FR-Train: A Mutual Information-Based Approach to Fair and Robust Training

Yuji Roh [1] **Kangwook Lee** [2] **Steven Euijong Whang** [1] **Changho Suh** [1]

Abstract

Trustworthy AI is a critical issue in machine learning where, in addition to training a model that is accurate, one must consider both *fair* and *robust* training in the presence of data bias and poisoning. However, the existing model fairness techniques mistakenly view poisoned data as an additional bias to be fixed, resulting in severe performance degradation. To address this problem, we propose FR-Train, which *holistically performs fair and robust model training*. We provide a mutual information-based interpretation of an existing adversarial training-based fairness-only method, and apply this idea to architect an additional discriminator that can identify poisoned data using a clean validation set and reduce its influence. In our experiments, FR-Train shows almost no decrease in fairness and accuracy in the presence of data poisoning by both mitigating the bias and defending against poisoning. We also demonstrate how to construct clean validation sets using crowdsourcing, and release new benchmark datasets[1].

1. Introduction

As machine learning becomes widespread in the Software 2.0 era (Karpathy, 2017), *trustworthy AI* is becoming increasingly critical. In addition to simply training accurate models, there is an urgent need to address multiple requirements including fairness, robustness, explainability, transparency, and accountability altogether (IBM, 2020). In particular, we focus on fairness and robustness, which are closely related issues that are affected by the same training data. For sensitive applications like healthcare, finance, and self-driving cars, a trained model must not discriminate customers based on sensitive attributes including age, sex, or religion. In addition, as applications often rely on external datasets for their training data, the model training must be resilient against noisy, subjective, or even adversarial data.

Traditionally, model fairness research (Venkatasubramanian, 2019; Chouldechova & Roth, 2018; Verma & Rubin, 2018) has focused on developing metrics such as disparate impact (Feldman et al., 2015), equalized odds (Hardt et al., 2016), and equal opportunity (Hardt et al., 2016), which capture various notions of discrimination. More recently, there has been a surge in *unfairness mitigation* techniques (Bellamy et al., 2018b), which improve the model fairness by either fixing the training data, training process, or trained model. Unfairness mitigation usually involves some tradeoff between the model's accuracy and fairness. Most recently, generative adversarial networks (GANs) are being adapted to a fairness setting (Zhang et al., 2018a). The architecture of GANs is suitable because accuracy and fairness are not always aligned, and it makes sense to simultaneously train two models: a classifier that predicts labels using input features and an adversary that predicts sensitive attributes using the classifier's predicted labels.

Robust model training is also important and needs to be concurrently taken into consideration. As dataset publishing is becoming mainstream as demonstrated by systems like Kaggle and Google Dataset Search (Noy et al., 2019), it is easy to publish data that is noisy, subjective, and even adversarial, which we hereafter refer to as *poisoned data*. As a result, there has been a proliferation of algorithms that make model training resilient to data poisoning as well (Natarajan et al., 2013; Biggio et al., 2011; Frénay & Verleysen, 2014). However, data poisoning attacks have become increasingly sophisticated, and defending against all of them is difficult (Koh et al., 2018).

Solving model fairness without addressing data poisoning may lead to a worse tradeoff between accuracy and fairness. For example, consider a banking system that is giving out loans where there are two sensitive groups: men and women. Suppose we use disparate impact (Feldman et al., 2015) as the fairness measure. If the model's positive prediction rate is M for men and W for women, the disparate impact is $\min\{\frac{M}{W}, \frac{W}{M}\}$ where a value of 1 is considered perfectly fair. Figure 1 shows a toy example of five men and five women

[1]School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea [2]Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, Wisconsin, USA. Correspondence to: Steven Euijong Whang <swhang@kaist.ac.kr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

[1]https://github.com/yuji-roh/fr-train

who need loans. Each person is associated with a single-dimensional feature x, and only the ones with a rounded box would pay back their loans (i.e., their labels are positive). Let us train a threshold classifier that divides the people into two groups where those on the left are denied loans and those on the right are granted loans. On the clean data above, a classifier that does not consider fairness (non-fair classifier, red dotted line) can have perfect accuracy at the cost of having a disparate impact of 0.5 because 40% of females are granted loans while 80% of males are granted loans. On the other hand, a fair classifier (blue solid line) can divide the people such that the disparate impact is perfect, but the accuracy is only 0.8. Now suppose we poison the data where we flip the labels of the 5^{th} and 7^{th} persons (both male) from positive to negative as shown below. While each classifier is trained on the poisoned data, its accuracy is measured using the clean data labels. For the non-fair classifier trained on this data, the results are mixed where the accuracy decreases from 1 to 0.9, but the disparate impact increases from 0.5 to 0.67. However, the fair classifier has strictly worse results where the accuracy decreases from 0.8 to 0.6 without any change in the disparate impact. Hence, the fair classifier's accuracy-fairness tradeoff is worse when the data is poisoned. One proposal is to sanitize the data prior to the model training, but it is known that removing poisoning without any knowledge of the model is extremely difficult (Koh et al., 2018).

Our main contribution is an integrated solution called FR-Train, which trains accurate models that are also fair and robust to poisoning. FR-Train extends a state-of-the-art fairness-only method called Adversarial Debiasing (AD) (Zhang et al., 2018a), which consists of a generator used for classification and a discriminator that distinguishes predictions from one sensitive group against others, similar to GANs (Goodfellow et al., 2014). The discriminator ensures that the prediction $\hat{y}$ is independent of the sensitive attribute z. We first provide interpretation of such an adversarial learning approach using mutual information. We then use the results as an inspiration to add a new robustness discriminator that uses mutual information to distinguish (training examples, predictions) of the training data from (validation examples, validation labels) of a separate and clean validation set. This discriminator ensures that the model predictions on the training data are "consistent" with labels on clean data, where the clean validation set acts as a reference to the training. In addition, we also utilize the robustness discriminator results to further improve the fairness training by re-weighting examples. In our experiments, we show that addressing robustness and fairness sequentially during model training is not as effective as addressing them concurrently as in FR-Train.

Another contribution is addressing the challenge of constructing a clean validation set and gracefully handling the case where it is small or unavailable. To this end, we demonstrate a practical crowdsourcing method using majority voting for constructing a clean validation set, which has less poisoning than the input data. We construct clean validation sets from real datasets using Amazon Mechanical Turk and release them as a community resource. In the worst case when the validation set is non-existent, we show how the parameters of FR-Train can be adjusted to still maintain reasonable accuracy and fairness.

