xFAIR: Better Fairness via Model-based Rebalancing of Protected Attributes

Kewen Peng, Joymallya Chakraborty Tim Menzies, *Fellow, IEEE*

Abstract—**Context**: Machine learning software can generate models that inappropriately discriminate against specific protected social groups (e.g., groups based on gender, ethnicity, etc). Motivated by those results, software engineering researchers have proposed many methods for mitigating those discriminatory effects. While those methods are effective in mitigating bias, few of them can provide explanations on what is the root cause of bias.
Objective: We aim at better detection and mitigation of algorithmic discrimination in machine learning software problems.
Method: Here we propose xFAIR, a *model-based* extrapolation method, that is capable of both mitigating bias and explaining the cause. In our xFAIR approach, protected attributes are represented by models learned from the other independent variables (and these models offer extrapolations over the space between existing examples). We then use the extrapolation models to relabel protected attributes later seen in testing data or deployment time. Our approach aims to offset the biased predictions of the classification model via rebalancing the distribution of protected attributes.
Results:The experiments of this paper show that, without compromising (original) model performance, xFAIR can achieve significantly better group and individual fairness (as measured in different metrics) than benchmark methods. Moreover, when compared to another instance-based rebalancing method, our model-based approach shows faster runtime and thus better scalability.
Conclusion: Algorithmic decision bias can be removed via extrapolation that smooths away outlier points. As evidence for this, our proposed xFAIR is not only performance-wise better (measured by fairness and performance metrics) than two state-of-the-art fairness algorithms.
Reproduction Package:In order to better support open science, all scripts and data used in this study are available on-line at https://github.com/anonymous12138/biasmitigation.

Index Terms—Software Fairness, Explanation, Bias Mitigation

1 Introduction

Increasingly, machine learning (ML) algorithms are applied in software engineering (SE) to assist decision-making, and some of the decisions take private information (e.g., race, gender, age) of human individuals into consideration. For example, ML models are used in software to assist determine which loan applications should get approved; which citizens should get bail; which patients can be released from the hospital. From an ethical perspective, using private information also makes such software under the exposure of unintentionally algorithmic discrimination, where the benefits of certain social groups are compromised. Many prior cases have shown the existence of such flaw: Google's sentimental analysis model was found to assign negative scores to homosexual or Jewish attributes in a sentence; In machine translation, translators wrongly re-label doctors as male and nurses female; In credit card applications, applicants with similar conditions are receiving significantly different credit lines based on their genders.

Many researchers are endeavoring to resolve the discrimination issue in ML software. Recent success with the Fair-SMOTE [1] of Chakraborty et al shows that it is possible to carefully rebalance the training data so as to mitigate bias in the data. Chakraborty et al. [1] conjectured that models are unfair when the training data does not equally represent all social groups. Fair-SMOTE uses a rebalancing method that adjusts the training data such that all values of ”protected attributes” are equally represented in the training data (and by “protected attributes” we mean information about age, gender, racial origins, veteran status, etc. that is used to identify a person as belonging to corresponding groups, some of which have suffered from social injustice in history).

While a successful system in its test domain, Fair-SMOTE has problems with *procedural justice*. By definition [2], [3], procedural justice requires not only fair results but also transparency of the decision-making process such that ones can verify whether the procedure guarantees fairness. One way to demonstrate procedural justice to a group of users is to ensure that an AI system never asks about protected attributes. Note that this is *not* the case with Fair-SMOTE since when its models are deployed, all the new examples must have the same format as the data seen during training. This means that, during deployment, the protected attributes data must be collected from users. Consequently, users can grow concerned that the model will not mitigate against bias (since it has access to the protected information).

Accordingly, this paper explores an alternative to Fair-SMOTE that, once developed, no longer needs to collect protected attributes from its users. In our concept of operation, our method only collects and uses those protected attributes to initially build its model (which assessed using widely accepted fairness metrics, see Table 2). After that, during deployment, our method does not demand access to protected attributes in any subsequent test data.

Removing protected attributes must be done carefully. Prior research has shown that it is insufficient to just remove projected attributes from data. If a model just ignores the protected attributes, then that can either (a) harm the performance of the prediction model due to information loss [4], [5], or (b) have a trivial influence on improving

- *K. Peng, J. Chakraborty and T. Menzies are with the Department of Computer Science, North Carolina State University, Raleigh, USA. E-mail:kpeng@ncsu.edu, jchakra@ncsu.edu, timm@ieee.org*

group fairness due to proxy discrimination [6], [7], [8]. Bias can persist due to the correlations between variables. Such correlations means that an unwanted bias can persist (even though the protected attribute is removed). For an example of proxy discrimination, consider how residential zip codes can be used to make biased decisions such as granting loan since zip codes might correlate to race given historical causes. [9].

To address these issues, our xFAIR works as follows:

- The algorithm will avoid accessing sensitive attributes during the deployment phase.
- It then artificially recreates those values via an *extrapolation model* learned from other non-protected attributes.

As shown by the results of this paper, xFAIR shows on-par or superior performance compared to the prior state-of-the art (Reweighing and Fair-SMOTE):

- xFAIR provided better bias reduction with as good or better predictive performance;
- xFAIR runs much faster (up to 600%) than Fair-SMOTE;
- xFAIR scales better to larger data sets;
- xFAIR handles multiple protected attributes very well (and the Fair-SMOTE paper notes that managing multiple attributes is an Achilles's heel of that algorithm).

Importantly, xFAIR ensures procedural justice. That is to say, xFAIR needs to access protected attributes during the initial commission stage, but not during deployment. That is, when this system is placed into production, it needs not access users' private information.

The rest of this paper is structured as follows. §3 provides a road-map of background knowledge and related work concerning fairness in ML software. §4 describes the motivation and methodology of our approach in this paper. §5 illustrates the experiment setup used to evaluate our approach along with other benchmarks. §6 shows experiment results. In 8, we elaborate the reasons why xFAIR should be promoted. §7 lists external and internal threats to validity in this paper. Finally, §9 presents our conclusions.

2 Frequently Asked Questions

2.1 Why Does xFAIR Work?

One frequently asked question is as follows. What is won by removing an attribute, then recreating its values via extrapolation from other attributes? Surely this extrapolation model just writes back the same values that were removed?

In reply, we say that the conclusions drawn from the extrapolated data is actually different, in certain small but crucial aspects, to the conclusions drawn from the raw data. In xFAIR, the relation between the protected and non-protected attributes are learned by an extrapolation model. When new data instances arrive in during the testing or deployment phase, xFAIR generates synthetic values for protected attributes to replace actual values. In that approach, any small variations in local data can be "smoothed-out" by sampling across all the data in the extrapolation model. Later in this paper, Figure 2 and Figure 3 show that this kind of smoothing has a critical and significant effect on mitigating bias. Specifically, in those two figures, we look at the unfairness suffered by different social groupings:

- In the test data, unprivileged groups have a much lower chance of receiving a favorable label, while having a much higher chance of receiving an unfavorable label.
- But when using our synthesized data generated from xFAIR, that bias has been dramatically removed, toward an ideal equilibrium between the privileged and unprivileged groups.

From this, we conjecture that biased decisions arises when a model occasionally using a protected attribute to make a decision, when it has no need to. Our experience suggests that we can remove those "occasional mistakes", and thus remove bias. To say that another way:

> *Bias arises when a model"reaches too far" towards some outlier region that was rarely encountered in training.*

Therefore:

> *Bias can be removed via extrapolation that smooths away those outlier points.*

Note that the above is only a brief sketch of how xFAIR works. For full details, see For full details, see §4.3.

2.2 Does xFAIR Handle All Unfairness?

When discussing this work with colleagues, we are often asked in xFAIR can mitigate against all the potential injustices that might be created by AI. In response, we say "no". Mitigating the untoward effects of AI is a much broader problem than just exploring bias in algorithmic decision making (as done in this paper). The general problem of fairness is that influential groups in our society might mandate systems that (deliberately or unintentionally) disadvantage sub-groups within that society. An software systems might satisfy all the evaluation metrics we use to evaluate fairness of (see §2) and still perpetuate social inequities. For example, (a) software license fees might be so expensive that only a small monitory of organizations can boast they are "fair"; or (b) the skills required to use a model's API might be so elaborate that even if the model is fair, only an elite group of programmers can use it.

