Training Fair Models in Federated Learning without Data Privacy Infringement

Xin Che  
McMaster University  
Hamilton, Canada  
chex5@mcmaster.ca

Jingdi Hu  
East China University of Science and Technology, Shanghai, China  
22013928@mail.ecust.edu.cn

Zirui Zhou  
Huawei Technologies Canada  
Burnaby, Canada  
zirui.zhou@huawei.com

Yong Zhang  
Huawei Technologies Canada  
Burnaby, Canada  
yong.zhang3@huawei.com

Lingyang Chu*  
McMaster University  
Hamilton, Canada  
chul9@mcmaster.ca

Abstract—Training fair machine learning models becomes more and more important. As many powerful models are trained by collaboration among multiple parties, each holding some sensitive data, it is natural to explore the feasibility of training fair models in federated learning so that the fairness of trained models, the data privacy of clients, and the collaboration between clients can be fully respected simultaneously. However, the task of training fair models in federated learning is challenging, since it is far from trivial to estimate the fairness of a model without knowing the private data of the participating parties, which is often constrained by privacy requirements in federated learning. In this paper, we first propose a federated estimation method to accurately estimate the fairness of a model without infringing the data privacy of any party. Then, we use the fairness estimation to formulate a novel problem of training fair models in federated learning. We develop FedFair, a well-designed federated learning framework, which can successfully train a fair model with high performance without data privacy infringement. Our extensive experiments on three real-world data sets demonstrate the excellent fair model training performance of our method.

Index Terms—federated learning, model fairness, data privacy, difference of generalized equal opportunities.

I. INTRODUCTION

Fairness and collaboration are among the top priorities in machine learning and AI applications. As accurate machine learning models are deployed in more and more important applications, enhancing and ensuring fairness in such models becomes critical for AI for social good [1].

At the same time, in order to build accurate machine learning models for sophisticated applications, such as finance and medicare, many parties, such as financial institutions and medical care organizations, have to collaborate and contribute their own private data [2], [3]. A key to successful collaborations is to fully protect the privacy of every party. However, it is a grand challenge to build fair machine learning models while preserving the data privacy of all parties in their collaboration.

There are many recent breakthroughs in enhancing fairness in machine learning models [1]. As discussed in Section II, the existing fairness enhancing methods (please see [1] for a survey) are mainly designed with an assumption of a unified available training data set, and thus cannot easily address the need for federated learning, such as distributed collaboration and privacy-preservation. The classic federated learning methods [3]–[7] do not consider the fairness of machine learning models at all. The collaborative fairness methods [8]–[14] focus on the fairness of the collaboration relationship among participating parties in the federated learning process, which is substantially different from the goal of training fair models.

Recently, there are some initiative fairness-aware federated learning methods [11], [15]–[19] conducted simultaneously with our work. These studies focus on improving the fairness of a trained model with respect to various notions of model fairness measurements that are substantially different from ours.

In this paper, we tackle the challenging problem of training fair models in federated learning. We systematically model the problem and develop a comprehensive solution that theoretically guarantees both fairness and privacy preservation requirements. Our key idea is to carefully design the fairness constraint based on a general fairness measurement named difference of generalized equal opportunity (DGEO) [20], so that the fairness constraint can be enforced in the whole learning process in a federated manner. In other words, the fairness constraint is implemented in a fully collaborative and privacy-preserving way. Moreover, the problem of training fair models in federated learning presents a challenging optimization problem to maximize accuracy and ensure fairness. To solve the problem we develop an elegant solution based on alternating gradient projection [21] in optimization.

We make the following contributions. First, we propose a federated estimation method to accurately estimate the fairness of a model without infringing the data privacy of any party. Second, when the data sets of all participating parties are identically and independently distributed (IID), we prove that the federated estimation method is more accurate than locally estimating the fairness of a model on individual parties and...
thus leads to a superior fair model training performance in federated learning. Third, by incorporating the model fairness estimation as a fairness constraint with a widely adopted federated loss function [4], we formulate our novel FedFair problem. We also develop an effective federated learning framework to tackle this problem without infringing the data privacy of any party. Last, we conduct extensive experiments on three real-world data sets to demonstrate that our method achieves the best performance in both cases when the data sets of all participating parties are IID and are not IID.

II. RELATED WORKS

There is a rich body of literature on federated learning. Please see [2] and [3] for excellent surveys. Here, we focus on connecting fair models and federated learning.

The classical federated learning methods [3]–[7] are mostly designed to protect privacy instead of training fair models. A straightforward extension to training fair models is to apply fairness constraints locally on each participating party. However, as discussed later in Section IV-C, since the local fairness on each party does not provide any guarantee of global fairness on all parties, this leads to an inferior fairness performance due to the inaccurate model fairness measures computed locally on each participating party.

The collaborative fairness methods [8]–[14] focus on balancing the rewards paid to the participating parties based on their contributions to the federated learning process. A typical line of works [8]–[10] combine incentive schemes with game theory to determine the payoff received by each party commensurate according to their contributions to the training process. Using a similar idea, Lysy et al. [13] and Sim et al. [14] achieve a better collaborative fairness performance by directly using model accuracy as the reward for the parties. Some other works [11], [12] advocate egalitarian equity in collaboration by optimizing the performance of the party with the worst performance.

The collaborative fairness task is substantially different from our study, because our task focuses on enhancing the fairness of a machine learning model, but the collaborative fairness task focuses on the fairness of contribution valuation among participating parties in the federated learning process.

A few recent works [11], [15]–[19] simultaneous to our work attempt to develop federated learning frameworks to train fair models with respect to different notions of model fairness. AgnosticFair [15] and FedFB [19] attempt to train models that are fair with respect to demographic parity [22]–[24]. FairFL [16] improves model fairness by reducing the difference of F1-scores between two groups of data instances. FLBGL [18] employs a fairness constraint named bounded group loss, which focuses on upper bounding the loss of each group rather than the loss difference between any two groups. FedMinMax [17] uses MinMax fairness to minimize the expected risk of the worst performing demographic group.

Unlike the above approaches, our work focuses on training fair models based on a different fairness measure named difference of generalized equal opportunities (DGEO) [20].

III. PROBLEM FORMULATION

In this section, we first review the fairness constraint based on a general fairness measure named difference of generalized equal opportunities (DGEO) [20]. Then, we formulate the problem of federated fair model training.

A. The DGEO Constraint

Denote by $(x, s, y)$ a triple of random variables drawn from an unknown distribution $D$, where $x \in \mathbb{R}^d$ is a $d$-dimensional vector of features, $s \in \{a, b\}$ is the membership of $x$ between a pair of pre-defined protected groups (e.g., $a$ for “female” and $b$ for “male”), and $y \in \{1, \ldots, C\}$ is the class label of $x$. A sample of $(x, s, y)$ is called a data instance (instance for short).

Denote by $f_0 : \mathbb{R}^d \rightarrow \mathbb{R}$ a model parameterized by a set of parameters $\theta$. For a loss function $\ell(f_0(x), y) \in \mathbb{R}$, the DGEO [20] of $f_0$ with respect to the instances contained in the protected groups $\{a, b\}$ and belonging to a protected class $c \in \{1, \ldots, C\}$ is defined as follows.

