal{O}$, the average intra-group distances are denoted by c. Between $\mathcal{I}$ and $\mathcal{O}$, the average distance is denoted by Δ s.t. $\Delta > c$. Then, for LOF to be able to bypass masking and flag all clustered true outliers, number of neighbors k needs to be chosen s.t. $k > |\mathcal{O}|$.*

Proof: We first prove that outliers would mask when $k \leq |\mathcal{O}|$. Consider an outlier point $p \in \mathcal{O}$. Since $\Delta > c$, p's k nearest neighbors consist only of outliers, i.e. are points within the set $\mathcal{O}$, $N_k(p) \subseteq \mathcal{O}$.

Then, the expected LOF (outlier score S) of p is calculated as follows.

$$\begin{aligned}
\mathbb{E}[\mathrm{LOF}_k(p)] &= \mathbb{E}\left[\frac{\sum_{o\in N_k(p)} \mathrm{LRD}_k(o)}{|N_k(p)| \cdot \mathrm{LRD}_k(p)}\right] && (3)\\
&= \frac{\sum_{o\in N_k(p)} \mathbb{E}\big[\mathrm{LRD}_k(o)\big]}{k \cdot \mathbb{E}\big[\mathrm{LRD}_k(p)\big]} && (4)\\
&= \frac{\sum_{o\in N_k(p)} \mathbb{E}\big[\frac{1}{\sum_{j\in N_k(o)} \mathrm{RD}_k(o,j)/|N_k(o)|}\big]}{k\mathbb{E}\big[\frac{1}{\sum_{m\in N_k(p)} \mathrm{RD}_k(m,p)/|N_k(p)|}\big]} && (5)\\
&= \frac{\sum_{o\in N_k(p)} \frac{1}{\sum_{j\in N_k(o)} \mathbb{E}[\mathrm{RD}_k(o,j)]/k}}{k\frac{1}{\sum_{m\in N_k(p)} \mathbb{E}[\mathrm{RD}_k(m,p)]/k}} && (6)\\
&\approx \frac{k \cdot \frac{1}{k\cdot c/k}}{k\frac{1}{k\cdot c/k}} = 1 && (7)
\end{aligned}$$

Similarly, for an inlier point $z \in \mathcal{I}$, z's k nearest neighbors consist only of inliers, when $k \leq |\mathcal{O}| < |\mathcal{I}|$. When $N_k(z) \subset \mathcal{I}$, the average reachability distances of z and z's neighbors are all c. Therefore, expected LOF of z is 1.

As a result, LOF algorithm fails to detect any outliers due to both outliers and inliers having the same expected outlier score S.

In contrast, when $|\mathcal{I}| > k > |\mathcal{O}|$, we instead have $|\mathcal{O}| \subseteq N_k(p)$. The numerator of Eq. (6), that is the local reachability density of neighborhood points becomes

$$\begin{aligned}
&\sum_{o\in\mathcal{O}} \frac{1}{\sum_{j\in N_k(o)} \mathbb{E}[\mathrm{RD}_k(o,j)]/k}\\
&+ \sum_{m\in N_k(p)\setminus\mathcal{O}} \frac{1}{\sum_{n\in N_k(m)} \mathbb{E}[\mathrm{RD}_k(m,n)]/k}\\
&= \sum_{o\in\mathcal{O}} \frac{1}{\Delta} + \sum_{m\in N_k(p)\setminus\mathcal{O}} \frac{1}{c} && (8)\\
&= \frac{|\mathcal{O}|}{\Delta} + \frac{k-|\mathcal{O}|}{c} && (9)
\end{aligned}$$

The denominator of Eq. (6), that is the local reachability density of p becomes

$$\begin{aligned}
&k\frac{1}{\sum_{m\in N_k(p)} \mathbb{E}[\mathrm{RD}_k(m,p)]/k} && (10)\\
&= \frac{k}{\sum_{m\in\mathcal{O}} \mathbb{E}[\mathrm{RD}_k(m,p)]/k + \sum_{n\in N_k(p)\setminus\mathcal{O}} \mathbb{E}[\mathrm{RD}_k(n,p)]/k} && (11)\\
&= \frac{k}{|\mathcal{O}| \cdot \Delta/k + (k-|\mathcal{O}|) \cdot \Delta/k} = \frac{k}{\Delta} && (12)
\end{aligned}$$