That said, as software developers, we cannot turn a blind eye to the detrimental social effects of our software. While no single paper can hope to fix all social inequities, this paper shows how to improve the model involved in assessing one particular kind of unfairness (algorithmic decision making bias).

As to other kinds of fairness, they need to be explored and, hopefully, research results like this one will motivate a larger community of researchers to take on the challenge of fairness.

3 Background and Related Work

In this section, we introduce fundamental theories about software fairness, metrics to measure it, and related works attempting to mitigate it.

3.1 Why Software Engineers Cares About Fairness

The rapid development of ML has greatly benefited SE practitioners, and examples of ML-assisted software can be found everywhere: defect prediction models used to locate the most error-prone code files in the upcoming releases; effort estimations tools used to better manage human

TABLE 1: Description of datasets used in this paper.

Dataset	#Features	#Rows	Domain	Protected Attribute	Favorable Label
Adult Census [10]	14	48,842	U.S. census information from 1994 to predict personal income	Sex, Race	Income $> \$50,000$
Compas [11]	28	7,214	Criminal history of defendants to predict re-offending	Sex, Race	Re-offend = false
German Credit [12]	20	1,000	Personal information to predict good or bad credit	Sex	Credit = good
Bank Marketing [13]	16	45,211	Marketing data of a Portuguese bank to predict term deposit	Age	Subscription = yes
Heart Health [14]	14	297	Patient information from Cleveland DB to predict heart disease	Age	Diagnose = yes
Default Credit [15]	23	30,000	Customer information in Taiwan to predict default payment	Sex	Payment = yes
MEPS15 [16]	1831	4,870	Surveys of household members and their medical providers	Race	Utilization >= 10

TABLE 2: Definitions and descriptions of fairness metrics used in this paper

Metric	Definition	Description

Average Odds Difference (AOD)	TPR = TP/(TP + FN), FPR = FP/(FP + TN)
	AOD= $((FPR_U - FPR_P) + (TPR_U - TPR_P))/2$
		Average of difference in False Positive Rates(FPR) and True Positive Rates(TPR) for unprivileged and privileged groups

Equal Opportunity Difference (EOD)	EOD = $TPR_U - TPR_P$	Difference of True Positive Rates(TPR) for unprivileged and privileged groups
Statistical Parity Difference (SPD)	SPD = P (Y = 1$\mid$PA = 0) − P (Y = 1$\mid$PA = 1)	Difference between probability of unprivileged group (protected attribute PA = 0) gets favorable prediction (Y = 1) & probability of privileged group (protected attribute PA = 1) gets favorable prediction (Y = 1)
Disparate Impact (DI)	DI = P [Y = 1—PA = 0]/P [Y = 1—PA = 1]	Similar to SPD but measuring ratio rather than the probability
Flip Rate (FR)	FLIP = Σ(L$\mid$L[PA=0] ≠ L[PA=1])/$total$	The ratio of instances whose predicted label (L) will change when flipping their protected attributes (e.g., PA=1 to PA=0)

and capital resources; multi-objective optimizers used to generate configuration solutions for system of enormous configurable options. Meanwhile, ethical concerns have also drawn increasing attention in the ML and SE communities.

While in many scenarios the only utility needs to be optimized is the performance of the models (in tasks about prediction, classification, ranking, etc.), other cases where private information of human beings are collected need to be handled more carefully. ML software systems have been deployed in many areas to assist make decisions that affect human individuals: Courts and corrections departments in US use software to determine sentence length for defendants [17]; algorithms are used to predict the default payments from credit card users [18]. During such procedure, private information such as age, ethnicity, and gender are collected. Moreover, it has been reveal in prior study that models learned from such data may contain algorithmic bias toward certain social groups.

In response to the above raising issues, IEEE has provoked ethical designs of AI-assisted systems [19] and European Union also announced the ethics guidelines of building trustworthy AI [20]. Fairness has been emphasized in both documents. Big industrial companies such as Facebook [21], Microsoft [22], and Google [23] also have begun to invest effort in ensuring fairness of their products. In academia, IEEE and ACM has set specific tracks [24], [25] for papers studying fairness problems .

3.2 Fairness in ML Software

In this work, we use binary classification models. We define some terms specific to the fairness of binary classification.

- A *favorable label* in a binary classification task is the label that grants the instance (usually human individuals) with privilege such as a job offer or being accepted for a loan.
- A *protected/sensitive attribute* reveals the social groups to which data instances belong, such as gender, race, and age. A binary protected attribute will divide the whole population into *privileged* and *unprivileged* groups in terms of the difference in receiving the favorable label.

The notion of bias comes if the outcome of the classification model gets significantly affected by sensitive/protected attributes. Table 1 shows seven fairness datasets used in this work. These datasets are very popular in the fairness domain and have been used by many prior researchers [27], [1], [28], [29], [30]. All of these datasets contain at least one protected attribute. Depending on that, the population is divided into two groups getting different benefits. For example, in the Adult [10] dataset, there are two protected attributes. Based on "sex", "male" is privileged; Based on "race", "white" is privileged.

The concept of fairness is complicated and very domain-specific. Narayanan [31] has defined 21 different versions of fairness. Based on prior literature [30], [27], [1], among these 21 versions, two specific versions of fairness are widely explored and given most importance. We have decided to explore the same two versions and chose different metrics to evaluate them.

- *Group fairness* requires the approximate equalization of certain statistical property across groups divided by the protected attribute. In this paper, we use 4 group fairness metrics that were widely used in prior research [6], [32], [27], [1], [30].
- *Individual fairness* requires that similar individuals should receive similar prediction outcomes by the ML model. The usual metric for measuring individual fairness is "consistency". But "consistency" is a collective metric based on nearest neighbors. That means it can be calculated for a set of data points, not for a single point. Chakraborty et al. [27] came with a new metric for measuring individual fairness where they measured the FLIP rate which computed the ratio of the population whose prediction outcomes are flipped (e.g., accepted to rejected) when reversing their protected attributes. We decided to use the same metric called *Flip Rate* (FR).

Table 2 contains mathematical definitions of 5 fairness metrics. All the group fairness metrics are calculated based on the confusion matrix of binary classification, which is consisted of four parts: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

3.3 Bias Mitigation

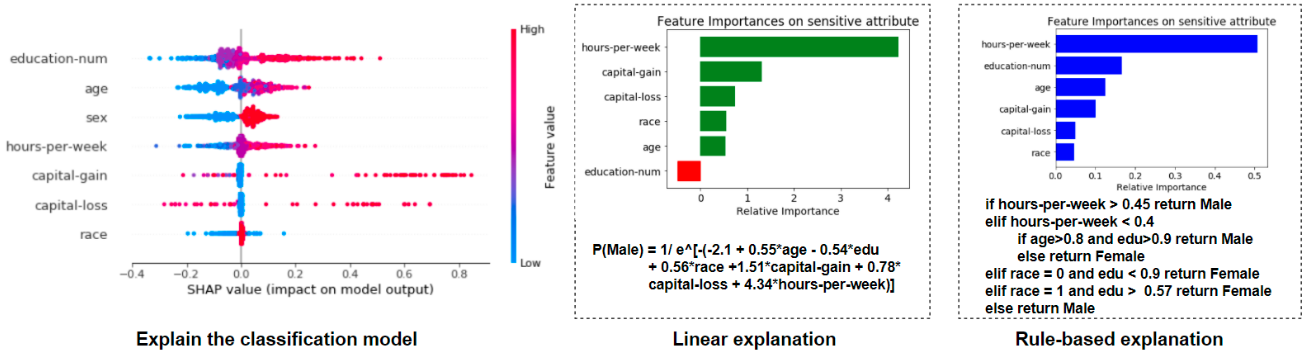
Many researchers endeavor to ensure fairness within their AI decision makeing software. the literature, one can categorize bias mitigation methods into three major groups, depending on when the mitigation procedure is performed.
Pre-processing: Pre-processing algorithms attempt to mitigate the bias of the model by pre-processing the training data that the model learns from. Reweighing was proposed by Kamiran et al. [6] to learn a probabilistic threshold that can generate weights to different instances in training samples according to the (protected and class attributes) combination that each of them belongs to. Fair-SMOTE [1] proposed by Chakraborty et al. [1] re-samples and generates synthetic instances among the training data so that the training data can reach equal distributions not only between different target labels but also among different protected attributes.
In-processing: In-processing methods generally take the optimization approach to mitigate bias. The dataset is typically divided into three parts: training, validation, and testing set. The learner is fitted on the training set and then optimized on the validation set using both performance and fairness metrics as the objectives. Kamishima et al. [33] developed Prejudice Remover which adds a discrimination-aware regularization to the learning objective of the prediction model.
Post-processing: This approach believes that bias can be removed by identifying and then reversing the biased outcomes from the classification model, which means that such methods typically only mutate outcomes of the classification model rather than the model itself. A "reject option classification" approach was proposed by Kamiran et al. [34] to firstly identify the model's decision boundary with the highest uncertainty. Within that region, the method will adjust the ratio between favorable labels on unprivileged groups and unfavorable labels on privileged groups.
This paper works in the same framework as Chakraborty et al [1]. Among the prior works introduced above, almost all approaches (except Fair-SMOTE) are shown to be useful in reducing the bias while also being exposed to the risk of degrading model performance (measured using metrics described in Table 2 and Table 4). As for Fair-SMOTE, Chakraborty et al. show that they can mitigate bias while maintaining the predictive power of the classification model. In summary, we decide to use Reweighing and Fair-SMOTE as baseline methods in this paper since both of them, like our approach, are pre-processing algorithms and Fair-SMOTE is the latest state-of-the-art to our knowledge.