Definition 1. Let $L^{a,c}(\theta) = \mathbb{E}[\ell(f_0(x), y) | s = a, y = c]$ be the expected conditional loss of a model $f_0$ on the samples in a protected group $s$ with a protected class label $c$. The difference of generalized equal opportunity (DGEO) of the model $f_0$ with respect to the class $c$ is the absolute difference between $L^{a,c}(\theta)$ and $L^{b,c}(\theta)$, that is, $|L^{a,c}(\theta) - L^{b,c}(\theta)|$.

The above DGEO is a continuous function to measure the fairness of two protected groups. A smaller DGEO means that $f_0$ commits more similar errors on the two protected groups of instances contained in a class $c$, which further indicates $f_0$ is fairer with respect to the instances in class $c$ [20]. On the opposite, a larger DGEO means that $f_0$ is more unfair.

Donini et al. [20] propose the notion of DGEO constraint. For a given threshold $\varepsilon \geq 0$ indicating the maximum unfairness we can tolerate, a model $f_0$ is said to be $\varepsilon$-fair if it satisfies

$$|L^{a,c}(f_0) - L^{b,c}(f_0)| \leq \varepsilon. \tag{1}$$

B. Federated Fair Model Training

Now we formulate the task of training fair models in federated learning (federated fair model training for short). The task allows a group of parties called clients to collaborate through a server to train a fair and accurate model without infringing the data privacy of any client.

Denote by $U_1, \ldots, U_N$ a group of $N$ clients, by $B_1, \ldots, B_N$ the private data sets owned by the clients, respectively, by $m_i$ the number of data instances in $B_i$, and by $D$ the unknown data distribution of $\bigcup_{i=1}^N B_i$, the union of the data of all clients. We define the federated fair model training task as follows.

Definition 2 (Federated Fair Model Training Task). Given a pair of protected groups $\{a, b\}$, a protected class $c \in \{1, \ldots, C\}$, and a threshold $\varepsilon \geq 0$ indicating the maximum unfairness that can be tolerated, the federated fair model training task is to develop a method that enables a group of clients $U_1, \ldots, U_N$ to train a model $f_0$ in collaboration, such that
1) The data instances in the private data set of each client, as well as any information about the distribution of the private data set, are not exposed to any other clients or any third party; and
2) $f_\theta$ is $c$-fair for the instances in the class $c$.

The first requirement is exactly the data-privacy-preserving requirement in conventional federated learning [3]–[7]. Here, we consider an honest but curious setting [25] for the data privacy of clients. Being honest means that every client and the server strictly implements the proposed federated learning algorithm without performing any malicious operation to disrupt the learning process. Being curious means that the clients and the server are allowed to actively collect information from all the contents (i.e., model parameters, gradients, etc.) they legally receive during the federated learning process.

The second requirement is about the fairness of the collaboratively trained model with respect to input data instances. Please note that, in this paper, we are not concerned about the fair valuation of contributions from clients, which is a completely different topic orthogonal to ours.

The above federated fair model training task can be formulated as a constrained optimization problem as follows.

$$\begin{align}
\min_{\theta} & \quad L(\theta) \\
\text{s.t.} & \quad |L^{a,c}(\theta) - L^{b,c}(\theta)| \leq \epsilon,
\end{align}$$

(2a)

(2b)

where $L(\theta) = \mathbb{E}[\ell(f_\theta(x), y)]$ is the expected loss of $f_\theta$ over the unknown distribution $\mathcal{D}$, and the DGEO constraint in Equation (2b) requires the trained model $f_\theta$ to be $c$-fair.

It is difficult to find a solution to Equation (2), because we cannot compute the exact values of $L(\theta)$ and $L^{a,c}(\theta) - L^{b,c}(\theta)$ without knowing $D$. To find a practical solution, we need to estimate the values of $L(\theta)$ and $L^{a,c}(\theta) - L^{b,c}(\theta)$ while preserving the data privacy of all the clients.

IV. FedFAIR: A PRACTICAL SOLUTION

In this section, we develop a practical solution to the federated fair model training task. We start with an analysis of the problem. Then, we present a baseline method that estimates $L^{a,c}(\theta) - L^{b,c}(\theta)$ locally on each client. Last, we develop our practical solution that estimates $L^{a,c}(\theta) - L^{b,c}(\theta)$ in a federated manner and discuss its advantage over the baseline method.

A. Problem Analysis

To develop a practical solution to the federated fair model training problem, let us first analyze the opportunities and challenges in estimating $L(\theta)$ and $L^{a,c}(\theta) - L^{b,c}(\theta)$.

Similar to many classical federated learning methods [4]–[7], we can compute a federated loss that estimates $L(\theta)$ by

$$\sum_{i=1}^{N} \frac{m_i}{m_{\text{total}}} \hat{L}_i(\theta),$$

(3)

where $m_{\text{total}} = \sum_{i=1}^{N} m_i$ is the total number of data instances owned by all clients, and $\hat{L}_i(\theta) = \frac{1}{m_i} \sum_{(x,s,y) \in B_i} \ell(f_\theta(x), y)$ is the empirical loss of $f_\theta$ on the private data set $B_i$.

Computing the federated loss does not infringe the data privacy of any client, because each empirical loss $\hat{L}_i(\theta)$ is privately computed by the corresponding client $U_i$, and all empirical losses are sent to the server to compute the federated loss. For every client, the server only knows the empirical loss of the client and the number of instances in the private data set. In other words, the client does not expose to the server any instance or any data distribution information.

The real challenge is to estimate $L^{a,c}(\theta) - L^{b,c}(\theta)$ without infringing the data privacy of any client. Without a carefully designed method to estimate $L^{a,c}(\theta) - L^{b,c}(\theta)$, a client may expose to the server some distribution information about the sensitive attribute value or the target class.

In the rest of this section, we first introduce a baseline method to estimate $L^{a,c}(\theta) - L^{b,c}(\theta)$ and discuss the limitations of this method. Then, we present our federated estimation method to estimate $L^{a,c}(\theta) - L^{b,c}(\theta)$ more accurately. Last, we develop an optimization method to find a good solution to the federated fair model training task.

B. A Local Estimation Approach

To estimate $L^{a,c}(\theta) - L^{b,c}(\theta)$ while keeping the data privacy of all clients, a baseline method is to let every client $U_i$ use its private data set $B_i$ to compute its local estimation by $\hat{L}_i^{a,c}(\theta) - \hat{L}_i^{b,c}(\theta)$, where $\hat{L}_i^{a,c}(\theta)$ and $\hat{L}_i^{b,c}(\theta)$ are the estimators of the expected conditional losses $L^{a,c}(\theta)$ and $L^{b,c}(\theta)$, respectively.

The estimators $\hat{L}_i^{a,c}(\theta)$ and $\hat{L}_i^{b,c}(\theta)$ are computed on the private data set $B_i$ by $\hat{L}_i^{a,c}(\theta) = \frac{1}{m_i} \sum_{(x,s,c) \in B_i} \ell(f_\theta(x), c)$, where $s \in \{a, b\}$ indicates the group membership of a data instance $(x, s, c)$, and $m_i^{s,c}$ is the number of data instances in $B_i$ that belong to a group $s$ and have a class label $c$.

