Fairness-aware Agnostic Federated Learning

Wei Du * Depeng Xu * Xintao Wu * Hanghang Tong †

Abstract

Federated learning is an emerging framework that builds centralized machine learning models with training data distributed across multiple devices. Most of the previous works about federated learning focus on the privacy protection and communication cost reduction. However, how to achieve fairness in federated learning is under-explored and challenging especially when testing data distribution is different from training distribution or even unknown. Introducing simple fairness constraints on the centralized model cannot achieve model fairness on unknown testing data. In this paper, we develop a fairness-aware agnostic federated learning framework (AgnosticFair) to deal with the challenge of unknown testing distribution. We use kernel reweighing functions to assign a reweighing value on each training sample in both loss function and fairness constraint. Therefore, the centralized model built from AgnosticFair can achieve high accuracy and fairness guarantee on unknown testing data. Moreover, the built model can be directly applied to local sites as it guarantees fairness on local data distributions. To our best knowledge, this is the first work to achieve fairness in federated learning. Experimental results on two real datasets demonstrate the effectiveness in terms of both utility and fairness under data shift scenarios.

1 Introduction

In traditional machine learning, the training data is usually stored in a central server. The central server needs to first collect the data from different sources and combines these data together to facilitate the learning. The rapid development of the various machine learning or deep learning models benefits significantly from the large-scale training data. However, the above training approach also raises data privacy concerns because the raw data needs to be uploaded to the server, which can lead to the possible sensitive information leakage.

To alleviate the above issues caused by centralized learning, federated learning is proposed as an attractive framework to handle complex data [7, 13], as it enables a new implementation of distributed deep learning over a large number of clients. Compared to the traditional centralized learning which collects all local data samples and builds the model at a central server, federated learning trains local models on local data samples and local clients exchange parameters to generate a global model. Although tremendous research has been done on the federated learning, how to achieve fairness in federated learning is under-explored. Fairness receives increasing attentions in machine learning. Previous research demonstrates that many machine learning models are often biased or unfair against some protected groups especially when they were trained on biased data. How to achieve fairness in federated learning, i.e., each client makes fair decisions for people irrespective of their protected attributes such as gender and race, is more urgent because the training data used in federated learning is often geo-distributed among various groups.

One challenge of achieving fairness in federated learning is due to the statistical challenge of the unknown testing data distribution. In this paper, we consider a supervised task with features $\mathbf{X}$ and labels Y and assume $P(y|\mathbf{x})$ is common across all clients. Our goal is to train a single global model that learns $P(y|\mathbf{x})$. The single global model can then be shared to clients or provided to new clients with no training data. However, in federated learning, the distributions of training data at different clients are often different, i.e., $P_i(\mathbf{x}, y) \neq P_j(\mathbf{x}, y)$. Data shift clearly exists between the training distribution of local data used to build the model and the unknown testing distribution. This data shift causes significant challenges for developing fair federated learning algorithms because the learned model may have poor performance on testing data and the learned model with fairness constraint on the training data cannot guarantee the fairness on the testing data. In addition, the global model may achieve fairness for some clients while fail for other clients due to different distribution shift of different clients.

It is beneficial to build a learning model that is robust against the possible unknown testing distribution in terms of both utility and fairness. The authors in [8] propose agnostic federated learning to deal with the unknown testing data distribution. They model the testing distribution as a mixture of the client data distributions and the mixture weight of one client is deviated from the proportion of its local data in the whole training data. They define the agnostic empirical loss with mixture weights and present a fast stochastic optimization algorithm. However, in their formulation, the model is optimized for the performance of the single worst client and does not take the fairness into consideration.

In this work, we propose fairness-aware agnostic fed-

*University of Arkansas, {wd005, depengxu, xintaowu}@uark.edu
†University of Illinois, htong@illinois.edu

erated learning (AgnosticFair) to deal with the challenge when the testing data distribution is unknown. We formulate AgnosticFair as a two-player adversarial minimax game between the learner and the adversary. The adversary aims to generate any possible unknown testing data distribution to maximize the classifier loss. We assign an individual reweighing value on each training sample and incorporate reweighing value in both agnostic loss function and agnostic fairness constraint. As a result, the global model learned in the minimax game achieves both high accuracy and fairness guarantee on unknown testing data. Moreover, each client can simply deploy the global model on its local site as the learned global model guarantees fairness on any local data. We conduct extensive experiments on two public datasets and compare our approach with several baselines. Evaluation results demonstrate the effectiveness of our proposed AgnosticFair in terms of accuracy and fairness under data shift scenarios. The main contributions of this work are summarized as follows. First, To the best of our knowledge, our research is the first one that formulates the problem of fairness-aware federated learning under the data distribution shift among the clients. Second, We propose to use kernel function parametrization in loss function and fairness constraints so that they are both agnostic to the data distribution shift. Third, We develop an efficient approach to optimize the agnostic loss function under the agnostic fairness constraints between the server and clients. During the optimization process, only parameters and coefficients are needed to exchange between clients and the server without disclosing any raw data. Fourth, We conduct extensive experiments to demonstrate that our approach can achieve fair prediction under distribution shift while maintaining high accuracy.

2 Related Work

Fairness-aware learning has received lots of attentions in the past few years. The developed methods can be categorized as pre-processing the training data to remove sensitive information about the protected attribute, post-processing the classifier to achieve fair prediction, and incorporating the fairness constraint into the classification model during the optimization process. Our work falls into the last category. However, most of the previous works on fairness learning are in the centralized setting and do not consider distribution shift.

There exist extensive research works on federated learning aiming to solve various challenges such as the limited bandwidth, privacy leakage and non-IID data among different clients since the pioneer work [7]. Regarding the non-IID distribution among different clients, some notable research works include [5, 6, 14]. To address the known distribution shift between training and testing data in the centralized learning, researchers have developed various reweighing methods, e.g., density ratio estimation is used to reweigh training data to represent known testing data [10], a sample reweighing approach is designed to correct the distribution shift [4], and reweighing values are used in loss optimization [11]. To address the unknown distribution shift (i.e., testing data are not available), the authors in [1] formulate a two-player minimax game by optimizing the worst case of expected loss and introduce using kernel reweighing functions to address unknown distribution shift.