Note that in Eq. (11), for both outlier or inlier neighbors of p, the reachability distance is Δ, since k-distance is the furthest reachable point for the k-neighbors (recall Definition 2). Combining both, the expected LOF becomes

$$\mathbb{E}[\mathrm{LOF}_k(p)] = \frac{\frac{|\mathcal{O}|}{\Delta} + \frac{k-|\mathcal{O}|}{c}}{\frac{k}{\Delta}} > 1 \qquad (13)$$

Since $k < |\mathcal{I}|$, the expected LOF remains 1 for any inlier. As such, $p \in \mathcal{O}$ is assigned a higher S than $z \in \mathcal{I}$ only when $|\mathcal{I}| > k > |\mathcal{O}|$, which concludes the proof. □

D.3.2 **Lemma 2.** *In clustered outlier setting, consider multiple outlier sets $\mathcal{O}_i$, $i \in \{1, ..., j\}$ and one inlier set $\mathcal{I}$, with the average intra-group distance for each $\mathcal{O}_i$ and $\mathcal{I}$ is denoted by c, and the average distance between outliers and inliers for each $\mathcal{O}_i$ is denoted Δ s.t. $\Delta > c$. Then, for LOF algorithm to detect all true outliers in the micro-clusters, the number of neighbors $k > \max(|\mathcal{O}_1|, ..., |\mathcal{O}_j|)$.*

Proof: Based on Lemma 1, to detect all outliers from $\mathcal{O}_i, i \in \{1, ..., j\}$, we need $k > |\mathcal{O}_i|$ for all i. Therefore, $k > \max(|\mathcal{O}_1|, ..., |\mathcal{O}_j|)$. □

D.3.3 **Proposition 1.** *In the clustered outliers setting, let groups a and b have equal size $n_a = n_b$ and equal base rate $P(Y = 1|a) = P(Y = 1|b)$. Let d and D denote intra-group distance between inlier pairs (same for outlier pairs) for group a and b, respectively, where $D > d$ as group b is sparser. Also denote by Δ_a and Δ_b the average distance between outliers and inliers in each group (See Fig. 17). Then, assuming LOF hyperparameter k is set s.t. $k > n_a \cdot P(Y = 1|a)$ and $\Delta_a = \Delta_b$, LOF tends to assign higher scores to group a-outliers, increasing flag rate $P(O = 1|a)$. Further, when $\Delta_b \approx D$, LOF score of b-outliers is ≈ 1, i.e. close to inlier scores, leading to low TPR for group b due to masking.*

Proof: Let us denote by $\mathcal{I}_a$, $\mathcal{I}_b$ for the sets of inliers and by $\mathcal{O}_a$, $\mathcal{O}_b$ the sets of outliers in groups a and b, respectively. Based on

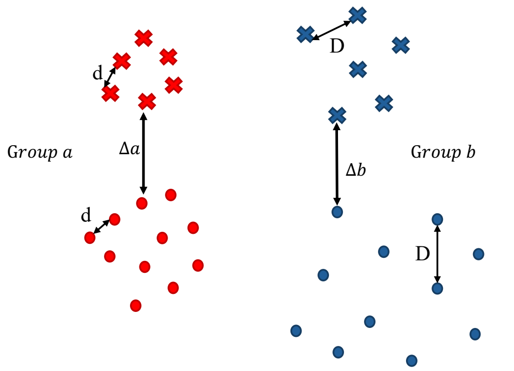

Figure 17: For clustered outliers, illustration of density variation between group a and b (Proposition 1).