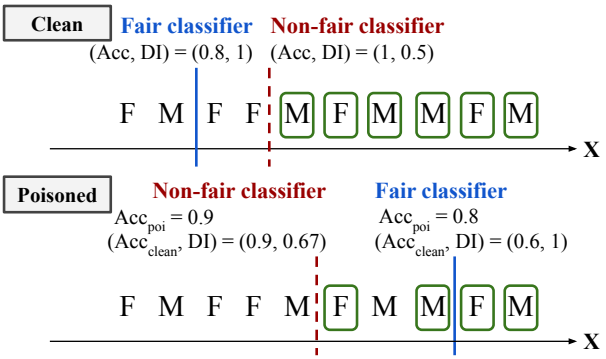


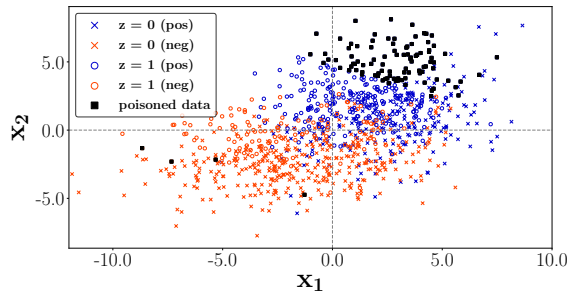
Figure 1. A small dataset of 10 people who need loans (F: female, M: male). A rounded box indicates a positive label. The clean data (above) is poisoned by flipping two labels (below). The vertical lines are the decision boundaries of non-fair and fair threshold classifiers. DI is disparate impact, and $\text{Acc}_{\text{clean}}$ (Acc_{poi}) is the accuracy on clean (poisoned) data.

In the following sections, we demonstrate the weaknesses of current fairness methods, propose FR-Train with experiments, and present the related work.

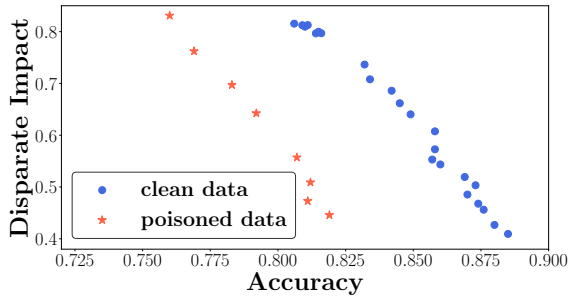
2. Vulnerability of Fairness Methods

We perform experiments to demonstrate that state-of-the-art fairness methods are indeed vulnerable even to simple poisoning attacks. We generate a synthetic dataset as shown in Figure 2a (see the generation details in Section 4.1). There are two non-sensitive attributes x_1 and x_2, which are reflected in the x-axis and y-axis, respectively. The examples are further divided into two classes based on the *sensitive* attribute z. For generation of poisoned data, we poison 10% of the training data by flipping the labels of examples that belong to a specific z attribute (for this experiment z = 1) so as to maximize the accuracy performance degradation. This approach is similar to an existing label flipping method (Paudice et al., 2018). To make a validation set, we randomly select clean examples that amount to 10% of the entire training data.

We use disparate impact as the fairness measure and evaluate a fairness method called Fairness Constraints (Zafar et al., 2017), which incorporates a regularization term that reflects fairness constraints in the context of convex margin-based classifiers such as logistic regression and support vector machines (SVMs). As this method involves a regularization



(a) Synthetic data with label-flipped poisoning



(b) Accuracy-fairness tradeoff curves for Fairness Constraints

Figure 2. The top figure shows a synthetic dataset with data poisoning. Examples are divided into z = 1 (marked with circles) and z = 0 (crosses) as per a sensitive attribute z. The blue points indicate positive labels while the red points denote negative ones. For the poisoning, we flipped labels of 10% of the examples with z = 1 so as to maximize the accuracy performance degradation (Paudice et al., 2018). The bottom figure shows that poisoning significantly worsens the accuracy-fairness tradeoff (i.e., the curve shifts to the left) of the Fairness Constraints method (Zafar et al., 2017).

factor λ that balances the accuracy and fairness objectives, we can obtain a tradeoff curve by adjusting its value. Figure 2b shows two accuracy-fairness tradeoff curves obtained with the clean and poisoned synthetic datasets. Notice that adding data poisoning clearly shifts the curve to the left, which means accuracy decreases. This coincides with our intuition. The poisoning confuses the model so that there are more biased examples to fix, which in turn makes it *overreact* and thus sacrifice more on accuracy. We also leave in the supplementary the accuracy-fairness tradeoff curves of Fairness Constraints on real datasets. The results clearly show that both accuracy and fairness decrease on the poisoned data. In Section 4, we will show how data poisoning affects other fairness methods.

3. FR-Train

We now describe FR-Train (see Figure 3). Unlike traditional GANs, the generator is a classifier that receives an example $x \in X$ and returns a prediction $\hat{y}$. There are two discriminators that respectively optimize fairness and robustness using mutual information. In addition, the outputs of the robustness discriminator can be used to further improve the

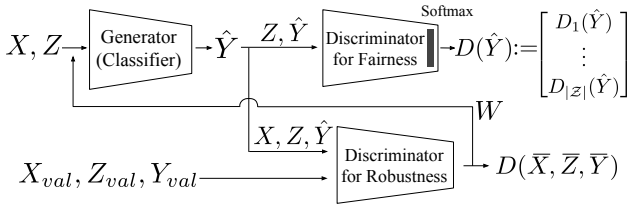


Figure 3. The architecture of FR-Train.

fairness training by re-weighting examples.

3.1. Fairness

We denote by $\mathcal{D}_{tr}$ the training data set. Suppose $\mathcal{D}_{tr}$ has m examples $\{(x^{(i)}, z^{(i)}, y^{(i)})\}_{i=1}^{m}$ where $x^{(i)}$ contains the non-sensitive attributes, $z^{(i)}$ contains the sensitive attributes, and $y^{(i)}$ is the label. Both the sensitive attribute and label can be multi-class, i.e., they can have one of multiple values. For notational simplicity, we assume there is one sensitive attribute, which can be viewed as a merged result of multiple sensitive attributes with a larger alphabet size. For illustrative purposes, we focus on disparate impact, leaving in the supplementary our formulation and experimental results for equalized odds and equal opportunity. Disparate impact aims for the same positive prediction ratio for each sensitive attribute $z \in \mathcal{Z}$ where $\mathcal{Z}$ is the set of possible sensitive attribute values. We use the following definition for disparate impact:

Definition 1. *(Disparate Impact)*
$P(\hat{Y} = 1|Z = z_1) = P(\hat{Y} = 1|Z = z_2)$, $\forall z_1, z_2 \in \mathcal{Z}$.