We use the local estimations computed on the private data sets of all the clients to build $N$ local DGEO constraints as $|\hat{L}_i^{a,c}(\theta) - \hat{L}_i^{b,c}(\theta)| \leq \epsilon, \forall i \in \{1, \ldots, N\}$.

By incorporating these constraints into the federated loss in Equation (3), we convert the problem in Equation (2) into the following locally constrained optimization (LCO) problem.

$$\begin{align}
\min_{\theta} & \quad \sum_{i=1}^{N} \frac{m_i}{m_{\text{total}}} \hat{L}_i(\theta) \\
\text{s.t.} & \quad |\hat{L}_i^{a,c}(\theta) - \hat{L}_i^{b,c}(\theta)| \leq \epsilon, \forall i \in \{1, \ldots, N\}
\end{align}$$

(4a)

(4b)

We show in Section IV-D that the LCO problem can be easily solved without infringing the data privacy of any client.

Assuming that the data instances in the data set $B_i$ are independently and identically drawn from $\mathcal{D}$, what is the chance that a feasible solution to the LCO problem may achieve a good fairness performance on the distribution $\mathcal{D}$?

Unfortunately, as indicated in the following analysis, the answer is not encouraging.

Theorem 1. Denote by $\sigma_i^2$, $i \in \{1, \ldots, N\}$, the variance of the estimation $\hat{L}_i^{a,c}(\theta) - \hat{L}_i^{b,c}(\theta)$ on the $i$-th client, and by $\sigma_{\text{min}}^2 = \min(\sigma_1^2, \ldots, \sigma_N^2)$ the minimum variance. If the
data instances in the data set \( B_i \) are independently and identically drawn from \( D \), then for any feasible solution \( \theta \) to the LCO problem and any real number \( r > 0 \),
\[
P \left( \left| L^{a,c}(\theta) - L^{b,c}(\theta) \right| < \epsilon + r \right) \geq 1 - \frac{\sigma_{\text{min}}^2}{r^2}.
\]

**Proof.** For a client \( U_i \), \( i \in \{1, \ldots, N\} \), define \( \mu(\theta) = L^{a,c}(\theta) - L^{b,c}(\theta) \) and \( R_i(\theta) = \hat{L}_{i}^{a,c}(\theta) - \hat{L}_{i}^{b,c}(\theta) \), where \( \mu(\theta) \) is a random variable because \( \theta \) is sampled from the random feasible region defined by Equation (4b).

Since all the data instances in the data set \( B_i \) are independently and identically drawn from \( D \), \( \mathbb{E}(R_i(\theta)) = \mu(\theta) \) always holds for every sample of \( \theta \). By substituting \( \mathbb{E}(R_i(\theta)) = \mu(\theta) \) into Chebyshev’s inequality, we have
\[
P \left( \left| R_i(\theta) - \mu(\theta) \right| < r \right) \geq 1 - \frac{\sigma^2}{r^2} \quad \text{for every sample of } \theta \text{ and any real number } r > 0.
\]

Since a feasible solution \( \theta \) to the LCO problem always satisfies \( |R_i(\theta)| \leq \epsilon \), if \( \theta \) satisfies \( |R_i(\theta) - \mu(\theta)| < r \), then \( |\mu(\theta)| < \epsilon + r \) always holds. This further indicates
\[
P (|\mu(\theta)| < \epsilon + r) \geq P (|R_i(\theta) - \mu(\theta)| < r),
\]

which means
\[
P (|\mu(\theta)| < \epsilon + r) \geq 1 - \frac{\sigma^2}{r^2}.
\]

Plugging \( \mu(\theta) = L^{a,c}(\theta) - L^{b,c}(\theta) \) into the above inequality, we have that the following holds for all \( i \) in \( \{1, \ldots, N\} \),
\[
P \left( \left| L^{a,c}(\theta) - L^{b,c}(\theta) \right| < \epsilon + r \right) \geq 1 - \frac{\sigma^2}{r^2}.
\]

This gives \( N \) independent inequalities that hold for the clients \( U_1, U_2, \ldots, U_N \), respectively. In all these \( N \) inequalities, the tightest one is
\[
P \left( \left| L^{a,c}(\theta) - L^{b,c}(\theta) \right| < \epsilon + r \right) \geq 1 - \frac{\sigma_{\text{min}}^2}{r^2},
\]

because the other \( N - 1 \) inequalities automatically hold when this inequality holds. The theorem follows. \( \square \)

According to Theorem 1, when the data instances in the data set \( B_i \) are independently and identically drawn from \( D \), a solution \( \theta \) to the LCO problem is \((\epsilon + r)\)-fair with a probability no smaller than a lower bound \( 1 - \frac{\sigma_{\text{min}}^2}{r^2} \).

Unfortunately, this lower bound may be small in practice, because the variances \( \sigma_1^2, \ldots, \sigma_N^2 \) may not be small when the numbers of private data instances owned by the clients are not large enough. As a result, solving the LCO problem may not have a large probability of achieving a good model fairness performance.

Moreover, as observed in the experiments in Section V-D, the LCO problem may even be infeasible when a large number of clients introduce a large number of local DGOE constraints. Because the intersection of the feasible regions induced by these constraints may have a good chance to be empty due to the variances \( \sigma_1^2, \ldots, \sigma_N^2 \).

**C. The Federated Estimation Method**

To tackle the above issues with the local DGOE constraints, we propose to estimate \( L^{a,c}(\theta) - L^{b,c}(\theta) \) by
\[
\frac{1}{N} \sum_{i=1}^{N} \left( \hat{L}_{i}^{a,c}(\theta) - \hat{L}_{i}^{b,c}(\theta) \right).
\]

We call this **federated estimation** because it is computed by taking the average of the local estimations computed by all the clients.

Similar to how we compute the federated loss, we compute the federated estimation on the server. This does not infringe on the data privacy of any client because the local estimations are privately computed by the clients before they are sent to the server.

We use the federated estimation to develop a federated DGOE constraint as
\[
\left| \frac{1}{N} \sum_{i=1}^{N} (\hat{L}_{i}^{a,c}(\theta) - \hat{L}_{i}^{b,c}(\theta)) \right| \leq \epsilon,
\]

and further incorporate this constraint with the federated loss in Equation (3) to convert the constrained optimization problem in Equation (2) into the following FedFair problem.

\[
\min_{\theta} \sum_{i=1}^{N} \frac{m_i}{m_{\text{total}}} \hat{L}_{i}(\theta)
\]

s.t.
\[
\frac{1}{N} \sum_{i=1}^{N} \left( \hat{L}_{i}^{a,c}(\theta) - \hat{L}_{i}^{b,c}(\theta) \right) \leq \epsilon
\]

Let us analyze the model fairness performance of a solution to the FedFair problem when the data instances in the data set \( B_i \) are independently and identically drawn from \( D \).

