Towards Fair and Robust Classification

Haipei Sun
Facebook Inc.
Seattle, WA
haipei@fb.com

Kun Wu
Stevens Institute of Technology
Hoboken, NJ
kwu14@stevens.edu

Ting Wang
Penn State University
University Park, PA
ting@psu.edu

Wendy Hui Wang
Stevens Institute of Technology
Hoboken, NJ
Hui.Wang@stevens.edu

***Abstract*—Robustness and fairness are two equally important issues for machine learning systems. Despite the active research on robustness and fairness of ML recently, these efforts focus on either fairness or robustness, but not both. To bridge this gap, in this paper, we design Fair and Robust Classification (FRoC) models that equip the classification models with both fairness and robustness. Meeting both fairness and robustness constraints is not trivial due to the tension between them. The trade-off between fairness, robustness, and model accuracy also introduces additional challenge. To address these challenges, we design two FRoC methods, namely FRoC-Pre that modifies the input data before model training, and FRoC-In that modifies the model with an adversarial objective function to address both fairness and robustness during training. FRoC-In is suitable to the settings where the users (e.g., ML service providers) only have the access to the model but not the original data, while FRoC-Pre works for the settings where the users (e.g., data owners) have the access to both data and a surrogate model that may have similar architecture as the target model. Our extensive experiments on real-world datasets demonstrate that both FRoC-In and FRoC-Pre can achieve both fairness and robustness with insignificant accuracy loss of the target model.**

***Index Terms*—Algorithmic fairness, adversarial robustness, classification, trustworthy machine learning.**

1. Introduction

Machine learning (ML) techniques are increasingly providing decision making and operational support across multiple domains and applications. One important concern for the adoption of ML techniques into operational decision processes is the trustworthiness of these techniques. Recent years have seen a proliferation of research in trustworthy ML. Two important issues of trustworthy ML are *fairness* and *robustness*. On one hand, ML models have been criticized for systemic biases that result in unintentional "unfair" decisions that favor particular individuals or groups of individuals while discriminating against others [1], [2]. On the other hand, ML models are vulnerable to those carefully crafted adversarial examples [3], [4] and thus can be easily misled and manipulated. The discovery that ML models are neither fair nor robust hinders significantly their practical deployment in the security-critical applications such as healthcare, finance, and transportation. Therefore, ensuring both fairness and robustness of ML models is paramount to the widespread adoption of ML in our society.

The goal of this paper is to equip the ML model with both fairness and robustness simultaneously. We consider classification models on tabular data as our target model. In terms of fairness, we consider statistical parity [5], a widely-used notion in the fairness literature, as our fairness definition. Intuitively, statistical parity specifies a *protected group* (e.g., females) and an *un-protected group* (e.g., males) by using a *protected attribute* (e.g., gender), and requires that both protected and un-protected groups should receive the positive outcome at equal rates. In terms of robustness, we consider the robustness against two popular evasion attacks, namely *Fast Gradient Sign Method* (FGSM) [6] and *Projected Gradient Descent* (PGD) attack [7]. The adversary of these two attacks aims to perturb test inputs to ML classifiers to cause misclassification.

There has been a considerable body of studies (e.g., [3], [5], [6], [8], [9]) that realize either fairness or robustness with classification models. Some recent works [10]–[12] have considered the interaction between robustness and fairness, but mainly focus on the image domain. Realizing both fairness and robustness on classification over tabular data has not been investigated yet. As a motivating example, consider the criminal justice scenario and a widely used criminal risk assessment tool named Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) in this scenario [13]. COMPAS predicts a defendant's risk of committing a misdemeanor or felony within two years of assessment from the information about the defendant and his/her prior criminal history (in tabular format). In this setting, the fairness concern is whether the classification model has discrimination towards a particular demographic group by overpredicting/underpredicting the recidivism risk of the members in the group [13], while the robustness issue is whether the attacker can help some particular defendants to escape from receiving the proper justice treatment by misleading the model to predict low recidivism risk for these defendants.

A straightforward solution to equip the target model with both fairness and robustness is to realize the two requirements on the target model in a sequential fashion, i.e., ensuring the classifiers satisfy one requirement first before handling the other. Although the solution is seemly sound, it is indeed incorrect due to the tension between fairness and robustness. For example, recent works [10], [11] have observed that equipping the ML models with fairness can make these models to be more susceptible

2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P) | 978-1-6654-1614-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/EUROSP53844.2022.00030

© 2022, Haipei Sun. Under license to IEEE.
DOI 10.1109/EuroSP53844.2022.00030

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

to data poisoning attacks. Our study (will be presented later in this paper) also shows that deploying the defense mechanisms against adversarial examples on fair ML classifiers can indeed bring bias and make the fair models to be unfair.

To address the tension between fairness and robustness, in this paper, we present the design of Fair and Robust classification (FROC) models that satisfy both fairness and robustness constraints jointly. We define two measurements, namely *bias score* and *robustness score*. The bias score measures model fairness in terms of statistical parity [5], while the robustness score measures model robustness against two types of evasion attacks (FGSM and PGD). Based on the two measurements, we formalize the definition of δ_F-*fairness* and δ_R-*robustness* as our fairness and robustness goals, which require that the bias score and the robustness score should meet the user-specified thresholds δ_F and δ_R respectively. Then we formalize our problem as an optimization problem that maximizes model accuracy while satisfying both δ_F-*fairness* and δ_R-*robustness*.

We design two different FROC solutions which are deployed at different phases of the ML pipeline to satisfy both δ_F-*fairness* and δ_R-*robustness* requirements: (1) A *pre-processing* method (FROC-PRE) that is deployed before training - it modifies the training data so that the models trained on that data will be fair and robust; and (2) An *in-processing* method (FROC-IN) that is deployed during training - it modifies the objective function of the target model to address both fairness and robustness constraints. Figure 1 illustrates both methods at high level. While these two methods can be applied at different phases of the ML pipeline in a centralized setting, they also can be applied to different parties in the distributed Machine-Learning-as-a-Service (MLaaS) setting [14], [15] where a third-party service provider (server) provides a cloud-based platform and machine learning tools as services to the end users. In particular, FROC-IN is suitable to the MLaaS service provider who has the access to the model but not to the training data. On the other hand, FROC-PRE is suitable to the data owner (client) who has the access to the training data but does not have access to the target model. However, the client may have the access to a surrogate model which approximates the target model's behaviors and output. To ensure that the server will generate a fair and robust model as the service, the client is willing to process her training data by utilizing the surrogate model before outsourcing the data to the server. The surrogate model can have the same loss function as the target model but of different architecture or the same architecture type but different architectural complexity. For example, consider a neural network (NN) model as the target model for binary classification, the data owner can construct either a logistic regression classifier or an NN whose architecture is much simpler than that of the target model (e.g., fewer layers and/or neurons).

For the FROC-IN method, we design two regularizers: (1) the *fairness regularizer* for statistical parity, and (2) the *robustness regularizer* for the evasion attacks (FGSM and PGD). We add both regularizers to the target model so that it is trained with an adversarial objective function. We address the trade-off between fairness, robustness, and accuracy by controlling the weights of both regularizers.

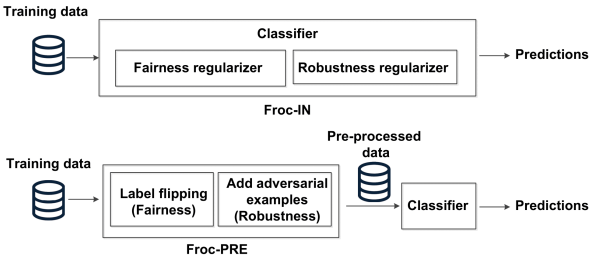


Figure 1: An illustration of FROC-PRE and FROC-IN

For the FROC-PRE method, we design an iterative approach that applies two different operations to modify the training data in several rounds, where a small portion of training data is modified in each round. These two data modification operations are: (1) flipping the binary labels of a set of original training samples (for fairness); and (2) augmenting the training data with a set of adversarial examples (for robustness). Since the users may not have the access to the target model, FROC-PRE considers a surrogate model that has either identical or similar architecture as the target model. In terms of fairness, FROC-PRE quantifies the *fairness influence score* of a training sample as the estimated change in fairness and accuracy of the surrogate model if the sample's label is flipped. Similarly, FROC-PRE quantifies the *accuracy influence score* of any adversarial example as the estimated change in the accuracy of the surrogate model if the adversarial example is inserted into the training data. Based on both influence scores, FROC-PRE iteratively selects a set of training samples that have the highest fairness influence scores for label flipping, as well as a set of adversarial examples that have the highest accuracy influence scores for insertion, until the model achieves both δ_F-*fairness* and δ_R-*robustness*. We design new *influence functions* that estimate both fairness and accuracy influence scores efficiently without model re-training.

We extensively evaluate the performance of both FROC-IN and FROC-PRE on multiple real-world datasets, and made the following observations from our experimental results.

- Imposing robustness and fairness constraints sequentially fails to meet both constraints, as the enforcement of one constraint can counteract the effect of the other. Therefore, both constraints should be dealt with simultaneously.
- Due to the correlation between attributes, simply excluding the protected attribute from training of the target model can eliminate neither the disparate impact on the target model nor the tension between fairness and robustness.
- Both FROC-IN and FROC-PRE address the trade-off between fairness, robustness, and model accuracy. In particular, both methods deliver small accuracy loss while ensuring the model satisfying δ_F-fairness and δ_R-robustness.

Our main contributions include the follows:

- We design the first influence functions that estimate the influence of both label flipping and insertion of adversarial examples on model fairness and accuracy.
- Based on the influence functions, we design two new algorithms named FROC-PRE and FROC-IN

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

that realize both fairness and robustness on classifiers simultaneously.
- We provide new insights into the interaction between fairness and robustness through extensive empirical study.

2. Related Work

Algorithmic fairness. Algorithmic fairness in ML has caught increasing attention from the ML community [16]. Several competing notions of algorithmic fairness have been recently proposed. These definitions can be categorized into two categories: (1) *Group fairness* that is concerned with the protected groups and requires that some statistic of interest be approximately equalized across groups [5], [9], [17]; and (2) *Individual fairness* [18] that prevents discrimination against individuals and requires similar individuals are treated similarly. In this paper, we focus on group fairness.

Techniques to design bias mitigation algorithms typically identify a fairness notion of interest first and modify a particular point of the ML pipeline to satisfy it. Methodologically, they fall broadly into three categories: (1) *pre-processing*: the bias in the training data is mitigated [5], [8], [19]; (2) *in-processing*: the ML model is modified by adding fairness as additional constraint [9], [20], [21]; and (3) *post-processing*: the results of a previously trained classifier are modified to achieve the desired results on different groups [17]. In this paper, we consider both pre-processing and in-processing methods for bias mitigation. We follow the general idea of using regularization in the literature for in-processing mitigation. Various fairness regularization terms have been designed for different fairness definitions. For example, the fairness regularization term in [22] penalizes the mutual information between the protected attribute and the outcome feature, while the fairness regularization in [20], [23] penalizes the difference in false positive rate and false negative rates between two groups. These approaches cannot be directly applied to our problem setting, as we consider statistical parity [5], [8] (i.e., matching of positive rate across different groups).

Robust machine learning. A considerably large amounts of research on adversarial ML and defense strategies have been performed recently. We refer the audience to some excellent surveys [24]–[26] of recent developments in robust ML. In this paper, we focus on two types of evasion attacks, namely proposed *Fast Gradient Sign Method* (FGSM) [6] and *Projected Gradient Descent* (PGD) attack [7]. We consider adversarial training as the defense mechanism.

Tension between fairness and robustness. There is very few study of the tension between robustness and fairness. Chang *et al.* [10] show that fairness impacts robustness - the fair models are noticeably less robust than unconstrained models against the data poisoning attacks during the training phase. They consider the data poisoning attacks during the training phase as well as *equalized odds* [17] as their fairness notion. Xu *et al.* [11] observe that robustness can impact fairness - the adversarial training algorithms tend to introduce disparity of accuracy and robustness between different groups of data. They also consider *equalized odds* [17] as their fairness notion, and design a debiasing algorithm to mitigate the accuracy/robustness disparity of adversarial training across different groups. Unlike these works, we consider the evasion attack during the inference phase, and use *statistical parity*, another widely-used fairness notion, as our fairness objective. While *equalized odds* [17] requires the same true positive rate across different groups, *statistical parity* requires the same positive rate across different groups. Furthermore, both [10], [11] aim to realize *robust fairness* (i.e., eliminating accuracy/robustness disparity in adversarial training), but our goal is to realize fairness and robustness simultaneously on the classification models. Sharma *et al.* [27] investigate the fairness and robustness issues of neural networks from the aspect of the data points' distance to boundaries. They give a new fairness definition which requires that different groups have equalized average distance to the boundaries. Their key idea is two-fold: (1) adjust the average distance to the decision boundary between groups so that the network is more fair with respect to the ability to obtain resources, and (2) increase the average distance of data points to the boundary to promote adversarial robustness. Both of their fairness definitions and robustness requirements are fundamentally different from ours.

3. Preliminaries

3.1. Algorithmic Fairness

In this paper, we mainly focus on group fairness. In general, the group fairness model is defined as following: given a dataset of domain $A \times X \times Y$, where A denotes the *protected attributes* (e.g., gender), X denotes the non-protected attributes, and Y is an outcome feature, the classifier model $\mathbf{M}$ learned from these samples should not have discriminatory effects towards the *protected groups* (e.g., female) defined by the values associated with the protected attribute compared with the *un-protected groups* (e.g., male). For simplicity, we only consider one protected/un-protected group in this paper. In the following discussions, we use $a = 0$ and $a = 1$ to indicate the protected and un-protected groups respectively.

3.2. Adversarial Robustness

There has been active research on adversarial attacks on ML in recent years. The main attacks can be categorized into two types [28]: (1) The *evasion attacks* by which the adversary tries to evade the system by adjusting malicious samples during testing phase; (2) The *poisoning attacks* by which an adversary tries to poison the training data by injecting carefully designed samples during training phase. In this paper, we only focus on the evasion attacks because we focus on the performance of models at inference time. Specifically, we consider two popular evasion attacks: *Fast Gradient Sign Method* (FGSM) [6] and *Projected Gradient Descent* (PGD) attack [7], which are explained below. Although FGSM is a weak attack [7], [29], we still consider it for the investigation of the impact of different attack power on fairness as well as the study of the trade-off between robustness and fairness.

Fast gradient sign method (FGSM) attack. FGSM [6] uses linear perturbation on the features of the testing samples. In particular, let θ be the parameter of the given model $\mathbf{M}$, and $\mathbf{L}()$ be the loss function. The perturbation

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

is performed by adding a noise vector η on the sample $\mathbf{x}$, where the noise is calculated as $\eta = \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{x}} \mathbf{L}(\theta, \mathbf{x}, y)\right)$. The parameter ϵ controls the intensity of the attack. The adversarial testing examples are generated as:

$$\widetilde{\mathbf{x}} = \mathbf{x} + \eta = \mathbf{x} + \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{x}} \mathbf{L}(\theta, \mathbf{x}, y)\right) \quad (1)$$

Projected gradient descent (PGD) attack. While FGSM is considered as a simple one-step scheme, PGD attack [7] is the multi-step variant of FGMS. Formally, the adversarial examples at the $(t+1)$-th iteration is generated as following:

$$\widetilde{\mathbf{x}}_{t+1} = \Pi_{\mathbf{x}+\mathcal{S}}\left(\widetilde{\mathbf{x}}_t + \xi \cdot \text{sign}\left(\nabla_{\mathbf{x}} \mathbf{L}\left(\theta, \mathbf{x}_t, y\right)\right)\right) \quad (2)$$

where Π is the projection operator, $\mathcal{S}$ is a set of perturbation candidates, and ξ is the parameter that controls the intensity of the attack in every iteration.