Different from all previous works, we first study the fairness-aware classification in federated learning. The most relevant works are [3], which considers the fairness in an online learning setting across different groups, and [8], which only mention concept of fairness in federated learning but do not propose any algorithms to achieve it. In contrast, our work firstly formulates the problem of fairness-aware federated learning under the data distribution shift, proposes a general framework by using a reweighing function on individual sample to solve the data distribution shift issue among different clients, and develops an effective algorithm that uses kernel function parametrization in both loss function and fairness constraints to achieve fairness guarantee and good accuracy under unknown testing data.

3 Fair Agnostic Federated Learning

3.1 Problem Formulation We first define the following notations used throughout the paper. Suppose there exist p local users $u_1, u_2, \ldots, u_p$ in the federated learning setting and each user is associated with a dataset $\mathcal{D}_k = \{\mathbf{X}, Y\}, k \in [p]$. The kth user contains n_k samples $t_1, t_2, \ldots, t_{n_k}$ and each sample is denoted as $t_i : \{\mathbf{x}_i, y_i\}, i \in [n_k]$. The total number of data samples is defined as $n = \sum_{k=1}^{p} n_k$. Let $\mathbf{X} \in \mathcal{X}$ be the input space and $Y \in \mathcal{Y}$ be the output space. We consider the binary classification that $\mathcal{Y} = \{0, 1\}$. S is one attribute of $\mathbf{X}$ and used to denote the sensitive attribute, where $S = 0$ represents the sensitive group and $S = 1$ denotes the non-sensitive group. The global model f predicts the label as $\hat{y} = f(\mathbf{x})$. The goal of the federated learning is to collaboratively train a machine learning model f by these p users. In this training process, each user only shares the model parameters and does not expose its raw data to others.

Following the standard federated learning framework that aims to minimize the empirical risk and learns the parameters $\mathbf{w} \in \mathcal{W}$, we write the objective function subject to the fairness constraint:

$$\begin{aligned} &\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) = \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k} l_i(f_k(\mathbf{x}_i;\mathbf{w}), y_i) \\ &\text{subject to} \quad g(\mathbf{x};\mathbf{w}) \le \epsilon, \end{aligned} \tag{3.1}$$

where f_k is the classifier of u_k, $g(\mathbf{x};\mathbf{w}) \le \epsilon$ is the fairness constraint, and $l_i(f_k(\mathbf{x}_i;\mathbf{w}), y_i)$ is the loss of sample t_i in u_k. The weight of each sample in Equation 3.1 from these p users is uniform. The underlying assumption of the standard federated learning framework is that the testing

data distribution is the same as the training data (union of data samples from p users) distribution. However, this assumption is rather restrictive and will lead to the following possible drawbacks. First, the performance of the trained model will be degraded if the distribution of the training data and that of the testing data do not coincide. Second, the fairness achieved on the training data does not guarantee the fairness on the testing data. Note that we only consider binary classification problem in this paper and the approach can be extended to multi-classification.

3.2 Agnostic Loss Function It is usually considered that the distribution shift exists between the training data $P_{tr}(\mathbf{X})$ and the testing data $P_{te}(\mathbf{X})$, whereas the conditional distribution $P(Y|\mathbf{X})$ indicating the prediction remains the same. To correct the distribution shift between $P_{tr}(\mathbf{X})$ and $P_{te}(\mathbf{X})$, a widely used approach is to reweigh the training samples in the learning process so that the learned model f can reflect the testing data distribution. We write the objective function in Equation 3.1 with the reweighed training samples as the following:

$$\min_{\mathbf{w}\in\mathcal{W}} L(\mathbf{w}) = \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}\theta(\mathbf{x}_i^k)l(f_k(\mathbf{x}_i^k;\mathbf{w}),y_i^k), \tag{3.2}$$

where $\theta(\mathbf{x})$ is the reweighing function to correct the distribution shift from $P_{tr}(\mathbf{X})$ to $P_{te}(\mathbf{X})$. There exist several methods to estimate the reweighing value $\theta(\mathbf{x})$ if the (unlabelled) testing data is given [9]. For example, applying density ratio estimation can compute the reweighing value $\theta(\mathbf{x}) = P_{te}(\mathbf{x})/P_{tr}(\mathbf{x})$ for each example, then the reweighing value can represent the possible testing distribution in real scenarios.

However, we cannot properly estimate the reweighing values if we do not have available testing data. Therefore, it is necessary to extend the above framework by building a classifier which is favorable to any unknown testing distribution. We define the agnostic loss over any unknown testing data as the following:

$$\min_{\mathbf{w}\in\mathcal{W}}\max_{\theta\in\Theta} L(\mathbf{w},\theta) = \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}\theta(\mathbf{x}_i^k)l(f_k(\mathbf{x}_i^k;\mathbf{w}),y_i^k). \tag{3.3}$$

Θ represents the set of unknown testing data distribution produced by the adversary. The formulation can be considered as a two-player adversarial game such that the adversary in Equation 3.3 tries to select a reweighing function $\theta \in \Theta$ to maximize the loss of the objective, whereas the learner tries to find parameters $\mathbf{w} \in \mathcal{W}$ to minimize the worst case loss over the unknown testing data distribution produced by the adversary.

The proposed framework by Equation 3.3 enjoys several advantages. First, under the independent and identically distributed (IID) data settings, the minimization of the robust reweighed loss is equivalent and dual to the empirical risk minimization (objective function in Equation 3.1) [1]. It indicates that the optimization of Equation 3.3 will not cause performance degradation when no distribution shift exists. Second, the optimal $\mathbf{w} \in \mathcal{W}$ is minimized for the worst case loss and the performance of the global model is robust with any unknown testing data.

