Computers & Security 132 (2023) 103336

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Improve individual fairness in federated learning via adversarial training

Jie Li[a], Tianqing Zhu[c,*], Wei Ren[a], Kim-Kwang Raymond[b]

[a] School of Computer Science, China University of Geosciences, No. 388 Lumo Road, Wuhan, Hubei 430074, PR China
[b] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX78249-0631, USA
[c] School of Computer Science, University of Technology Sydney, Australia

ARTICLE INFO

Article history:
Received 30 November 2022
Revised 23 March 2023
Accepted 9 June 2023
Available online 11 June 2023

Keywords:
Individual fairness
Federated learning
Adversarial training

ABSTRACT

Federated learning (FL) has been widely investigated these years. Since FL will be universally applied in the real world, the fairness issue involved is worthy of attention, while there are few relevant studies. Unlike previous work on group fairness in FL or fairness in centralized machine learning, this paper firstly considers both privacy and individual fairness and proposes promoting individual fairness in FL through distributionally adversarial training without violating data privacy. Specifically, we assume a model satisfying individual fairness as one robust to certain sensitive perturbations, which aligns with the goal of adversarial training. Then we transform the task of training an individually fair FL model into an adversarial training task. To obey the FL requirement of keeping data on clients privately, we execute the adversarial training task on the client side distributionally. Extensive experimental results on two real datasets collectively demonstrate the effectiveness of our proposed method, which not only improves individual fairness significantly but improves group fairness at the same time.

1. Introduction

Machine learning (ML) has developed in recent years, with many applications progressively transforming our life, such as speech recognition and recommendation systems. The great success of ML is attributed to the large amount of data collected from users. However, users are growing more concerned about their privacy as data leaks and privacy violations occur frequently. Additionally, some nations have enacted strict data regulation laws, such as the European General Data Protection Regulation (GDPR), making it more challenging to apply traditional centralized ML.

In response, a distributed ML framework named federated learning (FL) was proposed in 2016 (McMahan et al., 2017) to alleviate the conflict between user privacy and machine learning development, after which plenty of studies have made this technology more mature. FL enables many clients (e.g., mobile devices) to collaboratively train a model under the coordination of a central server without sharing their local raw data with the server. Following is a typical process of FL training: (1) The server selects some clients to participate in one training round; (2) the selected clients download the current global model from the server and train it with local data before uploading model updates; (3) the server collects all updates from the selected clients and computes the new global model weights using a method such as weighted average. FL has already been used in many real-world situations, such as enhancing Gboard's search suggestion quality (Yang et al., 2018).

Since ML has gradually been used to make high-risk decisions, its fairness is another security issue that has received widespread attention. Although it may seem logical that a model would make fair decisions without human bias, due to historical bias in training data (Suresh and Guttag, 2019) and some other sources of discrimination, the model trained on user-generated data would, in some cases harm the interests of minority groups. For instance, ProPublica reported that COMPAS, a widely used algorithm in American justice, would statistically punish blacks more severely than whites (Angwin et al., 2016). To mitigate human-induced bias in ML systems, researchers have proposed various mathematical definitions of fairness and corresponding methods to train ML models that satisfy these definitions. Current studies mainly focus on group fairness and individual fairness. The former aims to detect and mitigate bias and perform equally between different groups with respect to a sensitive attribute (Hardt et al., 2016; Kamiran and Calders, 2011), such as gender or race, and is more popular because it is easier to define and analyze. However, some works (Kleinberg et al., 2017) have proven that different definitions of group fairness are incompatible with one another. Moreover,

* Corresponding author.
E-mail addresses: leejie@cug.edu.cn (J. Li), tianqing.zhu@ieee.org (T. Zhu), weirencs@cug.edu.cn (W. Ren), raymond.choo@fulbrightmail.org (K.-K. Raymond).

https://doi.org/10.1016/j.cose.2023.103336
0167-4048/

these definitions have faced criticism for overlooking historical and structural factors that cause social unfairness and perpetuate existing biases. Individual fairness requires that similar samples should be treated similarly (Dwork et al., 2012), in which the similarity of samples is intractable to design and formalize, making it difficult to conduct follow-up studies. Fortunately, many recent works have investigated how to learn such similarity metrics from training data (Ilvento, 2020; Mukherjee et al., 2020; Yurochkin et al., 2020). Based on these current breakthroughs, this paper considers enforcing individual fairness. And surprisingly, experimental results show that our method can improve not only individual fairness metrics but also group fairness metrics. In contrast, previous methods of enforcing group fairness do not help enhance individual fairness.