4 Methodology

In this section, we illustrate our design of the proposed bias mitigation algorithm and the intuition behind it.

4.1 Explaining Bias

A prior study has been conducted to extrapolate the relation between the protected attribute and the target attribute (also known as the class label). Creager et al. [35] propose to use disentangled representation learning to identify potential bias-introducing latent in training data that contains mutual information of both targets and protected attributes. They then add regularization on the mutual information while also optimizing for the predictive power. Similarly, Park et al. [36] propose to disentangle information of the target attribute and protected attribute such that target-related information is preserved while protected-related information is removed. It is noteworthy that while both works were empirically tested effective in bias mitigation, the neural-network-based disentanglement approach is barely interpretable, which means the internal disentanglement process cannot be presented to users in a human-comprehensible manner. We view this as a transparency issue and propose an alternative approach. One of the most crucial presumptions in this paper (as well as the fairness domain) is that the protected attribute is essentially irrelevant to the classification problem (e.g., ideally the gender of an individual should not affect the result of the loan application). Based on this presumption, we can deduce that the protected attributes in the training data of some classification problems are informative only because it is a proxy of other relevant information. For example, prior studies believe that one cause of bias is the *negative legacy* which means the training data previously collected is either wrongly labeled or it reflects some discriminatory practices in the past [37]. Either way, when the classification model is trained on such data



P(Male) = 1/ e^[-(-2.1 + 0.55*age - 0.54*edu + 0.56*race +1.51*capital-gain + 0.78* capital-loss + 4.34*hours-per-week)]

```
if hours-per-week > 0.45 return Male
elif hours-per-week < 0.4
    if age>0.8 and edu>0.9 return Male
    else return Female
elif race = 0 and edu < 0.9 return Female
elif race = 1 and edu >  0.57 return Female
else return Male
```

Explain the classification model

Linear explanation

Rule-based explanation

Fig. 1: Example based on the Adult Income dataset. The left side is the explanation on the dependent attribute, in form of SHAP explanations [26] of the classification model. The middle and right blocks show two approaches (logistic regression and decision tree, respectively) applied in xFAIR to explain the influence of other independent attributes on the protected attribute.

and negative legacy disappears in data collect later (either because the data is correctly labeled or the discriminatory practices are eliminated), the model will generate biased outcomes that favor the privileged groups.

Therefore, to investigate the negative legacy potentially embedded in the training data, we decided to explore whether we can reverse engineer the proxy which is represented by the protected attribute. Here our conjecture is

- All the informative attributes have been included in the dataset (which may not be true in most real-world problems);
- Thus, we can infer the protected attributes into some combinations of other non-protected attributes.

For example, as shown in Figure 1, the rule list provided in the Adult income problem reveals that the privileged group (male) is more like to possess higher capital gain and working hours within training data. That is to say, it is possible that the classification model values the protected attribute only because it has a positive correlation with ”high capital balance” and ”more stable job”. That means the protected attribute is simply a kernel of a more profound relationship of multiple actually informative attributes. In that case, we could mitigate bias by removing the ”proxy” and re-emphasizing the importance of those attributes that are represented by the proxy.

Admittedly, our approach cannot guarantee the success of bias mitigation. We believe that, in such a case, the protected attribute might contain some information that has not been collected by the dataset so far. That is, the trade-off between fairness and model performance might be insolvable in that scenario. However, within the scope of the empirical study conducted in this paper, our approach is proven generally effective as supported by the results.

4.2 Using Explanations to Reduce Bias

Now as we can explain the cause of bias in terms of the relationship between the protected and non-protected attributes, we seek for means to mitigate such bias. Our intuition is simple:

> *If the prediction model exhibits bias that comes from data imbalance among protected attributes, we should offset such bias by relabeling the protected attributes (either assigned instances from privileged group to unprivileged group or vice versa) on certain instances.*

To identify the group of instances that require relabelling, we use the extrapolation model trained on the imbalanced training data. Using the Adult dataset as shown in Figure 1 as an example, one of the specific causes of bias here is that the imbalanced data shows a strong correlation between the privileged group ($sex{=}male$) and the number of working hours per week (*hours-per-week*), which is also positively related to the favorable class label. Now assume a new data instance in testing data possesses high *hours-per-week* yet an unprivileged protected attribute ($sex{=}female$). While a high *hours-per-week* attribute value increases the probability of a favorable label, the unprivileged protected attribute will conversely increase the probability of receiving an unfavorable label. Therefore, given information from the extrapolation model, a useful mitigation could be to reassign the instance to the privileged group.

To examine the applicability of our tactic, we conducted some experiments that lead to preliminary results shown in

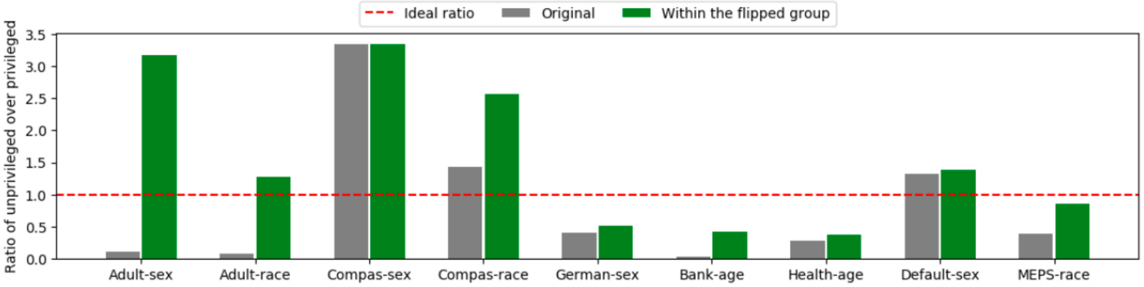


Fig. 2: Ratio for *favorable* labels. The gray bar shows the ratio of unprivileged instances receiving favorable labels among all testing data; The green bar shows the same ratio, but only among instances whose protected attribute values are flipped.

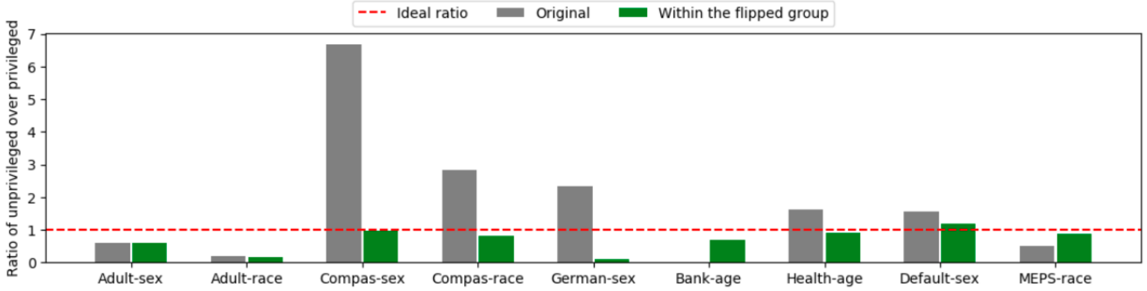


Fig. 3: Ratio for *unfavorable* labels. The gray bar shows the ratio of unprivileged instances receiving unfavorable labels among all testing data; The green bar shows the same ratio, but only among instances whose protected attribute values are mutated using our extrapolation model.