3.3 Kernel Function Parametrization The reweighing function $\theta \in \Theta$ can be chosen based on the prior knowledge or the application scenario. A reweighing function on individual data sample across clients usually corrects the distribution shift more accurately. We rewrite the agnostic loss in Equation 3.3 as:

$$\min_{\mathbf{w}\in\mathcal{W}}\max_{\boldsymbol{\alpha}\in\mathbf{R}^+} L(\mathbf{w},\boldsymbol{\alpha}) = \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k)l(f_k(\mathbf{x}_i^k;\mathbf{w}),y_i^k) \tag{3.4}$$

$$\text{subject to } \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) = 1.$$

$\theta_{\boldsymbol{\alpha}}(\mathbf{x})$ is the reweighing function that is linearly parametrized as the following:

$$\theta_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{m=1}^{M}\alpha_m K_m(\mathbf{x}), 0 \le \alpha_m \le B \tag{3.5}$$

where $K_m(\mathbf{x})$ is a basis function, M is the number of basis functions, and $\boldsymbol{\alpha}$ contains the mixing coefficients $\alpha_1, \alpha_2, \cdots, \alpha_M$. The sum-to-one constraint of the reweighing function $\theta_{\boldsymbol{\alpha}}(\mathbf{x})$ ensures that it can properly model the unknown distribution shift from the training data to the testing data. The coefficient α_m is non-negative and bounded by $B \in \mathbf{R}^+$, which constrains the value of $\theta_{\boldsymbol{\alpha}}(\mathbf{x})$ and controls the capacity of the adversary. The linearly parametrized reweighing function $\theta_{\boldsymbol{\alpha}}(\mathbf{x})$ has two advantages. First, the linear form allows us to choose multiple basis functions to capture many different possible uncertainties of the unknown testing data. Second, the optimization with linear form can be more easily solved by linear programming or convex programming tool.

In fact, there are many options to choose the form of basis functions. In this paper, we choose the Gaussian kernel

$$K_m(\mathbf{x}_i^k) = \exp(-\parallel \mathbf{b}_m - \mathbf{x}_i^k \parallel^2 /2\sigma^2) \tag{3.6}$$

with the basis $\mathbf{b}_m$ and the kernel width σ. The basis $\mathbf{b}_m$ can be chosen based on some prior knowledge, for example, we can set $\mathbf{b}_m$ as some possible centers of the testing data according to the hypothesis. σ is the width of the kernel. As the smaller variation of $\parallel \mathbf{b}_m - \mathbf{x} \parallel^2$ will cause larger value change of the kernel function and smaller σ indicates more possible testing distributions that the adversary can

generate [11]. Or the basis $\mathbf{b}_m$ could be an indicator function $\mathbb{1}_{[.]}$ which represents groups from different clients, ages, or domains. The value generated by each kernel function can be seen as a conditional probability $P(\mathbf{x}|m)$ of observing $\mathbf{x}$ given the class m in a mixture model. The mixing coefficients $\boldsymbol{\alpha} \in \mathcal{A}$ are usually bounded in the non-negative Euclidean space.

The agnostic federated learning framework proposed by [8] is a special case of our framework. As being said that the basis function could be an indicator function $\mathbb{1}_{[\cdot]}$ representing a group. In [8], it models the testing data distribution as an unknown mixture of p clients where each group is assigned with a uniform weight. The unknown testing data distribution in [8] can be constructed under our framework as the following:

$$\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) = \lambda_k \frac{n_k}{n}, 1 \leq i \leq n_k, 1 \leq k \leq p \tag{3.7}$$

where $\frac{n_k}{n}$ is the uniform weight of each data $\mathbf{x}_i^k$ before reweighing. More specifically, for each client u_k, the agnostic federated learning [8] assigns the same value λ_k to reweigh each data in $\mathcal{D}_k$. However, assigning reweighing value at the client level is insufficient to model the unknown distribution shift due to the following two reasons. First, the data from the same client also has diversity and needs different reweighing values. Second, different clients can have similar data and these similar data should be assigned with similar reweighing values. In our framework, we assign the reweighing value at the individual data level, which is more capable of modeling the unknown testing data distribution.

3.4 Agnostic Fairness Constraint The fairness constraint $g(\mathbf{x}; \mathbf{w})$ in standard federated learning (Equation 3.1) assigns uniform weight for each sample. When it comes to the unknown testing data, the fairness achieved by Equation 3.1 may not guarantee the fairness on unknown testing data. As being said, the adversary tries to produce a set of possible unknown testing distributions. It encourages us to construct the agnostic fairness constraint based on the unknown testing distributions by the adversary. Then, the optimization of the objective function is subject to the fairness constraint based on the unknown testing distribution.

There exist several notions for fairness $g(\mathbf{x}; \mathbf{w})$ and demographic parity is the most widely used notion in fairness machine learning [2]. It requires the prediction result by the model to be independent of the sensitive attribute S. Demographic parity is usually quantified by the risk difference, which measures the difference between positive predictions on the sensitive and the non-sensitive groups. The risk difference of a classifier f is expressed as:

$$RD(f) = |P(\hat{Y} = 1|S = 1) - P(\hat{Y} = 1|S = 0)|, \tag{3.8}$$

where $\hat{Y}$ is the predicted value of f. For each client, we define $\mathcal{D}_{ij}^k = \{\mathbf{x}|\hat{Y} = i, S = j\}$ where $i, j \in \{0, 1\}$. For notation convenience, we define $\mathcal{D}_{\cdot j}^k = \{\mathbf{x}|S = j\}$ where $j \in \{0, 1\}$ and $\cdot$ represents $\{0, 1\}$. Then we can write the expression for $RD(f)$ with uniform weight on training data as the following:

$$|\frac{\sum_{k=1}^p \sum \mathbb{1}_{\mathbf{x}_i^k \in \mathcal{D}_{11}^k}}{\sum_{k=1}^p \sum \mathbb{1}_{\mathbf{x}_i^k \in \mathcal{D}_{\cdot 1}^k}} - \frac{\sum_{k=1}^p \sum \mathbb{1}_{\mathbf{x}_i^k \in \mathcal{D}_{10}^k}}{\sum_{k=1}^p \sum \mathbb{1}_{\mathbf{x}_i^k \in \mathcal{D}_{\cdot 0}^k}}| \leq \epsilon, \tag{3.9}$$

where $\mathbb{1}_{[.]}$ is an indicator function and $\epsilon \in [0, 1]$ is a threshold for the fairness constraint. However, Equation 3.9 is constructed based on the training data and cannot preserve fairness on the unknown testing data. Hence, we use the same reweighing function to construct the agnostic fairness constraint as the following:

$$|\frac{\sum_{k=1}^p \sum_{\mathbf{x}_i^k \in \mathcal{D}_{11}^k} \theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k)}{\sum_{k=1}^p \sum_{\mathbf{x}_i^k \in \mathcal{D}_{\cdot 1}^k} \theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k)} - \frac{\sum_{k=1}^p \sum_{\mathbf{x}_i^k \in \mathcal{D}_{10}^k} \theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k)}{\sum_{k=1}^p \sum_{\mathbf{x}_i^k \in \mathcal{D}_{\cdot 0}^k} \theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k)}| \leq \epsilon. \tag{3.10}$$