The first discriminator in FR-Train distinguishes predictions w.r.t. one sensitive group from those in the others. Disparate impact intends the sensitive attribute to be independent of the model's prediction, i.e., $I(Z; \hat{Y}) = 0$.

We explain how FR-Train can enforce the above constraint. Let $P_Z(z)$ be the distribution of Z where $z \in \mathcal{Z}$. Let $\hat{Y}|Z = z \sim P_{\hat{Y}|z}(\cdot)$ and $\hat{Y} \sim P_{\hat{Y}}(\cdot)$. Then $P_{\hat{Y}}(\cdot) = \sum_{z\in\mathcal{Z}} P_Z(z)P_{\hat{Y}|z}(\cdot)$.

The following theorem asserts that mutual information is equivalent to the following function optimization where the optimal discriminator $D_z^\star(\hat{y}) = P_{Z|\hat{Y}}(z|\hat{y})$ and $\sum_{z\in\mathcal{Z}} D_z^\star(\hat{y}) = 1$, $\forall \hat{y} \in \mathcal{Y}$.

Theorem 1. $I(Z; \hat{Y}) =$
$$\max_{D_z(\hat{y}):\sum_z D_z(\hat{y})=1,\ \forall \hat{y}} \sum_{z\in\mathcal{Z}} P_Z(z)\mathbb{E}_{P_{\hat{Y}|z}}\Big[\log D_z(\hat{Y})\Big] + H(Z).$$

While deferring the detailed proof to the supplemental materials, we provide a brief overview of the proof. As the optimization problem in the RHS is convex, we find the optimal discriminator by solving the KKT conditions. We then show that the maximum value attained by the optimal discriminator is equal to the mutual information by using

the properties of mutual information and the generalized Jensen-Shannon divergence (Lin, 1991).

What is more involved than showing the above equality is designing the right optimization problem. One needs to carefully handcraft a plausible optimization problem so that its unique solution matches the desired quantity. Here, we design the optimization problem via a 'guess-&-check' approach aided by the structural insights across the KL divergences that appear in an alternative expression of mutual information.

We now discuss how to implement the above expression. Since we do not know $P_{\hat{Y}|z}(\cdot)$ exactly, we compute the following empirical version:

$$\max_{D_z(\hat{y}):\sum_z D_z(\hat{y})=1,\ \forall \hat{y}} \sum_{z\in\mathcal{Z}} P_Z(z) \sum_{i:z^{(i)}=z} \tfrac{1}{m_z} \log D_z(\hat{y}^{(i)}) + H(Z).$$

Now for sufficiently large m, the number m_z of examples with $z^{(i)} = z$ is approximately the same as $P_Z(z)m$. Therefore, the above expression becomes:

$$\max_{D_z(\hat{y}):\sum_z D_z(\hat{y})=1,\ \forall \hat{y}} \sum_{z\in\mathcal{Z}} \sum_{i:z^{(i)}=z} \frac{1}{m} \log D_z(\hat{y}^{(i)}) + H(Z).$$

Interestingly, this formulation is exactly the same as that in the original GAN (Goodfellow et al., 2014) when $|\mathcal{Z}| = 2$. We also remark that our formulation does not require a prior knowledge on $P_Z(z)$.

We note that Adversarial Debiasing (AD) (Zhang et al., 2018a) has an additional projection term that is used to force the classifier to never decrease the discriminator's loss. However, we do not use this term in FR-Train because it worsens the training stability in our experiments.

3.2. Robustness

The robustness discriminator ensures robust training by using mutual information to distinguish examples and predictions from a clean validation set. For now, let us assume such a validation set exists (in Section 4.2, we demonstrate how to construct one). The discriminator then distinguishes the training data with predictions $\{(x^{(i)}, z^{(i)}, \hat{y}^{(i)})\}_{i=1}^{m}$ from the validation set $\{(x_{\text{val}}^{(i)}, z_{\text{val}}^{(i)}, y_{\text{val}}^{(i)})\}_{i=1}^{m_{\text{val}}}$. Intuitively, if the classifier is confused by data poisoning in the training data, then its predictions will not be consistent with the labels of the clean data, and the discriminator would be able to detect that difference. Our use of a validation set is inspired by meta learning-based robust training algorithms (Ren et al., 2018), which also defends against poisoning attacks by using the validation data loss as a meta objective. However, a key difference is that we take an adversarial learning approach, which introduces a knob that controls the emphasis of robust training. We find that this knob enables FR-Train to be more robust to the validation set size (see details in Section 4.1). In Section 3.3, we also use the robustness discriminator to further improve the fairness training using example re-weighting.

We first define $\overline{X} = VX + (1-V)X_{\text{val}}$, $\overline{Z} = VZ + (1-V)Z_{\text{val}}$, and $\overline{Y} = V\hat{Y} + (1-V)Y_{\text{val}}$. Here, note that V is an indicator random variable that denotes whether an example is generated ($V = 1$) or comes from the validation set ($V = 0$). We then want to ensure that the distribution of $(X, Z, \hat{Y})$ matches that of $(X_{\text{val}}, Z_{\text{val}}, Y_{\text{val}})$. This can be done by enforcing $I(V; \overline{X}, \overline{Z}, \overline{Y}) = 0$, i.e., the predictions on the training data are indistinguishable from the labels of the validation set. Thus we can mimic the clean dataset while expecting an indirect sanitization effect.

Analogous to the fairness discriminator, we show that mutual information is equivalent to the following function optimization where the optimal discriminator $D_v^\star(x, z, y) = P_{V|\overline{X},\overline{Z},\overline{Y}}(v|x, z, y)$ and $\sum_{v\in\mathcal{V}} D_v^\star(x, z, y) = 1,\ \forall (x, z, y) \in \mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$. The proof is similar to that of Theorem 1.