Lemma 1, $k > |\mathcal{O}_a| = n_a \cdot P(Y=1|a) = |\mathcal{O}_b| = n_b \cdot P(Y=1|b)$. Then, for an outlier point $p_a \in \mathcal{O}_a$, the expected LOF is given as

$$\mathbb{E}[\mathrm{LOF}_k(p_a)] = \mathbb{E}\left[\frac{\sum_{o\in N_k(p_a)} \mathrm{LRD}_k(o)}{|N_k(p_a)| \cdot \mathrm{LRD}_k(p_a)}\right] \tag{14}$$

$$= \frac{\sum_{o\in N_k(p_a)} \mathbb{E}\big[\mathrm{LRD}_k(o)\big]}{k \cdot \mathbb{E}\big[\mathrm{LRD}_k(p_a)\big]} \tag{15}$$

$$= \frac{\sum_{o\in \mathcal{O}_a} \frac{1}{\sum_{j\in N_k(o)} \mathbb{E}[\mathrm{RD}_k(o,j)]/k}}{k \cdot \mathbb{E}\big[\mathrm{LRD}_k(p_a)\big]} \tag{16}$$

$$+ \frac{\sum_{m\in N_k(p_a)\setminus \mathcal{O}_a} \frac{1}{\sum_{n\in N_k(m)} \mathbb{E}[\mathrm{RD}_k(m,n)]/k}}{k \cdot \mathbb{E}\big[\mathrm{LRD}_k(p_a)\big]}$$

$$= \frac{\frac{|\mathcal{O}_a|}{\Delta_a} + \frac{k-|\mathcal{O}_a|}{d}}{\frac{k}{\Delta_a}} \ . \tag{17}$$

Using the same calculation, the expected LOF for $p_b \in \mathcal{O}_b$ is:

$$\mathbb{E}[\mathrm{LOF}_k(p_b)] = \frac{\frac{|\mathcal{O}_b|}{\Delta_b} + \frac{k-|\mathcal{O}_b|}{D}}{\frac{k}{\Delta_b}} \ . \tag{18}$$

Since $D > d$ (group b is sparser) and $\Delta_a = \Delta_b$,

$$\frac{\frac{|\mathcal{O}_a|}{\Delta_a} + \frac{k-|\mathcal{O}_a|}{d}}{\frac{k}{\Delta_a}} > \frac{\frac{|\mathcal{O}_b|}{\Delta_b} + \frac{k-|\mathcal{O}_b|}{D}}{\frac{k}{\Delta_b}} \tag{19}$$

$$\implies \mathbb{E}[\mathrm{LOF}_k(p_a)] > \mathbb{E}[\mathrm{LOF}_k(p_b)] \ . \tag{20}$$

Therefore, for the outlier scores S_a and S_b associated with p_a and p_b, we have $S_a > S_b$. Flag rate $P(O=1|G=a) = P(O=1|G=b, Y=1)P(Y=1|G=b) + P(O=1|G=b, Y=0)P(Y=0|G=b)$ increases with higher outlier scores assigned to group a-outliers as $P(O=1|G=a, Y=1)$ increases.

When $D \approx \Delta_b$, group b outliers $\mathcal{O}_b$ become indistinguishable from the inliers $\mathcal{I}_b$, where the expected LOF for p_b in Eq. (18) becomes

$$\mathbb{E}[\mathrm{LOF}_k(p_b)] = \frac{\frac{|\mathcal{O}_b|}{\Delta_b} + \frac{k-|\mathcal{O}_b|}{\Delta_b}}{\frac{k}{\Delta_b}} = 1 \ . \tag{21}$$

Then, group b-outliers mask as they blend with the inlier points with similar LOF score (≈ 1), leading to low TPR for group b. □