Adversarial training as defense. There have been significant efforts on designing defense techniques against the evasion attacks. *Adversarial training* (AT) is one of the most promising ways to obtain the adversarial robustness of learning models. The key idea of AT is to generate adversarial examples and augment these perturbed data while training the target model. The augmentation can be done either by feeding the model with both the original data and the crafted data [30], [31] or by learning with a modified objective function [6]. In this paper, we consider the latter approach as the defense against adversarial examples. Specifically, when training a model $\mathbf{M}$ with the utility loss function $\mathbf{L}$ on a training dataset $\{a_i, \mathbf{x}_i, y_i\}_{i=1}^n$, the new loss $\widetilde{\mathbf{L}}$ is defined as:

$$\widetilde{\mathbf{L}}(\mathbf{M}, \{\mathbf{x}\}, \{y\}) = \mathbf{L}(\mathbf{M}, \{\mathbf{x}\}, \{y\}) + \lambda \cdot \mathbf{L}(\mathbf{M}, \{\widetilde{\mathbf{x}}\}, \{y\}) \quad (3)$$

where $\{\widetilde{\mathbf{x}}\}_{i=1}^n$ are the adversarial examples generated by either FGSM or PGD attack, and λ controls the trade-off between robustness and model accuracy. Higher (lower) λ indicates higher (lower) robustness but worse (better) utility. $\lambda = 0$ indicates no robustness.

4. Problem Formulation

Consider a training dataset (A, X, Y) with m samples, where A, X, and Y are the protected attributes, non-protected attributes, and labels respectively. We consider a binary classifier $\mathbf{M}$ in this paper.

To prevent the impacts of the protected attribute on prediction, we use a simple yet widely-used bias mitigation method [5] that excludes the protected attributes from model training. Thus our analysis in the following discussions of model training does not take the protected attribute into consideration.

Model accuracy. Typically, the binary classification can be performed from the prediction of a posterior distribution (called posteriors) over the class labels. We use $\mathbf{M_c}$ to denote a probability classifier that outputs the posteriors in $[0, 1]$. Then $\mathbf{M}$ can be considered as a binary classifier that binarizes the output of $\mathbf{M_c}$. Formally, the model $\mathbf{M}$ makes the binary prediction as: $\mathbf{M}(\mathbf{x}) = \mathbb{1}(\mathbf{M_c}(\mathbf{x}) \geq 0.5), \forall \mathbf{x} \in X$, where $\mathbb{1}(\cdot)$ is the indicator function, which returns 1 if the condition holds, otherwise 0. Intuitively, $\mathbf{M}$ predicts $\hat{y} = 1$ if the posterior is no less than the threshold 0.5, otherwise $\hat{y} = 0$. Note that the protected attribute A is not included in training of $\mathbf{M}$ due to the disparate treatment.

There are various evaluation metrics to measure the accuracy of classification models. In this paper, we consider accuracy of the model $\mathbf{M}$ as the fraction of predictions that are correct. Formally:

$$\texttt{Acc}(X, Y; \mathbf{M}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(\mathbf{M}(\mathbf{x}_i) = y_i). \quad (4)$$

We use *binary cross entropy* (BCE) as the loss function. BCE is commonly used to measure the performance of a classification model whose output is a probability. Formally, the BCE loss function $\mathbf{L}_{\text{U}}$ is defined as follows:

$$\mathbf{L}_{\text{U}}(X, Y) = \frac{1}{n} \sum_{i=1}^{n} \left[y_i \log \mathbf{M_c}(\mathbf{x}_i) + (1 - y_i) \log(1 - \mathbf{M_c}(\mathbf{x}_i))\right] \quad (5)$$

Note that $\mathbf{L}_{\text{U}}(X, Y)$ does not consider the protected attribute due to the disparate treatment.

Fairness. As fairness is a complex and multi-faceted concept which depends on many factors (e.g., context and domains), many statistical definitions of fairness have been introduced in the literature [32]. And yet none is universally applicable. Therefore, we only consider a fairness definition, namely **statistical parity**, that is widely used in the fairness community. Statistical parity [5], [8] (also known as demographic parity) requires that the probability of being classified with positive labels should be the same across both protected and un-protected groups. Formally,

$$\Pr(\hat{y} = 1 | a = 1) = \Pr(\hat{y} = 1 | a = 0) \quad (6)$$

Following the definition of statistical parity, we define fairness as the difference in the positive rates of the protected and un-protected groups. Formally, given a classification model $\mathbf{M}$, a set of testing samples $\{(a_i^{\text{test}}, \mathbf{x}_i^{\text{test}}, y_i^{\text{test}})\}_{i=1}^n$ and their predictions $\hat{Y}^{\text{test}} = \{\hat{y}_i^{\text{test}}\}_{i=1}^n$ made by $\mathbf{M}$, the fairness of $\mathbf{M}$ is measured as the *bias score* $\mathcal{S}_B$ of $\hat{Y}$:

$$\mathcal{S}_B = \left| \frac{\sum_{i=1}^n \left[\mathbb{1}(a_i^{\text{test}}=0 \wedge \hat{y}_i^{\text{test}}=1)\right]}{\sum_{i=1}^n \mathbb{1}(a_i^{\text{test}}=0)} - \frac{\sum_{i=1}^n \left[\mathbb{1}(a_i^{\text{test}}=1 \wedge \hat{y}_i^{\text{test}}=1)\right]}{\sum_{i=1}^n \mathbb{1}(a_i^{\text{test}}=1)} \right| \quad (7)$$

Obviously, $\mathcal{S}_B \in [0, 1]$. The closer $\mathcal{S}_B$ is to 0, the more fair the model $\mathbf{M}$ is. We say *a model* $\mathbf{M}$ *is* δ_F*-fair* if its bias score $\mathcal{S}_B \leq \delta_F$, where δ_F is a user-specified threshold.

Robustness. Given a target model $\mathbf{M}$, a set of testing samples $\{(a_i^{\text{test}}, \mathbf{x}_i^{\text{test}}, y_i^{\text{test}})\}_{i=1}^n$ and their predictions $\hat{Y}^{\text{test}} = \{\hat{y}_i^{\text{test}}\}_{i=1}^n$ made by $\mathbf{M}$, we follow the metric of *adversarial accuracy* [7], [33], [34] to measure the *robustness score* $\mathbf{M}$. Informally, adversarial accuracy measures the accuracy of $\mathbf{M}$ on the adversarial examples $\{\widetilde{\mathbf{x}}_i^{\text{test}}\}$. Accordingly, we define the *robustness score* $\mathcal{S}_R$ as following:

$$\mathcal{S}_R = \texttt{Acc}\left(\{\widetilde{\mathbf{x}}_i^{\text{test}}\}, \{y_i^{\text{test}}\}; \mathbf{M}\right) \quad (8)$$

Intuitively, $\mathcal{S}_R$ measures the percentage of adversarial examples that are correctly predicted by $\mathbf{M}$ (i.e., they fail the evasion attack). Apparently, larger $\mathcal{S}_R$ indicates better robustness. We say *a model* $\mathbf{M}$ *is* δ_R*-robust* if its robustness score $\mathcal{S}_R \geq \delta_R$, where δ_R is a user-specified threshold.

Problem definition. The primary research question that we study in this paper is how to make the model

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

$\mathbf{M}$ δ_B-fair (in terms of statistical parity) and δ_R-robust against both evasion attacks without much sacrifice on model accuracy. Formally, the problem is defined by:

$$\begin{array}{ll} \max\limits_{\mathbf{M}} & \texttt{Acc}(X^{\text{test}}, Y^{\text{test}}; \mathbf{M}) \\ \text{s.t.} & \mathcal{S}_B(A^{\text{test}}, X^{\text{test}}; \mathbf{M}) \leq \delta_F \\ & \mathcal{S}_R(X^{\text{test}}, Y^{\text{test}}; \mathbf{M}) \geq \delta_R \end{array} \tag{9}$$

where δ_F and δ_R are the user-specified thresholds for bias score and robustness score respectively.

A seemly straightforward solution is to realize δ_F and δ_R *sequentially*. However, due to the tension between fairness and robustness [10], [11], the sequential methods (either robustness-then-fairness or fairness-then-robustness) fail, as the fair model will become unfair after adversarial training. Similarly, the robust model can become vulnerable after fairness enforcement. More details of the sequential approach can be found in Section 7. Therefore, we design the classification models that realize both fairness and robustness constraints jointly. Next, we present our two different solutions, namely FROC-IN and FROC-PRE. FROC-IN is an *in-processing* method which incorporates both fairness and robustness with the the learning process, while FROC-PRE is a *pre-processing* method that modifies the training data to improve the fairness and robustness of the trained model.

5. FROC-IN: Our In-processing Method

The key idea of FROC-IN is to equip the objective function of learning with both fairness and robustness. Quite a few prior works [6], [10], [27], [35], [36] have used adversarial regularization to realize either fairness or robustness constraint with the objective function. Inspired by these works, we design FROC-IN, which enforces fairness and robustness as the regularizers to the objective function of the target model. The new loss function of the model is defined as:

$$\mathbf{L}(A, X, Y) = \mathbf{L}_{\text{U}}(X, Y) + \lambda_F \cdot \mathbf{F}(A, X) + \lambda_R \cdot \mathbf{R}(X, Y) \tag{10}$$

where $\mathbf{L}_{\text{U}}$ is the accuracy loss of the target model (Equation 5), $\mathbf{F}$ and $\mathbf{R}$ are the regularizer terms of fairness and robustness respectively, and λ_F and λ_R are the parameters that control the trade-off between fairness, robustness, and accuracy. Larger λ_F (λ_R, resp.) indicates stronger fairness (robustness, resp.) but worse accuracy. Next, we explain how we design the two regularizers $\mathbf{F}$ and $\mathbf{R}$.

Fairness regularizer. Intuitively, the bias score (Eqn. 7) can be used to impose a penalty on the loss function as the fairness regularizer. However, it cannot be directly used as the fairness regularizer as the indicator function $\mathbb{1}(\cdot)$ is not continuous and thus its gradient cannot be properly calculated during training. Indeed, since the fairness condition of statistical parity (Eqn. 6) is non-convex, solving the constrained optimization problem defined by (9) is difficult. To overcome this difficulty, we relax the (non-convex) fairness condition (Eqn.7) into proxy conditions. First, we apply the following transformation

$$\mathbb{1}(a = z \wedge \hat{y} = 1) \Rightarrow \mathbb{1}(a = z) \cdot \mathbb{1}(\hat{y} = 1)(z \in \{0, 1\}).$$

As we only consider binary sensitive attributes, we can further apply the following transformation:

$$\mathbb{1}(a = z) = \begin{cases} 1 - a, & z = 0 \\ a, & z = 1 \end{cases} \tag{11}$$

Next, we apply approximate $\mathbb{1}(\hat{y} = 1)$ to make it outputs continuous values:

$$\mathbb{1}(\hat{y} = 1) \approx \mathbf{M_c}(\mathbf{x}). \tag{12}$$

where $\mathbf{M_c}$ outputs the posterior of x (defined in Section 4). In other words, the binary label output by $\mathbf{M}$ is approximated as the continuous probability output by $\mathbf{M_c}$.

Based on Equations (11) and (12), each component in Equation (7) is transformed to the followings:

$$\begin{cases} \mathbb{1}(a_i = 1) = a_i, \\ \mathbb{1}(a_i = 0) = 1 - a_i, \\ \mathbb{1}(a_i = 1 \wedge \hat{y}_i = 1) \approx a_i \mathbf{M_c}(\mathbf{x}_i), \\ \mathbb{1}(a_i = 0 \wedge \hat{y}_i = 1) \approx (1 - a_i)\mathbf{M_c}(\mathbf{x}_i) \end{cases}$$

Thus the fairness regularizer term $\mathbf{F}$ can be written as:

$$\mathbf{F}(A, X) = \left| \frac{\sum_{i=1}^{n} (1 - a_i)\mathbf{M_c}(\mathbf{x}_i)}{\sum_{i=1}^{n} (1 - a_i)} - \frac{\sum_{i=1}^{n} a_i \mathbf{M_c}(\mathbf{x}_i)}{\sum_{i=1}^{n} a_i} \right| \tag{13}$$

We further simplify Equation (13). Let $C_0 = \frac{1}{\sum_{i=1}^{n}(1-a_i)}$ and $C_1 = \frac{1}{\sum_{i=1}^{n} a_i}$. Apparently, both of them are constants for a given dataset. Thus Equation (13) can be rewritten as:

$$\begin{aligned} \mathbf{F}(A, X) &= \left| \sum_{i=1}^{n} C_0(1 - a_i)\mathbf{M_c}(\mathbf{x}_i) - \sum_{i=1}^{n} C_1 a_i \mathbf{M_c}(\mathbf{x}_i) \right| \\ &= \left| \sum_{i=1}^{n} (C_0 - a_i C_0 - a_i C_1)\mathbf{M_c}(\mathbf{x}) \right| \\ &= \left| \sum_{i=1}^{n} c_i \mathbf{M_c}(\mathbf{x}) \right| \end{aligned} \tag{14}$$

where $c_i = C_0 - a_i C_0 - a_i C_1$. With a set of given training samples, c_i is a constant and thus can be calculated by one traversal of the training samples.

Robustness regularizer. We follow [6] to define our robustness regularizer. Intuitively, during adversarial training, a set of adversarial examples are generated dynamically during training. These adversarial examples are derived from the parameters of the model in the previous epoch, and participate the current epoch as a penalty term. We adapt this idea to our setting and define the robustness regularizer as following:

$$\mathbf{R}(X, Y) = \mathbf{L}_{\text{U}}(\widetilde{X}, Y) \tag{15}$$

where $\mathbf{L}_{\text{U}}$ is the accuracy function (Equation (5)), and $\widetilde{X} = \{\widetilde{\mathbf{x}}\}$ are the adversarial examples.

Model training with both regularizers. The in-processing model with both regularizers (Eqn. (10)) can be trained by the stochastic gradient descent (SGD) method. The gradient for each iteration is calculated as:

$$\nabla_\theta \mathbf{L}(A, X, Y) = \nabla_\theta \mathbf{L}_{\text{U}}(X, Y) + \lambda_F \cdot \nabla_\theta \mathbf{F}(A, X) + \lambda_R \cdot \nabla_\theta \mathbf{R}(X, Y)$$

First, as the accuracy loss $\mathbf{L}_{\text{U}}(X, Y)$ is defined as a *binary cross entropy* function, its gradient is calculated as:

$$\nabla_\theta \mathbf{L}_{\text{U}}(X, Y) = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} [y_i \nabla_\theta \log \mathbf{M_c}(\mathbf{x}_i) + (1 - y_i) \log \nabla_\theta (1 - \mathbf{M_c}(\mathbf{x}_i))]$$

Next, the gradient of the fairness term in Eqn. (14) is computed as:

$$\nabla_\theta \mathbf{F}(A, X) = \texttt{sign}\left(\left| \sum_{i=1}^{n} c_i \mathbf{M_c}(\mathbf{x}) \right| \right) \cdot \frac{1}{n} \sum_{i=1}^{n} c_i \nabla_\theta \mathbf{M_c}(\mathbf{x})$$

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

Finally, the gradient of the robustness term in Eqn. (15) is computed as:

$$\nabla_\theta \mathbf{R}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i \nabla_\theta \log \mathbf{M_c}(\widetilde{\mathbf{x}}_i) + (1-y_i)\log \nabla_\theta (1-\mathbf{M_c}(\widetilde{\mathbf{x}}_i))\right]$$

where $\widetilde{\mathbf{x}}_i$ is the adversarial example generated by either FGSM or PGD attack.