Theorem 2. $I(V; \overline{X}, \overline{Z}, \overline{Y}) =$

$$\max_{D_v(x,z,y):\sum_v D_v(x,z,y)=1,\ \forall (x,z,y)} \sum_{v\in\mathcal{V}} P_V(v)\mathbb{E}_{P_{\overline{X},\overline{Z},\overline{Y}|v}}\left[\log D_v(\overline{X}, \overline{Z}, \overline{Y})\right] + H(V).$$

3.3. Architecture

We describe the FR-Train architecture in Figure 3. For the loss function of the generator, we employ cross entropy:

$$L_1 = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)} \log \hat{y}^{(i)} - (1-y^{(i)}) \log(1-\hat{y}^{(i)}).$$

We set the loss function w.r.t. the fairness discriminator as:

$$L_2 = \max_{D(\cdot)} \sum_{z\in\mathcal{Z}} \sum_{i:z^{(i)}=z} \frac{1}{m} \log D_z(\hat{y}^{(i)}) + H(Z)$$

where $D(\cdot) := (D_1(\cdot), \ldots, D_{|\mathcal{Z}|}(\cdot))$. The condition $\sum_{z\in\mathcal{Z}} D_z^\star(\hat{Y}) = 1$ can be enforced by adding a softmax layer to the discriminator.

Finally, implementing $I(V; \overline{X}, \overline{Z}, \overline{Y})$, we set the loss function w.r.t. the robustness discriminator as:

$$L_3 = \max_{D^r(\cdot)} \sum_{i:v^{(i)}=0} \frac{1}{m} \log D^r(x_{val}^{(i)}, z_{val}^{(i)}, y_{val}^{(i)}) + \sum_{i:v^{(i)}=1} \frac{1}{m} \log(1 - D^r(x^{(i)}, z^{(i)}, \hat{y}^{(i)})) + H(V).$$

The final objective function is the weighted sum of these value functions:

$$\min_{G(\cdot)} L_1 + \lambda_1 L_2 + \lambda_2 L_3.$$

Here λ_1 and λ_2 are tuning knobs that play roles to emphasize fair and robust training, respectively.

Example Re-weighting for Fairness Training In addition to the above architecture, we also utilize the decision values $D^r(X, Z, \hat{Y})$ of the robustness discriminator as example weights to further improve the fairness training (in Figure 3, the arrow from the robustness discriminator's output to the classifier's input). In particular, the two losses L_1 and L_2 are now computed using the example weights. The intuition is that, by giving more weight to the clean examples, we can improve the accuracy-fairness tradeoff. A question is when to apply these weights. If we apply the weights too early, then $D(X, Z, \hat{Y})$ may not be accurate enough and actually harm the fairness training. Intuitively, we would like to use the discriminator's results when we know it is performing at least as well as the classifier. Hence, for a more reliable signal, we use the *relative performance* between the classifier and robustness discriminator to generate the weights. Given the classifier's loss L_c and the robustness discriminator's loss L_d, we compute the final example weights as $W = R + D(X, Z, \hat{Y}) \times (1 - R)$ where $R = \sigma(\frac{L_c}{L_d} - C)$ is a conversion of the loss ratio into a probability using the sigmoid function σ and hyperparameter C. We note that C acts as a threshold on the loss ratio.

4. Experiments

We provide experimental results for FR-Train. For the fairness measure, we use disparate impact, while leaving in the supplementary the results for equalized odds and equal opportunity. We evaluate all models on separate clean test sets. In our experiments, we use two sensitive attributes z_1 and z_2, and disparate impact is measured as the ratio $\min\{\frac{P(\hat{Y}=1|Z=z_1)}{P(\hat{Y}=1|Z=z_2)}, \frac{P(\hat{Y}=1|Z=z_2)}{P(\hat{Y}=1|Z=z_1)}\}$. We use PyTorch (Paszke et al., 2017), and all experiments are performed on a server with Intel i7-6850 CPUs. More implementation details are in the supplementary.

4.1. Synthetic Data Results

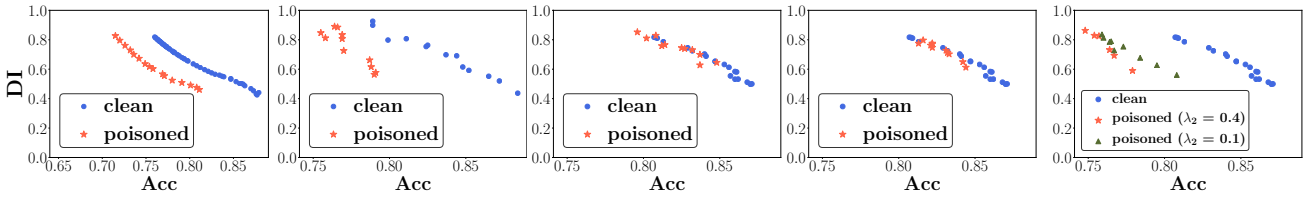
For the synthetic data, we generate 2,000 examples with two non-sensitive attributes x_1 and x_2, a sensitive attribute z, and a label y, using a method similar to the algorithm proposed by (Zafar et al., 2017). Both z and y are binary, and the (x_1, x_2) pair consists of two normal distributions: $(x_1, x_2)|y = 0 \sim \mathcal{N}([-2; -2], [10, 1; 1, 3])$ and $(x_1, x_2)|y = 1 \sim \mathcal{N}([2; 2], [5, 1; 1, 5])$. The z attribute has the Bernoulli distribution $p(z = 1) = p((x'_1, x'_2)|y = 1)/[p((x'_1, x'_2)|y = 0) + p((x'_1, x'_2)|y = 1)]$ where $(x'_1, x'_2) = (x_1 \cos(\pi/4) - x_2 \sin(\pi/4), x_1 \sin(\pi/4) + x_2 \cos(\pi/4))$. Finally for each example, the x_1 and x_2 values are sampled as per the normal distribution associated with the y. For data poisoning, we flip the labels of examples with $z = 1$ so as to maximize the accuracy performance degradation as described in Section 2, and the amount of poisoning is 10% of $\mathcal{D}_{tr}$. In the supplementary, we also perform FR-Train varying the amount of poisoning from 10% to 40%.

Table 1. Accuracy and fairness performances on the synthetic test datasets w.r.t. disparate impact (DI). Two types of methods are compared: (1) fairness methods: FC (Zafar et al., 2017), LBC (Jiang & Nachum, 2020), and AD (Zhang et al., 2018a) where "RML+" denotes the application of sanitization using RML (Ren et al., 2018) beforehand; (2) non-fairness methods: LR and RML. For FR-Train and RML, the validation set is 10% of $\mathcal{D}_{tr}$. The amount of poisoning is 10% of $\mathcal{D}_{tr}$. For each result of the poisoned data, we make a comparison with the clean data result and show the percentage increase or decrease.