D.3.4 **Proposition 2.** *In the clustered outliers setting, let sp denote an iTree's split, and $P(x_a < sp < x_b)$ denote the probability for a split to occur between points x_a and x_b. Let groups a and b have equal base rate $P(Y=1|a) = P(Y=1|b)$ and equal average distance between outliers and inliers $\Delta_a = \Delta_b$. As group b is sparser, $d < D$ denotes the intra-group distances between inlier and outlier pairs for groups a and b, respectively (See Fig. 18). Let $\{\mathcal{O}_a, \mathcal{I}_a\}$, $\{\mathcal{O}_b, \mathcal{I}_b\}$ represent the outliers, inliers sets for groups a and b. Then, for $o_b, p_b \in \mathcal{O}_b$, and $o_a, p_a \in \mathcal{O}_a$, we have $P(p_b < sp < o_b) > P(p_a < sp < o_a)$, indicating an iTree is more likely to split among group-b outliers. The difference between $P(p_b < sp < o_b)$ and $P(p_a < sp < o_a)$ becomes larger when iTree is built in higher dimensions. In addition, for $q_b \in \mathcal{I}_b$, $q_a \in \mathcal{I}_a$, $P(o_b < sp < q_b) = P(o_a < sp < q_a)$, i.e. an iTree is equally likely to split between the inliers and outliers clusters.*

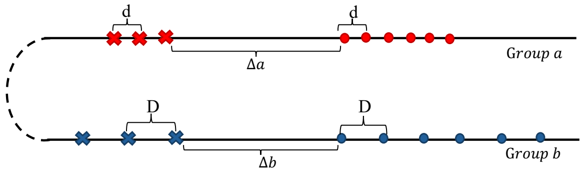

Figure 18: For clustered outliers, illustration of density variation between group a and b in 1-d for iForest (Proposition 2).

Proof: For simplicity, we first consider the one-dimensional setting. Since iTree is built based on uniformly random splits between points and $D > d$, for $o_b, p_b \in \mathcal{O}_b$, and $o_a, p_a \in \mathcal{O}_a$, we have

$$P(p_b < sp < o_b) = \frac{D}{\max_{x,y\in \mathcal{I}\cup\mathcal{O}} d(x,y)} \tag{22}$$

$$P(p_a < sp < o_a) = \frac{d}{\max_{x,y\in \mathcal{I}\cup\mathcal{O}} d(x,y)} \tag{23}$$

$$\implies P(p_b < sp < o_b) > P(p_a < sp < o_a) \ . \tag{24}$$

In words, iTree is more likely to split among b-outliers than a-outliers. iForest assigns higher outlier scores to points that are isolated via fewer splits, yielding shorter root-to-leaf paths. Then, based on Eq. (24), an iTree is more likely to isolate b-outliers and thus associate them with shorter paths, which explains why $P(O=1|G=b) > P(O=1|G=a)$ for iForest.

On the other hand, for $p_b \in \mathcal{O}_b$, $q_b \in \mathcal{I}_b$, and $p_a \in \mathcal{O}_a$, $q_a \in \mathcal{I}_a$, we have

$$P(p_b < sp < q_b) = \frac{\Delta_b}{\max_{x,y\in \mathcal{I}\cup\mathcal{O}} d(x,y)} \tag{25}$$

$$= \frac{\Delta_a}{\max_{x,y\in \mathcal{I}\cup\mathcal{O}} d(x,y)} = P(p_a < sp < q_a) \tag{26}$$

That is, iTree is equally likely to split between the inliers and outliers for both groups a and b. This single split is crucial for both groups for ruling out the outliers collectively as a cluster. Due to this special split, iForest can flag certain fraction of a-outliers, leading to smaller decrease of TPR for group a.

Extending the analysis to the multiple h-dimensional setting, we assume the intra-group distance remains $D > d$ on each dimension, and distance $\Delta_a = \Delta_b$ holds between outliers and inliers for both groups a and b. Then, for $\mathbf{o}_b, \mathbf{p}_b \in O_b$, and $\mathbf{o}_a, \mathbf{p}_a \in O_a$, we have

$$P(\mathbf{p}_b < sp < \mathbf{o}_b) - P(\mathbf{p}_a < sp < \mathbf{o}_a) \tag{27}$$

$$= \frac{D^h}{\prod_{j=1}^{h}(\max_{\mathbf{x},\mathbf{y}\in \mathcal{I}\cup O} d(\mathbf{x}_j, \mathbf{y}_j))} - \frac{d^h}{\prod_{j=1}^{h}(\max_{\mathbf{x},\mathbf{y}\in \mathcal{I}\cup O} d(\mathbf{x}_j, \mathbf{y}_j))} \tag{28}$$