6. FRoC-PRE: Our Pre-Processing Method

A possible weakness of FROC-IN is that it requires the access to the target model, which may not be possible when the target model is owned by a third-party (e.g., an MLaaS service provider) and is not accessible to the users. Thus we design another method named FROC-PRE, which assumes that the users have the access to the training data and a surrogate model which have similar architecture as the target model. We will discuss how to choose the surrogate model in Section 8.

The key idea of FROC-PRE is to modify the training data so that the model trained on the modified data is fair and robust. We consider two types of data modification: (1) flip the binary labels of a set of original training samples $D_F \subseteq D_{train}$; and (2) augment D_{train} with a set of adversarial examples D_R. Label flipping aims to remove the bias from the training data, while augmenting with adversarial examples is to enhance the robustness of the model. Intuitively, we aim to find D_F and D_R that make the model $\mathbf{M}$ trained on the pre-processed dataset (A', X', Y') satisfy both requirements of δ_F-fairness and δ_R-robustness.

Although both δ_F-fairness and δ_R-robustness are required on the testing data (Eqn. (9)) which may not be available during the pre-processing phase, a model that is δ_F-fair and δ_R-robust on the training data should be δ_F-fair and δ_R-robust on the testing data too, due to the assumption that both training and testing data have the same distribution. Thus we formalize the pre-processing problem as an optimization problem defined as following:

$$\begin{array}{ll} \max\limits_{A,X,Y} & \texttt{Acc}(X,Y;\mathbf{M}) \\ \text{s.t.} & \mathbf{M} = \arg\max_{\mathbf{M}'} \texttt{Acc}(X,Y;\mathbf{M}) \\ & \mathcal{S}_B(A,X;\mathbf{M}) \le \delta_F \\ & \mathcal{S}_R(X,Y;\mathbf{M}) \ge \delta_R \end{array} \tag{16}$$

where the function ACC, $\mathcal{S}_B$ and $\mathcal{S}_R$ are the accuracy, bias score and robustness score of the model $\mathbf{M}$ respectively, and δ_F and δ_R are the user-specified thresholds for bias score and robustness score. Our experimental results show that, FROC-PRE ensures that the model is δ_F-fair and δ_R-robust on the testing data. More details can be found in Section 7.

Choosing both D_F and D_R in one shot may be too rigid and lead to significant accuracy loss. Therefore, we take a greedy, sequential approach to pick samples of D_F and D_R in multiple trials. In each trial, we pick and modify one small portion of training data, and observe the change of model fairness and robustness by the modification. If the model achieves both requirements δ_F-fairness and δ_R-robustness, we terminate the modification. Otherwise, we continue with the next trial.

Following this idea, we design an iterative method that picks the data samples for modification. Specifically, consider the original training data D^0_{train}. At the i-th iteration, FROC-PRE applies the following two steps on the current dataset D^i_{train} and generates the dataset D^{i+1}_{train} for the next iteration:

- **Step 1.** If δ_F-fairness is not satisfied, select ℓ_1 samples D^i_F from D^0_{train}. Flip the labels of each sample in D^i_F.
- **Step 2.** If δ_R-robustness is not satisfied, select ℓ_2 samples D^i_R from D^0_{train}, generate ℓ_2 adversarial examples D^i_R accordingly. Augment D^i_{train} with D^i_R (i.e., $|D^{i+1}_{train}| = |D^i_{train}| + \ell_2$);

The algorithm repeats the two steps until either both δ_F-fairness and δ_R-robustness are met or the number of iterations has reached a pre-defined budget. Apparently, $D_R = \cup_{D^i_R}$ and $D_F = \cup_{D^i_F}$. Both ℓ_1 and ℓ_2 values control the impacts of fairness and robustness on model. To ensure equal impact of fairness and robustness, we use $\ell_1 = \ell_2$ in the algorithm, and use $\ell = \ell_1(\ell_2)$ in the following discussions.

A naive solution to the optimization problem in Eqn. (16) is to enumerate all choices and pick the one that returns the best accuracy. Given m samples in the training data, there are $\binom{m}{\ell}$ choices to pick ℓ samples. Apparently this is unacceptable due to its high complexity. The main challenge is thus to design an efficient solution that picks the samples for label flipping and generation of adversarial examples.

To address the computational challenge, we design a greedy algorithm to solve the optimization problem. Intuitively, first, for each training sample, we estimate the "influence" of flipping its label on both model fairness and accuracy, and pick ℓ samples of the largest influence by label flipping. Next, we estimate the "influence" of each adversarial example on model accuracy, and pick ℓ ones that have the minimal influence for data augmentation. We note that the model that FROC-PRE estimates the influence on is not necessarily the same as the target model $\mathbf{M}$. We denote this model as the *surrogate model* $\mathbf{M}_S$, which is also a binary classification model that may have identical or similar architecture as $\mathbf{M}$. Since both $\mathbf{M}$ and $\mathbf{M}_S$ are classification models, they use the same loss function $\mathbf{L}_\text{U}$ (Eqn. (5)). In the following discussion, we first present how to estimate these two types of influence on a particular classification model. Then we discuss the FROC-PRE in details.

6.1. Estimating Influence of Label Flipping on Model Fairness and Accuracy

To estimate the influence of flipping the label of a particular training sample on model fairness and accuracy, we first estimate the influence of label flipping on model fairness alone. Then we estimate the "combined" influence of a label flipping on model fairness and accuracy together. Next, we present the details of these two steps.

Estimating fairness influence of label flipping. Changing the label of a training sample can impact the model's bias score. Intuitively, the influence of a particular label flipping on model fairness can be measured by asking the counterfactual: *how would model fairness change if the model sees a different label of this sample during training?* A simple solution of estimating a label's influence on model fairness is to flip the label, re-train the model from scratch, and measure the bias score of the

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

re-trained model. Apparently, this process is not acceptable due to its high computation cost. Unfortunately, the influence function in the literature [37] only estimates the impact of a training sample on model accuracy. It cannot estimate the influence of label flipping on model fairness.

In this paper, we design a new influence function that estimates the influence of flipping a label y on model fairness. The estimation is not straightforward, as the bias score (Eqn. (7)) is computed from binary labels. To facilitate our estimation, we consider the approximation of the bias score in Eqn. (14). However, Eqn. (14) does not involve Y, which makes difficult to estimate the influence on model fairness. Therefore, we estimate the influence on fairness in two steps: First, we estimate the change on the model parameters θ by flipping a label. Then we estimate the influence on bias score of the model by parameter changes. We use $\bar{y}$ to denote the flipped label of y.

Step 1: Estimate the change on model parameters. The change in model parameters due to flipping a label in the training set can be formalized as $\theta_{\bar{y}} - \theta$. To estimate this change efficiently, we adapt the basic idea of the influence function [38] to our setting. The idea is to compute the parameter change if considering flipping label y as y being upweighted by some small τ, so that

$$\theta_\tau = argmin_\theta \frac{1}{n}\sum_{i=1}^{n} \mathbf{L}_{\mathrm{U}}(z_i,\theta) + \tau \mathbf{L}_{\mathrm{U}}(z,\theta)$$

The influence of upweighting y on the parameters θ can be computed as the following [38]:

$$\mathcal{I}_{up}(\mathbf{z})|_{\tau=0} = -H_\theta^{-1} \nabla_\theta \mathbf{L}_{\mathrm{U}}(\mathbf{x}, y),$$

where $H_\theta = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 \mathbf{L}_{\mathrm{U}}(X, Y)$ is the Hessian and is positive by assumption. Now, consider flipping the label of a sample $\mathbf{z} = (\mathbf{x}, y)$ to $\overline{\mathbf{z}} = (\mathbf{x}, 1-y)$. The change of parameters θ is computed as:

$$I_\theta^f(\mathbf{z}) = \mathcal{I}_{up}(\overline{\mathbf{z}}) - \mathcal{I}_{up}(\mathbf{z}) = -H_\theta^{-1}\left[\nabla_\theta \mathbf{L}_{\mathrm{U}}(\mathbf{x}, 1-y) - \nabla_\theta \mathbf{L}_{\mathrm{U}}(\mathbf{x}, y)\right] \tag{17}$$

Step 2: Estimate the change on bias score. We estimate the influence of parameter changes (by Step 1) on the approximate bias score (Eqn. (14)) as following:

$$\mathcal{I}_f = -\nabla_\theta \mathbf{F}(A, X) \tag{18}$$

Then, the *fairness influence score* of flipping the label y of a particular training sample on model fairness is estimated as:

$$\mathcal{I}_F(y) = \mathcal{I}_f^\top I_\theta^f(\mathbf{z}), \tag{19}$$

where a negative (resp., positive) $\mathcal{I}_F(y)$ value indicates that flipping y will lower (resp., increase) the bias. The absolute value $|\mathcal{I}_F(y)|$ indicates the intensity of influence.

Estimating fairness-utility influence of label flipping. As the optimization problem (Eqn. (16)) aims to minimize the accuracy loss, we estimate the influence of flipping a label y on model accuracy loss as:

$$\mathcal{I}_U^f(y) = I_u^\top I_\theta^f(\mathbf{z}) \tag{20}$$

where $I_u = -\nabla_\theta \mathbf{L}_{\mathrm{U}}(X, Y)$. Intuitively, a positive (resp., negative) $\mathcal{I}_U(y)$ indicates a demotion (resp., promotion) of model accuracy by flipping y.

Next, we combine I_F and I_U into one influence score, namely the *fairness-accuracy influence score* $I_{FU}(y)$ that quantifies the influence of flipping a particular label y on both model fairness and accuracy:

$$\mathcal{I}_{FU}(y) = \mathcal{I}_F(y) \cdot \exp\left(-\left|\mathcal{I}_U^f(y)\right|\right) \tag{21}$$

A negative (resp., positive) I_{FU} indicates that flipping y will lower (resp., increase) the bias. Furthermore, For a negative $I_{FU}(y)$ value, the larger $|I_{FU}(y)|$ is, the more improvement to fairness and the less impact on model accuracy when the label y is flipped. Thus we pick the labels of the negative $I_{FU}(y)$ that with the largest absolute value for flip. We take an exponential term on $\mathcal{I}_U^f(y)$ to amplify the effect of label flipping on utility.

6.2. Estimating Adversarial Example's Influence on Model Accuracy

Generating adversarial examples. For a given data sample $\{a, x, y\}$, we generate its adversarial example as $\{a, \widetilde{\mathbf{x}}, y\}$. In other words, the adversary example only perturbs the non-protected attributes X, but keeps the protected attributes A and labels Y unchanged. We do not perturb the protected attribute A because it is not included in training of the target model due to the disparate treatment. Next, we explain how to generate $\widetilde{\mathbf{x}}$.

For any numerical value x, we utilize the PGD attack to generate its perturbed value $\widetilde{\mathbf{x}}$. For any categorical value x, we follow [39] to generate adversarial examples of categorical data in the discrete domain. Briefly speaking, any given categorical feature X is approximated as a Concrete random variable, which has the categorical probability $p_1, p_2, \ldots, p_d$ encoded as a one-hot vector in a d-dimensional space $\mathbb{R}^d$. Then $\widetilde{\mathbf{x}}$ is selected from the one-hot vector as the one of the highest probability that maximizes the likelihood of the attack success.

Estimating influence of adversarial examples on model accuracy. Intuitively, adding adversarial examples into the training data can either improve or downgrade model accuracy. A naive method to measure the impact of an adversarial sample on model accuracy is to add it into the training data and retrain the model. Apparently this method is computationally expensive. Therefore, given a training sample $(\mathbf{x}, y)$ and its adversarial example $(\widetilde{\mathbf{x}}, y)$, we estimate the *accuracy influence score* of inserting $(\widetilde{\mathbf{x}}, y)$ into the training data as the following:

$$I_U(\widetilde{\mathbf{x}}) = \left\|\nabla_\theta \mathbf{L}_{\mathrm{U}}\left(\widetilde{\mathbf{x}}, y\right)\right\|_2 \tag{22}$$

where $\mathbf{L}_{\mathrm{U}}$ is the accuracy function (Eqn. (5)). Intuitively, the adversarial examples that have lower accuracy should have smaller influence on model accuracy if they are added into the training dataset.

6.3. Algorithm Details

Applying the two steps in different orders can lead to different samples to be picked for label flipping as well as different adversarial examples to be generated. However, our empirical results show that performance of FROC-PRE under different orders of the two sub-steps is very similar. Thus in the paper, we only focus on the order that label flipping is performed before generation of adversarial examples.

The pseudo-code is shown in Algorithm 1. We highlight some details that were not covered in the previous discussions. First, when the algorithm picks training

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: FROC-PRE algorithm

Input : Training data (A, X, Y) with m samples; Fairness threshold δ_F; Robustness threshold δ_R; # of iterations L; # of samples to modify per iteration l; Surrogate model $\mathbf{M}_S$

Output : Pre-processed training dataset $(A^{'}, X^{'}, Y^{'})$

$(A^{(0)}, X^{(0)}, Y^{(0)}) = (A, X, Y)$;
Initialize the set of flipped labels $\mathcal{F} = \emptyset$;
Initialize the set of adversarial examples $\mathcal{R} = \emptyset$;
for $i = 1$ *to* L **do**
  Train model $\mathbf{M}_S^{(i)}$ on dataset $(A^{(i-1)}, X^{(i-1)}, Y^{(i-1)})$;
  Calculate $\mathcal{S}_B$ (Eqn. (7)) and $\mathcal{S}_R$ (Eqn. (8)) of $\mathbf{M}_S^{(i)}$;
  if $\mathcal{S}_R \geq \delta_R$ **and** $\mathcal{S}_B \leq \delta_F$ **then** `// Meet both constraints`
    break;
  $(A^{(i)}, X^{(i)}, Y^{(i)}) = (A^{(i-1)}, X^{(i-1)}, Y^{(i-1)})$;
  if $\mathcal{S}_B > \delta_F$ **and** $|\mathcal{F}| < m$ **then** `// Fairness`
    $Z_F = \left\{ y_j \middle| y_j \in Y^{(0)} \wedge j \notin \mathcal{F} \right\}$;
    for $y_j \in Z_F$ **do**
      Calculate $\mathcal{I}_{FU}(y_i)$ (Eqn. (21));
    Pick top-l labels P_1 of the smallest negative $\mathcal{I}_{FU}(y_i)$ (i.e., largest fairness-accuracy influence score);
    $\mathcal{F} = \mathcal{F} \cup \{i | y_i \in P_1\}$;
    Flip $\left\{ y_j \in Y^{(i)} \middle| y_j \in P_1 \wedge \left(\mathbf{x}_j \in X^{(i)} \vee \widetilde{\mathbf{x}}_j \in X^{(i)} \right) \right\}$;
  if $\mathcal{S}_R < \delta_R$ **and** $|\mathcal{R}| < n$ **then** `// Robustness`
    $\widetilde{Z}_R = \left\{ \texttt{Adv}(\mathbf{x}_j, y_j; \mathbf{M}_S^{(i)}) \middle| \mathbf{x}_j \in X^{(0)} \wedge j \notin \mathcal{R} \right\}$;
    for $\widetilde{\mathbf{x}}_j \in Z_R$ **do**
      Calculate $I_U(\widetilde{\mathbf{x}}_j)$ (Eqn. (22));
    Pick top-l adversarial examples P_2 of the smallest accuracy influence score ;
    $\mathcal{R} = \mathcal{R} \cup \{i | \widetilde{\mathbf{x}}_i \in P_2\}$;
    Augment $\{(a_j, \widetilde{\mathbf{x}}_j, y_i) | \widetilde{\mathbf{x}}_j \in P_2\}$ with $(A^{(i)}, X^{(i)}, Y^{(i)})$;
return $(A^{(i)}, X^{(i)}, Y^{(i)})$;

samples to flip, it does not pick any sample that has been flipped in previous iterations. Second, when the algorithm flips the labels of particular training samples, it also flips the labels of the adversarial examples of these training samples if there is any. Otherwise the generated adversarial example will have the opposite labels and lose its ability to enhance model robustness.