Method	Clean data		Poisoned data	
	DI	Acc.	DI	Acc.
FC	.822	.806	.831 (1.1% ↑)	.760 (5.7% ↓)
LBC	.819	.760	.827 (1.0% ↑)	.715 (5.9% ↓)
AD	.807	.811	.834 (3.4% ↑)	.769 (5.2% ↓)
RML+FC	.822	.806	.802 (2.4% ↓)	.529 (34.% ↓)
RML+LBC	.819	.760	.810 (1.1% ↓)	.752 (1.1% ↓)
RML+AD	.807	.811	.808 (0.1% ↑)	.756 (6.8% ↓)
LR	.409	.885	.446 (9.1% ↑)	.819 (7.5% ↓)
RML	.471	.876	.395 (16.% ↓)	.869 (0.8% ↓)
FR-Train	**.818**	**.807**	**.827 (1.1% ↑)**	**.814 (0.9% ↑)**

Accuracy and Fairness We compare FR-Train with various baselines. First, there are the fairness methods: Fairness Constraints (Zafar et al., 2017) (FC), Label Bias Correction (Jiang & Nachum, 2020) (LBC), and Adversarial Debiasing (Zhang et al., 2018a) (AD). As described in the previous sections, FC adds a penalty term that captures the prediction differences across sensitive groups, while AD utilizes adversarial learning to achieve high fairness. LBC is an example re-weighting algorithm, which assumes the existence of true *unbiased yet unknown* labels. LBC provides theoretical guarantees that training on the resulting loss corresponds to training on the true unbiased labels, which yields a fair model. While there exist other re-weighting techniques including (Agarwal et al., 2018), we choose LBC because it performs the best in experiments (Jiang & Nachum, 2020).

Since FR-Train is to our knowledge the first method to address both fairness and robustness in model training, there is no fairness method that also performs data sanitization using a clean validation set. However, (Ren et al., 2018) is a state-of-the-art robust training method based on meta learning using a clean validation set, which we call RML. For a fair comparison, we thus extend the three fairness methods by first performing RML and then utilizing the example weights in the fairness training in a straightforward fashion. In addition, we compare with non-fairness methods: logistic regression (LR) and RML.



(a) LBC (b) AD (c) FR-Train (Val. set size = 10%) (d) FR-Train (Val. set size = 5%) (e) FR-Train (Val. set size = 0.1%)

Figure 4. Accuracy-fairness tradeoff curves. Figures (a) and (b) show that the poisoning worsens the accuracy-fairness tradeoffs of LBC (Jiang & Nachum, 2020) and AD (Zhang et al., 2018a). Figures (c) and (d) show that FR-Train maintains the tradeoffs even with a 5% validation set. When the validation set is too small (Figure (e)), FR-Train can adjust λ_2 to reduce the adverse effect on training.

Table 1 compares FR-Train with the baselines. We use a validation set that amounts to 10% of $\mathcal{D}_{tr}$. We also apply proper hyperparameters so that the disparate impacts are similar (around 0.8) across all methods, if possible. When setting λ_1 and λ_2 for FR-Train, we usually fix λ_2 to some value and then adjust λ_1 using one-round cross validation. There is no hyperparameter tuning for logistic regression and the meta learning-based robust training algorithm, as they have no knobs for adjusting fairness. The results show that for the fairness methods, data poisoning aggravates accuracy-fairness trade-offs. For example, the accuracy of FC falls by 5.7%, while the disparate impact of it remains a similar value. On the other hand, the performance for FR-Train does not degrade: disparate impact and accuracy increase by 1.1% and 0.9%, respectively. Table 1 also shows that combining the fairness methods with RML (rows 4–6) does not always yield better accuracy and fairness. In fact, using sanitization may lower the accuracy or fairness (e.g., RML+FC has an accuracy of 0.529 on poisoned data while FC has 0.760). The results suggest that removing poisoning and then bias is not that effective.

We observe how accuracy trades off with fairness on clean and poisoned datasets. The results for FC are shown in Figure 2b. For LBC, we employ the number of training as a knob to trade accuracy off fairness since LBC gradually improves fairness by repeatedly updating example weights per training. As shown in Figure 4a, the tradeoff curve shifts to the left, which demonstrates a clear tradeoff degradation. For AD, we employ the α parameter (Zhang et al., 2018a) analogous to λ_1 as a knob to trade accuracy off fairness. Figure 4b shows the tradeoff curve again shifts to the left.

Validation Set Size Figures 4c to 4e show how the validation set size affects the robustness of FR-Train. In particular, we compare the accuracy-fairness tradeoff of FR-Train on clean data and that on poisoned data while varying the size of the validation set. When running on poisoned data, we fixed $\lambda_2 = 0.4$ and varied λ_1. We see that even a 5% validation set (Figure 4d) is sufficient to maintain the accuracy and fairness obtained on the clean data. When using 0.1% (Figure 4e), the validation set is too small and has an adverse effect on the training. However, by decreasing the tuning knob λ_2 down to 0.1, we can de-emphasize robust training, thereby avoiding the adverse effect (Figure 4e, green triangles). This is in contrast to RML, which suffers from a non-negligible performance degradation for a very small validation set. See details in the supplementary.

4.2. Real Data Results

We use two real datasets: ProPublica COMPAS (Angwin et al., 2016) and AdultCensus (Kohavi, 1996), which have 7,214 and 45,222 examples, respectively. We use the same preprocessing as in IBM's AI Fairness 360 (Bellamy et al., 2018a) and use the sensitive attribute SEX for both datasets. For data poisoning, we use the same method employed on synthetic data: flipping the labels with $z = 1$ so as to maximize the accuracy performance degradation. The amount of poisoning is 10% of $\mathcal{D}_{tr}$.

While we assumed that a small yet clean validation set is available in the previous synthetic data experiments, such an assumption does not hold in practice. Thus, for real-data experiments, we consider a scenario where one first constructs a small (which amounts to 5% of $\mathcal{D}_{tr}$) validation set based on crowdsourcing, and then uses it for FR-Train. We provide details on how to construct this validation set in Section 4.5.

Summarized in Tables 2 and 3 are the fairness and accuracy performances of various training algorithms on the COMPAS and AdultCensus datasets, respectively. As in Table 1, we apply proper hyperparameters so that the disparate impacts are similar across all distinct methods, both for the clean and poisoned datasets. The results are similar to Table 1: the three fairness methods have worse disparate impact and accuracy due to data poisoning; LR and RML exhibit poor disparate impacts; and FR-Train again shows little degradation both in fairness and accuracy. Tables 2 and 3 also show that combining the fairness methods with sanitization using RML (rows 4–6) does not always yield better accuracy and fairness and may even lower them, which is consistent with the results on synthetic data. One may wonder if the fairness baselines would perform better if they are

Table 2. Accuracy and fairness performances on COMPAS test data w.r.t. disparate impact (DI) where the training data is poisoned using the label flipping attack. Two types of methods are compared: (1) fairness methods: FC, LBC, and AD where "RML+" denotes the application of sanitization using RML beforehand; (2) non-fairness methods: LR and RML. For FR-Train and RML, the validation set is 5% of $\mathcal{D}_{tr}$. The amount of poisoning is 10% of $\mathcal{D}_{tr}$. For each result of the poisoned data, we compare with the clean data result and show the percentage increase or decrease.