Figure 2 and Figure 3. Figure 2 plots, within testing data of each dataset, the ratio of unprivileged over-privileged groups in receiving favorable labels; Figure 3 plots the ratio of the same two groups in receiving unfavorable labels. The ideal equilibrium is $ratio = 1$, where privileged and unprivileged groups are evenly distributed in both classes. However, as revealed in Figure 2, the unprivileged group is highly under-represented in 6 out of 9 cases. For the other 3 cases (Compas and Default datasets), Figure 3 shows that the unprivileged group suffers from an extremely higher probability of receiving unfavorable labels than the privileged group does. Fortunately, such an imbalanced ratio is diminished within the group of instances whose protected attributes are flipped by our extrapolation model (represented as green bars). Presented by both figures, the flipped group shows either (a) an increased ratio of an unprivileged group receiving favorable labels, or (b) a decreased ratio of an unprivileged group receiving unfavorable labels in each dataset. It is especially noteworthy that in certain datasets (Bank and MEPS) where the ratio is below 1 in both scenarios, the flipped group constantly shows a tendency of moving towards the ideal equilibrium. This indicates that our tactic is self-adaptive for efficiently handling various types of imbalance.

4.3 Our approach: xFAIR

Fair-SMOTE is the prior state-of-the-art method that uses over-sampling algorithms to offset the imbalance among training data. Not only focusing on re-balancing the difference between different target attributes (class labels), Fair-SMOTE also fills the gap among groups of all combinations of protected and target attributes. Despite the effectiveness of Fair-SMOTE, we discovered some shortcomings caused by its design:

- Fair-SMOTE requires generating new samples to offset data imbalance, which can dramatically increase the size of training samples if the original data space is huge and the imbalance is severe. This limits the scalability of Fair-SMOTE.
- When over-sampling data, Fair-SMOTE uses differential evolution process (select, crossover, mutation) to generate synthetic instances. The cost of this mutation process also slows down and limits the scalability of Fair-SMOTE. Moreover, the ground truth of synthetic data is unverifiable, which might introduce noise to training data.
- Since Fair-SMOTE relies on different evolution, there are many parameters and the cost of tuning them can be high depending on the dimensionality of original data.

Inspired by prior approaches, in this paper we proposed a bias mitigation algorithm using the following design choices:

- To facilitate the interpretability of the mitigation model, our approach uses an extrapolation model to extrapolate the correlations among dependent variables that might cause bias in training data. Our current implementation includes two options for the model: decision tree and logistic regression.
- To ensure individual fairness, our approach guarantees absolute procedural justice by not using protected attributes after the model is trained. Instead, our approach will rely on the extrapolation model to generate synthetic protected attributes.
- To increase the scalability, our approach will not oversample the training data. Instead, we will use the extrapolation model to mutate the distribution of protected attributes in testing data so that the originally biased predictions due to imbalanced training data can be offset.

The overview of the proposed approach is shown in Figure 4. Algorithm 1 describes the pseudocode of xFAIR.

Algorithm 1: xFAIR pseudocode

```
Data: $X_{train}$ contains training data without dependent
      attributes; $X_{test}$ contains testing data without
      dependent attributes; $budget$ is the number of
      extrapolation models that can be used for
      weighted-vote the synthetic values
Result: Testing data with synthetic protected attribute
        values $X'_{test}$
begin
    $P_{train}, NP_{train} \leftarrow X_{train}$ // Divide independent
      attributes into protected and non-protected
      attributes
    $M \leftarrow$ InitializeModels($budget$)
    for $i \leftarrow 0$ to $budget$ do
        $(P'_{train}, NP'_{train} \leftarrow$ SMOTE$(P_{train}, NP_{train})$
        $M_i \leftarrow$ FitModel$(P'_{train}, NP'_{train})$
    $P_{test}, NP_{test} \leftarrow X_{test}$
    $P'_{test} \leftarrow$ M.weightedVote$(NP_{test})$
    $X'_{test} \leftarrow$ Append$(P'_{test}, NP_{test})$
    return $X'_{test}$
```

The advantages of our approach over prior methods are obvious: By deploying an extrapolation model to both explain and mitigate bias, xFAIR can offer concise insights on the potential cause of bias. As presented in Figure 1, either linear coefficients or rule-based summaries can be provided to explain why the protected attribute might "deceive" the classification model. Moreover, since xFAIR does not require generating additional synthetic data samples to balance/distort the original training samples, its runtime is much faster than the benchmark methods. Note that xFAIR only uses SMOTE [38] when training the extrapolation model (to better predict protected attributes), and will not affect the training data of the classification model.

5 Experiment Setup

In this section, we describe the data preparation for the experiment as well as the general experiment setup.

5.1 Data

This paper uses collected datasets that are widely used in prior related research (see Table 1). After data collection, we firstly need to pre-process the data. For most of the datasets used in this paper (German, Bank, Heart, Default, and MEPS15), there is no feature engineering required because either the features are all numerical or a standard procedure is adopted by all prior practitioners. For Adult and Compas datasets, there are some variants of pre-processing being proposed by past researchers. Here we did not follow the pre-processing steps mentioned in AIF360 [39], which includes one-hot encoding non-ordinal categorical features.

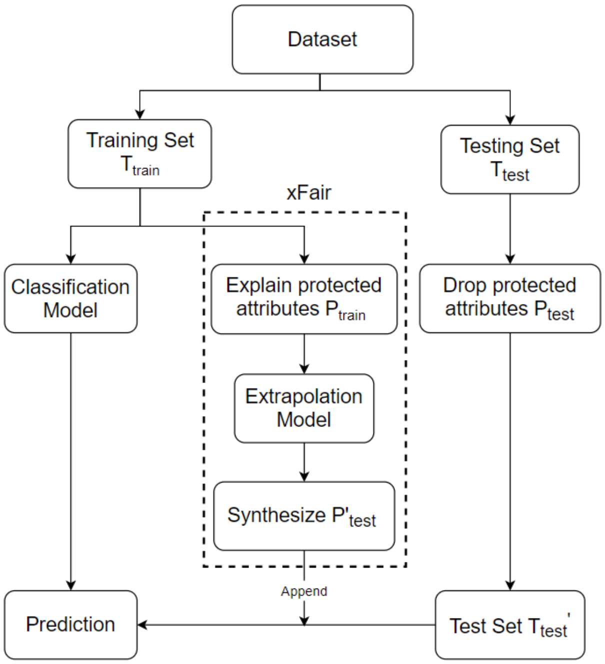


Fig. 4: An overview of xFAIR and the experiment rig in this paper. Note that the synthesized protected attributes are only used for model prediction whereas the real protected attributes are used in computing fairness metrics.

This is because many prior works, including Fair-SMOTE [1] the benchmark method used in this paper, are only applicable on numerical and ordinal categorical features. For example, Fair-SMOTE applies deferential evolutionary algorithms to generate mutants for the purpose of oversampling. Such methods cannot cope with the restraints from one-hot encoded features, and therefore may generate invalid mutants. In short, we removed all non-ordinal categorical features in these two datasets. Similar approaches can also be found in many other previous works [6], [32], [40], [1], [41].

Finally, we apply min-max scaling to transform each dataset. For each experiment trial, we split the data into 80% training data and 20% testing data, using the same set of random seeds on all methods to control the variable of comparison. We repeat this procedure 20 times for statistical analysis.

5.2 Model Selection

In xFAIR. we must select a **classification** model and an **extrapolation** model: The classification model is used to predict the dependent variable in the task using independent variables. The extrapolation model is used by xFAIR to explain and mitigate the bias of the classification model. In Table 3, we explore the interplay of different classification models and extrapolation models in three of the datasets used in this paper. Our initial choices of classification models include random forest (RF), 2-layer neural network (as known as multi-layer perceptron, MLP), and naive Bayes (NB). As for the extrapolation model, we include two highly interpretable models: logistic regression (LR) and classification and regression tree (CART). Indicated by the result, we cannot find an absolute winner among the classification models, which can outperform others in all cases. Moreover, the choice of the extrapolation model has a trivial influence on the final result. In short, our general insight from Table 3 is that (a) the choice of the classification model varies among different datasets, and (b) the performance of xFAIR is robust regardless of the choice of extrapolation model. Hence, in the following experiment, we will use RF as the classification model and CART as the extrapolation model.