Then our proposed fairness-aware agnostic federated learning (AgnosticFair) is the combination of Equation 3.4 and Equation 3.10. The fairness constraint in Equation 3.10 is constructed based on the unknown testing distribution, so when it comes to the unknown testing data, the trained classifier can still preserve the fairness. Another benefit is that even though the distributions of the local clients and the server side do not coincide, the classifier can still guarantee the fairness on each local client due to the agnostic fairness constraint.

The optimal solution of Equation 3.4 under the fairness constraint by Equation 3.10 is computationally intractable to obtain because the fairness constraint contains the indicator function. An alternative fairness constraint is defined as the covariance between the sensitive attribute and the signed distance from the non-sensitive attribute vector to the decision boundary. It has been proved that the decision boundary fairness is a concept of risk difference [12]. We write this alternative definition $C_{\mathcal{D}}(\mathbf{x}; \mathbf{w})$ as:

$$C_{\mathcal{D}}(\mathbf{x}; \mathbf{w}) = \frac{1}{n} \sum_{k=1}^{p} \sum_{i=1}^{n_k} (s_{\mathbf{x}_i^k} - \bar{s}) d_{\mathbf{w}(\mathbf{x}_i^k)}, \tag{3.11}$$

where $s_{\mathbf{x}_i^k}$ is the value of the sensitive attribute of the sample $\mathbf{x}_i^k$, $d_{\mathbf{w}(\mathbf{x}_i^k)}$ is the distance to the decision boundary of the classifier f, $\bar{s}$ is the mean value of the sensitive attribute over $\mathcal{D}$ that is $\frac{\sum_{k=1}^p \sum_{i=1}^{n_k} s_{\mathbf{x}_i^k}}{n}$. To achieve fair classification, it is required that $|C_{\mathcal{D}}(\mathbf{x}; \mathbf{w})| \leq \tau$ where $\tau \in \mathbf{R}^+$. Incorporating the reweighing values into the fairness constraint gives:

$$C_{\mathcal{D}}(\boldsymbol{\alpha}; \mathbf{x}; \mathbf{w}) = \frac{1}{n} \sum_{k=1}^{p} \sum_{i=1}^{n_k} (s_{\mathbf{x}_i^k} - \bar{s}) \theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) d_{\mathbf{w}(\mathbf{x}_i^k)}. \tag{3.12}$$

3.5 Solving Fairness-aware Agnostic Federated Fairness Learning Now we are ready to formulate our agnostic federated learning under the decision boundary fairness constraint as:

(3.13)

$$\min_{\mathbf{w}\in\mathcal{W}} \max_{\boldsymbol{\alpha}\in\mathbf{R}^+} \quad L(\mathbf{w},\boldsymbol{\alpha}) = \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) l(f_k(\mathbf{x}_i^k;\mathbf{w}), y_i^k)$$

$$\text{subject to} \quad \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) = 1, 0 \le \alpha_m \le B$$

$$|\frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}(s_{\mathbf{x}_i^k} - \bar{s})\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) d_{\mathbf{w}(\mathbf{x}_i^k)}| \le \tau.$$

The optimization of Equation 3.13 includes two sets of parameters, $\boldsymbol{\alpha}$ and $\mathbf{w}$. The minimax expression encourages us to alternatively optimize $\boldsymbol{\alpha}$ and $\mathbf{w}$ in an iterative way. The client and server will collaboratively optimize $\mathbf{w}$ and $\boldsymbol{\alpha}$ to solve the minimax problem. The general pipeline is that the client optimizes $\mathbf{w}$ with fixed $\boldsymbol{\alpha}$, while the server optimizes $\boldsymbol{\alpha}$ with fixed $\mathbf{w}$. One challenge is how both the server and the clients conduct optimization iteratively through sharing parameters or intermediate results (rather than raw data), as required in federated learning.

Throughout this paper, we use ϕ to denote the coefficient vector of $\mathbf{w}$ (with fixed α), ψ to denote the coefficient vector of α (with fixed $\mathbf{w}$). We use the subscript L, θ and C denote the loss function, equality constraint, and inequality constraint, and further add the subscript k for the coefficients from client u_k. Moreover, we use the superscript t to express the coefficients at step t during the optimization. For example, $\psi_{C,k}^t$ denotes the coefficient vector of α (with fixed $\mathbf{w}$) in the inequality constraint formula for client u_k at step t.

The objective loss $L(\mathbf{w},\boldsymbol{\alpha})$ (abbreviated as L) shown in Equation 3.13 can be written as a function of $\boldsymbol{w}$ with corresponding coefficients ϕ_L when $\boldsymbol{\alpha}$ is fixed. More importantly, the second summation over samples in client u_k, $\sum_{i=1}^{n_k}\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) l(f_k(\mathbf{x}_i^k;\mathbf{w}), y_i^k)$, can be similarly expressed as a function $\boldsymbol{w}$ with corresponding coefficients $\phi_{L,k}$. We can easily see $\phi_L = \sum_{k=1}^{p}\phi_{L,k}$ and hence each client can simply send $\phi_{L,k}$ (rather than any raw data) to the server. Similarly, when $\boldsymbol{w}$ is fixed, $L(\mathbf{w},\boldsymbol{\alpha})$ can be written as a function of $\boldsymbol{\alpha}$ with corresponding coefficients ψ_L, the second summation over samples in client u_k has coefficients $\psi_{L,k}$, and $\psi_L = \sum_{k=1}^{p}\psi_{L,k}$.