Method	Clean data		Poisoned data	
	DI	Acc.	DI	Acc.
FC	.777	.682	.794 (2.2% ↑)	.612 (10.% ↓)
LBC	.866	.671	.838 (2.8% ↓)	.671 (0.0% -)
AD	.846	.680	.813 (6.1% ↓)	.570 (16.% ↓)
RML+FC	.777	.682	.560 (28.% ↓)	.645 (5.4% ↓)
RML+LBC	.866	.671	.869 (0.4% ↑)	.646 (3.7% ↓)
RML+AD	.846	.680	.820 (3.1% ↓)	.573 (16.% ↓)
LR	.465	.674	.454 (5.0% ↓)	.631 (6.4% ↓)
RML	.493	.680	.575 (17.% ↑)	.646 (5.0% ↓)
FR-Train	**.838**	**.676**	**.846 (1.0% ↑)**	**.670 (0.9% ↓)**

trained on the clean validation set. In the supplementary, we show that the performances are actually worse than those in Tables 2 and 3. This is because the clean validation set is too small to be used as a stand-alone train data. Indeed, a similar observation is made in (Zhang et al., 2018b).

Table 3. Accuracy and fairness results on AdultCensus test data w.r.t. disparate impact (DI). Other settings are identical to Table 2.

Method	Clean data		Poisoned data	
	DI	Acc.	DI	Acc.
FC	.825	.826	.741 (10.% ↓)	.801 (3.0% ↓)
LBC	.825	.825	.760 (7.9% ↓)	.792 (4.0% ↓)
AD	.850	.767	.755 (11.% ↓)	.563 (27.% ↓)
RML+FC	.825	.826	.821 (0.5% ↓)	.780 (5.6% ↓)
RML+LBC	.825	.825	.762 (7.6% ↓)	.788 (4.5% ↓)
RML+AD	.850	.767	.834 (1.9% ↓)	.647 (16.% ↓)
LR	.328	.847	.189 (42.% ↓)	.819 (3.3% ↓)
RML	.327	.846	.268 (18.% ↓)	.840 (0.7% ↓)
FR-Train	**.828**	**.824**	**.847 (2.3% ↑)**	**.809 (1.8% ↓)**

Table 4 shows the confusion matrix comparison for disparate impact between FR-Train, AD, and FC with sanitization using RML, using the poisoned AdultCensus dataset. The results are reported when FR-Train, AD, and FC achieve (Acc, DI) = (0.809, 0.847), (0.647, 0.834), and (0.780, 0.821), respectively. FR-Train outperforms AD and FC in all aspects because its robustness discriminator is more effective in sanitizing poisoned data.

Table 4. Confusion matrix on poisoned AdultCensus dataset w.r.t. disparate impact. Other settings are identical to Table 2.

Method		Female		Male	
		$\hat{y}=0$	$\hat{y}=1$	$\hat{y}=0$	$\hat{y}=1$
RML+FC	$y=0$	2,990	78	4,842	71
	$y=1$	238	147	1,952	313
RML+AD	$y=0$	2,345	723	3,966	947
	$y=1$	289	96	1,792	473
FR-Train	$y=0$	2,761	307	4,730	183
	$y=1$	113	272	1,428	837

Table 5. Ablation study for FR-Train on COMPAS test data w.r.t. disparate impact (DI) where the training data is poisoned using the label flipping attack. Four methods are compared: (1) FR-Train without R ($\lambda_2 = 0$), (2) FR-Train without F ($\lambda_1 = 0$), (3) FR-Train without example re-weighting (Without RW), and (4) FR-Train. For rows 2–4, the validation set is 5% of $\mathcal{D}_{tr}$.

Method	Clean data		Poisoned data	
	DI	Acc.	DI	Acc.
Without R	.846	.678	.802 (5.2% ↓)	.580 (14.% ↓)
Without F	.482	.681	.420 (13.% ↓)	.632 (7.2% ↓)
Without RW	.832	.677	.840 (1.0% ↑)	.624 (7.8% ↓)
FR-Train	**.838**	**.676**	**.846 (1.0% ↑)**	**.670 (0.9% ↓)**

4.3. Ablation Study

In Table 5, we perform an ablation study to investigate the effect of each component of FR-Train. **(Without 'R')** When $\lambda_2 = 0$ (i.e., no robust training), disparate impact is high, but accuracy is low on the poisoned data, just like the other fairness-only methods (Table 2, rows 1–3). **(Without 'F')** On the other hand, when $\lambda_1 = 0$ (i.e., no fair training), the accuracy is high, but the disparate impact is low, just like the other non-fairness methods (Table 2, rows 7–8). **(Without Re-weighting)** Finally, when not using example re-weighting, both accuracy and disparate impact are similar to or worse than FR-Train.

In summary, *only a holistic* framework like FR-Train *can achieve both* excellent model fairness and training robustness. In comparison, other methods tailored for only one of these objectives lose either accuracy, fairness or both.

4.4. Error range of FR-Train

We investigate the error range of FR-Train. All the FR-Train experiments on the poisoned data are re-conducted with ten different random seeds to generate error ranges with mean (m) and standard deviation (s) values. The performances are reported in the form of $m \pm s/2$ in Table 6. On the synthetic and AdultCensus datasets, the lowest performances

(i.e., $m - s/2$) of FR-Train are still better than the second-best performances in Tables 1 and 3, respectively. For the COMPAS dataset, the lowest performance of FR-Train is slightly worse than those of the LBC-related algorithms, which can be explained by the fact that the LBC algorithms were not affected much by the poisoning in the first place.

Table 6. Error range of FR-Train on the poisoned datasets w.r.t. disparate impact (DI). The poisoned settings are identical to the previous experiments.