5.3 Evaluation Criteria

To evaluate the predictive performance of each method, we use metrics that can be computed via the confusion matrix of binary classification: accuracy, precision, recall, and F1 score. These criteria were selected since (a) they are widely used in both software analytics [42] and fairness literature [43], [44], [29], [30], [45] and (b) we are comparing our results to that of Chakraborty et al.'s FSE'21 paper, which uses the same set of criteria. The definitions of performance metrics are shown in Table 4.

To assess the effectiveness of mitigating bias, we use fairness metrics as introduced in Table 2, some of which are also computed based on the confusion matrix of binary classification. The group fairness metrics aim to evaluate whether different social groups, as identified by their protected attributes, receive statistically similar prediction outcomes by the classification model. The individual fairness metric, denoted as flip rate, is designed based on the intuition of procedural justice. By definition, when individuals that are similar to each other regardless the protected attributes, they shall receive similar prediction outcomes (in this case of binary classification, the same outcome). To assess this criterion, we use the following situation testing tactic:

- For each instance in testing data, flip the protected attribute.
- Pass the edited data instances into the classification model
- Record the times where the new prediction outcome differs from the original one.

It is noteworthy that the situation testing is also used in Fair-SMOTE. The major difference is that Fair-SMOTE uses situation testing as a technique to identify and remove bias-introducing instances whereas in this paper we only use it to assess the extent of individual fairness for all the methods examined in the experiment.

5.4 Statistical Analysis

To compare the predictive performance and ability in mitigating bias among all algorithms on every dataset, we use a non-parametric significance test along with a non-parametric effect size test. Specifically, we use the Scott-Knott test [46] that sorts the list of treatments (in this case, the benchmark bias-mitigation methods and our approach) by their median scores. After the sorting, it then splits the list into two sub-lists. The objective for such a split is to maximize the expected value of differences $E(\Delta)$ in the observed performances before and after division [47]:

$$E(\Delta) = \frac{|l_1|}{|l|} abs(E(l_1) - E(l))^2 + \frac{|l_2|}{|l|} abs(E(l_2) - E(l))^2 \quad (1)$$

where $|l_1|$ means the size of list l_1.

TABLE 3: Preliminary result on choice of the extrapolation model in xFAIR. Here, cells marked in darker colors are better than those marked in lighter colors within the same dataset block. For each dataset, we repeat the experiment for 20 runs and report the median values. The ranks indicated by colors are determined by the Scott-Knott test as described in §5.4.

Dataset: Protected Attribute	Classification Model	Extrapolation Model	Accuracy	Precision	Recall	F1	AOD	EOD	SPD	DI
Adult: Sex	RF	CART	0.83	0.68	0.56	0.61	0.02	0.06	0.11	0.46
		LR	0.83	0.68	0.53	0.59	0.02	0.06	0.11	0.47
	MLP	CART	0.82	0.65	0.51	0.57	0.03	0.06	0.1	0.48
		LR	0.82	0.66	0.48	0.55	0.03	0.07	0.11	0.5
	NB	CART	0.8	0.67	0.3	0.42	0.02	0.01	0.06	0.47
		LR	0.8	0.66	0.31	0.42	0.02	0.01	0.06	0.47
Adult: Race	RF	CART	0.82	0.72	0.53	0.61	0	0.03	0.07	0.36
		LR	0.83	0.71	0.53	0.61	0.01	0.02	0.07	0.35
	MLP	CART	0.83	0.71	0.48	0.57	0	0.02	0.06	0.37
		LR	0.83	0.71	0.49	0.58	0	0.02	0.06	0.35
	NB	CART	0.8	0.67	0.3	0.42	0.02	0.01	0.03	0.28
		LR	0.8	0.67	0.31	0.42	0.02	0.01	0.03	0.3
Compass: Sex	RF	CART	0.65	0.66	0.72	0.69	0	0.05	0.08	0.12
		LR	0.64	0.66	0.74	0.7	0	0.03	0.06	0.1
	MLP	CART	0.67	0.67	0.79	0.73	0.02	0.06	0.11	0.16
		LR	0.68	0.68	0.79	0.73	0.01	0.06	0.11	0.16
	NB	CART	0.67	0.66	0.82	0.73	0.03	0.08	0.14	0.17
		LR	0.68	0.67	0.81	0.73	0.01	0.08	0.13	0.17
Compass: Race	RF	CART	0.64	0.66	0.73	0.69	0.02	0.09	0.13	0.19
		LR	0.64	0.66	0.72	0.69	0.02	0.09	0.13	0.19
	MLP	CART	0.69	0.7	0.76	0.73	0.03	0.12	0.18	0.25
		LR	0.69	0.7	0.76	0.73	0.03	0.12	0.18	0.25
	NB	CART	0.68	0.68	0.79	0.73	0.04	0.11	0.18	0.24
		LR	0.68	0.68	0.78	0.73	0.04	0.11	0.18	0.24
German: Sex	RF	CART	0.7	0.73	0.92	0.81	0.05	0.04	0.08	0.09
		LR	0.69	0.71	0.92	0.8	0.05	0.06	0.09	0.1
	MLP	CART	0.69	0.71	0.93	0.81	0.06	0.04	0.08	0.09
		LR	0.69	0.71	0.92	0.8	0.06	0.04	0.09	0.09
	NB	CART	0.6	0.79	0.59	0.67	0.02	0.11	0.11	0.18
		LR	0.6	0.79	0.59	0.67	0.02	0.11	0.11	0.18

TABLE 4: Performance metrics used in this paper.

Metrics	Definition
Accuray	(TP+TN)/(TP+TN+FP+FN)
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
F1 score	2 × (Precision × Recall)/(Precision + Recall)

The Scott-Knott test assigns ranks to each result set; the higher the rank, the better the result. Two results will be ranked the same if the difference between the distributions is not significant. In this expression, Cliff's Delta estimates the probability that a value in list A is greater than a value in list B, minus the reverse probability [48]. A division passes this hypothesis test if it is not a "small" effect ($Delta \geq 0.147$). This hypothesis test and its effect sizes are supported by Hess and Kromery [49].

6 Results

To assess the effectiveness of our proposed approach as compared to other benchmark methods, we design the experiment evaluation around 3 research questions (RQs).

> **RQ1**: Can we provide human-comprehensible interpretations on the cause of bias?

Prior works either (a) do not offer interpretations on the cause of bias [35], [36], [6], or (b) offer instance-based summary [1] on the cause of bias. Although the latter one is human-comprehensible, we aim for generating more concise and structured interpretations via the extrapolation model. As shown in §4, we proposed an approach that explains the cause of bias in training data by extrapolating the correlation between the protected attributes and non-protected attributes. The result has provided evidence that supports the presumption, which is that the privileged group may share a similar latent with the favorable-labeled group [35], [36], as indicated by the explanations on both the target attribute (label) and the protected attribute. Such similarity within training data may mislead the classification model to wrongly emphasize the importance of the protected attribute, which is essentially a proxy of a combination of other informative attributes. In short, the answer to RQ1 is: **Yes, we can provide human-comprehensible interpretations on the cause of bias.**

> **RQ2**: Can we use the **RQ1** results to mitigate bias?

After obtaining interpretations on the cause of bias, we further attempt to use that knowledge to offset the biased behavior of the classification model. More specifically, we utilize the interpretations learned from extrapolation to relabel the protected attribute of testing data, such that we

TABLE 5: Results for RQ3. RF denotes the default random forest learner. For all performance metrics, greater is better; for all fairness metrics, smaller is better. Here, cells marked in darker colors are better than those marked in lighter colors within the same dataset block. For each dataset, we repeat the experiment for 20 runs and report the median values. The ranks indicated by colors are determined by the Scott-Knott test as described in §5.4.