As a variant of ML, FL suffers from the same bias issue. Moreover, the decentralized structure of FL may introduce some additional biases. Approaches that work well in centralized ML cannot be directly applied to FL because they always presuppose full access to the entire training dataset. Several existing studies that have noticed fairness issues in FL only concentrate on group fairness (Abay et al., 2020; Cui et al., 2021; Du et al., 2021; Zhang et al., 2020), ignoring individual fairness. To fill this gap, we aim to train an individually fair FL model in this work. First, to define the similarity metric of individual fairness, we extend the idea of learning from training data in ML (Yurochkin et al., 2020) to FL while keeping the training data decentralized. After that, we consider examples that violate the definition of individual fairness as adversarial attack examples and then convert the task of training an individually fair model into an adversarial training task on each client side. We evaluate this method on two datasets and compare it with baselines and other methods. The results show that our method can significantly improve model fairness without compromising accuracy and privacy. The main contributions of this paper are:

- We present an adversarial training approach for training FL models that satisfy individual fairness without requiring direct access to training data or the underlying data distribution, thereby protecting user privacy at the same time.
- We show that our approach can guarantee fairness on the global model as well as on client-side (local) data distribution because we conduct adversarial training on the client-side distributedly.
- We implement and measure the proposed method on two datasets while testing various hyper-parameters, and the experimental results demonstrate its effectiveness. We also investigate the impact of the fairness enhancement algorithm on data privacy by conducting several membership inference attacks.

The remaining part of this paper proceeds as follows. Section 2 briefly reviews the work related to ours. Section 3 provides the essential background information and notion used in this paper. Section 4 presents the detailed individual fairness enhancement methodology in the context of FL. Section 5 analyses the results on different datasets and demonstrates the effectiveness of the proposed method. Finally, Section 6 concludes this paper.

2. Related work

Up to now, few studies have considered fairness in FL. Most of them discuss fairness in resource allocation between clients, i.e., ensuring the global model learns and performs equally on each client side (Li et al., 2020; Mohri et al., 2019). In particular, Mohri et al. (2019) designed an agnostic FL framework that minimizes the loss of the worst-performing client. While the optimization objective in Li et al. (2020) is a weighted loss based on each client's performance, where a client with a larger local empirical loss will be assigned a heavier weight. It is more flexible, nevertheless, both of them did not evaluate the global model's performance on fairness metrics related to sensitive attributes, known as algorithmic fairness. Furthermore, much of the current literature pays particular attention to algorithmic fairness in the FL context (Abay et al., 2020; Cui et al., 2021; Du et al., 2021; Zhang et al., 2020). Cui et al. (2021) put forward a FL framework taking both algorithmic fairness and fairness resource allocation into account and trained such a fair model through reaching Pareto optimality of this multi-objective optimization problem. This provides insights into achieving multiple targets at the same time. Abay et al. (2020) analyzed causes of bias in FL. Some are inherent to traditional centralized ML, while others are introduced by FL's distinctive operations, such as client selection and model aggregation. After that, they proposed several bias mitigation techniques based on reweighing and fairness-aware regularization, demonstrating the feasibility of adapting these methods from centralized ML to FL with some modifications. Considering the challenge of unknown testing distribution, Du et al. (2021) developed a fairness-aware agnostic FL framework, which involved using kernel reweighing functions to reweigh each training sample in the context of the loss function and the fairness constraint. Their work firstly gives attention to fairness-aware FL under the data distribution shift. Zhang et al. (2020) addressed the fairness issue in FL with a principled deep multi-agent reinforcement learning framework that determines whether a client is selected to participate in this training round based on the fairness performance of the global model. They leverage the particular structure of FL to benefit it. However, all of these works remain narrow in focus dealing only with group fairness and lacking knowledge about individual fairness.