After the iterations, the algorithm picks ℓ_F labels in total of the largest fairness-accuracy influence score (Eqn. (21)) to flip, and ℓ_R adversarial examples of the smallest accuracy influence score (Eqn. (22)) to be added into the training data. The value of ℓ_F is not necessarily the same as ℓ_R. Our empirical results show that both flipped labels and adversarial examples take a small portion of the training data (at most 17.1% flipped labels and at most 13.3% as adversarial examples). More details can be found in Section 7.

Since FROC-PRE executes a fixed number of iterations, the output data may not allow the model to satisfy both δ_F- fairness and δ_R-robustness constraints. However, our experimental results show that this only happens when either the fairness or the robustness requirement is too strong. More details are included in Section 7.

Dataset	# of samples	# of attributes	Protected Attribute		Label
			Race	Gender	
Adult	45,222	14	non-white$^+$ white$^-$	F$^+$ M$^-$	Binary
Hospital	52,778	122			
COMPAS	20,000	9		M$^+$, F$^-$	

TABLE 1: Summary of the real-world datasets ($+$ and $-$ indicate the protected and un-protected groups, M and F stands for male and female).

7. Experiments

7.1. Experimental Setup

All the experiments are performed on a server with Ubuntu 18.04.4, two Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz, 384GB memory, and four NVIDIA Quadro RTX 6000 graphic cards. Codes are implemented and executed in Python 3.7.9 with PyTorch 1.7.1 and Scikit-learn 0.23.2. All the code and data for our experiments can be found at Github[1].

Datasets. We use three real-world datasets: *Adult* dataset [40], *COMPAS* dataset [41] and *Hospital* dataset [42]. We consider these three datasets because they are widely used in the fairness literature [5], [20], [43]. The statistics as well as the fairness setup of these datasets are shown in Table 1. More details of these datasets can be found in Appendix A. All categorical values in the datasets are transformed to numerical ones by one-hot encoding.

Network capacity. We consider a simple neural network that consists of a convolutional layer with two filters, followed by another convolutional layer with four filters, and a fully connected hidden layer with 64 units. The network is trained with 500 epochs, a batch size of 256 and the learning rate of 0.01. We follow [7] to construct the adversarial examples with ξ=0.3. All models are trained on the training dataset that consists of 70% data samples randomly selected from each dataset, and tested on the remaining 30% data samples.

Parameter setting of FROC-PRE algorithm. We use $L = 1,000$, $\delta_F \in [0.01, 0.2]$, and $\delta_R \in [0, 1]$ for the FROC-PRE algorithm.

Evaluation metrics. We use the bias score (Eqn. (7)), robustness scores (Eqn. (8)), and the accuracy (Eqn. (4)) to evaluate fairness, robustness, and model accuracy respectively.

Baseline - sequential methods. We consider the alternative solution that enforces fairness and robustness independently in a sequence. We consider two different sequential methods: (1) **Robustness-before-Fairness (RbF)** method: we generate a number of adversarial examples by PGD attack, and augment the training data with these examples. Then we apply fairness-enhancing methods to train the model on the training data with adversarial examples; (2) **Fairness-before-Robustness (FbR)** method: we apply fairness-enhancing methods on the training data to remove bias. Then we use PGD-adversarial training to improve model robustness.

For the implementation of fairness solutions, we choose two fairness-enhancing algorithms from IBM's AIF360 fairness toolbox[2] that provide statistical parity: (1)

1. https://github.com/fatml-res/robustness-and-fairness
2. IBM AIF360 fairness toolbox: https://aif360.mybluemix.net

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

Reweighing (RW) [44] method is a pre-processing method that associates a *weight* value with each training sample to indicate the independence between the protected attribute and the label; and (2) *Disparate Impact Remover* (DIR) [5] method is a pre-processing method that modifies the non-protected attributes so that their distribution across different groups is similar.

7.2. Failure of Sequential Methods

We evaluated the performance of both sequential methods. We vary the value of the parameter $\lambda = \{0, 0.01, 0.05, 0.1, 0.5, 1.0\}$ for adversarial training (Eqn. (3)) to control the degree of robustness. We show the results on Adult and COMPAS datasets with adversarial examples generated by PGD attack in Figure 2. The results of Hospital dataset can be found in Appendix F.

First, for the RbF methods (Figure 2 (a) - (d)), when the model is not equipped with robustness (i.e. λ = 0), the bias score of the model by applying both fairness enhancing methods (RW and DIR) is reduced compared with the original model. This shows the effectiveness of both RW and DIR in terms of fairness enhancement. However, when these two fairness enhancing methods are employed on the data with adversarial examples, the bias score becomes very close to that without fairness for both RW and DIR methods. Even worse, the bias score can be even higher than that under no-fairness setting after robustness is equipped with the model for COMPAS dataset with gender as the protected attribute (Figure 2 (d)). This demonstrates that applying robustness before fairness can make the existing fairness enhancing methods become ineffective on the data with adversarial examples.

Similarly, for the FbR methods (Figure 2 (e) - (h)), the bias score also increases after the model is equipped with robustness. In some settings (e.g., COMPAS dataset), the bias score increasingly grows with stronger robustness (i.e., when the robustness parameter λ increases). In other words, for a model that was fair, making it robust can deteriorate its fairness.

Next, we analyze the reason why robustness hurts fairness. Model unfairness can spur from two sources: (1) biases in the training data; and (2) class imbalance (i.e., the protected group's data is not sufficiently represented) [45]. Therefore, we analyze the bias in the adversarial examples as well as their distributions across different groups. We found that, first, the adversarial examples generated by PGD attack are indeed biased. Their labels are highly dependent on the protected attributes. The Pearson correlation between the labels and the protected attributes of the adversarial examples can be as high as 0.85. Second, since the adversarial examples use the same values of the protected attributes of their original ones, the distribution of adversarial examples mimics the distribution of the original data and thus is also imbalanced across different groups. Therefore, adding such adversarial examples to the training data that was imbalanced at first place exacerbates the imbalance between different groups. Augmenting such biased adversarial examples of imbalanced distribution counteracts the fairness-enhancing effect by RW and DIR, and thus downgrades model fairness.

We also compare both sequential methods with FROC-IN in terms of their fairness and robustness. We do not consider FROC-PRE for comparison because its fairness and robustness parameters δ_F and δ_R are not comparable to the parameter λ for adversarial training by both sequential methods and FROC-IN.

The fairness performance of the three approaches for both Adult and COMPAS datasets are shown in Figure 2. The results on Hospital dataset are shown in Appendix F. The main observation is that, in general, FROC-IN outperforms both sequential methods in most of the settings. The only exception is the setting of COMPAS dataset with large λ value (e.g., $\lambda \geq 0.5$) (Figure 2 (g) & (h)), where the DIR method, one of the FbR methods, has better bias score than FROC-IN. However, in these settings, the bias score of FROC-IN is only slightly higher than that by DIR. Nevertheless, FROC-IN still outperforms the two RbF methods in terms of robustness (Figure 2 (c) & (d)).

7.3. Performance of FRoC-In Method

Trade-off between fairness, robustness, and accuracy. We measure model accuracy, fairness, and robustness of FROC-IN on the three datasets. The results on Adult and COMPAS datasets for PGD attack are shown in Figure 3. The results on FGSM attack and for Hospital dataset can be found in Appendice D and F respectively. Unsurprisingly, the trade-off always exists between fairness, robustness and accuracy. In all the settings of Adult dataset, the accuracy downgrades when the robustness scores grows (i.e., stronger robustness). Similarly, the accuracy decreases when the bias scores decreases (i.e., more fairness). However, FROC-IN well addresses the trade-off between fairness, robustness and accuracy. For example, for Adult dataset, the model accuracy decreases at most 4.24% in all the settings, even when the robustness score as high as 0.83 and the bias score is as low as 0.009. Interestingly, higher accuracy is also achieved with stronger fairness or robustness on COMPAS dataset. We found that the reason is that the original model on COMPAS dataset was overfitting. Adding both fairness and robustness regularizers eliminate such overfitting and thus leads to higher accuracy.

Interaction between fairness and robustness regularizers. In this part of experiments, we try to answer the research question: *How fairness and robustness interact with each other during model training?* We measure the angle between $\nabla_\theta \mathbf{F}$ and $\nabla_\theta \mathbf{R}$ (i.e., the gradients of fairness and robustness regularizers) during model training. Intuitively, an angle that is greater than 90° indicates that fairness and robustness compete with each other, otherwise they are aligned with each other. Given that our neural network is small and low-dimensional, the orthogonal angle between the gradient vectors of fairness and robustness regularizers can demonstrate the interaction between fairness and robustness.

We tried various settings of $\lambda_{\mathbf{F}}$ and $\lambda_{\mathbf{R}}$ values. We ran 500 epochs of FROC-PRE, and monitored the angel between $\nabla_\theta \mathbf{F}$ and $\nabla_\theta \mathbf{R}$ during these epochs. Figure 4 shows the results for Adult and COMPAS datasets when PGD is the attack. The results for FGSM attack and for Hospital dataset are included in Appendices D and F respectively. First, we observe the angle between $\nabla_\theta \mathbf{F}$ and $\nabla_\theta \mathbf{R}$ remains in the range of $[70°, 110°]$ after the initial 20 epochs, for all the settings. The angle grows fast at first, and quickly becomes larger than 90° in the first 20 epochs. This indicates that the fairness and robustness

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

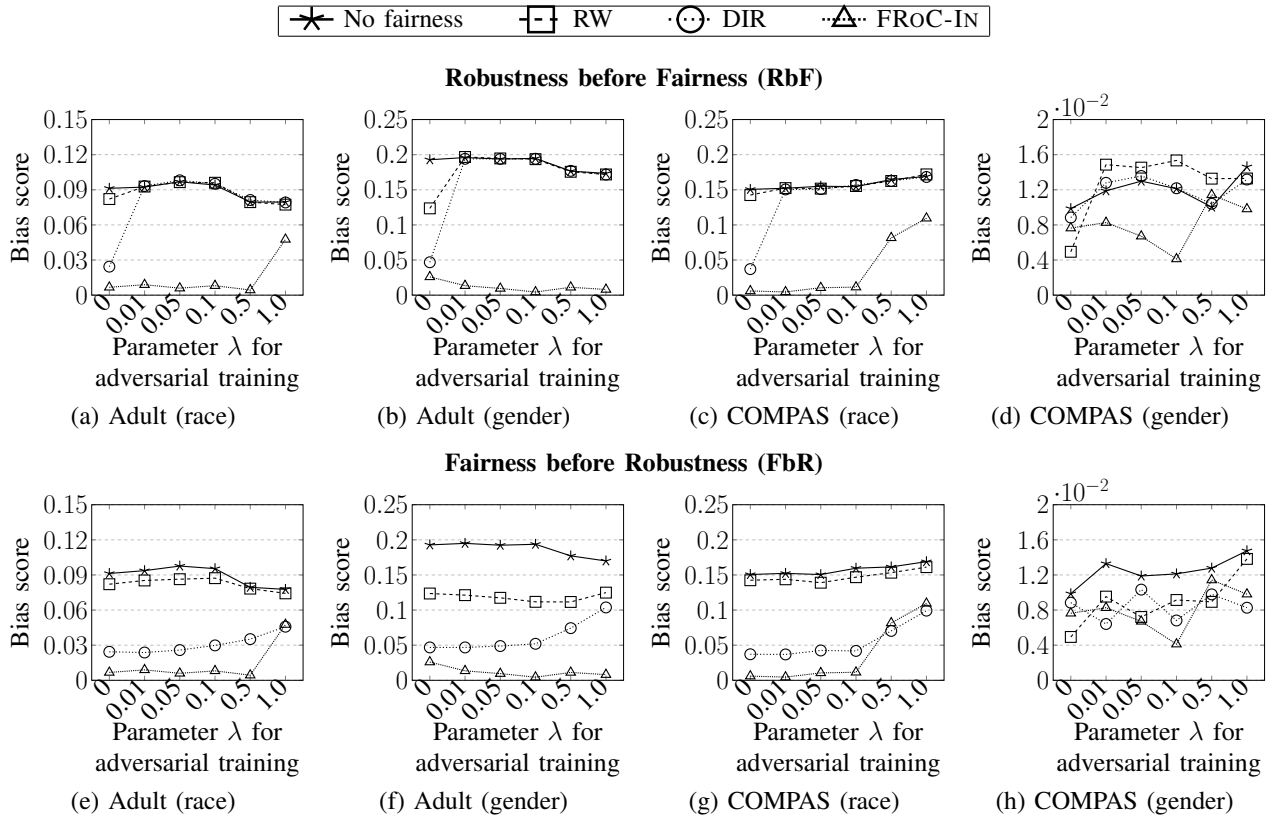


Figure 2: Comparison between performance of both sequential methods and FROC-IN. Robustness is implemented by inserting adversarial examples generated by the PGD attack. Fairness is implemented by pre-processing training data with either Reweighing (RW) [44] or Disparate Impact Remover method (DIR) [5].

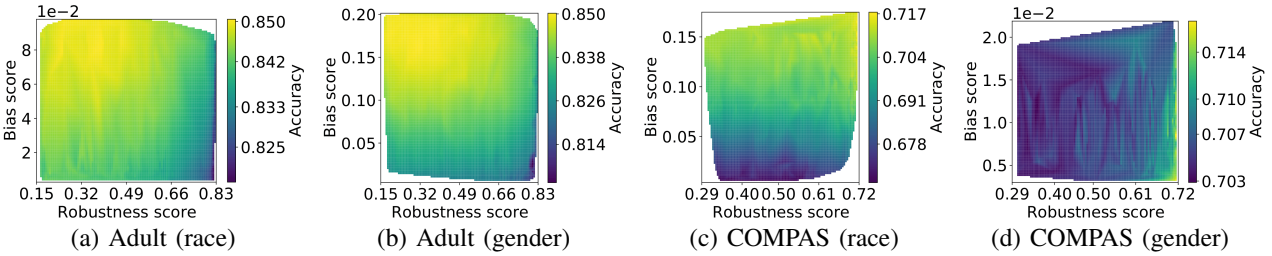


Figure 3: Model fairness, robustness, and accuracy of FROC-IN method (PGD attack). X- and y- axis show robustness and bias scores respectively. Accuracy is visualized in colors; light (deep) color indicates higher (lower) accuracy

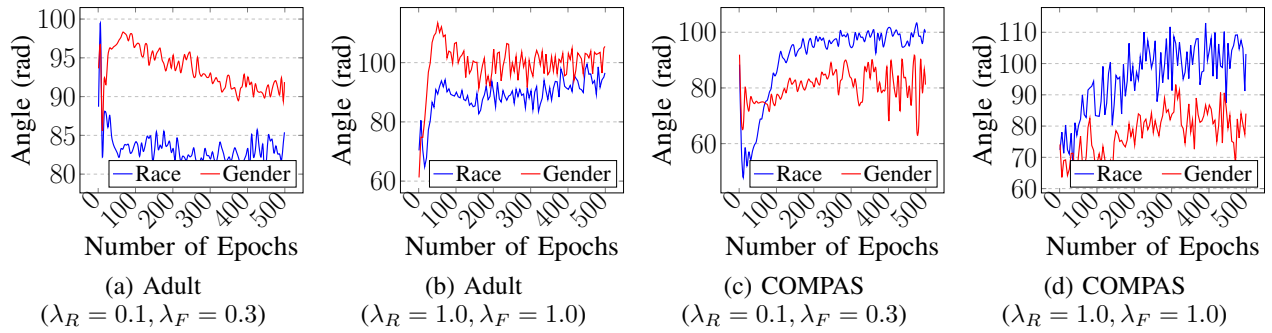


Figure 4: Interaction between fairness and robustness regularizers during training of FROC-IN method (PGD attack). The interaction is measured as the angle between the gradients of both fairness and robustness regularizers

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

regularizers are competing with each other at beginning of model training. Then the angle eventually becomes relatively stable when more epochs are executed. In particular, the stabilized angle (for both protected attributes) is close to 90° for most of the settings. Such alignments facilities the model to achieve both fairness and robustness requirements with small model accuracy loss. The two exceptions include the COMPAS dataset with *race* as the protected attribute (blue line in Figure 4 (c) & (d)), and the Adult dataset with *gender* as the protected attribute and $\lambda_R = 1$ (red line in Figure 4 (b)). The main reason behind these exceptions is that the model has not reached convergence for these settings yet when the number of epochs reaches 500.