Since $D > d$, $D^h - d^h$ grows with increasing h. While the denominator is roughly the same, the discrepancy between two probabilities increases with larger h. □

D.3.5 **Proposition 3.** *In the clustered outliers setting, let groups a and b have equal size $n_a = n_b$ and equal base rate $P(Y = 1|a) = P(Y = 1|b)$. Assume group b-inliers form f smaller populations, $\mathcal{I}_b^i$, for $i \in [\![1, f]\!]$, and group b-outliers form g smaller populations, O_b^j for $j \in [\![1, g]\!]$. Let d and D denote intra-group distance between inlier pairs (same for outlier pairs) for group a and each sub-population in b, respectively, where $D = d$. Denote Δ_a as the average distance between outliers and inliers in group a, and Δ_b as the average distance between each subpopulation in group b for both inliers and outliers (See Fig. 19). Then, assuming LOF hyperparameter k is set s.t. $k > n_a \cdot P(Y = 1|a)$, and $\Delta_a = \Delta_b$ while $\Delta_a > d$ and $\Delta_b > D$, LOF tends to assign higher scores to group b-outliers, increasing flag rate $P(O = 1|b)$. Further, when $|\mathcal{I}_b^j| < |O_b^i| < k$, LOF score of b-inliers is larger than of b-outliers, leading to both high FPR and low TPR for group b.*

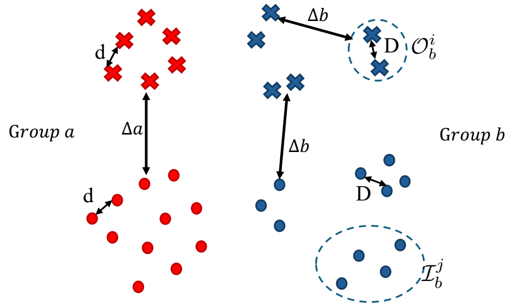

Figure 19: For clustered outliers, illustration of membership obfuscation bias in group b (Proposition 3).

Proof: For simplicity, we assume each subpopulation in group b maintains the average intra-group distance D, and the distances between each subpopulation remains at Δ_b. However, altering certain subpopulation's intra-group distance or distance to other subpopulations can cause one subpopulation to have lower/higher LOF outlier scores, thus complicating the analysis on detecting group a and b outliers.

Let us denote by $\mathcal{I}_a$ and O_a the sets of inliers and outliers in group a. Based on Lemma 2, LOF hyperparameter $k > n_a \cdot P(Y = 1|a) > \max(|O_b^1|, ..., |O_b^f|)$. For an outlier point p_b^i in subpopulation O_b^i for any $i \in [1, ..., f]$, the average local reachability density is $\frac{k}{\Delta_b}$ by the same calculation as Eq. (12). On the other hand, the local reachability density of p_b^i's k-neighborhood is calculated as

$$\frac{|O_b^i|}{\Delta_b} + \frac{k - |O_b^i|}{D} \tag{29}$$

In comparison to the expected LOF score of an outlier point $p_a \in O_a$, with $\Delta_a = \Delta_b$, $d = D$, $\Delta_a > d$, and $\Delta_b > D$, we derive the following inequality.

$$\frac{\frac{|O_b^i|}{\Delta_b} + \frac{k-|O_b^i|}{D}}{\frac{k}{\Delta_b}} > \frac{\frac{|O_a|}{\Delta_a} + \frac{k-|O_a|}{d}}{\frac{k}{\Delta_a}} \tag{30}$$

$$\implies \mathbb{E}[\mathrm{LOF}_k(p_b^i)] > \mathbb{E}[\mathrm{LOF}_k(p_a)]\ . \tag{31}$$

Therefore, p_b^i from group b receives higher outlier score than p_a ($S_b > S_a$). Flag rate $P(O = 1|G = b)$ increases due to increased $P(O = 1|G = b, Y = 1)$.