Dataset: Protected Attribute	Method	Accuracy	Precision	Recall	F1 score	AOD	EOD	SPD	DI	FR
Adult: Sex	RF	0.83	0.72	0.53	0.61	0.08	0.24	0.18	0.78	0.20
	RF+Random	0.83	0.75	0.48	0.57	0.01	0.04	0.11	0.52	0.07
	RF+Reweighing	0.75	0.48	0.71	0.57	0.01	0.07	0.15	0.37	0.02
	RF+Fair-SMOTE	0.79	0.54	0.71	0.61	0.06	0.22	0.20	0.54	0.18
	RF+xFAIR	0.83	0.67	0.56	0.61	0.02	0.06	0.10	0.46	0
Adult: Race	RF	0.84	0.73	0.53	0.61	0.03	0.10	0.09	0.49	0.09
	RF+Random	0.83	0.70	0.51	0.59	0	0.03	0.08	0.43	0.08
	RF+Reweighing	0.76	0.49	0.72	0.59	0.03	0.05	0.05	0.12	0.04
	RF+Fair-SMOTE	0.79	0.54	0.72	0.62	0.02	0.07	0.11	0.37	0.16
	RF+xFAIR	0.82	0.72	0.53	0.61	0	0.03	0.07	0.36	0
Compas: Sex	RF	0.65	0.67	0.73	0.70	0.05	0.10	0.14	0.19	0.28
	RF+Random	0.64	0.66	0.71	0.68	0	0.08	0.11	0.16	0.27
	RF+Reweighing	0.62	0.64	0.67	0.66	0.03	0.07	0.12	0.18	0.30
	RF+Fair-SMOTE	0.65	0.67	0.70	0.68	0	0.06	0.09	0.17	0.21
	RF+xFAIR	0.65	0.66	0.72	0.69	0	0.06	0.08	0.12	0
Compas: Race	RF	0.65	0.67	0.73	0.70	0.02	0.10	0.14	0.20	0.24
	RF+Random	0.64	0.66	0.73	0.69	0.02	0.10	0.14	0.20	0.26
	RF+Reweighing	0.63	0.66	0.66	0.66	0.01	0.02	0.05	0.09	0.19
	RF+Fair-SMOTE	0.65	0.68	0.70	0.69	0.03	0.04	0.13	0.15	0.18
	RF+xFAIR	0.64	0.67	0.73	0.69	0.02	0.08	0.13	0.19	0
German: Sex	RF	0.70	0.72	0.93	0.81	0.05	0.07	0.11	0.11	0.14
	RF+Random	0.67	0.71	0.90	0.79	0.02	0.01	0.08	0.09	0.13
	RF+Reweighing	0.64	0.77	0.71	0.73	0.08	0	0.15	0.26	0.08
	RF+Fair-SMOTE	0.58	0.79	0.55	0.65	0.06	0.06	0.09	0.18	0.18
	RF+xFAIR	0.70	0.73	0.92	0.81	0.05	0.04	0.08	0.09	0
Bank: Age	RF	0.80	0.78	0.82	0.8	0.06	0.09	0.26	0.55	0.31
	RF+Random	0.80	0.77	0.81	0.79	0.07	0.10	0.10	0.21	0
	RF+Reweighing	0.77	0.74	0.79	0.76	0.04	0.02	0.20	0.40	0.03
	RF+Fair-SMOTE	0.80	0.77	0.82	0.80	0.02	0.06	0.22	0.44	0.13
	RF+xFAIR	0.80	0.78	0.81	0.80	0.05	0.07	0.11	0.22	0
Health: Age	RF	0.83	0.86	0.78	0.81	0.10	0.06	0.32	0.55	0.07
	RF+Random	0.83	0.85	0.78	0.81	0.08	0.02	0.24	0.44	0.03
	RF+Reweighing	0.76	0.71	0.82	0.75	0.11	0.11	0.38	0.54	0.02
	RF+Fair-SMOTE	0.84	0.86	0.79	0.82	0.08	0.04	0.28	0.48	0.03
	RF+xFAIR	0.84	0.86	0.79	0.82	0.05	0.04	0.24	0.43	0
Default: Sex	RF	0.82	0.66	0.37	0.47	0.01	0.02	0.02	0.20	0.03
	RF+Random	0.81	0.65	0.37	0.47	0.01	0.02	0.02	0.18	0.02
	RF+Reweighing	0.42	0.26	0.88	0.40	0	0	0.02	0.03	0
	RF+Fair-SMOTE	0.82	0.62	0.42	0.50	0	0.02	0.03	0.15	0.02
	RF+xFAIR	0.82	0.65	0.37	0.47	0	0	0.02	0.16	0
MEPS: Race	RF	0.87	0.66	0.39	0.49	0.02	0.07	0.04	0.33	0.01
	RF+Random	0.86	0.65	0.39	0.49	0.01	0.03	0.02	0.23	0.01
	RF+Reweighing	0.76	0.4	0.76	0.52	0.01	0.05	0.05	0.14	0
	RF+Fair-SMOTE	0.87	0.64	0.47	0.54	0.02	0.05	0.04	0.30	0.02
	RF+xFAIR	0.87	0.67	0.39	0.49	0.01	0.03	0.03	0.25	0

TABLE 6: Summarized result of RQ3. Each cell is the mean rank across all datasets. Lower ranks are better and highlighted in darker colors. xFAIR constantly obtains top ranks in all metrics.

	Accuracy	Precision	Recall	F1 Score	AOD	EOD	SPD	DI	FR
RF	1.0	1.2	1.8	1.2	2.3	2.8	2.6	2.8	2.8
RF+Random	1.1	1.2	2.1	1.6	1.6	1.7	1.4	2.1	2.3
RF+Reweighing	2.1	2.3	1.8	2.1	1.8	1.6	2.0	1.9	1.9
RF+Fair-SMOTE	1.6	1.6	1.8	1.3	1.7	1.9	2.2	2.2	2.6
RF+xFAIR	1.0	1.3	1.8	1.2	1.3	1.3	1.2	1.6	1.0

expect certain data instances (that receive biased outcomes due to their protected attributes) to receive a different classification outcome due to the change of the protected attribute. Figure 2 and Figure 3 show our preliminary result. Among the relabeled testing data, the unprivileged group has an increased possibility to receive a favorable label, and the gap of the favorable label rates between the privileged and unprivileged group is reduced. This result is consistent with the following:

- The bias of training data is related to the imbalanced distribution between the privileged and unprivileged group in receiving favorable and/or unfavorable labels.
- Such bias can be largely offset when relabeling the protected attribute assigns instances to a different group.

Thus, our answer to RQ2 is: **Yes, it is viable to use such interpretations to further mitigate bias.**

> **RQ3**: How is the performance of our approach compared to other benchmarks, including state-of-the-art algorithms?

Table 5 compares xFAIR against other baselines. Reweighing [6] is proposed by Kamiran et al. (introduced in §3.3) to mitigating bias via adjusting the instance weights for training samples in different groups (as determined by both their labels and protected attributes). Fair-SMOTE [1] is proposed by Chakraborty et al. to reduce bias by not only handling the data imbalance between target labels but also imbalanced distribution among different protected attributes. We chose them as our benchmark methods because (a) like xFAIR, they both belong to the pre-processing category, and (b) prior results [28] have shown that some widely cited in-processing and post-processing methods are outperformed by a pre-processing method, which is later beat by Fair-SMOTE as well. In addition to these state-of-the-art methods, we also implement a ”naive” baseline in our experiment, denoted as ”Random”, that randomly shuffles the protected attributes.

Comparing xFAIR against other baselines, we observe that the result is either (a)xFAIR outperforms other methods in fairness metrics while maintaining the performance (in most cases, the default learner has top-ranked performance), or (b) xFAIR reaches on-par fairness measures with some other baselines but obtains better performance at the same time. The summarized result is presented in Table 6, where we can find that xFAIR constantly obtains top ranks in both fairness and performance. It is also noteworthy that xFAIR, by its design choices, can achieve perfect individual fairness while other baselines fail to improve it (even worsen in some cases).

Thus, our answer to RQ3 is **xFAIR performs better or similar to the two state-of-the-art algorithms in terms of both fairness and performance.**

> **RQ4**: Is xFAIR more scalable than Fair-SMOTE in terms of runtime complexity?

xFAIR is built upon design choices that avoid synthetic data generation. This not also avoids the potential risk of introducing noise but also makes the whole framework more light-weighted. Figure 5 presents the runtime of xFAIR as compared to that of Fair-SMOTE on every dataset. While the datasets are sorted by their sizes, we do not see a proportional relationship between the size and runtime in either method. This could be because the runtime of the model is also influenced by the dimensionality of the training space. Moreover, since Fair-SMOTE requires generating additional data, its runtime also depends on the extent of data imbalance: In cases where the data distribution is severely imbalanced, more synthetic data are required. Nevertheless, despite the variables described above, we can still observe apparent domination that xFAIR runs constantly faster than Fair-SMOTE, which aligns with our design expectation. Thus, our answer to RQ4 is **xFAIR has a significantly shorter runtime, and therefore more scalable than Fair-SMOTE.**

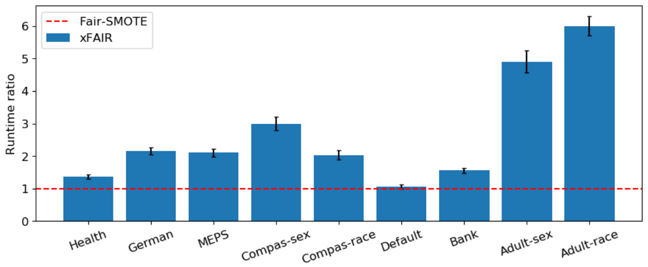


Fig. 5: Result for RQ4. The ratio is calculated by dividing the runtime of Fair-SMOTE over that of xFAIR. The datasets are sorted by the size in an ascending order.