Note that we have excluded the protected attribute from training due to disparate treatment. Then why fairness and robustness still interact with each? To explain this, we measured the correlation between each un-protected attribute and the protected attribute for the three datasets (results in Appendix C). Since some attributes are categorical, we consider *Cramer's V measurement*, which measures the relation between two categorical variables. It returns a value in the range [0, 1]. In general, a Cramer's V measurement value $v \in [0.1, 0.3]$ indicates a weak association, $v \in [0.3, 0.5]$ indicates a medium association, and $v > 0.5$ indicates a strong association. Our results show a strong association between the un-protected and protected attributes on Adult dataset (e.g., the Cramer's V is 0.65 between the un-protected *relationship* attribute and the protected *gender* attribute). Thus simply removing the protected attribute did not eliminate its disparate impact on the decision, and thus leaving the model in a fierce competition between the fairness and robustness regularizers during training, leading to the gradients between them as large as 110° at the initial 100 epochs (Figure 4 (a) & (b)). On the other hand, the Cramer's V correlation between all the un-protected attributes and the protected attribute *gender* on COMPAS dataset is weak (the strongest correlation is 0.193). These weak/medium correlations still cannot remove the disparate impact on the target model (See Appendix C for more discussions). However, the weak correlation leads to a milder competition between fairness and robustness - the angel between the gradients between fairness and robustness regularizers (red lines in Figure 4) for COMPAS datasets is smaller than that on Adult dataset. The observation on the *race* protected attribute is similar - stronger correlation between un-protected and protected attributes leads to higher tension between fairness and robustness.

7.4. Performance of FRoC-Pre Method

Trade-off between fairness, robustness, and accuracy. Figure 5 shows the model accuracy, fairness, and robustness of FROC-PRE for both Adult and COMPAS datasets when PGD is the attack. The results for FGSM attack and Hospital dataset are similar and can be found in Appendix E and F respectively. The results shows that, first, FROC-PRE addresses the trade-off between fairness, robustness, and model accuracy. In particular, the accuracy decreases when either the fairness constraint or the robustness constraint gets stronger. Nevertheless, the accuracy loss remains insignificant. For example, the accuracy decreases at most 7.46 % for Adult dataset and 10.26 % for COMPAS dataset. We note that non-negligible accuracy loss is expected to achieve both robustness and fairness. Indeed, the previous literature [20] has shown that even achieving statistical parity fairness alone can incur the accuracy loss as high as 40%. Compared with this, our 10% loss is not significant.

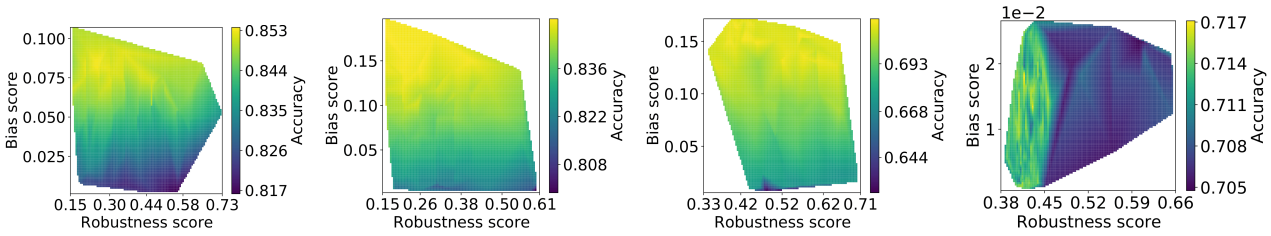
We also measured the number of labels to be flipped as well as the number of adversarial examples to be added by FROC-PRE for the three datasets. The results show that only a small percentage of labels are flipped as well as the adversarial examples are inserted. In particular, for Adult dataset, the algorithm flips 1,995 (6.3%) labels and adds 4,940 (15.6%) adversarial examples. For COMPAS dataset, the algorithm flips 2,395 (17.1%) labels and adds 1,858 (13.3%) adversarial examples. For Hospital dataset, the algorithm flips 810 (2.2%) labels and adds 4,406 (11.9%) adversarial examples.

Varying target and surrogate models. One advantage of FROC-PRE is that it considers a surrogate model, so it can be used for different target classification models. We consider the following five classification models as the target model: (1) logistic regression (LR), (2) neural networks with 1 hidden layer and 64 neurons (NN1x64), (3) neural networks with 1 hidden layer and 128 neurons (NN1x128), which is the same as the surrogate model in FROC-PRE, (4) neural networks with 1 hidden layer and 256 neurons (NN1x256); and (5) neural networks with 2 hidden layers and 128 neurons on each layer (NN2x128). We also consider three types of surrogate models, namely LR, NN1x128, and NN2x128. We measure the architectural complexity of the neural networks as the total number of neurons. All the NN models use the same activation function. We tried various threshold values of δ_R and δ_F, and picked those that deliver the best model accuracy. We also consider three different settings of the fairness and robustness thresholds for *strong*, *medium*, and *weak* fairness-robustness settings. The details of δ_F and δ_R of these settings can be found in Figure 6.

We present the results for Adult and COMPAS datasets when adversarial examples are generated by PGD attack in Figure 6. The results for FGSM attack on these two datasets can be found in Appendix E. We use the colored rectangles to indicate the δ_F-fairness and δ_R-robustness requirements. Intuitively, the (bias score, fairness score) points that lie either on the edges of the rectangles or inside of the rectangle indicate that the model satisfies δ_F-fairness and δ_R-robustness, otherwise it fails either δ_F-fairness or δ_R-robustness or both.

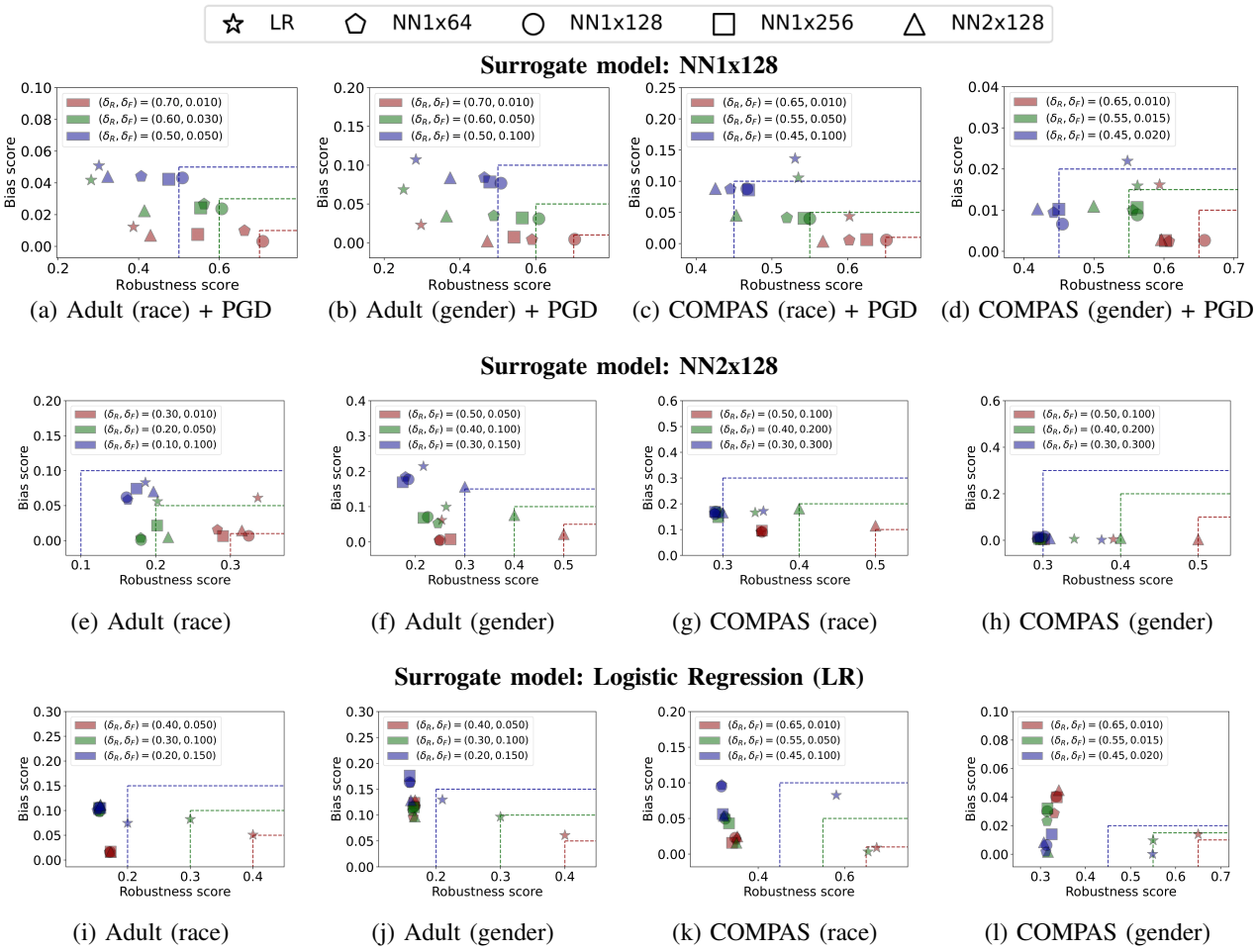
We have the following main observations. First, when the surrogate model has identical architecture as the target model, it always satisfies both fairness and robustness requirements in all the settings. Second, when the surrogate model has the same type but different architectural complexity from the target model (e.g., both are neural networks), the output may fail to meet both fairness and robustness simultaneously. Furthermore, the performance depends on the difference in the architectural complexity of the surrogate and target models. In particular, when the architecture of the surrogate model is simpler than the target model, the NN-based surrogate model that has the same number of layers but different number of neurons always has the closest performance as the target model. For example, when NN1x128 is the surrogate model

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.



(a) Adult (race) (b) Adult (gender) (c) COMPAS (race) (d) COMPAS (gender)

Figure 5: Model fairness, robustness, and accuracy of FROC-PRE method (PGD attack). Accuracy is visualized in colors; light (deep) color indicates higher (lower) accuracy



(a) Adult (race) + PGD (b) Adult (gender) + PGD (c) COMPAS (race) + PGD (d) COMPAS (gender) + PGD

(e) Adult (race) (f) Adult (gender) (g) COMPAS (race) (h) COMPAS (gender)

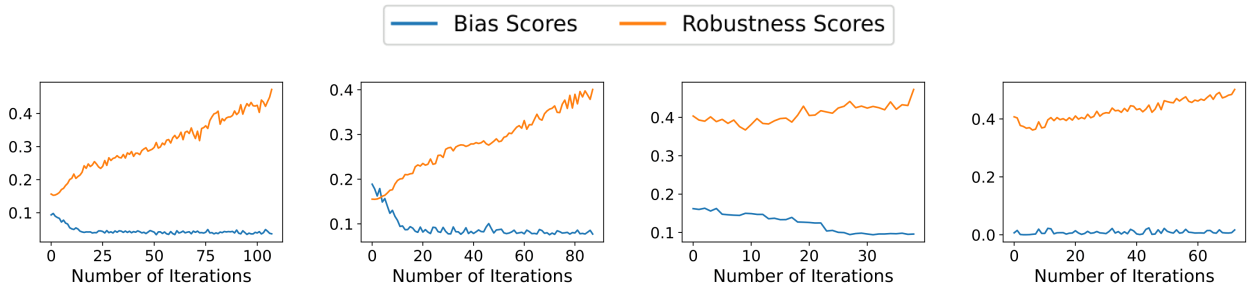
(i) Adult (race) (j) Adult (gender) (k) COMPAS (race) (l) COMPAS (gender)

Figure 6: Performance of FROC-PRE method on various target and surrogate models (PGD attack). The colored rectangles indicate the δ_F-fairness and δ_R-robustness requirements for different δ_F and δ_R thresholds. The (bias score, fairness score) points that lie either on the borders of the rectangles or inside the rectangle indicate that the model satisfies δ_F-fairness and δ_R-robustness, otherwise it fails either δ_F-fairness or δ_R-robustness or both.

(Figure 6 (a) - (d)), the performance for NN1x256 as the target model (squares in Figure 6 (a) - (d)) is better than the performance for NN2x128 (triangles in Figure 6 (a) - (d)). On the other hand, when the architecture of the surrogate model is more complex than the target model, the model that has the closest complexity to the target model has the best performance. For example, consider NN2x128 as the surrogate model (Figure 6 (e) - (h)) and Adult dataset, the performance of NN1x128 as the target model (circles in Figure 6 (e) & (f)) is better than the performance for NN1x64 (pentagons in Figure 6 (e) & (f)). On the other hand, all surrogate models whose architecture is simpler than the target model have similar performance on COMPAS dataset (Figure 6 (g) - (h)). Third, when the surrogate model is fundamentally different from the target model (e.g., LR as the surrogate model and neural networks as the target model), the output fails to meet both fairness and robustness in most of the settings. To summarize, FROC-PRE is effective for those surrogate models of similar architecture as the target model, especially the ones of the same number of layers of NN. Number of neurons at these layers does not impact significantly on the fairness and robustness performance.

Interaction between fairness and robustness during iterations. To better understand how fairness and robustness constraints interact with each other, we plot the

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.



(a) Adult (race) ($\lambda_R = 0.45$, $\lambda_F = 0.05$)
(b) Adult (gender) ($\lambda_R = 0.4$, $\lambda_F = 0.1$)
(c) COMPAS(race) ($\lambda_R = 0.45$, $\lambda_F = 0.10$)
(d) COMPAS (gender) ($\lambda_R = 0.5$, $\lambda_F = 0.02$)

Figure 7: Interaction between fairness and robustness of FROC-PRE (PGD attack). The interaction is illustrated as the change of bias and robustness scores during iterations of FROC-PRE.

change of bias and robustness scores during the iterations of FROC-PRE when PGD is the attack in Figure 7. The results of FGSM attack can be found in Appendix E. From the results in Figure 7, we observe that the competition exists between both constraints in FROC-PRE, as both scores are not consistently improved over the iterations. Indeed, the two scores can change in opposite directions, i.e., one score increases while the other decreases, in the same iteration. However, the competition is weak as eventually both scores reach their threshold requirements. We also observe that both scores are improved at different speeds. In particular, the robustness score is improved faster than that of the bias score in most of the settings, as the slope of the robustness score curve is more steep than that of the bias score curve.