In addition, when $|\mathcal{I}_b^j| < k$, an inlier $p_b^j \in \mathcal{I}_b^j$ has the following expected LOF score.

$$\mathbb{E}[\mathrm{LOF}_k(p_b^j)] = \frac{\frac{|\mathcal{I}_b^j|}{\Delta_b} + \frac{k-|\mathcal{I}_b^j|}{D}}{\frac{k}{\Delta_b}} \tag{32}$$

If $|\mathcal{I}_b^j| < |O_b^i|$, then $\mathbb{E}[\mathrm{LOF}_k(p_b^j)] > \mathbb{E}[\mathrm{LOF}_k(p_b^i)]$, where $p_b^i \in O_b^i$ is a true outlier; i.e. LOF score of b-inliers is higher than for b-outliers, in this case. As $P(O = 1|G = b, Y = 0) > (O = 1|G = b, Y = 1)$, FPR increases and TPR drops for group b. □

D.3.6 **Proposition 4.** *In the clustered outliers setting, let sp denote an iTree's split point, and $P(x_a < sp < x_b)$ the probability for a split to occur between points x_a and x_b. Let groups a and b have equal base rates, $P(Y = 1|a) = P(Y = 1|b)$. Let $\{O_a, \mathcal{I}_a\}$, $\{O_b, \mathcal{I}_b\}$ represent the outlier and inlier sets for groups a and b. Assume group b-inliers form f smaller populations, $\mathcal{I}_b^i$, for $i \in [\![1, f]\!]$, and b-outliers form g smaller populations, O_b^j for $j \in [\![1, g]\!]$. Let d and D denote intra-group distance between inlier pairs (same for outlier pairs) for group a and b among both $\mathcal{X}_g$ and $\mathcal{X}_c$ dimensions where $D = d$. Also denote by Δ_a the average distance between outliers and inliers for group a among $\mathcal{X}_c$, by Δ_b the average distance between outliers and inliers for group b among $\mathcal{X}_c$, and by Δ_g the average distance among outlier subpopulations and inlier subpopulations of group b among $\mathcal{X}_g$, s.t. $\Delta_g > d$ (See Fig. 20). Then among $\mathcal{X}_g$ features, we have $P(p_b < sp < o_b) > P(p_a < sp < o_a)$, where $o_b \in O_b^i$, $p_b \in O_b^l$, $l \neq i$, and $o_a, p_a \in O_a$. That is, an iTree split on $\mathcal{X}_g$ between group b-subpopulations is more likely than between group a-outliers. In addition, among $\mathcal{X}_c$, we have $P(p_b < sp < q_b) = P(p_a < sp < q_a)$, where $p_b \in O_b$, $q_b \in \mathcal{I}_b$, $p_a \in O_a$, $q_a \in \mathcal{I}_a$. That is, an iTree split on $\mathcal{X}_c$ between inlier and outlier clusters is equally like for both groups.*

Proof: When group b exhibits subpopulations due to group-membership obfuscation or naturally, it fragments into inlier and outlier micro-clusters based on varying feature values within $\mathcal{X}_g$. Then among $\mathcal{X}_g$ features, for $o_b \in O_b^i$, $p_b \in O_b^l$, $l \neq i$, and $o_a, p_a \in O_a$, we have

$$P(p_b < sp < o_b) = \frac{\Delta_g}{\max_{x,y\in \mathcal{I}\cup O} d(x, y)} \tag{33}$$

$$> \frac{d}{\max_{x,y\in \mathcal{I}\cup O} d(x, y)} = P(p_a < sp < o_a) \tag{34}$$

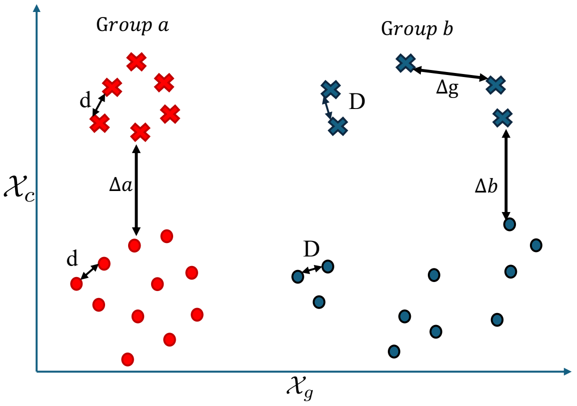

Figure 20: For clustered outliers, illustration of group membership obfuscation bias in group b is only in $\mathcal{X}_g$ (x-axis) but not in $\mathcal{X}_c$ (y-axis), for iForest (Proposition 4).