Although there has been more research on group fairness, its shortcomings cannot be overlooked. As proven in Kleinberg et al. (2017), different definitions of group fairness are incompatible. Additionally, even if a model achieves group fairness, it may still be unfair from the perspective of individual users. And for this reason, the notion of individual fairness was considered. Dwork et al. (2012) proposed the definition of individual fairness in 2011, which states that similar samples should be treated similarly. Due to the difficulty of determining an appropriate metric function to quantify the similarity of two inputs (also named fair metric), few studies on individual fairness, and many of them tried to avoid the original definition. Some broaden the concept of "individual" to an example with multiple attribute labels (Hébert-Johnson et al., 2018; Kearns et al., 2018; 2019), while some assume an individual fairness violations detector to circumvent a fair metric (Kim et al., 2018; Sharifi-Malvajerdi et al., 2019). Thankfully, recent research has investigated the feasibility of learning fair metrics from training data (Ilvento, 2020; Yurochkin et al., 2020). Moreover, a few works based on this have achieved promising performance (Vargo et al., 2021; Yurochkin et al., 2020; Yurochkin and Sun, 2021). Our work follows their steps. Unfortunately, all the aforementioned studies focus on centralized ML, which cannot be directly applied to the FL setting due to raw data sharing.

Our approach to fair training is also analogous to adversarial training (Goodfellow et al., 2015; Madry et al., 2018), which enhances the robustness of ML models against adversarial attacks. The most similar to our work is Sinha et al. (2018), in which they defend adversarial attacks through the principled lens of distributionally robust optimization(DRO).

3. Preliminary

In this section, we will provide some necessary background information and notation used in our work.

Federated learning In a conventional FL setting, many clients with local private data and computational resources work together to train a global model under the coordination of a central server (Kairouz et al., 2021). Suppose there are K clients and each client has a dataset $D_k \triangleq \{(\boldsymbol{x}_i^k, y_i^k) \mid \boldsymbol{x}_i^k \in \mathcal{X}, y_i^k \in \mathcal{Y}\}$ of size N_k, where $\mathcal{X}$ and $\mathcal{Y}$ are inputs spaces and outputs spaces respectively. The goal of FL is to search a shared parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ to solve:

$$\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) \triangleq \sum_{k=1}^{K} p_k F_k(\boldsymbol{\theta}), \tag{1}$$

where F is a objective function and $p_k = \frac{N_k}{\sum_{k=1}^{K} N_k}$. F_k, the local optimization objective of client k, is typically an empirical risk function over D_k, i.e. $F_k(\boldsymbol{\theta}) \triangleq \frac{1}{N_k} \sum_{i=1}^{N_k} \ell(h(\boldsymbol{x}_i^k), y_i^k)$, where hypothesis function $h : \mathcal{X} \to \mathcal{Y}$ is a model parameterized by $\boldsymbol{\theta}$ and $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a loss function. To achieve this goal, McMahan et al. (2017) proposed FedAvg, which has become one of the most popular algorithms. Algorithm 1 provides the details of the algorithm. First, the server randomly selects m clients as a set S_t to participate in each global training round t. The most up-to-date global model θ_t is then sent to clients in set S_t, where it can be updated locally using each user's datasets D_k through E steps of stochastic gradient descent (SGD). After that, the server gathers all the updated models $\boldsymbol{\theta}_{t+1}^k$ and uses a weighted algorithm to compute a new global model $\boldsymbol{\theta}_{t+1}$.

Algorithm 1 ***FedAvg***. K clients indexed by k, training datasets D_k of size N_k, local minibatch size B, the number of local epochs E, global training round T, learning rate η, client fraction p.

Server:
  initialize $\boldsymbol{\theta}_0$
  for each round $t = 1, 2, \ldots, T$ **do**
    $m \leftarrow \lfloor p \cdot K \rfloor$
    $S_t \leftarrow$ (selected randomly set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $\boldsymbol{\theta}_{t+1}^k \leftarrow \text{ClientUpdate}(k, \boldsymbol{\theta}_t)$
    $\boldsymbol{\theta}_{t+1} \leftarrow \sum_{k=1}^{K} p_k \boldsymbol{\theta}_{t+1}^k$