7.5. FROC-IN versus FROC-PRE

We compare model accuracy, fairness, and robustness of FROC-IN and FROC-PRE. We show the results for Adult and COMPAS datasets in Figure 8. The results for Hospital dataset can be found in Appendix F. For ease of demonstration, we use different colors to illustrate the performance of two algorithms. In particular, the red area shows the cases that FROC-PRE can meet both fairness and robustness thresholds but FROC-IN cannot. The blue area shows the cases that only FROC-IN can meet both fairness and robustness thresholds but FROC-PRE cannot. The purple area shows the cases that FROC-PRE outperforms FROC-IN in terms of model accuracy when they have comparable fairness and robustness performance. And the green area shows the cases that FROC-IN outperforms FROC-PRE in terms of model accuracy under the same fairness and robustness performance.

From the results, we observe that, first, the "winner" of FROC-PRE and FROC-IN varies even on the same dataset but with different protected attributes. This is not surprising as the performance of both methods depends on the overall data distributions as well as the distributions of both protected and un-protected groups. Second, FROC-PRE is more likely to outperform FROC-IN in terms of model accuracy under FGSM setting (the purple areas in Figure 8 (a), (b), (e)) and (f)), but loses to FROC-IN in terms of model accuracy under PGD setting, except for COMPAS dataset with gender as the protected attribute. Furthermore, when the robustness threshold is very large (i.e., strong robustness requirement), FROC-PRE is more likely to fail to meet both robustness and fairness requirements than FROC-IN. On the other hand, when the bias threshold is very small (e.g., less than 0.005), FROC-IN may fail to meet both robustness and fairness requirements while FROC-PRE can satisfy (e.g., the red areas in Figure 8 (b), (d), (f), and (h)). Therefore, FROC-IN is more suitable for the strong robustness settings, while FROC-PRE is more suitable for the settings of very strong fairness requirements.

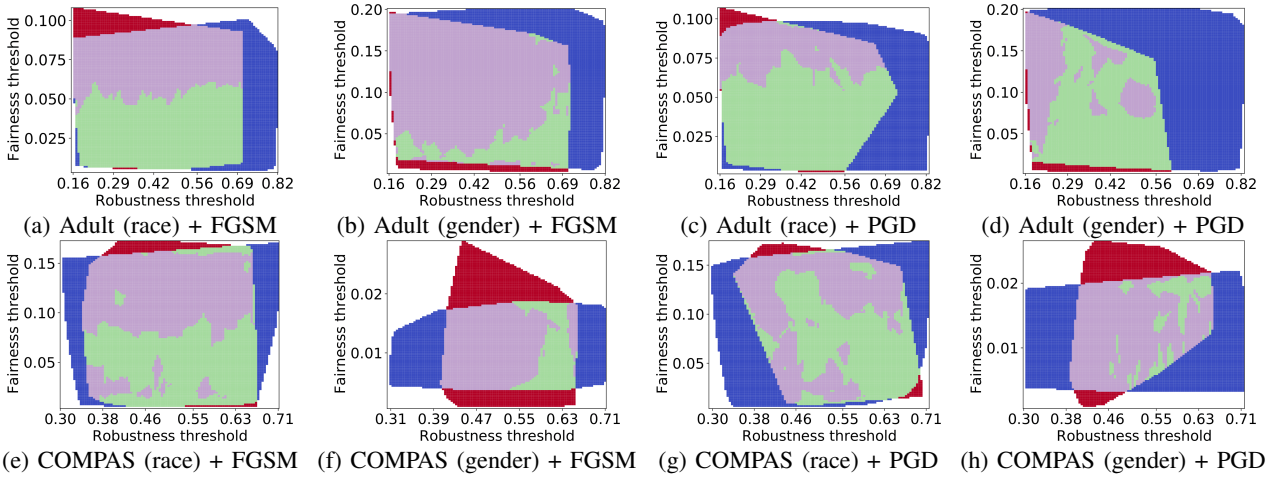
8. Discussions

Incremental learning for FROC-PRE method. One weakness of FROC-PRE is that it has to re-train the model at each iteration for the estimation of influence scores. One possible optimization is to let the model $\mathbf{M}^{(i)}$ at the i-th iteration first inherit the parameters of the model $\mathbf{M}^{(i-1)}$ from the last iteration, and use *incremental learning* techniques [46] that allows remodelling the network in an incremental way without retraining. We can adapt the stepwise updating algorithm in [46] to remodel the network by only computing the pseudoinverse of the flipped label. Since we only flip a small portion labels in each iteration, the incremental learning approach should be cost-efficient.

Adapting FROC methods to other fairness definitions. In this paper, we only consider statistical parity. However, both FROC-IN and FROC-PRE methods can be easily adapted to other fairness definitions (e.g., equal opportunity [17] and equalized odds [17]). The fairness regularizer $\mathbf{F}$ (Eqn. (13)) has to be re-designed for these fairness definitions. The fairness influence scores of FROC-PRE can be efficiently estimated by taking the gradient of the re-designed $\mathbf{F}$ accordingly.

Robustness against the data poisoning attacks. In this paper, we only consider the robustness against the evasion attacks. If the attack models change to the data poisoning attacks [47], [48], we can apply the following method to realize both fairness and robustness. First, we identify the poisoned data points and filter them out by the existing methods [49]–[51]. Then we apply the existing bias mitigation methods (e.g., [5], [8], [19]) on the cleaned data to meet the fairness requirement. Our claim is that applying bias mitigation after cleaning of poisoned samples will not counteract model robustness, as it will not insert any poisoned samples into the training data. This is different from the sequential method (Section 7.2) that provides robustness against the evasion attack, which

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.



(a) Adult (race) + FGSM (b) Adult (gender) + FGSM (c) Adult (race) + PGD (d) Adult (gender) + PGD

(e) COMPAS (race) + FGSM (f) COMPAS (gender) + FGSM (g) COMPAS (race) + PGD (h) COMPAS (gender) + PGD

Figure 8: Comparison between FROC-IN and FROC-PRE. Red area: only FROC-PRE can satisfy both fairness and robustness thresholds; Blue area: only FROC-IN can satisfy both fairness and robustness thresholds; Purple area: FROC-PRE outperforms FROC-IN in terms of model accuracy; Green area: FROC-IN outperforms FROC-PRE in terms of model accuracy

is destined to fail as it has to insert adversarial examples which may bring new bias into the model.

Extension to non-binary protected attributes. Both FROC-IN and FROC-PRE only consider binary protected attributes. When extending to non-binary protected attributes, FROC-IN has to re-design the approximation of bias score (Eqn. (11)) as well as the fairness regularizer (Eqn. (14)). FROC-PRE can utilize the same revised approximation of bias score (Eqn. (11)) to deal with the non-binary protected attributes.

Extension to problem-space adversarial attacks. By problem-space attacks, the adversary has to generate adversarial examples in the problem-space [52]–[54]. A set of problem-space constraints under problem-space attacks have been identified in [55]. Intuitively, by incorporating these problem-space constraints, FROC-PRE can be extended to defend against the problem-space data poisoning attacks by searching for the adversarial examples in the problem space that incurs the minimum influence on model accuracy (Eqn. (22)) when inserting into the data. The challenge is how to design efficient search strategies. We will leave this to the future work.

Possibility of accessing surrogate models in practice. We have shown that FROC-PRE performs well if the surrogate model has similar architecture as the target model. However, the users may not have the knowledge of target model architecture in practice. In the absence of the prior knowledge to select a surrogate model, the users can choose the surrogate model based on their experience or the popularity of the models. However, this may lead to poor performance of FROC-PRE. A more accurate but also more costly way to choose the surrogate model is to use surrogate modelling selection technique (SMTS) [56], [57] that uses machine learning techniques to learn and predict the best surrogate model whose output mimics the output of the target model with the given input.

Generalizing to multi-label classifiers. So far we only consider binary classifiers. Extending our work to multi-label classifiers can face the following two challenges. First, it can incur significant classification accuracy loss, as enforcing fairness alone on multi-label classifiers can incur heavy accuracy loss [58]. Second, by adapting both fairness regularizer (Eqn. (13)) and robustness regularizer (Eqn. (15)) to the multi-class setting, FROC-IN may suffer from high computational complexity and slow convergence, as the convergence rate of the multi-class classifier with statistical parity fairness alone can be very slow on large data [58]. The concern of slow convergence also applies to FROC-PRE.

9. Conclusion and Future Work

In this paper, we study the problem of equipping the classification models with both fairness and adversarial robustness against evasion attacks. We design two algorithms, namely FROC-IN and FROC-PRE. FROC-IN is an in-processing method that adds fairness and adversarial robustness as two regularizers to the objective function of the model, while FROC-PRE is a pre-processing method that modifies the training data to remove data bias and add adversarial examples. Our experimental results show that both FROC-IN and FROC-PRE address the trade-off among fairness, robustness, and model accuracy.

For the future work, we will take privacy, another important issue of trustworthy ML, into consideration. We will consider various types of privacy inference attacks (e.g., membership inference attack [59], [60], attribute inference attack [61], [62], and model inversion attack [63]) as well as their impacts on fairness and robustness. We will also address the trade-off between fairness, robustness, privacy, and model accuracy.

Acknowledgment

This material is based upon Wendy Hui Wang's work supported by the National Science Foundation (NSF) under Grant No. 2029038 and 2135988. Ting Wang is partially supported by NSF under Grant No. 1951729, 1953813, and 1953893.

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

References

[1] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine bias: there's software used across the country to predict future criminals. and it's biased against blacks," ProPublica, 2016.

[2] C. O'neil, *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books, 2016.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[4] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv preprint arXiv:1801.02610*, 2018.

[5] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 259–268.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[8] T. Calders, F. Kamiran, and M. Pechenizkiy, "Building classifiers with independency constraints," in *International Conference on Data Mining Workshops*, 2009, pp. 13–18.

[9] T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," *Data Mining and Knowledge Discovery*, vol. 21, no. 2, pp. 277–292, 2010.

[10] H. Chang, T. D. Nguyen, S. K. Murakonda, E. Kazemi, and R. Shokri, "On adversarial bias and the robustness of fair machine learning," *arXiv preprint arXiv:2006.08669*, 2020.

[11] H. Xu, X. Liu, Y. Li, A. Jain, and J. Tang, "To be robust or to be fair: Towards fairness in adversarial training," in *International Conference on Machine Learning*, 2021, pp. 11 492–11 501.

[12] H. Chen, H. Zhang, D. Boning, and C.-J. Hsieh, "Robust decision trees against adversarial examples," in *International Conference on Machine Learning*, 2019, pp. 1122–1131.

[13] J. Dressel and H. Farid, "The accuracy, fairness, and limits of predicting recidivism," *Science advances*, vol. 4, no. 1, 2018.

[14] M. Ribeiro, K. Grolinger, and M. A. Capretz, "Mlaas: Machine learning as a service," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 896–902.

[15] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, "Chiron: Privacy-preserving machine learning as a service," *arXiv preprint arXiv:1803.05961*, 2018.

[16] A. Chouldechova and A. Roth, "The frontiers of fairness in machine learning," *arXiv preprint arXiv:1810.08810*, 2018.

[17] M. Hardt, E. Price, N. Srebro *et al.*, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, 2016, pp. 3315–3323.

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

[19] F. Kamiran and T. Calders, "Classifying without discriminating," in *2nd International Conference on Computer, Control and Communication*, 2009, pp. 1–6.

[20] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1171–1180.

[21] G. Goh, A. Cotter, M. Gupta, and M. P. Friedlander, "Satisfying real-world goals with dataset constraints," in *Advances in Neural Information Processing Systems*, 2016, pp. 2415–2423.

[22] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2012, pp. 35–50.

[23] Y. Bechavod and K. Ligett, "Learning fair classifiers: a regularization-inspired approach," *CoRR*, vol. abs/1707.00044, 2017.

[24] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.

[25] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14 410–14 430, 2018.

[26] E. Tabassi, K. Burns, M. Hadjimichael, A. Molina-Markham, and J. Sexton, "A taxonomy and terminology of adversarial machine learning," *NIST IR*, 2019.

[27] S. Sharma, A. H. Gee, D. Paydarfar, and J. Ghosh, "Fair-n: Fair and robust neural networks for structured data," *arXiv preprint arXiv:2010.06113*, 2020.

[28] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 4, pp. 984–996, 2013.

[29] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019.

[30] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[31] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*, 2017.

[32] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[33] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2021–2031.

[34] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *arXiv preprint arXiv:1904.12843*, 2019.

[35] C. Lyu, K. Huang, and H.-N. Liang, "A unified gradient regularization family for adversarial examples," in *IEEE international conference on data mining*, 2015, pp. 301–309.

[36] U. Shaham, Y. Yamada, and S. Negahban, "Understanding adversarial training: Increasing local stability of neural nets through robust optimization," *arXiv preprint arXiv:1511.05432*, 2015.

[37] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International Conference on Machine Learning*, 2017, pp. 1885–1894.

[38] R. D. Cook and S. Weisberg, *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

[39] P. Yang, J. Chen, C.-J. Hsieh, J.-L. Wang, and M. I. Jordan, "Greedy attack and gumbel attack: Generating adversarial examples for discrete data." *Journal of Machine Learning Research*, vol. 21, no. 43, pp. 1–36, 2020.

[40] R. Kohavi and B. Becker, "Uci machine learning repository: Adult data set," https://archive.ics.uci.edu/ml/datasets/Adult.

[41] L. K. Jeff Larson, Surya Mattu and J. Angwin, "Data and analysis for 'machine bias'," https://github.com/propublica/compas-analysis/.

[42] T. D. of State Health Services, "Hospital discharge data use agreement," https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm.

[43] Y. Li, H. Sun, and W. H. Wang, "Towards fair truth discovery from biased crowdsourced answers," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 599–607.

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.

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

[45] S. Yan, H.-t. Kao, and E. Ferrara, "Fair class balancing: enhancing model fairness without observing sensitive attributes," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1715–1724.

[46] C. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 1, pp. 62–72, 1999.

[47] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[48] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, and F. Roli, "Support vector machines under adversarial label contamination," *Neurocomputing*, vol. 160, pp. 53–62, 2015.

[49] R. Laishram and V. V. Phoha, "Curie: A method for protecting svm classifier from poisoning attack," *arXiv preprint arXiv:1606.01584*, 2016.

[50] J. Chen, X. Zhang, R. Zhang, C. Wang, and L. Liu, "De-pois: An attack-agnostic defense against data poisoning attacks," *IEEE Transactions on Information Forensics and Security*, 2021.

[51] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," *arXiv preprint arXiv:1706.03691*, 2017.

[52] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," *arXiv preprint arXiv:1804.07998*, 2018.

[53] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," *arXiv preprint arXiv:1812.05271*, 2018.

[54] E. Quiring, A. Maier, and K. Rieck, "Misleading authorship attribution of source code using adversarial learning," in *28th USENIX Security Symposium*, 2019, pp. 479–496.

[55] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ml attacks in the problem space," in *IEEE Symposium on Security and Privacy*, 2020, pp. 1332–1349.

[56] B. S. Saini, M. López-Ibáñez, and K. Miettinen, "Automatic surrogate modelling technique selection based on features of optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 1765–1772.

[57] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, "Surrogate-based analysis and optimization," *Progress in aerospace sciences*, vol. 41, no. 1, pp. 1–28, 2005.

[58] C. Denis, R. Elie, M. Hebiri, and F. Hu, "Fairness guarantee in multi-class classification," *arXiv preprint arXiv:2109.13642*, 2021.

[59] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symposium on Security and Privacy*, 2017, pp. 3–18.

[60] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *23rd USENIX Security Symposium*, 2014, pp. 17–32.

[61] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137–150, 2015.

[62] N. Z. Gong and B. Liu, "You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors," in *25th USENIX Security Symposium*, 2016, pp. 979–995.