**Theorem 2.** Denote by \( \sigma_i^2 \), \( i \in \{1, \ldots, N\} \), the variance of the local estimation \( \hat{L}_{i}^{a,c}(\theta) - \hat{L}_{i}^{b,c}(\theta) \) on client \( U_i \). Let \( \sigma_{\text{avg}}^2 = \frac{1}{N} \sum_{i=1}^{N} \sigma_i^2 \). If the data instances in the data set \( B_i \) are independently and identically drawn from \( D \), then for any feasible solution \( \theta \) to the FedFair problem and any \( r > 0 \),
\[
P \left( \left| L^{a,c}(\theta) - L^{b,c}(\theta) \right| < \epsilon + r \right) \geq 1 - \frac{\sigma_{\text{avg}}^2}{N r^2}.
\]

**Proof.** Since the private data sets \( B_1, \ldots, B_N \) are independently and identically drawn from \( D \), the local estimations computed on these private data sets are independent of each other. Therefore, the variance of the federated estimation (see Equation (5)) is
\[
\sigma_{\text{fed}}^2 = \text{Var} \left[ \frac{1}{N} \sum_{i=1}^{N} (\hat{L}_{i}^{a,c}(\theta) - \hat{L}_{i}^{b,c}(\theta)) \right] = \frac{\sigma_{\text{avg}}^2}{N}.
\]

By Chebyshev’s inequality and following the proof of Theorem 1, we have
\[
P \left( \left| L^{a,c}(\theta) - L^{b,c}(\theta) \right| < \epsilon + r \right) \geq 1 - \frac{\sigma_{\text{fed}}^2}{r^2}.
\]

Since \( \sigma_{\text{fed}}^2 = \frac{\sigma_{\text{avg}}^2}{N} \), the theorem follows. \( \square \)

According to Theorem 2, when the data instances in the data set \( B_i \) are independently and identically drawn from \( D \), a feasible solution \( \theta \) to the FedFair problem is \((\epsilon + r)\)-fair with a probability no smaller than a lower bound \( 1 - \frac{\sigma_{\text{avg}}^2}{N r^2} \). Interestingly, this means that the solution \( \theta \) has a higher probability to be \((\epsilon + r)\)-fair when the number of clients increases, because the variable \( N \) in the lower bound \( 1 - \frac{\sigma_{\text{avg}}^2}{N r^2} \) is the number of clients.

Based on Theorems 1 and 2, we can compare the model fairness performance of the FedFair approach and the LCO approach when the data instances in the data set \( B_i \) are independently and identically drawn from \( D \). The key is to analyze the difference between \( \sigma_{\text{avg}}^2 \) and \( \sigma_{\text{min}}^2 \), which depends on the differences among \( \sigma_1^2, \ldots, \sigma_N^2 \).

Since we use the same formula to compute every local estimation based on the data instances drawn from the same
distribution \( D \), the differences among \( \sigma_1^2, \ldots , \sigma_N^2 \) are determined by the differences among the numbers of data instances owned by different clients. In typical federated learning applications [3]–[5], clients often have similar amounts of data. This produces small differences among \( \sigma_1^2, \ldots , \sigma_N^2 \), which further means \( \sigma_{avg}^2 \) and \( \sigma_{min}^2 \) are close to each other.

More often than not, the number of clients \( N \) is large in typical federated learning scenarios [2], [3]. Thus, the variance \( \sigma_{fed}^2 = \frac{\sigma_{avg}^2 - \sigma_{min}^2}{2} \) of the federated estimation is much smaller than the minimum variance \( \sigma_{min}^2 \) of the local estimations. In consequence, based on Theorems 1 and 2, a solution to the FedFair problem can have a much higher probability of achieving good model fairness performance than a solution to the LCO problem.

Theorems 1 and 2 explain the advantage of FedFair over LCO when the data instances in the data set \( B_i \) are independently and identically drawn from \( D \). Although our above analysis is on independently and identically sampled data, as to be discussed later in Section V, we conduct extensive experiments on data sets that are drawn independently and identically from \( D \) and on data sets that are not drawn independently and identically from \( D \). FedFair achieves the best performance on all the data sets.

**D. Solving the FedFair and LCO Problems**

In this subsection, we discuss how to solve the FedFair problem and the LCO problem without infringing on the data privacy of any client. We first convert the FedFair problem to a nonconvex-concave min-max problem. Then, we apply the alternating gradient projection (AGP) algorithm [21] to tackle the min-max problem without infringing the data privacy of any client. We also extend AGP to tackle the LCO problem in a similar privacy-preserving way.

AGP [21] is designed to obtain a \( \delta \)-stationary point of a nonconvex-concave min-max problem in \( O(\delta^{-4}) \) iterations. It alternatively updates primal and dual variables by employing simple projected gradient steps at each iteration.

Denote by \( \bar{D}_i(\theta) = L_i^{\bar{b}}(\theta) - L_i^{\bar{a}}(\theta), i \in \{1, \ldots , N\} \), the local estimation of client \( U_i \). We rewrite the federated DCEO constraint in the FedFair problem to \(-\varepsilon \leq \frac{1}{N} \sum_{i=1}^{N} \bar{D}_i(\theta) \leq \varepsilon \), which represents two constraints on \( \theta \). In this way, the FedFair problem can be equivalently converted to a *min-max problem*

\[
\min_{\theta} \max_{\lambda_a, \lambda_b \geq 0} \mathcal{L}(\theta, \lambda_a, \lambda_b),
\]

where

\[
\mathcal{L}(\theta, \lambda_a, \lambda_b) = \frac{1}{N} \sum_{i=1}^{N} \frac{m_i}{m_{total}} \bar{L}_i(\theta) + \lambda_a \left( \frac{1}{N} \sum_{i=1}^{N} \bar{D}_i(\theta) - \varepsilon \right) - \lambda_b \left( \frac{1}{N} \sum_{i=1}^{N} \hat{D}_i(\theta) + \varepsilon \right)
\]

(7)

is a Lagrangian function with the Lagrange multipliers \( \lambda_a, \lambda_b \). Since the Lagrangian function is nonconvex in terms of \( \theta \) and concave in terms of \( \lambda_a \) and \( \lambda_b \), the min-max problem can be directly solved using the AGP method [21].

**Algorithm 1: Tackling the FedFair Problem**

**Input:** The clients \( U_1, \ldots , U_N \) holding the private data sets \( B_1, \ldots , B_N \), respectively; a pre-defined pair of groups \( \{a, b\} \); a class \( c \in \{1, \ldots , C\} \); the hyperparameters \( \alpha, \beta, \gamma \) of AGP [21]; and a server \( Q \).

**Output:** A fair model \( f_0 \) that achieves a small DEGO on \( D \) with respect to the groups \( \{a, b\} \) and the class \( c \).

1. \( Q \) initializes \( \theta^0, \lambda_a^0 \) and \( \lambda_b^0 \) in the same way as AGP [21].
2. for \( k = 1, 2, \ldots , K \) do
3. \( Q \) broadcasts \( \theta^k, \lambda_a^k \) and \( \lambda_b^k \) to the clients.
4. Each client \( U_i \) uses \( B_i \) to compute \( \hat{D}_i(\theta^k), \nabla \hat{D}_i(\theta^k) \)
and \( \nabla L_i(\theta^k) \), and send them to \( Q \).
5. \( Q \) updates \( \theta^{k+1}, \lambda_a^{k+1} \) and \( \lambda_b^{k+1} \) by (8), (10) and (11), respectively.
6. end
7. return The fair model \( f_0 \).