In words, iTree is more likely to split among group b-outliers than a-outliers, due to larger distances between group b's subpopulations. Therefore, $P(O=1|G=b) > P(O=1|G=a)$ for iForest. Moreoever, among $\mathcal{X}_g$ dimensions, for $r_b \in \mathcal{I}_b^i, s_b \in \mathcal{I}_b^l, l \neq i$,

$$P(r_b < sp < s_b) = \frac{\Delta_g}{\max_{x,y \in \mathcal{I} \cup \mathcal{O}} d(x,y)} \tag{35}$$

$$= P(p_b < sp < o_b) \tag{36}$$

That is, iTree is equally likely to split between group b's outlier micro-clusters and inlier micro-clusters among $\mathcal{X}_g$. Therefore, iForest can have $P(O=1|G=b,Y=1) = P(O=1|G=b,Y=0)$ and incur an increase in group b's FPR, if only $\mathcal{X}_g$ dimensions alone were considered. When combined with $\mathcal{X}_c$ dimensions, for $p_b \in O_b$, $q_b \in \mathcal{I}_b$, $p_a \in O_a$, $q_a \in \mathcal{I}_a$, we can show that $P(p_b < sp < q_b) = P(p_a < sp < q_a)$ using the same calculation as Eq. (26). Then, because iTree is equally likely to split between inlier and outlier clusters for both groups a and b among $\mathcal{X}_c$, iForest successfully assigns high outlier scores to points in both O_b and O_a. However, as group b's flag rate is increased due to increased FPR, flag rate of a is reduced leading to lower TPR for group a. □

D.3.7 **Proposition 5.** *For scattered outliers, let groups a and b have equal size $n_a = n_b$ and equal base rate $P(Y=1|a) = P(Y=1|b)$. Instead of forming a single cluster, scattered outliers form local neighbors (See Fig. 21). We assume that group a's outliers form g local neighborhoods, each denoted by $\mathcal{M}_a^i$, $i \in \{1,...,g\}$, and group b's outliers form f local neighborhoods, each denoted by $\mathcal{M}_b^j$, $j \in \{1,...,f\}$. In other words, we have $\cup_{1,...,f}\mathcal{M}_b^j = O_b$ and $\cup_{1,...,g}\mathcal{M}_a^i = O_a$. Let d_{in} and D_{in} denote the average intra-group distance between inliers for group a and b, respectively, where $d_{in} < D_{in}$ (group b is sparser). Similarly, let d_{out} and D_{out} denote the average intra-group distance between outliers of the local neighborhoods for group a and b, respectively, where $d_{out} < D_{out}$. We also have $d_{out} > d_{in}$ and $D_{out} > D_{in}$ due to the scattered nature of the outliers. Let Δ_a and Δ_b denote the average distance between outliers and inliers in each group, and m the maximum size of the local neighborhoods. Then, assuming LOF hyperparameter k is set s.t. $k > m$ and $\Delta_b = \Delta_a \geq D_{out}$, LOF tends to assign higher scores to group a-outliers, increasing flag rate $P(O=1|a)$.*

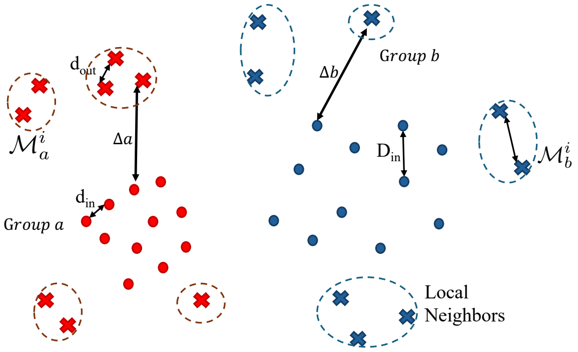

Figure 21: For scattered outliers, illustration of density variation between group a and b (Proposition 5).