[63] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

Data	PA	Acc.	$\mathcal{S}_B$	$\mathcal{S}_R$	
				FGSM	PGD
Adult	Race	0.8472	0.0904	0.1579	0.1477
	Gender	0.8473	0.1933	0.1577	0.1460
COMPAS	Race	0.7096	0.1523	0.3015	0.2867
	Gender	0.7043	0.0194	0.3035	0.2834
Hospital	Race	0.6593	0.0153	0.3407	0.3328
	Gender	0.6572	0.0175	0.3430	0.3299

TABLE 2: Testing accuracy (Acc.), fairness ($\mathcal{S}_B$), and robustness ($\mathcal{S}_R$) of three datasets with the average of 20 repeats (PA = protected attribute)

Appendix A. Details of Real-World Datasets

We use the following datasets in the experiments: (1) **Adult dataset** [40] includes 45,222 instances and 14 attributes (such as age, gender, education, marital status, occupation, working hours, and native country) that describe the information about individuals from the 1994 U.S. census. The prediction task is to determine whether a person makes over $50K annually. (2) **COMPAS dataset** [41] contains criminal history, jail and prison time, demographics and COMPAS (which stands for Correctional Offender Management Profiling for Alternative Sanctions) risk scores for defendants from Broward County, Florida. The prediction task is to infer a criminal defendant's likelihood of becoming a recidivist (i.e., a criminal who re-offend) within two years. (3) **Hospital dataset** [42] is released by the Texas Department of State Health Services. It contains records of inpatient stays in some health facilities. The features include types of external causes of injury, diagnosis, the procedures the patient underwent, and demographic information such as gender, age, and race. The classification task is to predict the patient's main procedure. We categorize the main procedures into two groups (corresponding to the prediction labels): cardiology and pulmonology.

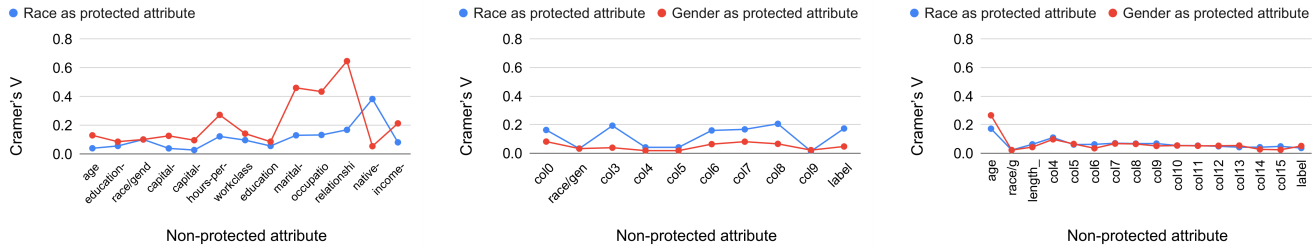
Appendix B. Fairness and Robustness on Original Data

We measure testing accuracy, fairness, and robustness of the target model on the three datasets, and show the results in Table 2. It can be observed that, first, the target model has noticeable bias ($\mathcal{S}_B \in [0.0153, 0.19]$) in its prediction results. Second, the target model is not robust against both FGSM and PGD attacks ($\mathcal{S}_R \in [0.146, 0.343]$).

Appendix C. Correlation between Protected and Non-protected Attributes

To understand why excluding the protected attributes from the training of the target model cannot eliminate the tension between fairness and robustness, we measure the correlation between protected and non-protected attributes. Since some attributes are categorical, we consider *Cramer's V measurement*, which measures the relation between two categorical variables. It returns a value in the

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.



(a) Adult (b) COMPAS (c) Hospital

Figure 9: Correlation between the protected and non-protected attributes.

range [0, 1]. In general, a Cramer's V value $v \in [0.1, 0.3]$ indicates a weak association, $v \in [0.3, 0.5]$ indicates a medium association, and $v > 0.5$ indicates a strong association. The results of Cramer's V measurement is shown in Figure 9. We observe a strong association between the non-protected attributes and the protected attributes on Adult dataset. For example, the Cramer's V between the non-protected *relationship* attribute and the protected *gender* attribute is 0.65. On the other hand, the correlation between the protected and the non-protected attributes is weak on COMPAS and Hospital datasets (no more than 0.2 on COMPAS dataset and 0.3 on Hospital datasets). However, even with such weak correlations, simply removing the protected attribute did not eliminate its disparate impact on the decision, which can be observed from the high bias score on COMPAS and Hospital dataset (e.g., the bias score of the original classifier on COMPAS dataset when Race is the protected attribute is as high as 0.1523). Thus it still leads to the tension between fairness and robustness.

Appendix D. Additional Results of FROC-IN Method

Model fairness, robustness, and accuracy. Figure 10 shows the performance results of FROC-IN when FGSM is the adversarial attack on Adult and COMPAS datasets. The observations are similar to that of PGD attack (Figure 3). We omit the discussions due to limited space.

Interaction between fairness and robustness regularizers. Figure 11 shows the interaction between fairness and robustness regularizers during training of FROC-IN method when FGSM is the attack. We omit the discussions as they are similar to PGD as the attack (Figure 4).

Appendix E. Additional Results of FROC-PRE

Model fairness, robustness, and accuracy. Figure 12 shows model fairness, robustness, and accuracy for FGSM attack. The results are similar to that for PGD attack (Figure 5).

Interaction between bias and robustness. Figure 13 illustrates the interaction between bias and robustness scores over the iterations of FROC-PRE when FGSM is the attack. The observations are similar to that for PGD as the attack (Figure 7). We omit the discussions due to the limited space.

Various target and surrogate models. Figure 14 presents the performance of FROC-PRE on various surrogate models when FGSM is the attack. The observations are similar as Figure 6 and thus are omitted due to limited space.

Appendix F. Additional Results on Hospital Dataset

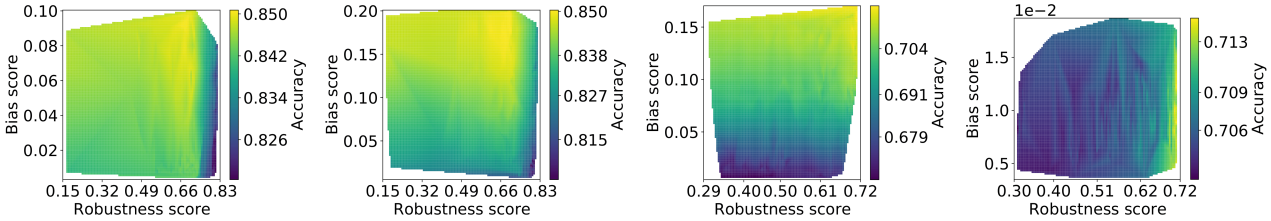
Performance of sequential methods The performance of mode fairness by both sequential methods on Hospital dataset is shown in Figure 15. The observation is similar to that on Adult and COMPAS datasets: fairness can be deteriorated by robustness, as the bias score of the model with robustness deployment can be higher than that of the model without fairness. In all settings, the bias score is the same as that for the model without fairness when the parameter $\lambda = 1$ (i.e., the strongest robustness). In other words, strong robustness can eliminate the fairness effect by both DIR and RW on the model.

The performance of model robustness by two sequential methods on both Adult and COMPAS datasets are shown in Figure 16, where fairness is implemented by pre-processing the training dataset with Disparate Impact Remover method (DIR) [5]. Surprisingly, for both methods, the robustness score stays stable regardless the change of fairness. Our analysis shows that DIR does not change the distribution of data near the boundary of the classifiers; thus the robustness score remains stable.

Performance of FROC-IN. Figure 17 shows the performance of FROC-PRE on Hospital dataset. Our observation is similar to that on Adult and COMPAS datasets (Figure 3) - FROC-IN addresses the trade-off between fairness, robustness and accuracy. In most of the settings, the accuracy downgrades when the robustness scores grows (i.e., stronger robustness). Similarly, the accuracy decreases when the bias scores decreases (i.e., more fairness).

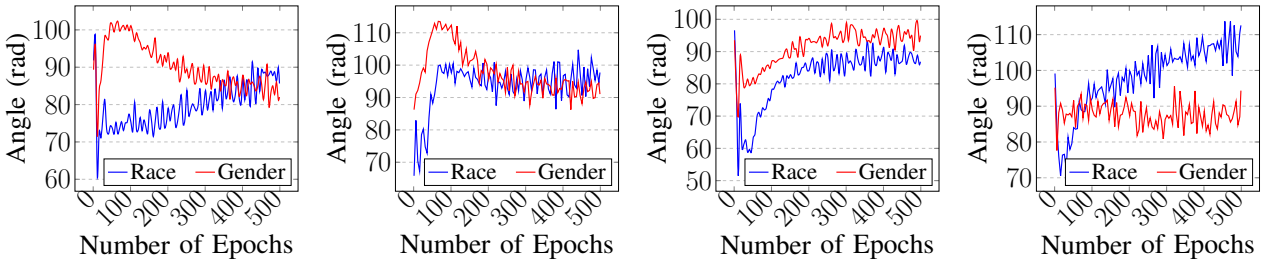
We also measure the interaction between fairness and robustness regularizers during training of FROC-IN. Figure 18 shows the angle between fairness and robustness regularizers during training. Due to the medium correlation between un-protected attributes and the protected attribute on Hospital dataset (the strongest correlation is 0.3), the competition between the two regularizers can be tense, which leads to the angle between the gradients larger than $100°$ at the initial 100 epochs (red line in Figure 18 (a) & (b)) However, similar to Adult and COMPAS dataset, the angle between $\nabla_\theta \mathbf{F}$ and $\nabla_\theta \mathbf{R}$ shrinks into the

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.



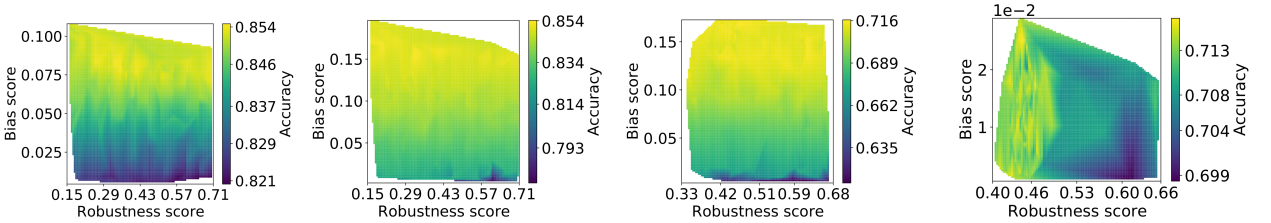
(a) Adult (race) (b) Adult (gender) (c) COMPAS (race) (d) COMPAS (gender)

Figure 10: Model fairness, robustness, and accuracy of FROC-IN method (FGSM attack). X- and y- axis show robustness scores and bias scores respectively. Accuracy is visualized in colors; light (deep) color indicates higher (lower) accuracy



(e) Adult ($\lambda_R = 0.1, \lambda_F = 0.3$) (f) Adult ($\lambda_R = 1.0, \lambda_F = 1.0$) (g) COMPAS ($\lambda_R = 0.1, \lambda_F = 0.3$) (h) COMPAS ($\lambda_R = 1.0, \lambda_F = 1.0$)

Figure 11: Interaction between fairness and robustness regularizers during training of FROC-IN method (FGSM attack). The interaction is measured as the angle between the gradients of both fairness and robustness regularizers



(a) Adult (race) + FGSM (b) Adult (gender) + FGSM (c) COMPAS (race) + FGSM (d) COMPAS (gender) + FGSM

Figure 12: Model fairness, robustness, and accuracy of FROC-PRE method (FGSM attack). Accuracy is visualized in colors; light (deep) color indicates higher (lower) accuracy

range of $[70°, 110°]$ after the initial 20 epochs for all the settings of Hospital dataset, and then keeps stable when it is close to $90°$.

Performance of FROC-PRE. Figure 19 shows the results of our FROC-PRE for Hospital dataset. It can be observed that FROC-PRE well addresses the trade-off among fairness, robustness, and model accuracy. The accuracy decreases at most 1.31 % for Hospital dataset in all settings.

FROC-PRE versus FROC-IN From the results in Figure 20, we have the following observations: First, the "winner" of FROC-PRE and FROC-IN varies even on the same dataset but with different protected attributes. Second, when the robustness threshold is very large (i.e., strong robustness requirement), FROC-PRE is more likely to fail to meet both robustness and fairness requirements than FROC-PRE. Similarly, when the fairness threshold is very small (i.e., strong fairness requirement), FROC-PRE may fail to meet both robustness and fairness requirements more frequently than FROC-IN. This shows that FROC-IN is more suitable than FROC-PRE on Hospital dataset.

	Age	Education	Marital-status	Hours-per-week
Example 1				
Ori.	0.39	Assoc-acdm	Divorced	0.39
Adv.	0.29	10th	Married-spouse-absent	0.31
Example 2				
Ori.	0.05	HS-grad	Never-married	0.14
Adv.	0.15	Prof-school	Married-civ-spouse	0.24

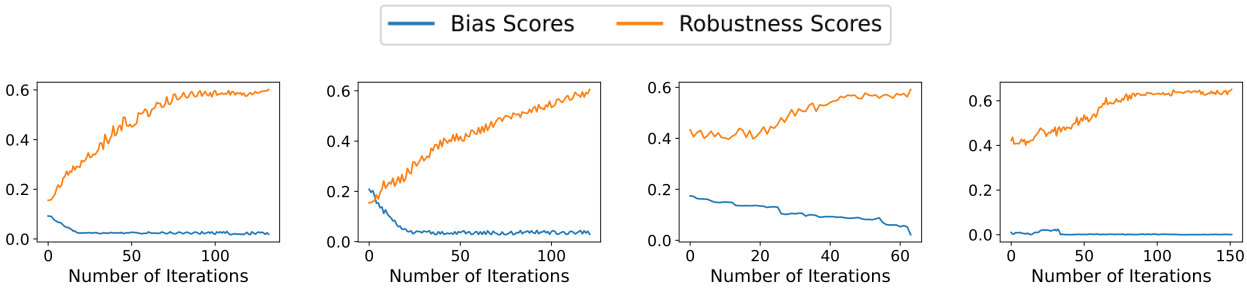
TABLE 3: Examples of adversarial examples ($\lambda_R = 0.1$). The values of both *Age* and *Hours-per-week* attributes are normalized.

Appendix G. Generated Adversarial Examples

As explained in Section 6.2, we follow [39] to generate adversarial examples in the discrete domain. We use the implementation[3] of [39], and illustrate two adversarial examples of Adult dataset in Table 3. The examples include two numerical attributes *Age* and *Hours-per-week*, and two categorical attributes *Education* and *Marital-status*. The values of both numerical attributes are normalized.