Each iteration of AGP updates \( \theta, \lambda_a, \) and \( \lambda_b \) using the gradients computed from a regularized loss function \( \mathcal{L}(\theta, \lambda_a, \lambda_b) = \mathcal{L}(\theta, \lambda_a, \lambda_b) - \frac{\gamma}{2} \lambda_a^2 + \frac{\lambda_b^2}{2} \) is a regularization term that makes \( \mathcal{L}(\theta, \lambda_a, \lambda_b) \) strongly concave with respect to \( \lambda_a \) and \( \lambda_b \) to achieve a faster convergence speed, and \( \gamma \geq 0 \) is a small hyperparameter to control the level of regularization.

Specifically, the \( k \)-th iteration of AGP consists of one gradient step to update \( \theta \) and one projected gradient descent step to update \( \lambda_a \) and \( \lambda_b \).

The first step updates \( \theta \) with a step size \( \alpha \) by

\[
\theta^{k+1} = \theta^k - \alpha \sum_{i=1}^{N} \nabla \hat{L}_i(\theta^k), \tag{8}
\]

where

\[
\nabla \hat{L}_i(\theta^k) = \frac{m_i}{m_{total}} \nabla \hat{L}_i(\theta^k) + \frac{\lambda_a^k - \lambda_b^k}{N} \nabla \hat{D}_i(\theta^k) \tag{9}
\]

is the gradient computed by client \( U_i \). Here, \( \nabla \hat{L}_i(\theta^k) \) and \( \nabla \hat{D}_i(\theta^k) \) are respectively the gradients of \( \hat{L}_i(\theta^k) \) and \( \hat{D}_i(\theta^k) \) with respect to \( \theta^k \). These gradients are computed by client \( U_i \) using \( B_i \).

The second step updates \( \lambda_a \) and \( \lambda_b \) with a step size \( \beta \) by

\[
\lambda_a^{k+1} = \max \left( 1 - \gamma \beta \lambda_b^k + \frac{\beta}{N} \sum_{i=1}^{N} \hat{D}_i(\theta^k) - \beta \varepsilon, 0 \right) \tag{10}
\]

and

\[
\lambda_b^{k+1} = \max \left( 1 - \gamma \beta \lambda_a^k - \frac{\beta}{N} \sum_{i=1}^{N} \hat{D}_i(\theta^k) - \beta \varepsilon, 0 \right). \tag{11}
\]

The above iteration continues until AGP converges. Algorithm 1 shows how we deploy AGP in a federated learning framework to tackle the FedFair problem. The LCO problem can be solved in a similar way as the FedFair problem using AGP [21]. We first introduce a pair of Lagrange multipliers, denoted by \( \lambda_{a_i} \) and \( \lambda_{b_i} \), \( i \in \{1, \ldots , N\} \), for each local DCEO constraint rewritten as

\[
-\varepsilon \leq L_i^{\bar{b}}(\theta^k) - L_i^{\bar{a}}(\theta^k) \leq \varepsilon \tag{11}
\]

Then, we adapt Algorithm 1 by replacing Equation (9) by \( \nabla \hat{L}_i(\theta^k) \) by \( \frac{m_i}{m_{total}} \nabla \hat{L}_i(\theta^k) + (\lambda_a^k - \lambda_b^k) \hat{D}_i(\theta^k) \) in (8), (10) and (11).
for each client based on its private training data set. This will
obtain the sixth and the seventh methods in our experiments.

As the fourth and the fifth methods in our experiments, we
evaluate the performance of our proposed methods for feder-
ated fair model training, and compare with the state-of-the-art
baselines.

V. EXPERIMENTAL RESULTS

In this section, we conduct a series of experiments to
evaluate the performance of our proposed methods for feder-
ated model training, and compare with the state-of-the-art
baselines.

A. Experiment Settings

In our experiments, we evaluate the performance of a series of
seven methods. First of all, FedFair is our proposed method
solving the FedFair Problem in Equation (6). Second, LCO is
the method solving the LCO problem in Equation (4). Third,
we use the state-of-the-art federated fair model training
method AgnosticFair (AF) [15] as a baseline.

As the fourth and the fifth methods in our experiments, we
compare with a separate training (ST) method and the classical
federated learning method FedAvg [4]. We also extend FedAvg
by two post-processing fairness enhancing methods, equalized
odds (EO) [27] and calibrated equalized odds (CEO) [28], to
obtain the sixth and the seventh methods in our experiments.

The ST method simply asks each client \( U_i \) to use its own
private training data set \( B_i \) to train a fair model by solving
the following constrained optimization problem:

\[
\min_{\theta} \hat{L}_i(\theta) \\
\text{s.t. } |\hat{L}_i^{a,c}(\theta) - \hat{L}_i^{b,c}(\theta)| \leq \epsilon
\]

(12a)

(12b)

The post-processing methods, such as EO and CEO, are
used to separately post-process the model trained by FedAvg
for each client based on its private training data set. This will
produce \( N \) fairness enhanced models for \( N \) clients. We report
the average performance and the variance of those \( N \) models.

For all the methods except AF, we use two types of models
for \( f_0 \). One is logistic regression, the other one is a fully
connected neural network with two hidden layers, where the
first and the second hidden layers contain eight and four ReLU
units, respectively.

We do not report the performance of AF for the neural
network model because the source code of AF is not applicable
to training neural networks.

The source codes of all the baseline methods are provided
by the authors of [4], [15], [27], [28]. FedFair and LCO are
implemented in PyTorch 1.4, the source code is available at:
https://bit.ly/fairfed. All the experiments are conducted on Dell
Alienware with Intel(R) Core(TM) i9-9980XE CPU, 128G
memory, NVIDIA 1080Ti, and Ubuntu 16.04.

B. Data Sets

We adopt the following three benchmark data sets that have
been widely used in literature to evaluate the performances of
fair model training methods [20], [29].

For each of the following data sets, we use an IID setting
and a non-IID setting. For the IID setting, we uniformly
sample the data for every client, such that the data of all clients
are independently and identically distributed. For the non-
IID setting, we adopt the non-IID data partitioning strategy
proposed by Huang et al. [7], such that the data of clients are
not independently and identically distributed.

The ADULT data set [30] contains 45,222 data instances.
Each data instance consists of 14 features of a person. The binary
class label of a data instance indicates whether a person’s
annual income is above 50,000 dollars or not. Following the
settings of Hardt et al. [27], we use “female” and “male” as
the pair of protected groups, and use “above 50,000 dollars”
as the protected class. For both the IID setting and the non-
IID setting, we use 40,000 data instances as the training data,
use 5,222 data instances as the testing data, and use 50 clients
each holding \( \frac{1}{50} \) of the training data as the private training
data set.

The COMPAS data set [31] contains 5,278 data instances.
Every data instance consists of 16 features of a person. The binary
class label of a data instance indicates whether the person is a recidivist or not. Following the settings of
Hardt et al. [27], we use “African-American” and “Caucasian”
as the pair of protected groups, and use “not a recidivist” as
the protected class. For both the IID setting and the non-IID
setting, we use 4,800 data instances as the training data set,
478 data instances as the testing data, and 20 clients each
holding \( \frac{1}{20} \) of the training data as the private training data set.

The DRUG data set [32] contains 1,885 data instances. Each
data instance represents a person characterized by 14 features.
The binary class label of a data instance indicates whether the
person abuses volatile substance or not. We use “white” and
“non-white” as the pair of protected groups, and use “not abuse
volatile substance” as the protected class. For both the IID
setting and the non-IID setting, we use 1,600 data instances
produced by the federated DGEO constraint is the time for each client to compute and communicate \( \hat{D}_i(\theta^k) \) and \( \nabla \hat{D}_i(\theta^k) \). Such extra time cost also scales linearly with respect to the number of clients.