Similarly, the equality constraint for the reweighing functions (abbreviated as θ) and the inequality constraint for the decision boundary fairness (abbreviated as C) in Equation 3.13 can be expressed as functions with corresponding coefficients. We note that the equality constraint only involves variable $\boldsymbol{\alpha}$ and does not have coefficient vector ϕ_θ. The relationships of the coefficients are shown in the following equation.

$$\begin{aligned}\phi_L^t &= \sum_{k=1}^{p}\phi_{L,k}^t, \phi_C^t = \sum_{k=1}^{p}\phi_{C,k}^t \\ \psi_L^t &= \sum_{k=1}^{p}\psi_{L,k}^t, \psi_\theta^t = \sum_{k=1}^{p}\psi_{\theta,k}^t, \psi_C^t = \sum_{k=1}^{p}\psi_{C,k}^t.\end{aligned} \tag{3.14}$$

Algorithm 1 AgnosticFair: Fairness-aware Agnostic Federated Learning

Require:
$\mathcal{D}_k$ from client u_k, $k = 1, \cdots, p$;
Training steps T;
Initial parameters $\mathbf{w}^0$ and $\boldsymbol{\alpha}^0$;
Ensure:
Global model parameter vector $\mathbf{w}$;
Initialize parameters $\mathbf{w}^0$ and $\boldsymbol{\alpha}^0$ for all clients;
$t = 0$;
While $t \le T$ **do**
  Client Side:
  for $k = 1 : p$ **do**
    Client k receives averaged $\bar{\mathbf{w}}^t$, $\boldsymbol{\alpha}^t$ and ϕ_C^t;
    Client k computes optimal $\mathbf{w}_k^{t+1}$ using Equation 3.16 and uploads to server;
    Client k computes $\phi_{C,k}^t, \psi_{L,k}^{t+1}, \psi_{\theta,k}^{t+1}, \psi_{C,k}^{t+1}$ and uploads to server;
  Server Side:
    Server aggregates $\psi_L^{t+1} = \sum_{k=1}^{p}\psi_{L,k}^{t+1}$, $\psi_\theta^{t+1} = \sum_{k=1}^{p}\psi_{\theta,k}^{t+1}, \psi_C^{t+1} = \sum_{k=1}^{p}\psi_{C,k}^{t+1}$;
    Server computes optimal $\boldsymbol{\alpha}^{t+1}$ using Equation 3.17;
    Server aggregates $\phi_C^{t+1} = \sum_{k=1}^{p}\phi_{C,k}^{t+1}$ and averages $\bar{\mathbf{w}}^{t+1} = \frac{1}{p}\sum_{k=1}^{p}\mathbf{w}_k^{t+1}$;
    Server sends back $\bar{\mathbf{w}}^{t+1}$, $\boldsymbol{\alpha}^{t+1}$ and ϕ_C^{t+1};
  $t = t + 1$;
return $\bar{\mathbf{w}}^T$

In the following, we present details about the optimization process between the server and client. We show those key parameters and coefficients exchanged between the client and the server as well as their calculations in Figure 1.

Client side: In standard federated learning, each client computes $\mathbf{w}$ based on its local data and exchanges the updated $\mathbf{w}$ with other clients via the server. Here we also follow this standard approach that each client computes the optimal values of $\mathbf{w}$ locally using the fixed $\boldsymbol{\alpha}$ received from the server. We can decompose the part of Equation 3.13

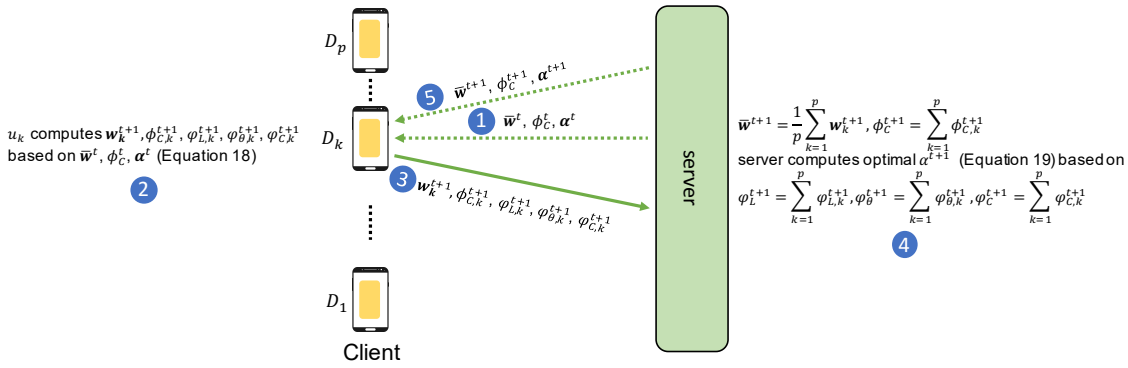

Figure 1: The interaction between the client and server in federated learning. The client side optimizes its local $\mathbf{w}$ and server side optimizes the $\boldsymbol{\alpha}$.

related to $\mathbf{w}$ as the following:

(3.15)
$$\min_{\mathbf{w}\in\mathcal{W}} \quad L(\mathbf{w}) = \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k} \theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) l(f_k(\mathbf{x}_i^k;\mathbf{w}), y_i^k)$$
$$\text{subject to} \quad |\frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}(s_{\mathbf{x}_i^k} - \bar{s})\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) d_{\mathbf{w}(\mathbf{x}_i^k)}| \le \tau.$$

It can be seen that given the fixed $\boldsymbol{\alpha}$, the optimization of $\mathbf{w}$ is only subject to the inequality constraint. The optimization of Equation 3.15 depends on the choice of loss function and the learning model. For example, the loss function of linear regression is convex and the inequality constraint of $\mathbf{w}$ is also linear so the convex programming tool can be used to solve it. However, the loss function over $\mathbf{w}$ of many other machine learning models (e.g., deep learning models) is not convex. Hence, it is challenging to optimize the non-convex function subject to the constraint. We observe that the inequality constraint in Equation 3.15 is used to guarantee the fairness of the updated $\mathbf{w}$ during the optimization. Instead of optimizing non-convex loss function subject to the constraints, we can transform the inequality constraint to a penalty term on the loss function.