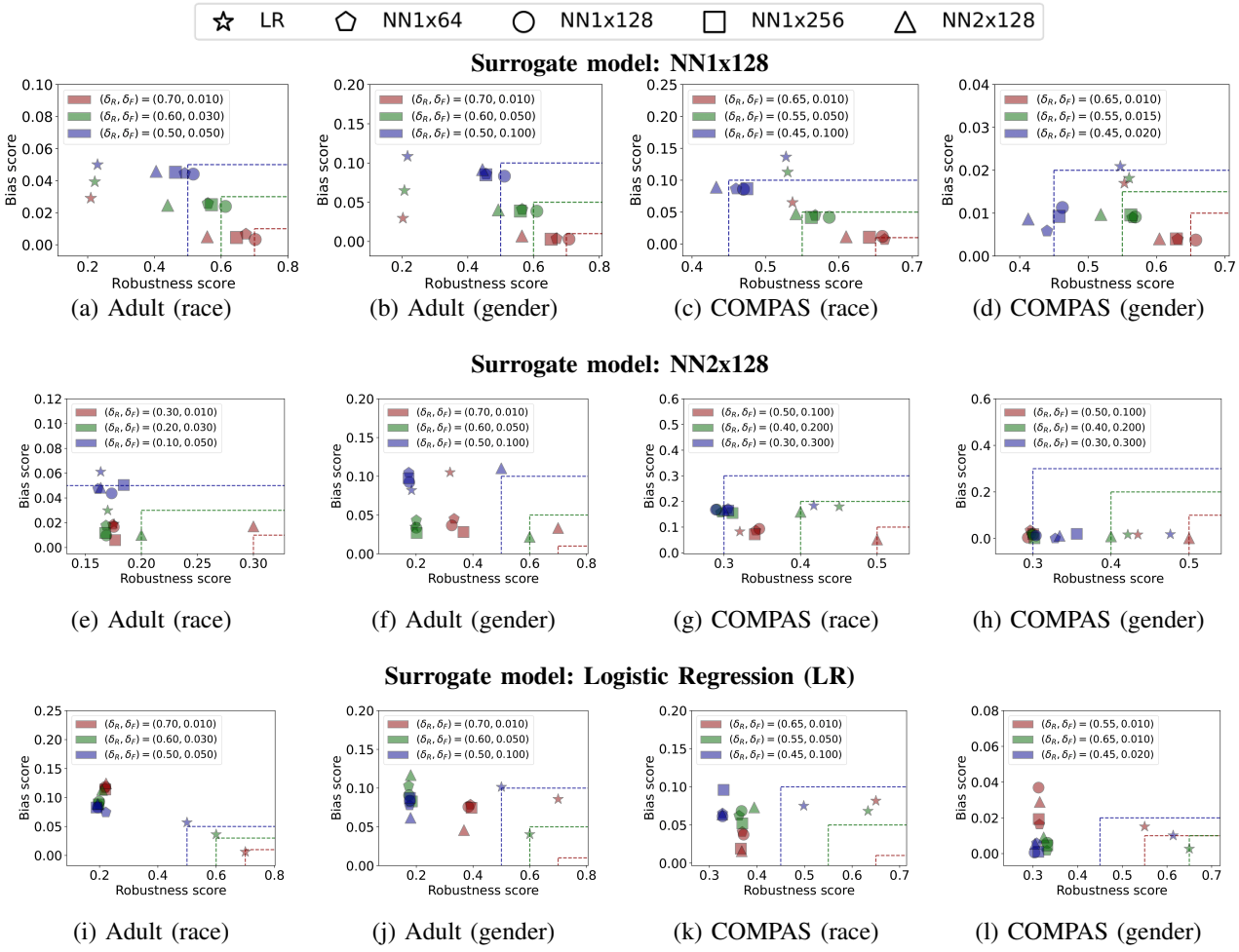
3. https://github.com/ggcodeanonymous/Greedy-Attack-and-Gumbel-Attack

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.



(a) Adult (race) ($\lambda_R = 0.6$, $\lambda_F = 0.03$)
(b) Adult (gender) ($\lambda_R = 0.6$, $\lambda_F = 0.05$)
(c) COMPAS (race) ($\lambda_R = 0.55$, $\lambda_F = 0.05$)
(d) COMPAS (gender) ($\lambda_R = 0.65$, $\lambda_F = 0.01$)

Figure 13: Interaction between fairness and robustness of FROC-PRE (FGSM attack). The interaction is illustrated as the change of bias and robustness scores during iterations of FROC-PRE



(a) Adult (race) (b) Adult (gender) (c) COMPAS (race) (d) COMPAS (gender)

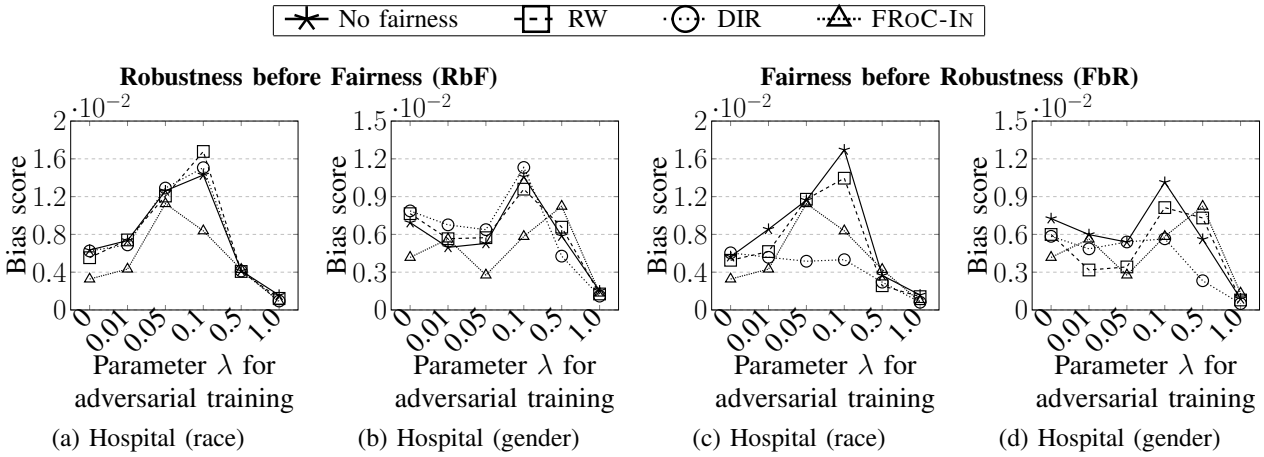
(e) Adult (race) (f) Adult (gender) (g) COMPAS (race) (h) COMPAS (gender)

(i) Adult (race) (j) Adult (gender) (k) COMPAS (race) (l) COMPAS (gender)

Figure 14: Performance of FROC-PRE method with various target and surrogate models (FGSM attack, Adult & COMPAS datasets). The colored rectangles indicate the δ_F-fairness and δ_R-robustness requirements for different δ_F and δ_R thresholds. The (bias score, fairness score) points that lie either on the edges of the rectangles or inside of the rectangle indicate that the model satisfies δ_F-fairness and δ_R-robustness, otherwise it fails either δ_F-fairness or δ_R-robustness or both. NN2x128 model is the surrogate model.

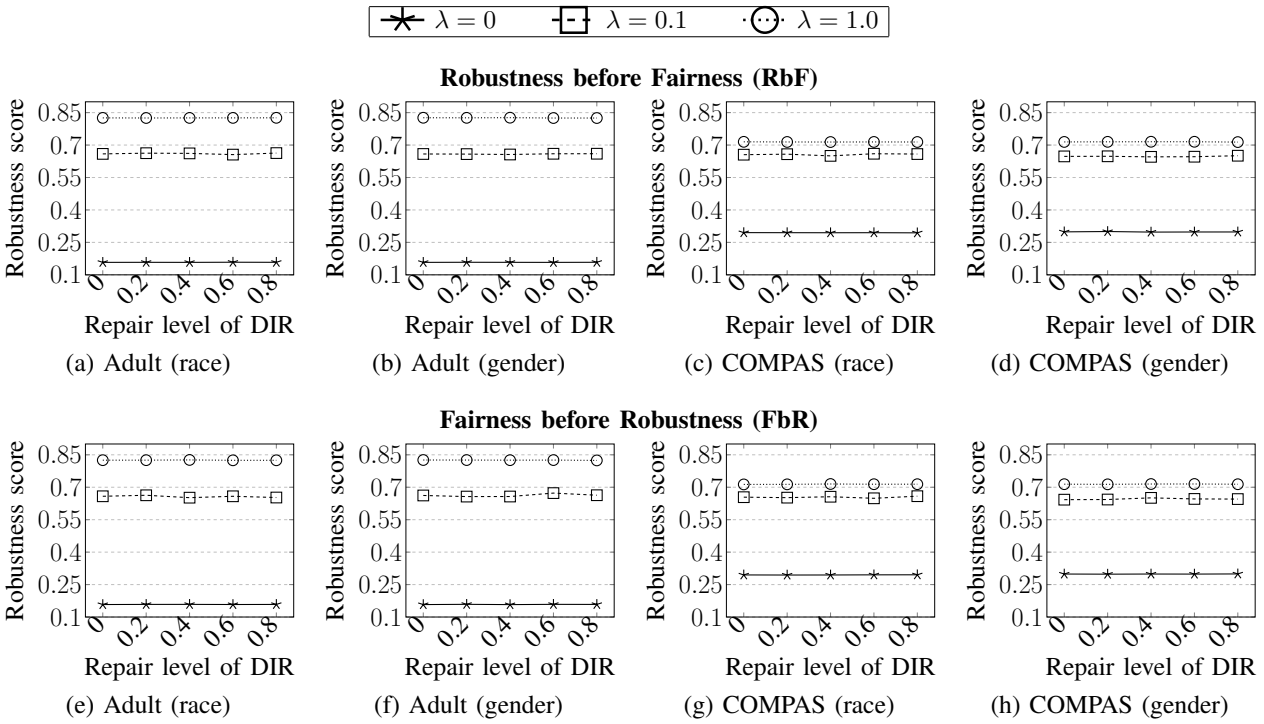
The adversarial examples are generated from the discrete domain of both categorical attributes. For example, in Example 1, the adversarial example of "Divorced" is "Married-spouse-absent", where the indices of these two values is one position away in the domain of the attribute *Marital-status*.

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.



(a) Hospital (race) (b) Hospital (gender) (c) Hospital (race) (d) Hospital (gender)

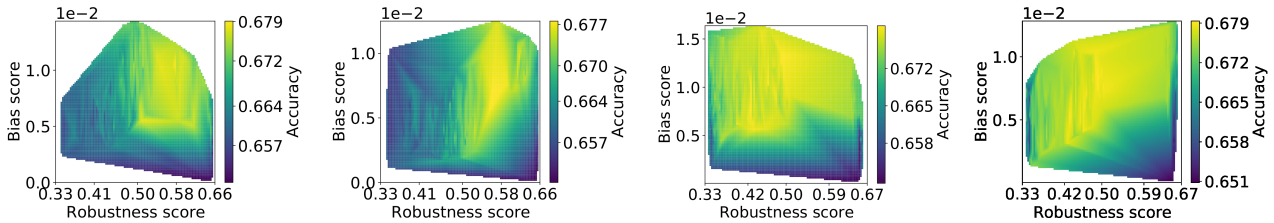
Figure 15: Comparison between performance of both sequential methods and FROC-IN on Hospital dataset. Robustness is implemented by inserting adversarial examples generated by the PGD attack. Fairness is implemented by pre-processing training data with either Reweighing (RW) [44] or Disparate Impact Remover method (DIR) [5]



(a) Adult (race) (b) Adult (gender) (c) COMPAS (race) (d) COMPAS (gender)

(e) Adult (race) (f) Adult (gender) (g) COMPAS (race) (h) COMPAS (gender)

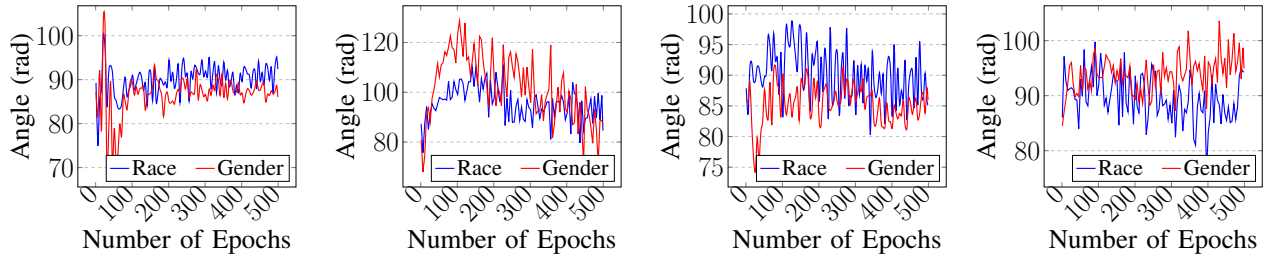
Figure 16: Robustness of two sequential methods. Robustness is implemented as adding adversarial examples generated by PGD attack. Fairness is implemented by pre-processing the training dataset with Disparate Impact Remover method (DIR) [5]



(a) Hospital (race) + FGSM (b) Hospital (gender) + FGSM (c) Hospital (race) + PGD (d) Hospital (gender) + PGD

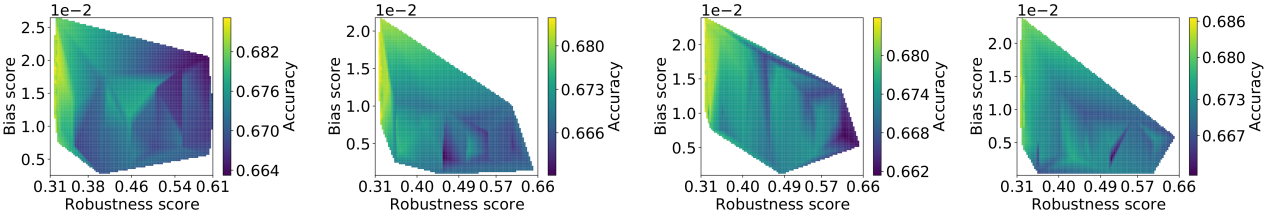
Figure 17: Model fairness, robustness, and accuracy of FROC-IN method on Hospital dataset. Accuracy is visualized in colors; light (deep) color indicates higher (lower) accuracy

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.



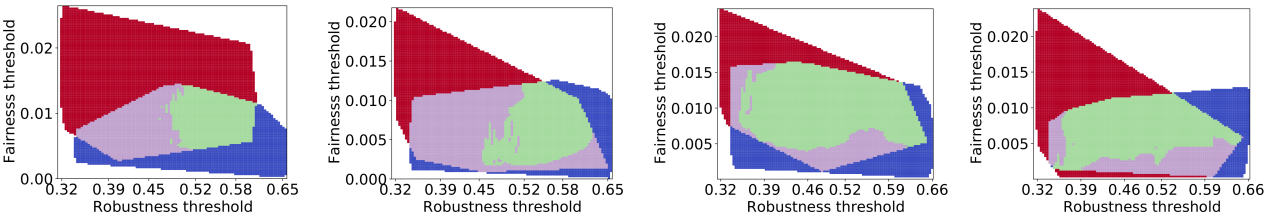
(a) Hospital (race) + FGSM (b) Hospital (gender) + FGSM (c) Hospital (race) + PGD (d) Hospital (gender) + PGD

Figure 18: Interaction between fairness and robustness regularizers during training of FROC-IN, measured as the angle between the gradients of the two regulariziers (Hospital dataset)



(a) Hospital (race) + FGSM (b) Hospital (gender) + FGSM (c) Hospital (race) + PGD (d) Hospital (gender) + PGD

Figure 19: Model fairness, robustness, and accuracy of FROC-PRE (Hospital dataset). Accuracy is visualized in colors; light (deep) color indicates higher (lower) accuracy



(a) Hospital (race) + FGSM (b) Hospital (gender) + FGSM (c) Hospital (race) + PGD (d) Hospital (gender) + PGD

Figure 20: Comparison between FROC-IN and FROC-PRE on Hospital dataset. Red area: the cases that only FROC-PRE can satisfy both fairness and robustness thresholds; Blue area: the cases that only FROC-IN can satisfy both fairness and robustness thresholds; Purple area: FROC-PRE outperforms FROC-IN in terms of model accuracy; Green area: FROC-IN outperforms FROC-PRE in terms of model accuracy

Authorized licensed use limited to: Penn State University. Downloaded on August 16,2022 at 15:46:19 UTC from IEEE Xplore. Restrictions apply.