Algorithm 1 protects the data privacy of the clients in a
similar way as [4], [7], [26]. In each iteration, there is no
communication between the clients, and the information sent
from a client to the server only contains \( D_i(\theta^k) \), \( \nabla D_i(\theta^k) \), and \( \nabla L_i(\theta^k) \), which do not expose the private data sets of
any clients or any information about the distribution of the
private datasets.

The overall time cost of Algorithm 1 scales linearly with
respect to the number of clients, because every client conducts
the same computational process and there is no communica-
tion between clients. Comparing to the classic federated
learning [4] which does not involve the federated DGEO con-
straint, the major extra time cost introduced by the federated
DGEO constraint is the time for each client to compute and
communicate \( D_i(\theta^k) \) and \( \nabla D_i(\theta^k) \). Such extra time cost also scales linearly with respect to the number of clients.

The DRUG data set [32] contains 1,885 data instances. Each
data instance represents a person characterized by 14 features.
The binary class label of a data instance indicates whether the
person abuses volatile substance or not. We use “white” and
“non-white” as the pair of protected groups, and use “not abuse
volatile substance” as the protected class. For both the IID
setting and the non-IID setting, we use 1,600 data instances
produced by the federated DGEO constraint is the time for each client to compute and communicate \( \hat{D}_i(\theta^k) \) and \( \nabla \hat{D}_i(\theta^k) \). Such extra time cost also scales linearly with respect to the number of clients.

Algorithm 1 protects the data privacy of the clients in a
similar way as [4], [7], [26]. In each iteration, there is no
communication between the clients, and the information sent
from a client to the server only contains \( D_i(\theta^k) \), \( \nabla D_i(\theta^k) \), and \( \nabla L_i(\theta^k) \), which do not expose the private data sets of
any clients or any information about the distribution of the
private datasets.

The overall time cost of Algorithm 1 scales linearly with
respect to the number of clients, because every client conducts
the same computational process and there is no communica-
tion between clients. Comparing to the classic federated
learning [4] which does not involve the federated DGEO con-
straint, the major extra time cost introduced by the federated
DGEO constraint is the time for each client to compute and
communicate \( D_i(\theta^k) \) and \( \nabla D_i(\theta^k) \). Such extra time cost also scales linearly with respect to the number of clients.
as the training data, 285 data instances as the testing data, and 10 clients each holding \( \frac{1}{10} \) of the training data as the private training data set.

### C. Fairness Measure

In our experiments, the fairness of a trained model \( f_\theta \) is evaluated by

\[
\text{fairness} = 1 - \text{DEO}(f_\theta),
\]

where \( \text{DEO}(f_\theta) \) is the difference of equal opportunity (DEO) [20] of the trained model \( f_\theta \), and it is computed by

\[
\text{DEO}(f_\theta) = \left| P\{ f_\theta(x) > 0 \mid y = c, s = a \} - P\{ f_\theta(x) > 0 \mid y = c, s = b \} \right|. \quad (14)
\]

Essentially, for an instance \( x \) with a ground truth label \( y = c, c \) is the protected class, \( \text{DEO}(f_\theta) \) measures the absolute difference of the probabilities of \( x \) being predicted by \( f_\theta \) to be in the class \( c \) when \( x \) belongs to the groups \( a \) and \( b \), respectively. A smaller value of \( \text{DEO}(f_\theta) \) means \( f_\theta \) is more fair and a larger value means \( f_\theta \) is less fair.

According to [20], \( \text{DEO}(f_\theta) \) is equivalent to DGEO when the loss function is a discrete function, that is, \( \ell(f_\theta(x), y) = \mathbb{I}_{y \cdot f_\theta(x) \leq 0} \). The value range of \( \text{DEO}(f_\theta) \) is \([0, 1]\), but the value range of DGEO is unbounded. Therefore, \( \text{DEO}(f_\theta) \) is a better choice to develop an evaluation metric of fairness in our experiments. As a result, we use \( \text{DEO}(f_\theta) \) to develop the “fairness” in Equation (13). Obviously, “fairness” also has a value range of \([0, 1]\) and a larger value of fairness means \( f_\theta \) is more fair.

We also evaluate the overall performance of \( f_\theta \) by the harmonic mean of its fairness and accuracy, which is simply referred to as the harmonic mean (HM) when the context is clear. For ST, EO and CEO that produce \( N \) different models for \( N \) clients, respectively, we report the average accuracy, the average fairness, and the average of the harmonic means of the \( N \) models on the testing data sets.

### D. Fairness and Accuracy of Trained Models

Each of ST, LCO, AF, and FedFair has some hyper-parameters that control the trade-off between fairness and accuracy. To comprehensively compare the performance of these methods, we conduct a grid search on the hyper-parameters. This produces multiple trained models for every method, where each model is trained using a unique value of the hyper-parameter. For every method, we report the performance of the models that achieve the top-5 largest harmonic means. For each trained model, the fairness and the accuracy are plotted as a single point in Figure 1. Since we train 5 models for each method,
Table I

The performance of all the methods on the data sets in the IID setting. The accuracy, fairness, and harmonic mean are denoted by AC, FR, and HM, respectively. “LR” means that the trained model is a logistic regression model and “NN” means that the trained model is a neural network. The column OPT reports the optimal hyper-parameter $\tau$ for AF and the optimal hyper-parameter $\epsilon$ for ST, LCO, and FedFair, to achieve the best harmonic mean of fairness and accuracy.

| Method   | DRUG (IID), LR | COMPASS (IID), LR | ADULT (IID), LR |
|----------|----------------|------------------|-----------------|
|          | OPT | AC | FR | HM | OPT | AC | FR | HM | OPT | AC | FR | HM |
| FedAvg   | n/a | 0.72 ± 0.02 | 0.85 ± 0.07 | 0.78 ± 0.04 | n/a | 0.65 ± 0.01 | 0.71 ± 0.01 | n/a | 0.84 ± 0.00 | 0.89 ± 0.01 | 0.87 ± 0.01 |
| EO       | n/a | 0.61 ± 0.09 | 0.67 ± 0.05 | 0.68 ± 0.03 | n/a | 0.63 ± 0.02 | 0.68 ± 0.04 | n/a | 0.68 ± 0.01 | 0.70 ± 0.02 | 0.68 ± 0.02 |
| CEO      | 0.01 | 0.66 ± 0.02 | 0.97 ± 0.02 | 0.97 ± 0.02 | n/a | 0.65 ± 0.01 | 0.71 ± 0.01 | n/a | 0.84 ± 0.00 | 0.89 ± 0.01 | 0.87 ± 0.01 |
| ST       | 0.07 | 0.61 ± 0.01 | 0.82 ± 0.02 | 0.97 ± 0.02 | n/a | 0.65 ± 0.01 | 0.71 ± 0.01 | n/a | 0.84 ± 0.00 | 0.89 ± 0.01 | 0.87 ± 0.01 |
| LCO      | 0.40 | 0.72 ± 0.02 | 0.73 ± 0.03 | 0.81 ± 0.02 | n/a | 0.60 ± 0.01 | 0.68 ± 0.04 | n/a | 0.68 ± 0.01 | 0.70 ± 0.02 | 0.68 ± 0.02 |
| FedFair  | 0.05 | 0.72 ± 0.02 | 0.73 ± 0.03 | 0.81 ± 0.02 | n/a | 0.60 ± 0.01 | 0.68 ± 0.04 | n/a | 0.68 ± 0.01 | 0.70 ± 0.02 | 0.68 ± 0.02 |