We choose the square term for fairness inequality constraint as a penalty and rewrite the loss function of $\mathbf{w}$ for client k as the following:

(3.16)
$$\min_{\mathbf{w}\in\mathcal{W}} L(\mathbf{w}) = \frac{1}{n_k}\sum_{i=1}^{n_k}[\theta(\mathbf{x}_i^k) l(f_k(\mathbf{x}_i^k;\mathbf{w}), y_i^k)] + \lambda(\frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}(s_{\mathbf{x}_i^k} - \bar{s})\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) d_{\mathbf{w}(\mathbf{x}_i^k)} - \tau)^2$$

where λ is a hyperparameter controlling the trade-off between the classification accuracy and the fairness. Equation 3.16 includes two terms. The first term is the loss of each client based on its own data while the second term is the global fairness constraint.

Suppose client u_k receives the average parameters $\bar{\mathbf{w}}^t$ and $\boldsymbol{\alpha}^t$ from the server at the tth step, it can compute $\phi_{L,k}^t$ using $\boldsymbol{\alpha}^t$ and local data $\mathcal{D}_k$. For the second term computation, it needs to receive $\phi_C^t = \sum_{k=1}^{p} \phi_{C,k}^t$ from the server, where each client can compute $\phi_{C,k}^t$ independently using local data $\mathcal{D}_k$. In this process, the raw data of each client is not exposed, which fulfills the requirement of federated learning. Given $\bar{\mathbf{w}}^t$, $\phi_{L,k}^t$ and ϕ_C^t, client u_k obtains the complete form of Equation 3.16 and can compute optimal $\mathbf{w}_k^{t+1}$ based on $\mathcal{D}_k$. Based on $\mathbf{w}_k^{t+1}$ and fixed $\boldsymbol{\alpha}^t$, it can compute $\psi_{L,k}^{t+1}, \psi_{\theta,k}^{t+1}, \psi_{C,k}^{t+1}$, and $\phi_{C,k}^{t+1}$ and upload them to the server.

Server side: The optimization of $\boldsymbol{\alpha}$ is subject to both equality and inequality constraints, which is expressed as:

(3.17)
$$\max_{\boldsymbol{\alpha}\in\mathbf{R}^+} \quad L(\boldsymbol{\alpha}) = \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k} \theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) l(f_k(\mathbf{x}_i^k;\mathbf{w}), y_i^k)$$
$$\text{subject to} \quad \frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) = 1, 0 \le \alpha_m \le B$$
$$|\frac{1}{n}\sum_{k=1}^{p}\sum_{i=1}^{n_k}(s_{\mathbf{x}_i^k} - \bar{s})\theta_{\boldsymbol{\alpha}}(\mathbf{x}_i^k) d_{\mathbf{w}(\mathbf{x}_i^k)}| \le \tau.$$

Given fixed $\mathbf{w}$, $\theta_{\boldsymbol{\alpha}}$ is a linear function subject to linear equality and inequality constraints. The server aggregates coefficient vector $\psi_L^{t+1}, \psi_\theta^{t+1}, \psi_C^{t+1}$ of $\boldsymbol{\alpha}$ (Equation 3.14) to obtain the complete form of Equation 3.17. The server then uses linear programming tool to obtain the optimal values $\boldsymbol{\alpha}^{t+1}$ and sends back to each client. In addition, the server also averages parameters $\bar{\mathbf{w}}^{t+1} = \frac{1}{p}\sum_{k=1}^{p} \mathbf{w}_k^{t+1}$, aggregates $\phi_C^{t+1} = \sum_{k=1}^{p} \phi_{C,k}^{t+1}$, and sends them back to each client for next round iteration.

We also present the pseudo code of our proposed fairness-aware agnostic federated learning (AgnosticFair) in Algorithm 1. It can be seen that each client optimizes $\mathbf{w}$ at

the local side and the server optimizes $\boldsymbol{\alpha}$. The final classifier with fair prediction is achieved through the iterative optimization process.

4 Experiments

4.1 Experimental Setup Datasets. We evaluate our proposed approach AgnosticFair on two widely used datasets, Adult dataset and Dutch dataset. Adult dataset collects the personal information from different people including age, education level, race, gender, and so forth. The prediction task is to determine whether the income of a person is over 50K or not. Dutch dataset collects personal information of the inhabitants in Netherlands and the task is also to classify the individual into high income or low income. For both datasets, we set "gender" as the sensitive attribute. For non-sensitive attributes, we apply one-hot encoding to convert the categorical attributes into vectors and normalize numerical attributes to the range within $[0, 1]$. After preprocessing, Adult dataset consists of 45222 data samples and each data sample has 40 features, whereas Dutch dataset consists of 60420 data samples and each data sample has 35 features.

To create the distribution shift scenarios from the training set to the testing set, we artificially split each dataset as the following. For Adult, the training set $\mathcal{D}^{tr}$ contains 80% data of people working in private company and 20% data of people working in other organizations, and the testing set $\mathcal{D}^{te}$ contains the rest of the data. Hence, $\mathcal{D}^{tr}$ of Adult is dominated by data of people working in private company, while $\mathcal{D}^{te}$ of Adult is dominated by data of people working in other organizations. We consider 2 local clients in our experiment u_1 and u_2. u_1 only contains data of people working in private company from $\mathcal{D}^{tr}$ while u_2 only contains data of people working in other groups. Similarly for Dutch, the training set $\mathcal{D}^{tr}$ contains 80% data of people who are married with children and 40% data of people from other groups, and $\mathcal{D}^{te}$ contains the rest of the data. We also consider 2 local clients, u_1 contains data of people who are married with children while u_2 contains of people in other groups.

Hyperparameters. In our experiment, we use Gaussian kernel in Equation 3.6 as the reweighing function to construct the unknown testing data distribution. The upper bound B for $\boldsymbol{\alpha}$ is set as 5 and σ is chosen to be 1. In fact, the upper bound of B is rarely reached in practical optimization so that it will not limit the power of the adversary too much. The basis of the Gaussian kernel is chosen from training data and the number of kernels is set as 200. The threshold τ in Equation 3.13 is set as a constant 0.05 and λ in Equation 3.16 is set as 2.