Dataset	Poisoned data	
	DI	Acc.
Synthetic	0.795 ± 0.019	0.805 ± 0.008
COMPAS	0.827 ± 0.027	0.653 ± 0.005
AdultCensus	0.871 ± 0.034	0.796 ± 0.006

4.5. Constructing a Clean Validation Set

We now demonstrate how to construct a clean validation set using crowdsourcing. We construct validation sets for the COMPAS and AdultCensus datasets using Amazon Mechanical Turk (AMT). Although these datasets have labels, we assume that they are not available to use as clean data. We also release the datasets as a community resource (see the supplementary for the description and data) and believe our construction can be generalized to other datasets. While crowdsourcing is not the only way to construct a clean validation set, it is sufficient for our purposes.

We design the AMT task for each dataset by asking a worker to classify each example. For the AdultCensus dataset, a worker looks at various attributes of a person and predicts if a person has an income of at least \$50K. Instead of a yes/no answer, the answer must be on a scale of 1 to 4, which reflects the worker's opinion more accurately. The COMPAS dataset has a similar setting where the only difference is that the workers need to predict if a criminal will reoffend in two years. Each task displays about 30 questions where we pay 3 cents per answer. For quality control, each task also contains quizzes to educate the workers, and some questions are used to evaluate the performance of the workers. After collecting answers, we filter out poor performers, take the average of at most a fixed number of N responses per question, and compare with the threshold 2.5 to produce the final labels. The number of answers per question can be fewer than N if inaccurate workers are filtered out. We used workers of all demographics in the US, Canada, and UK. While this majority voting approach already works well in our experiments, one could additionally apply various quality control techniques like peer-reviewing that are known to further reduce bias (Karger et al., 2011).

The important questions are how accurate the crowdsourced labels are and whether the constructed validation set results in high accuracy and fairness for FR-Train. Table 7 shows the crowdsourced labels accuracies when N increases from 1 to 11. Even for the highest accuracies, the predictions are not perfect because the workers are looking at limited information (i.e., only the features) without any other context. To see if the workers can do better, we also train logistic regression models on ground truth labels and show their accuracies on test data as upperbounds. As a result, the accuracies are comparable when $N = 11$ for both the COMPAS and AdultCensus datasets. We thus use this setting for all experiments. Table 8 shows how useful our constructed validation set is compared to using a "perfect" validation set of the same size made of ground truth labels. For both datasets, using a ground truth validation set results in slightly higher, but comparable disparate impacts while obtaining near-identical accuracies, justifying the use of crowdsourced validation sets for FR-Train.

Table 7. Accuracy comparison of the crowdsourced labels (N: number of answers averaged per example) and predictions of a logistic regression model trained on ground truth labels.

Dataset	Crowdsourcing			Trained Model
	$N = 1$	$N = 5$	$N = 11$	
COMPAS	0.609	0.656	0.667	0.659
AdultCensus	0.645	0.721	0.743	0.804

Table 8. Accuracy and fairness of FR-Train when using crowdsourced labels versus ground truth labels for the validation set. The training data is poisoned as in Tables 2 and 3.

Dataset	Validation set	DI	Acc.
COMPAS	Crowdsourcing	0.846	0.670
	Ground truth	0.899	0.674
AdultCensus	Crowdsourcing	0.847	0.809
	Ground truth	0.864	0.809

5. Related Work

Model Fairness The notion of discrimination has many definitions and usually comes from certain social goals that one wants to guarantee. As a result, many fairness measures have been proposed (Verma & Rubin, 2018). While we focus on group fairness, which ensures similar statistics between two sensitive groups, an interesting future work is to consider individual fairness (Dwork et al., 2012), which guarantees similar prediction results across nearby examples. Recently, there has also been a surge of research on unfairness mitigation techniques (Bellamy et al., 2018b). Depending on where a fix occurs, there are mainly three approaches: (1) *pre*-processing techniques (Kamiran & Calders, 2011; du Pin Calmon et al., 2017; Zemel et al., 2013; Feldman et al., 2015) that fix the training data; (2) *in*-processing techniques (Zafar et al., 2017; Jiang & Nachum, 2020; Zhang

et al., 2018a; Kamishima et al., 2012; Cotter et al., 2019; 2018; Agarwal et al., 2018) that address the issue during model training; and (3) *post*-processing techniques (Hardt et al., 2016; Pleiss et al., 2017; Kamiran et al., 2012; Chzhen et al., 2019) that manipulate predictions while maintaining the model. Among the three, the in-processing techniques have the advantages that one can work with any data and that there is more control on model training (Venkatasubramanian, 2019).

Although not our immediate focus, there are other noteworthy directions in fairness research. Causality-based fairness (Kilbertus et al., 2017; Kusner et al., 2017; Zhang & Bareinboim, 2018; Nabi & Shpitser, 2018; Khademi et al., 2019; Khademi & Honavar, 2020) suggests how to understand the causal relationship between attributes to overcome the limitations of non-causal approaches. Just as non-causal fairness can be captured by mutual information, we suspect there may be a connection between causal fairness and directed information. Another important approach (Hashimoto et al., 2018) is based on distributionally robust optimization (DRO) (Sinha et al., 2017), which focuses on when the sensitive attribute z is unknown. The DRO-based fairness approach ensures fair results by equalizing risks over all distributions without the knowledge of z, but it does not directly minimize the fairness metrics such as disparate impact and equalized odds. In comparison, FR-Train assumes full knowledge of z and utilizes it to directly minimize the fairness metrics.

As we demonstrate in Section 2, the existing fairness techniques are not tailored for robust training, so they are vulnerable to data poisoning attacks. In comparison, FR-Train addresses both model fairness and robust training within the same model training process because they are closely related and affected by the same training data.

Robust Training There is a heavy literature on how to make the model training robust against noisy or even adversarial data (Natarajan et al., 2013; Biggio et al., 2011; Frénay & Verleysen, 2014; Kurakin et al., 2017). A major challenge is that there can be a wide range of data poisoning attacks that keep on evolving. While sanitizing the training data before model training is an option, defending against all possible attacks seems fundamentally infeasible as demonstrated by (Koh et al., 2018). A more recent trend is to develop general defense algorithms for any attack *during model training* using meta learning (Veit et al., 2017; Li et al., 2017; Xiao et al., 2015; Hendrycks et al., 2018). Our FR-Train framework is inspired by robustness training with meta learning (Ren et al., 2018), but employs a GAN-based model to support fair and robust training without using meta learning. In particular, the design of FR-Train's robustness discriminator is based on mutual-information-based theoretical insights (Section 3.2). Another line of research is defending against adversarial attacks during *test* time (Biggio et al., 2013; Goodfellow et al., 2015; Wong & Kolter, 2018). In comparison, our focus is on defending against data poisoning on the *training* data.