Table II

The performance of all the methods on the data sets in the non-IID setting. The accuracy, fairness, and harmonic mean are denoted by AC, FR, and HM, respectively. “LR” means that the trained model is a logistic regression model and “NN” means that the trained model is a neural network. The column OPT reports the optimal hyper-parameter $\tau$ for AF and the optimal hyper-parameter $\epsilon$ for ST, LCO, and FedFair, to achieve the best harmonic mean of fairness and accuracy.

| Method   | DRUG (non-IID), LR | COMPASS (non-IID), LR | ADULT (non-IID), LR |
|----------|-------------------|----------------------|---------------------|
|          | OPT | AC | FR | HM | OPT | AC | FR | HM | OPT | AC | FR | HM |
| FedAvg   | n/a | 0.61 ± 0.00 | 0.67 ± 0.02 | 0.78 ± 0.01 | n/a | 0.72 ± 0.00 | 0.87 ± 0.01 | n/a | 0.93 ± 0.00 | 0.92 ± 0.01 | 0.89 ± 0.01 |
| EO       | n/a | 0.61 ± 0.01 | 0.66 ± 0.02 | 0.78 ± 0.01 | n/a | 0.72 ± 0.00 | 0.87 ± 0.01 | n/a | 0.93 ± 0.00 | 0.92 ± 0.01 | 0.89 ± 0.01 |
| CEO      | 0.01 | 0.66 ± 0.02 | 0.76 ± 0.03 | 0.71 ± 0.02 | n/a | 0.72 ± 0.00 | 0.87 ± 0.01 | n/a | 0.93 ± 0.00 | 0.92 ± 0.01 | 0.89 ± 0.01 |
| ST       | 0.05 | 0.61 ± 0.00 | 0.67 ± 0.02 | 0.78 ± 0.01 | n/a | 0.72 ± 0.00 | 0.87 ± 0.01 | n/a | 0.93 ± 0.00 | 0.92 ± 0.01 | 0.89 ± 0.01 |
| LCO      | 0.40 | 0.72 ± 0.01 | 0.73 ± 0.03 | 0.81 ± 0.02 | n/a | 0.60 ± 0.01 | 0.68 ± 0.04 | n/a | 0.68 ± 0.01 | 0.70 ± 0.02 | 0.68 ± 0.02 |
| FedFair  | 0.05 | 0.72 ± 0.01 | 0.73 ± 0.03 | 0.81 ± 0.02 | n/a | 0.60 ± 0.01 | 0.68 ± 0.04 | n/a | 0.68 ± 0.01 | 0.70 ± 0.02 | 0.68 ± 0.02 |

Every method has 5 points in Figure 1. Some methods show better results, because they can explicitly impose the required trade-off between accuracy and fairness in the objective functions to find a good balance during training [33].

The accuracies of the models trained by LCO, AF, and FedFair are mostly comparable, but FedFair always achieves the best fairness performance. This demonstrates the outstanding performance of FedFair in training fair models with high accuracies.

The accuracies of the logistic regression models in Figure 1(a) are slightly better than the accuracies of the corresponding neural network models in Figure 1(b) because the neural network models tend to overfit the small training data of DRUG.

We can also conclude from Figure 1 that the fairness of the models trained by LCO is weaker than that of FedFair because the LCO problem quickly becomes infeasible when $\epsilon$ gets small, which makes it impossible to improve the fairness of the models trained by LCO. To the contrary, the FedFair
problem stays feasible for small values of $\epsilon$, thus it is able to train models with a much higher fairness than LCO with only slight overhead in accuracy.

Tables I and II summarize the performance of all the compared methods on the data sets in the IID setting and the non-IID setting, respectively. To analyze the variance of the performance, for each of the IID setting and the non-IID setting, we repeat the random data set construction process introduced in Section V-B to construct four additional data sets for each of DRUG, COMPAS, and ADULT. Every additional data set is randomly constructed using a unique random seed to generate the $N$ private training data sets and the testing data set. For each of DRUG, COMPAS, and ADULT, including the first data set we constructed at the beginning, in total we have a collection of five randomly constructed data sets for the IID setting and another collection of five randomly constructed data sets for the non-IID setting.

For FedAvg, AF, LCO, and FedFair, we train five models on the five random data sets in the same collection. Then, we evaluate the mean and standard deviation of the accuracies, fairness, and harmonic means of the five models. For EO, CEO, and ST, we train one model per client on each of the five random data sets in the same collection. This produces $5N$ models for each collection of random data sets, where $N$ is the number of clients. Then, we evaluate the mean and standard deviation of the accuracies, fairness, and harmonic means of the $5N$ models. For ST, AF, LCO, and FedFair, we use the optimal hyper-parameter that achieves the best harmonic mean performance. FedAvg, EO, and CEO are evaluated with the default settings because they do not provide a hyper-parameter to control the trade-off between accuracy and fairness.

As shown in Tables I and II, in most cases, the accuracies of the models trained by FedFair are comparable with the other methods, but FedFair achieves the best fairness in most of the cases and always achieves the best harmonic mean. This demonstrates the outstanding performance of FedFair in training fair models with high accuracies.

### E. Effect of Parameter $\epsilon$

In this subsection, we first analyze how the parameter $\epsilon$ affects the fairness and accuracy of the logistic regression models trained by FedFair, then we analyze the effect of $\epsilon$ on the neural network models trained by FedFair.

Figure 2 shows the performance of the logistic regression models trained by FedFair. For each data set, we use FedFair to train a logistic regression model $f_0$ with different values of $\epsilon$. The sets of $\epsilon$ we use are $\{10^{-4}, 10^{-3}, 0.01, 0.07, 0.13, 0.19\}$ for DRUG and $\{10^{-4}, 10^{-3}, 0.01, 0.1, 0.2, 0.4\}$ for COMPASS and ADULT. We report the performance of $f_0$ in fairness and accuracy on the testing data.

As it is shown in Figures 2(a), 2(c), 2(e), and 2(g), reducing $\epsilon$ significantly improves the fairness of $f_0$, since a smaller $\epsilon$ requires $f_0$ to have a smaller DGE0, that is, tending to be fairer. However, a smaller $\epsilon$ does not always produce a better fairness here. The reason is that the evaluation metric of fairness is defined based on the notion of DEO($f_0$), which is equivalent to DGE0 only if a discrete loss function $\ell(f_0(x), y) = \mathbb{1}_{y \neq f_0(x)} \leq 0$ is used [20]. The discrete loss function cannot be used to train the logistic regression model $f_0$, so we use a continuous loss function, which makes DGE0 not equivalent to DEO($f_0$).