Baselines. In our experiment, we use the logistic regression model to evaluate our proposed algorithm. We compare the performance of our proposed AgnosticFair with the following baselines: (a) standard federated learning (FL) without fairness constraint; (b) standard federated learning with fairness constraint (FairFL); (c) agnostic federated learning [8] that assigns the reweighing value at the client level. To conduct meaningful comparison between our model with baselines, we introduce several variations: AgnosticFair-a that optimizes agnostic loss (Equation 3.4) without any fairness constraint; AgnosticFair-b that optimizes agnostic loss subject to unweighted fairness constraint.

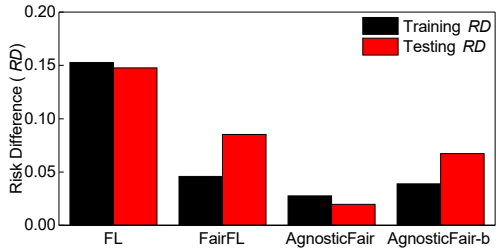

Figure 2: Model Fairness under data distribution shift (Adult)

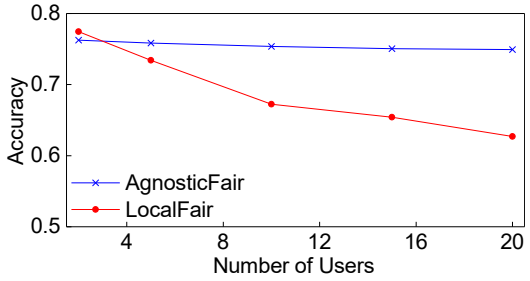

Figure 3: The accuracy of the global model with different number of local clients (Adult).

Metrics. We evaluate our proposed framework and baselines based on utility and fairness. We use accuracy to measure the utility and risk difference (RD) to measure the fairness. A fair classifier usually has $RD(f) \leq 0.05$. We run all experiments 20 times and report the average results.

4.2 Comparison under Unknown Data Shift In our framework, the reweighing function is designed to improve the performance of the classifier if the data distributions of the training and the testing set do not coincide. The use of the reweighing function in the fair constraint can also achieve fairness guarantee under unknown testing data.

Accuracy. We report the experimental results in Table 1 that demonstrate the accuracy improvement by our framework. We summarize several interesting points regarding accuracy as the following. The testing accuracy of AgnosticFair-a is 0.8264 on Adult and 0.8162 on Dutch, while the accuracy of FL is 0.7998 on Adult and 0.7951 on Dutch. More specifically, AgnosticFair-a outperforms FL by 0.0296 on Adult and by 0.0211 on Dutch. It demonstrates that the agnostic loss function in Equation 3.3 improves the performance of

Table 1: Model performance under data distribution shift (Adult and Dutch) Acc: accuracy

Methods	Adult Dataset			Dutch Dataset		
	Training Acc	Testing Acc	Testing RD	Training Acc	Testing Acc	Testing RD
FL	0.7500	0.7998	0.1477	0.8133	0.7951	0.1945
AgnosticFair	0.7413	0.7626	0.0196	0.7478	0.7205	0.0371
AgnosticFair-a	0.7820	0.8294	0.1306	0.8259	0.8162	0.2154
FairFL	0.7537	0.7534	0.0852	0.6899	0.7011	0.0961
[8]	0.7761	0.7774	0.1150	0.8170	0.7738	0.1238

Table 2: Local fairness and global fairness under data distribution shift (Adult)

Methods	u_1 Testing RD	u_2 Testing RD	Global Testing RD	Global Testing Accuracy
AgnosticFair	0.0208	0.0177	0.0196	0.7626
AgnosticFair-b	0.0450	0.0795	0.0673	0.7885

the model when it comes to the distribution change.

Compared to [8], AgnosticFair-a enjoys higher accuracy. As discussed before, assigning a reweighing value at the client level optimizes the client with the worst loss which will reduce its generalization ability on the unknown testing data. AgnosticFair-a assigns a reweighing value for each data sample across all clients and puts more weights on difficult data samples with higher loss. In fact, the difficult data samples can come from any client and are taken into the consideration during the training. Hence, the generalization ability in the testing stage of AgnosticFair-a is increased.

Fairness. In this experiment, we will show that our proposed AgnosticFair can achieve fairness guarantee under unknown data distribution shift. In Table 1, the two columns "Testing RD" show the risk difference values of our AgnosticFair and four baselines over the testing data of both Adult and Dutch. We also draw a plot in Figure 2 to show achieving fairness on the training data by baselines cannot guarantee fairness on testing data whereas our AgnosticFair can achieve the guarantee. First, experimental results show that both AgnosticFair and AgnosticFair-b can achieve fairness on the training data, but only AgnosticFair can guarantee the fairness on the testing data. AgnosticRegFair-b uses the agnostic loss function, but its fairness constraint is unweighted. It can be concluded that using the agnostic loss function only cannot guarantee the fairness when it comes to the unknown testing data. Second, FL achieves high accuracy but cannot achieve the fairness. The RD of the FL is 0.1477, whereas a fair learning model usually requires RD to be less than 0.05. The fairness constraint of FairFL does not consider data distribution shift. The results show that FairFL achieves fairness on the training data but fails on the testing data.

Federated Learning with Different Number of Clients. Our proposed AgnosticFair can also achieve fairness for local clients when the distribution shift exists between the local clients and the global server. In fact, the distribution shift from the global training data to the local client data is a special case of the unknown testing data distribution. We use the same data split setting and report the result of local fairness and global fairness in Table 2. It can be seen that AgnosticFair-b cannot guarantee the fairness on the unknown testing data because it fails on u_2 ($RD = 0.0795$). Due to the agnostic fairness constraint of AgnosticFair, it can achieve fairness for both local clients. To demonstrate the stability of our proposed AgnosticFair, we show its accuracy under different number of clients in Figure 3. We use the same data split setting and distribute the data evenly to each client without overlap. The accuracy of the global classifier is recorded when fairness is achieved on all clients. It can be seen that the performance of AgnosticFair is independent of the number of clients. For comparison, we also use another straightforward approach LocalFair that achieves fairness for each local client by adding a local fairness constraint based on its own data. It can guarantee the fairness for local client, however, the drawback is to add a local fairness constraint for each client, which will reduce the utility of the global model if more clients are included. Figure 3 also shows the accuracy curve of LocalFair, we can see that its performance degrades significantly with the increasing number of clients.