Proof: Let $\mathcal{I}_a$, $\mathcal{I}_b$ denote the sets of inliers in group a and group b. We have that $\Delta_a \geq d_{out}$ and $\Delta_b \geq D_{out}$. Otherwise, when a local neighborhood has intra-neighborhood distance larger than distances between outliers and inliers (i.e. $d_{out} > \Delta_a$, $D_{out} > \Delta_b$), the local neighborhood can instead form several smaller neighborhoods. Based on Lemma 1 and 2, LOF could flag micro-clusters of local neighborhood outliers when $k > m$, the size of the largest local neighborhood for scattered outliers in both groups.

For an outlier point in any of group a's local neighborhoods, $p_a^i \in \mathcal{M}_a^i$ for $i \in \{1,...,g\}$, the expected LOF is

$$\mathbb{E}[\text{LOF}_k(p_a^i)] = \mathbb{E}\left[\frac{\sum_{o \in N_k(p_a^i)} \text{LRD}_k(o)}{|N_k(p_a^i)| \cdot \text{LRD}_k(p_a^i)}\right] \tag{37}$$

$$= \frac{\sum_{o \in N_k(p_a^i)} \mathbb{E}\left[\text{LRD}_k(o)\right]}{k \cdot \mathbb{E}\left[\text{LRD}_k(p_a^i)\right]} \tag{38}$$

$$= \frac{\frac{|\mathcal{M}_a^i|}{\Delta_a} + \frac{k-|\mathcal{M}_a^i|}{d_{in}}}{\frac{k}{\Delta_a}} \tag{39}$$

In Eq. (39), at least $k - |\mathcal{M}_a^i|$ neighbors of p_a^i contain only inliers and therefore the average reachability distance is d_{in}. Assume that there exists a $p_b^j \in \mathcal{M}_b^j$, where $|\mathcal{M}_b^j| = |\mathcal{M}_a^i|$, Then, we have $\mathbb{E}[\text{LOF}_k(p_a^i)] > \mathbb{E}[\text{LOF}_k(p_b^j)]$ due to $d_{in} < D_{in}$. That is, if local neighborhoods are of similar sizes, LOF assigns higher score to group-a outliers, increasing flag rate $P(O=1|G=a)$.

When $D_{out} \approx \Delta_b$, outliers are equally distanced from one another and from the inliers, and thus local neighborhoods disappear. However for $p_b \in O_b$, as long as $\alpha > 0$ number of group b-inliers exist among the k-neighbors of p_b, the expected LOF of p_b is

$$\mathbb{E}[\text{LOF}_k(p_b)] = \frac{\frac{k-\alpha}{\Delta_a} + \frac{\alpha}{D_{in}}}{\frac{k}{\Delta_a}} \tag{40}$$

which prevents LOF to assign an outlier score of 1 for p_b. The adjacency to inliers prevents LOF from incurring very low TPR on group b, unlike the clustered outlier setting in Proposition 1. □

D.4 Limitations of Analysis

While our theoretical analyses provide explanations for various OD outcome discrepancies between protected groups *a* and *b*, we acknowledge they exhibit several limitations.

First, while some mechanistic/non-learning algorithms (LOF and iForest in our study) can be explained by analyzing their working mechanisms, it is hard to directly analyze the behavior of other algorithms in reaction to different data biases. Specifically, the learning-based algorithms DeepAE and FairOD that rely on a multi-layer autoencoder structure are hard to analyze mathematically due to randomness in initialization and training.

Second, while we explicitly and clearly stated, our theoretical analyses are based on certain assumptions. Specifically, we assumed that each group or subpopulation of outliers and inliers follow the same average pairwise intra-group distances, which is smaller than the distances between inliers and outliers. The uniformity of data distributions generally does not hold true, especially for real-world datasets. We also assumed the groups to be well separated in the proxy feature space so that inter-group distances do not interfere with the behavior of LOF and iForest on each group. While it may be a weaker assumption that groups are not too alike, the extent of inter-group similarity could affect our propositions in practice.