The value of $\epsilon$ also has an effect on the accuracy of $f_0$. As it is shown in Figures 2(a), 2(e), and 2(i), on the data sets in the IID setting. The accuracy of $f_0$ drops when $\epsilon$ becomes smaller, because reducing $\epsilon$ induces a tighter federated DGE0 constraint, which shrinks the feasible region of the FedFair problem and makes it harder to find a good solution. As shown
in Figure 2(c), on the DRUG data set in the non-IID setting, the accuracy of $f_0$ does not always drop when $\epsilon$ becomes smaller, because the local data sets on different clients are not identically and independently distributed. However, this does not affect the practical performance of FedFair much. As discussed in Section V-D, FedFair always achieves the best performance on all data sets in both the IID and the non-IID settings.

Next, we analyze the effect of $\epsilon$ on the performance of the neural network models trained by FedFair in Figure 2. The values taken by $\epsilon$ are $\{10^{-4}, 10^{-5}, 0.005, 0.01, 0.02, 0.07, 0.13, 0.19, 0.22, 0.25\}$ for DRUG and $\{10^{-4}, 10^{-3}, 0.01, 0.1, 0.2, 0.4\}$ for COMPAS and ADULT.

We can see in Figure 2 that the accuracies of the neural network models do not always drop with the decrease of $\epsilon$. This is largely due to the high model complexity and high non-linearity of the neural network models, where a tighter federated DGEO constraint induced by a smaller $\epsilon$ may act as a regularization term to improve the testing accuracies of the neural network models. We can also observe that a smaller $\epsilon$ does not always induce a larger fairness for the neural network models. This is because the notion of DEO($f_0$), from which we develop fairness, is not equivalent to DGEO when we use cross-entropy loss to train neural network models [20].

VI. CONCLUSIONS

In this paper, we tackle the task of training fair models in federated learning. We first propose an effective federated estimation method to accurately estimate the fairness of a model and analyze why it achieves a smaller estimation variance than locally estimating the fairness of a model on every client. Based on the federated estimation, we develop a fairness constraint that can be smoothly incorporated into an effective federated learning framework to train high-performance fair models without infringing the data privacy of any client.

REFERENCES

[1] D. Pessach and E. Shmueli, “Algorithmic fairness,” arXiv preprint arXiv:2001.09784, 2020.
[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings et al., “Advances and open problems in federated learning,” Foundations and Trends® in Machine Learning, vol. 14, no. 1–2, pp. 1–210, 2021.
[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” ACM Transactions on Intelligent Systems and Technology, vol. 10, no. 2, pp. 1–19, 2019.
[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Artificial Intelligence and Statistics, 2017, pp. 1273–1282.
[5] Y. Chen, L. Su, and J. Xu, “Distributed statistical machine learning in adversarial settings: Byzantine gradient descent,” ACM Measurement and Analysis of Computing Systems, vol. 1, no. 2, pp. 1–25, 2017.
[6] H. Wang, M. Urochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, “Federated learning with matched averaging,” in International Conference on Learning Representations, 2020.
[7] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, “Personalized cross-silo federated learning on non-iid data,” in AAAI Conference on Artificial Intelligence, vol. 35, no. 9, 2021, pp. 7865–7873.
[8] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, “On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing,” IEEE Journal on Selected Areas in Communications, vol. 35, no. 4, pp. 832–847, 2017.

[9] S. Gollapudi, K. Kollias, D. Panigrahi, and V. Platisika, “Profit sharing and efficiency in utility games,” in Annual European Symposium on Algorithms, 2017.
[10] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, “A fairness-aware incentive scheme for federated learning,” in Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 393–399.
[11] M. Mohri, G. Sivek, and A. T. Suresh, “Agnostic federated learning,” in International Conference on Machine Learning, 2019, pp. 4615–4625.
[12] T. Li, M. Sanjabi, A. Beirami, and V. Smith, “Fair resource allocation in federated learning,” arXiv preprint arXiv:1905.10497, 2019.
[13] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, and K. S. Ng, “Towards fair and privacy-preserving federated deep models,” IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 11, pp. 2524–2541, 2020.
[14] R. H. L. Sim, Y. Zhang, M. C. Chan, and B. K. H. Low, “Collaborative machine learning with incentive-aware model rewards,” in International Conference on Machine Learning, 2020, pp. 8927–8936.
[15] W. Du, D. Xu, X. Wu, and H. Tong, “Fairness-agnostic federated learning,” arXiv preprint arXiv:2010.05057, 2020.
[16] D. Y. Zhang, Z. Kou, and D. Wang, “FairFPL: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models,” in IEEE International Conference on Big Data, 2020, pp. 1051–1060.
[17] A. Papadaki, N. Martinez, M. Bertran, G. Sapiro, and M. Rodrigues, “Minimax demographic group fairness in federated learning,” arXiv preprint arXiv:2201.08304, 2022.
[18] S. Hu, Z. S. Wu, and V. Smith, “Provably fair federated learning via bounded group loss,” arXiv preprint arXiv:2203.10190, 2022.
[19] Y. Zeng, H. Chen, and K. Lee, “Improving fairness via federated learning,” arXiv preprint arXiv:2110.13545, 2021.
[20] M. Donini, L. Oneto, S. Ben-David, J. S. Shawe-Taylor, and M. Pontil, “Empirical risk minimization under fairness constraints,” in Advances in Neural Information Processing Systems, 2018, pp. 2791–2801.
[21] Z. Xu, H. Zhang, Y. Xu, and G. Lan, “A unified single-loop alternating gradient projection algorithm for nonconvex-concave and convex-nonconcave minimax problems,” arXiv preprint arXiv:2006.02032, 2020.
[22] T. Calders, F. Kamiran, and M. Pechenizkiy, “Building classifiers with indiscernibility constraints,” in IEEE International Conference on Data Mining Workshops, 2009, pp. 13–18.
[23] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, “Fairness through awareness,” in Theoretical Computer Science Conference, 2012, pp. 214–226.
[24] T. Calders and S. Verwer, “Three naive bayes approaches for discrimination-free classification,” Data Mining and Knowledge Discovery, vol. 21, no. 2, pp. 277–292, 2010.
[25] S. Kadhe, N. Rajaraman, O. O. Koyluoglu, and K. Ramchandran, “Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning,” arXiv preprint arXiv:2009.11248, 2020.
[26] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” Machine Learning and Systems, vol. 2, pp. 429–450, 2020.
[27] M. Hardt, E. Price, and N. Srebro, “Equality of opportunity in supervised learning,” in Advances in Neural Information Processing Systems, 2016, pp. 3315–3323.
[28] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, “On fairness and calibration,” Advances in Neural Information Processing Systems, vol. 30, 2017.
[29] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” arXiv preprint arXiv:1905.10497, 2019.
[30] R. Kohavi, “Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid,” in International Conference on Knowledge Discovery and Data Mining and Data Mining, 1996, pp. 221–227.
[31] D. A. Larson and Mattu, Surya and Kirchner, Lauren and Angwin, Julia, “Compas analysis,” in Github: https://github.com/propublica/compas-analysis/, 2016.
[32] E. Fehrman, A. K. Muhammad, E. M. Mirkes, V. Egan, and A. N. Gorban, “The five factor model of personality and evaluation of drug consumption risk,” in Data science, 2017, pp. 231–242.
[33] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro, “Learning non-discriminatory predictors,” in Conference on Learning Theory, 2017, pp. 1920–1953.