> **RQ5**: Can xFAIR handle multiple protected attributes?

While under-represented in past research, it is a possible case scenario that a dataset contains more than one protected attributes in a dataset (just like Adult and Compas datasets). Fortunately, our design choices of xFAIR makes itself extremely easy to be applied to cases with more than one protected attribute.

To examine the effectiveness of xFAIR, we conducted experiments on Adult and Compas datasets, both of which contain two protected attributes: race and sex. Following our framework described in 4, we now need to build two extrapolation models for the two protected attributes respectively. After that, we will drop both protected attributes from the test data. As shown in Table 7, xFAIR can improve fairness in both protected attributes simultaneously while maintaining the predictive performance. It is noteworthy that Fair-SMOTE can also handle bias mitigation for multiple protected attributes. However, since it uses the over-sampling tactic to reduce bias (in order to achieve balance among different combinations of protected and target attributes), the number of samples needed to over-sample explodes exponentially as the dimensionality of protected attributes increases. Thus, our answer to RQ5 is **xFAIR shows significant efficiency in bias mitigation when handling more than one protected attributes simultaneously.**

7 Threats to Validity

Sampling Bias - While experimenting with other datasets may yield different results, we believe our extensive study

TABLE 7: Result for RQ5. In Adult and Compas datasets, xFAIR can mitigate bias for two protected attributes simultaneously. Similar to Table 5, here cells with significantly better results are marked in a darker color.

Dataset	Method	Protected Attribute	Accuracy	Precision	Recall	F1	AOD	EOD	SPD	DI	FR
Adult	RF	Sex	0.83	0.72	0.53	0.61	0.08	0.24	0.18	0.78	0.20
		Race					0.03	0.10	0.09	0.49	0.09
	RF+xFAIR	Sex	0.83	0.69	0.52	0.59	0.02	0.06	0.11	0.49	0
		Race					0	0.02	0.06	0.34	0
Compas	RF	Sex	0.65	0.67	0.73	0.70	0.05	0.10	0.14	0.19	0.28
		Race					0.02	0.10	0.14	0.20	0.24
	RF+xFAIR	Sex	0.65	0.66	0.73	0.69	0.01	0.05	0.09	0.14	0
		Race					0.02	0.09	0.13	0.19	0

here has shown the constant effectiveness of xFAIR in various cases. Most of the prior works [4], [50], [51], [34], [28] used one or two datasets where we used seven well-known datasets in our experiments. We have also observed other emerging datasets in the fairness fields, and we will try to extend our research scope once we verify the validity of the new datasets. In the future, we will explore more datasets and more learners.

Evaluation Bias - We used the five fairness metrics in this study, covering both definitions of group and individual fairness. Prior works [27], [43], [33] only used two or three metrics whereas IBM AIF360 [39] contains more than 50 metrics. More evaluation criteria will be examined in future work.

Conclusion Validity - Our experiments are based on the assumption that test data is unbiased and correctly labeled. Prior fairness studies also made the similar assumption [1], [30], [28].

Internal Validity - We used random forest model with mostly off-the-shelf parameters. However, hyperparameters play a crucial role in the performance of ML models. Therefore, we cannot rule out the possibility that other ML models, after fine tuning, can achieve superior results. In the future, we will endeavor to address hyperparameter optimization for performance improvement. Moreover, our feature processing step during the experiment follows procedures found in prior works, especially those that are compared in this paper only in order to make a fair comparison. While other benchmark methods may have certain limitation in selecting features, our approach is actually applicable to all kinds of features.

External Validity - Our work is limited to binary classification and tabular data which are very common in AI software. However, all the methods used in this paper can easily be extended in case of multi-class classification, and regression problems. In the future, we will try to extend our work to other domains of SE and ML.

8 Discussion: Why xFAIR?

In this section, we discuss what makes xFAIR novel and distinguishable from prior works in this research domain.

8.1 Procedural Justice

The idea of procedural justice is originally defined in law practice [2], [52], and recently became a part of the discussion of building fairer ML models [3], [53]. In this paper, we found that our design choice of improving individual fairness has made xFAIR a suitable fit to satisfy procedural justice. By definition, procedural justice requires not only fair results but also transparency of the decision-making process such that ones can verify whether the procedure guarantees fairness [2], [3]. As stated in our introduction, xFAIR's concept of operation offers that transparency:

- xFAIR only collects and uses those protected attributes *to initially build its model*
- After that, during deployment, our method does not demand access to protected attributes in any subsequent test data.

Also note that group fairness metrics are more likely to reflect *distributive justice*, which concerns fairness in terms of the distribution of rights [54] (in our case, the distribution of favorable labels among different social groups). From this perspective, xFAIR satisfies the standard of procedural justice in a manner that does not require the access of protected attributes from testing data. This ensures that the decision-making of xFAIR will not be affected by the change of protected attributes at all, as also reflected by the Flip Rate (FR). As indicated by the experiment result, no other benchmark method is designed to address this problem.

8.2 Ethical Concerns?

We can foresee the potential criticism that our approach might face: does xFAIR merely *hide* the bias since we while our data no longer has (e.g.) gender, race, etc. it still retains the influences of those attributes? We say that this is demonstrably not true. An important feature of the assessment methodology is that:

- When we assess the fairness of xFAIR (in §5)...
- .... that assessment uses all the protected attributes on the unaltered data.

Hence we can assert that our synthesis approach not only enables the procedural justice (discussed in the introduction) but it also reduces the measurable effects of unfairness.

8.3 Explainable Extrapolation

One of the major contributions of xFAIR that differs itself from prior work is the interpretability of the internal process. Prior state-of-the-art methods (Fair-SMOTE and Reweighing) mitigate bias via either over-sampling or adjusting instance weights among training data. Despite the effectiveness, such approaches make it difficult for human users to reason the mitigation process because they cannot summarize the exact representation of bias in the existing data.

xFAIR, on the other hand, makes its mitigations more obvious. As shown in Figure 1, xFAIR can provide different kinds of interpretations based on the type of extrapolation models that users can customize.

9 Conclusion

Fairness in machine learning software has become a serious concern in the software engineering community. Many fairness methods synthesize new samples [1], [55], [56] in order to better balance training data (expecting to remove certain biases). This paper tested the conjecture that such synthesis might work better if it was *model-based* rather than *instance-based* (since the latter is more susceptible to minor variations in the data).

Our results explored that conjecture. We found that we can endorse Chakraborty et al. [1] findings that bias might come from imbalanced data distributions. Moreover, instead of generating new data samples, we proposed xFAIR. a better and faster approach that outperforms Fair-SMOTE. In addition, our approach also guarantees absolute procedural fairness. That is to say, by avoiding to use the real protected attributes and synthesizing our own ones, our model can ensure that individuals that are only different in protected attributes (the real ones) will receive the same predictions. This is a significant improvement, because as revealed by our situational testing, sometimes such individual unfairness can exist among up to 20% of the test data. On experimentation, we found that:

- xFAIR is performance-wise better (measured by fairness and performance metrics) than two state-of-the-art fairness algorithms
- xFAIR is also more interpretable than them in the way it can present concise explanations on the cause of bias.
- When looking at individual fairness (as indicated by *flip rate*), xFAIR can ensure perfect individual fairness while other benchmarks cannot.

Based on the above, we conclude that:

- We can recommend xFAIR for bias mitigation.
- Rather than a black-box model, we can make mitigation procedure interpretable to achieve not only better fairness but also transparency and accountability of the mitigation model.
- We ensure that our model shows zero individual unfairness: Two individuals who only differ in the protected attributes will always receive the same prediction outcomes.