6. Conclusion

We proposed FR-Train, which is a holistic framework for trustworthy AI by performing both unfairness mitigation and robust training. Our key contribution is providing interpretation of an adversarial learning approach using mutual information and proposing a novel GAN architecture that enjoys the *synergistic effect* of combining two approaches: (1) employing a fairness discriminator that distinguishes predictions w.r.t. one sensitive group from others and (2) employing a robustness discriminator that distinguishes training data with predictions from a clean validation set and is also used to further improve the fairness training through example re-weighting. In addition, we demonstrated how a clean validation set can be constructed using crowdsourcing and released two new datasets built from Amazon Mechanical Turk as a community resource. In our experiments, we showed that existing fairness methods are vulnerable to data poisoning, even when combined with data sanitization. In comparison, FR-Train is robust to the poisoning and can be adjusted to maintain reasonable accuracy and fairness even if the validation set is too small or unavailable.

Acknowledgements

Yuji Roh and Steven E. Whang were supported by a Google AI Focused Research Award and by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921). This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-19-1-4050.

References

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. M. A reductions approach to fair classification. In *ICML*, pp. 60–69, 2018.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias: There's software used across the country to predict future criminals. And its biased against blacks., 2016.

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

Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J. T., Saha, D., Sattigeri, P., Singh, M., Varshney, K. R., and Zhang, Y. AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *CoRR*, abs/1810.01943, 2018b.

Biggio, B., Nelson, B., and Laskov, P. Support vector machines under adversarial label noise. In *ACML*, pp. 97–112, 2011.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *ECML PKDD*, pp. 387–402, 2013.

Chouldechova, A. and Roth, A. The frontiers of fairness in machine learning. *CoRR*, abs/1810.08810, 2018.

Chzhen, E., Denis, C., Hebiri, M., Oneto, L., and Pontil, M. Leveraging labeled and unlabeled data for consistent fair binary classification. In *NeurIPS*, pp. 12760–12770. 2019.

Cotter, A., Gupta, M. R., Jiang, H., Srebro, N., Sridharan, K., Wang, S., Woodworth, B. E., and You, S. Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. *CoRR*, abs/1807.00028, 2018.

Cotter, A., Jiang, H., and Sridharan, K. Two-player games for efficient non-convex constrained optimization. In *ALT*, pp. 300–332, 2019.

du Pin Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R. Optimized pre-processing for discrimination prevention. In *NeurIPS*, pp. 3995–4004, 2017.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *ITCS*, pp. 214–226, 2012. ISBN 978-1-4503-1115-1.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *KDD*, pp. 259–268, 2015.

Frénay, B. and Verleysen, M. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Netw. Learning Syst.*, 25(5):845–869, 2014.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *NeurIPS*, pp. 2672–2680, 2014.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *NeurIPS*, pp. 3315–3323, 2016.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. Fairness without demographics in repeated loss minimization. In *ICML*, pp. 1929–1938, 2018.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, pp. 10477–10486, 2018.

IBM. Trusting ai. https://www.research.ibm.com/artificial-intelligence/trusted-ai/, 2020.

Jiang, H. and Nachum, O. Identifying and correcting label bias in machine learning. In *AISTATS*, pp. 702–712, 2020.

Kamiran, F. and Calders, T. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.*, 33(1):1–33, 2011.

Kamiran, F., Karim, A., and Zhang, X. Decision theory for discrimination-aware classification. In *ICDM*, pp. 924–929, 2012.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. Fairness-aware classifier with prejudice remover regularizer. In *ECML PKDD*, pp. 35–50, 2012.

Karger, D. R., Oh, S., and Shah, D. Iterative learning for reliable crowdsourcing systems. In *NIPS*, pp. 1953–1961. 2011.

Karpathy, A. Software 2.0. https://medium.com/@karpathy/software-2-0-a64152b37c35, 2017.

Khademi, A. and Honavar, V. G. Algorithmic bias in recidivism prediction: A causal perspective (student abstract). In *AAAI*, pp. 13839–13840, 2020.

Khademi, A., Lee, S., Foley, D., and Honavar, V. Fairness in algorithmic decision making: An excursion through the lens of causality. In *WWW*, pp. 2907–2914, 2019.

Kilbertus, N., Rojas-Carulla, M., Parascandolo, G., Hardt, M., Janzing, D., and Schölkopf, B. Avoiding discrimination through causal reasoning. In *NeurIPS*, pp. 656–666, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koh, P. W., Steinhardt, J., and Liang, P. Stronger data poisoning attacks break data sanitization defenses. *CoRR*, abs/1811.00741, 2018.

Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD*, pp. 202–207, 1996.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. In *ICLR*, 2017.

Kusner, M. J., Loftus, J., Russell, C., and Silva, R. Counterfactual fairness. In *NeurIPS*, pp. 4066–4076. 2017.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L. Learning from noisy labels with distillation. In *ICCV*, pp. 1928–1936, 2017.

Lin, J. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

Nabi, R. and Shpitser, I. Fair inference on outcomes. *AAAI*, pp. 1931–1940, 2018.

Natarajan, N., Dhillon, I. S., Ravikumar, P., and Tewari, A. Learning with noisy labels. In *NeurIPS*, pp. 1196–1204, 2013.

Noy, N., Burgess, M., and Brickley, D. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *28th Web Conference (WebConf 2019)*, 2019.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Paudice, A., Muñoz-González, L., and Lupu, E. C. Label sanitization against label flipping poisoning attacks. In *ECML PKDD*, pp. 5–15, 2018.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J. M., and Weinberger, K. Q. On fairness and calibration. In *NeurIPS*, pp. 5684–5693, 2017.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *ICML*, pp. 4331–4340, 2018.

Sinha, A., Namkoong, H., and Duchi, J. C. Certifying some distributional robustness with principled adversarial training. In *ICLR*, 2017.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. J. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pp. 6575–6583, 2017.

Venkatasubramanian, S. Algorithmic fairness: Measures, methods and representations. In *PODS*, pp. 481, 2019.

Verma, S. and Rubin, J. Fairness definitions explained. In *FairWare@ICSE*, pp. 1–7, 2018.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, pp. 5283–5292, 2018.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *CVPR*, pp. 2691–2699, 2015.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, pp. 962–970, 2017.

Zemel, R. S., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *ICML*, pp. 325–333, 2013.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *AIES*, pp. 335–340, 2018a.

Zhang, J. and Bareinboim, E. Fairness in decision-making - the causal explanation formula. In *AAAI*, 2018.

Zhang, X., Zhu, X., and Wright, S. J. Training set debugging using trusted items. In *AAAI*, 2018b.