4.3 Comparison under IID Setting In our last experiment, we also test the performance of our model under the IID data setting. In this experiment, we randomly split two datasets, Adult and Dutch. For each dataset, we use 80% of the data as the training set and the rest 20% as the testing set. The number of local clients is set to be 2 and the training data is evenly distributed to each local client. Table 3 shows the experimental results.

First, AgnosticFair-a achieves same level of performance with FL if no data distribution shift exists. For Adult, the testing accuracy of FL (AgnosticFair-a) is 0.8130 (0.8111). For Dutch, the testing accuracy of FL (AgnosticFair-a) is 0.8096 (0.8089). This result also echoes the theoretical statement in [1]: under IID data, the mini-

Table 3: Model performance of IID data(Adult and Dutch) Acc: accuracy

Methods	Adult Dataset			Dutch Dataset		
	Training Acc	Testing Acc	Testing RD	Training Acc	Testing Acc	Testing RD
FL	0.8129	0.8130	0.1490	0.8116	0.8096	0.1698
AgnosticFair	0.7938	0.7749	0.0299	0.7338	0.7322	0.0270
AgnosticFair-a	0.8083	0.8111	0.1515	0.8135	0.8089	0.1526
FairFL	0.7731	0.7723	0.0235	0.7564	0.7346	0.0325
[8]	0.7774	0.7785	0.1484	0.7925	0.7892	0.1547

mization of the robust reweighed loss (Equation 3.2 in our framework) is equivalent and dual to the empirical risk minimization (Equation 3.1 in our framework).

Second, our AgnosticFair-a also achieves higher accuracy than [8] under the IID setting. More specially, the testing accuracy is 0.7785 (0.7892) for Adult (Dutch), which is still lower than that of FL. As aforementioned, [8] assigns a different weight for each client. The optimization process as stated in their work will improve the worst loss of the individual client, whereas the global generalization on the testing data will be weakened. Finally, our AgnosticFair achieves fairness guarantees while preserve good accuracy.

5 Conclusions and Future Work

In this paper, we have proposed a fairness-aware agnostic federated learning framework to deal with unknown testing data distributions. We apply kernel reweighing functions to parametrize the loss function and fairness constraint. Hence our framework can achieve both good model accuracy and fairness on unknown testing data. We conducted a series of experiments on two datasets and experimental results demonstrated three benefits of the trained centralized model by our fairness-aware agnostic federated learning. First, it can improve the prediction accuracy under the distribution shift from the training data to the testing data. Second, it can guarantee fairness on the unknown testing data. Third, it can guarantee the fairness of each local client. In our future work, we will extend our framework to cover other commonly used fairness notations. e.g., equalized odds and equalized opportunity [2], and incorporate surrogate functions in agnostic fair constraints of our framework to address the challenge of the indicator function used in fairness notations. We will also study kernel function parametrization with different basis functions. Our proposed framework can also be adapted to the centralized fairness-aware learning where the training and testing data differ. Moreover, the proposed framework can also be applied in the fair transfer learning where distribution shift usually exists between the source domain and target domain.

Acknowledgement

This work was supported in part by NSF 1920920, 1937010, 1946391, and 1939725.

References

[1] P. D. GRÜNWALD, A. P. DAWID, ET AL., *Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory*, The Annals of Statistics, (2004).

[2] M. HARDT, E. PRICE, N. SREBRO, ET AL., *Equality of opportunity in supervised learning*, in NeurIPS, 2016.

[3] T. B. HASHIMOTO, M. SRIVASTAVA, H. NAMKOONG, AND P. LIANG, *Fairness without demographics in repeated loss minimization*, in ICML, 2018.

[4] J. HUANG, A. GRETTON, K. BORGWARDT, B. SCHÖLKOPF, AND A. J. SMOLA, *Correcting sample selection bias by unlabeled data*, in NIPS, 2007.

[5] X. LI, K. HUANG, W. YANG, S. WANG, AND Z. ZHANG, *On the convergence of fedavg on non-iid data*, in ICLR, 2020.

[6] Y. MANSOUR, M. MOHRI, J. RO, AND A. T. SURESH, *Three approaches for personalization with applications to federated learning*, arXiv preprint arXiv:2002.10619, (2020).

[7] H. B. MCMAHAN, E. MOORE, D. RAMAGE, S. HAMPSON, ET AL., *Communication-efficient learning of deep networks from decentralized data*, in AISTATS, 2016.

[8] M. MOHRI, G. SIVEK, AND A. T. SURESH, *Agnostic federated learning*, in ICML, 2019.

[9] J. QUIONERO-CANDELA, M. SUGIYAMA, A. SCHWAIGHOFER, AND N. D. LAWRENCE, *Dataset shift in machine learning*, The MIT Press, 2009.

[10] H. SHIMODAIRA, *Improving predictive inference under covariate shift by weighting the log-likelihood function*, Journal of statistical planning and inference, (2000).

[11] J. WEN, C.-N. YU, AND R. GREINER, *Robust learning under uncertain test distributions: Relating covariate shift to model misspecification.*, in ICML, 2014.

[12] Y. WU, L. ZHANG, AND X. WU, *On convexity and bounds of fairness-aware classification*, in ACM WWW, 2019.

[13] Q. YANG, Y. LIU, T. CHEN, AND Y. TONG, *Federated machine learning: Concept and applications*, ACM TIST, (2019).

[14] Y. ZHAO, M. LI, L. LAI, N. SUDA, D. CIVIN, AND V. CHANDRA, *Federated learning with non-iid data*, arXiv preprint arXiv:1806.00582, (2018).