Acknowledgements

This work was partially funded by a research grant from the Laboratory for Analytical Sciences, North Carolina State University.

References

[1] J. Chakraborty, S. Majumder, and T. Menzies, “Bias in machine learning software: Why? how? what to do?” in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 429–440. [Online]. Available: https://doi.org/10.1145/3468264.3468537

[2] T. R. Tyler and E. A. Lind, “Procedural justice,” in *Handbook of justice research in law*. Springer, 2002, pp. 65–92.

[3] M. K. Lee, A. Jain, H. J. Cha, S. Ojha, and D. Kusbit, “Procedural justice in algorithmic fairness: Leveraging transparency and outcome control for fair algorithmic mediation,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. CSCW, pp. 1–26, 2019.

[4] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, and K. R. Varshney, “Optimized pre-processing for discrimination prevention,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3992–4001. [Online]. Available: http://papers.nips.cc/paper/6988-optimized-pre-processing-for-discrimination-prevention.pdf

[5] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[6] F. Kamiran and T. Calders, “Data preprocessing techniques for classification without discrimination,” *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, 2012.

[7] ——, “Classifying without discriminating,” in *2009 2nd international conference on computer, control and communication*. IEEE, 2009, pp. 1–6.

[8] N. Kilbertus, M. Rojas-Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf, “Avoiding discrimination through causal reasoning,” *arXiv preprint arXiv:1706.02744*, 2017.

[9] L. Zhang, Y. Wu, and X. Wu, “A causal framework for discovering and removing direct and indirect discrimination,” *arXiv preprint arXiv:1611.07509*, 2016.

[10] “Uci:adult data set,” 1994. [Online]. Available: http://mlr.cs.umass.edu/ml/datasets/Adult

[11] “propublica/compas-analysis,” 2015. [Online]. Available: https://github.com/propublica/compas-analysis

[12] “Uci:statlog (german credit data) data set,” 2000. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

[13] “Bank marketing uci,” 2017. [Online]. Available: https://www.kaggle.com/c/bank-marketing-uci

[14] “Uci:heart disease data set,” 2001. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Heart+Disease

[15] “Uci:default of credit card clients data set,” 2016. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[16] “Medical expenditure panel survey,” 2015. [Online]. Available: https://meps.ahrq.gov/mepsweb/

[17] A. Feller, E. Pierson, S. Corbett-Davies, and S. Goel, “A computer program used for bail and sentencing decisions was labeled biased against blacks. it’s actually not that clear,” *The Washington Post*, vol. 17, 2016.

[18] I.-C. Yeh and C.-h. Lien, “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

[19] K. Shahriari and M. Shahriari, “Ieee standard review—ethically aligned design: A vision for prioritizing human wellbeing with artificial intelligence and autonomous systems,” in *2017 IEEE Canada International Humanitarian Technology Conference (IHTC)*. IEEE, 2017, pp. 197–201.

[20] E. Commission, C. Directorate-General for Communications Networks, and Technology, *Ethics guidelines for trustworthy AI*. Publications Office, 2019.

[21] “Facebook says it has a tool to detect bias in its artificial intelligence,” 2018. [Online]. Available: https://qz.com/1268520/facebook-says-it-has-a-tool-to-detect-bias-in-its-artificial-intelligence/

[22] “Fate: Fairness, accountability, transparency, and ethics in ai,” 2018. [Online]. Available: https://www.microsoft.com/en-us/research/group/fate/

[23] T. Simonite, “Google offers to help others with the tricky ethics of ai,” *Ars Technica*, vol. 29, 2020.

[24] “Acm conference on fairness, accountability, and transparency (acm fat*).” [Online]. Available: https://fatconference.org/

[25] “Explain 2019.” [Online]. Available: https://2019.ase-conferences.org/home/explain-2019

[26] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in neural information processing systems*, 2017, pp. 4765–4774.

[27] J. Chakraborty, S. Majumder, Z. Yu, and T. Menzies, “Fairway: A way to build fair ml software,” in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 654–665. [Online]. Available: https://doi.org/10.1145/3368089.3409697

[28] J. Chakraborty, T. Xia, F. M. Fahid, and T. Menzies, “Software engineering for fairness: A case study with hyperparameter optimization,” 2019.

[29] J. Chakraborty, K. Peng, and T. Menzies, “Making fair ml software using trustworthy explanation,” in *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1229–1233. [Online]. Available: https://doi.org/10.1145/3324884.3418932

[30] S. Biswas and H. Rajan, “Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness,” *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Nov 2020. [Online]. Available: http://dx.doi.org/10.1145/3368089.3409704

[31] A. Narayanan, “Tl;ds - 21 fairness definition and their politics by arvind narayanan,” 2019.

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

[33] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Machine Learning and Knowledge Discovery in Databases*, P. A. Flach, T. De Bie, and N. Cristianini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 35–50.

[34] F. Kamiran, S. Mansha, A. Karim, and X. Zhang, "Exploiting reject option in classification for social discrimination control," *Inf. Sci.*, 2018.

[35] E. Creager, D. Madras, J.-H. Jacobsen, M. Weis, K. Swersky, T. Pitassi, and R. Zemel, "Flexibly fair representation learning by disentanglement," in *International conference on machine learning*. PMLR, 2019, pp. 1436–1445.

[36] S. Park, D. Kim, S. Hwang, and H. Byun, "Readme: Representation learning by fairness-aware disentangling method," *arXiv preprint arXiv:2007.03775*, 2020.

[37] T. Kamishima, S. Akaho, and J. Sakuma, "Fairness-aware learning through regularization approach," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 643–650.

[38] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, 2002.

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

[40] F. P. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3995–4004.

[41] R. Salazar, F. Neutatz, and Z. Abedjan, "Automated feature engineering for algorithmic fairness," *Proceedings of the VLDB Endowment*, vol. 14, no. 9, pp. 1694–1702, 2021.

[42] T. Menzies, E. Kocaguneli, B. Turhan, L. Minku, and F. Peters, *Sharing data and models in software engineering*. Morgan Kaufmann, 2014.

[43] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," 2016.

[44] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," *arXiv preprint arXiv:1709.02012*, 2017.

[45] P. Zhang, J. Wang, J. Sun, G. Dong, X. Wang, X. Wang, J. S. Dong, and T. Dai, "White-box fairness testing through adversarial sampling," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 949–960.

[46] N. Mittas and L. Angelis, "Ranking and clustering software cost estimation models through a multiple comparisons algorithm," *IEEE Transactions on software engineering*, vol. 39, no. 4, pp. 537–551, 2012.

[47] T. Xia, R. Krishna, J. Chen, G. Mathew, X. Shen, and T. Menzies, "Hyperparameter optimization for effort estimation," *arXiv preprint arXiv:1805.00336*, 2018.

[48] G. Macbeth, E. Razumiejczyk, and R. D. Ledesma, "Cliff's delta calculator: A non-parametric effect size program for two groups of observations," *Universitas Psychologica*, vol. 10, no. 2, pp. 545–555, 2011.

[49] M. R. Hess and J. D. Kromrey, "Robust confidence intervals for effect sizes: A comparative study of cohen'sd and cliff's delta under non-normality and heterogeneous variances," in *annual meeting of the American Educational Research Association*, 2004, pp. 1–30.

[50] S. Galhotra, Y. Brun, and A. Meliou, "Fairness testing: testing software for discrimination," *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2017*, 2017. [Online]. Available: http://dx.doi.org/10.1145/3106237.3106277

[51] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," 2018.

[52] L. B. Solum, "Procedural justice," *S. CAl. l. reV.*, vol. 78, p. 181, 2004.

[53] S. K. Ötting and G. W. Maier, "The importance of procedural justice in human–machine interactions: Intelligent systems as new decision agents in organizations," *Computers in Human Behavior*, vol. 89, pp. 27–39, 2018.

[54] K. S. Cook and K. A. Hegtvedt, "Distributive justice, equity, and equality," *Annual review of sociology*, vol. 9, no. 1, pp. 217–241, 1983.

[55] V. Zelaya, P. Missier, and D. Prangle, "Parametrised data sampling for fairness optimisation," *KDD XAI*, 2019.

[56] F. Kamiran and T. Calders, "Classification with no discrimination by preferential sampling," in *Proc. 19th Machine Learning Conf. Belgium and The Netherlands*. Citeseer, 2010, pp. 1–6.