ClientUpdate($k, \boldsymbol{\theta}$): // *Run on client k*
  $iter \leftarrow \lfloor N_k / B \rfloor$
  for each local epoch i from 1 to E **do**
    for each iteration j from 1 to $iter$ **do**
      $b \leftarrow$minibatch of size B sampled from D_k
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla \ell(\boldsymbol{\theta}; b)$
  return $\boldsymbol{\theta}$

Individual fairness The key concept of individual fairness is that two similar samples should be treated similarly. Mathematically, for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X} \subseteq \mathbb{R}^d$, the definition of individual fairness (Dwork et al., 2012) is:

$$d_y(h(\boldsymbol{x}_1), h(\boldsymbol{x}_2)) \leq L \cdot d_x(\boldsymbol{x}_1, \boldsymbol{x}_2), \tag{2}$$

where d_x and d_y are metrics on spaces of inputs and outputs and $L \geq 0$ is a Lipschitz constant. According to this mathematical formula, a model h that satisfies individual fairness should also have similar predictions on two inputs with a high degree of similarity. On the other hand, $d_y(\cdot)$ can be large if $d_x(\cdot)$ is large.

Adversarial training Some machine learning models are vulnerable to adversarial attacks (Goodfellow et al., 2015), which degrade model accuracy by adding small but deliberately worst-case perturbations to examples from the dataset. For this reason, adversarial training is proposed to train models that are more resistant to such adversarial attacks (Goodfellow et al., 2015). This paper follows the procedure of Sinha et al. (2018) and conducts adversarial training through the perspective of distributionally robust optimization. Thus, the objective of adversarial training can be formulated as follows:

$$\min_{\boldsymbol{\theta}} \sup_{P \in \mathcal{P}} \mathbb{E}_P[\ell(\boldsymbol{\theta}; \boldsymbol{z})], \tag{3}$$

where $\boldsymbol{z} \in \mathcal{Z}$, $\mathcal{Z} \triangleq \mathcal{X} \times \mathcal{Y}$, $\mathcal{P}$ is a collection of distributions around the data-generating distribution P_0. To solve this min-max problem, we need to find a distribution P that generates adversarial examples with the highest expected loss. After that, the adversarial examples are used to optimize the current value of θ. This allows us to obtain a more robust model.

4. Fairness enhancement methodology

In this section, we aim to improve individual fairness in the FL setting via adversarial training. The challenge in implementing individual fairness is to find an appropriate fair metric d_x. And different from traditional machine learning, data are distributed on clients separately in FL so that an overview of all data is not available, making it harder to define d_x and conduct adversarial training. So in the rest of this section, we will first introduce the way to determine a fair metric extended from previous work (Mukherjee et al., 2020) on machine learning. Afterward, we will present how to adversarially train an individually fair model based on the fair metric while strictly complying with the privacy restrictions of FL.

4.1. Fair metric definition

For the sake of simplicity, we will only consider the binary classification task here, i.e., $\mathcal{Y} = \{0, 1\}$. It is easy to define d_y as the outputs are discrete values. However, it is hard to define the fair metric d_x because the inputs are usually high-dimension data and different input attributes represent different significance. So we extend the method proposed in centralized ML (Mukherjee et al., 2020) to FL, which learns a sensitive subspace that encodes all sensitive information of training dataset and treats samples that differ only on the sensitive subspace as similar. Suppose that we already have a sensitive subspace A, P is the projection matrix of A. What we need when measuring the similarity of samples is other information that is not correlated with sensitive attributes. So if we project samples onto the orthogonal complement space of A, i.e., removing all the sensitive information in samples, they can be evaluated by general numerical calculation methods such as p-norms. Therefore, the fair distance $d_x(\boldsymbol{x}_1, \boldsymbol{x}_2)$ between $\boldsymbol{x}_1, \boldsymbol{x}_2$ can be defined as the Euclidean distance of $\boldsymbol{x}_1, \boldsymbol{x}_2$ projected to the orthogonal complement space of A. Formally,

$$d_x(\boldsymbol{x}_1, \boldsymbol{x}_2) \triangleq \|(I - P) \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2)\|_2. \tag{4}$$

$d_x(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is 0 iff $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are different only in sensitive subspace. This fair metric encodes the intuition of similar samples as similarity in demographic information not relevant to sensitive information. Now the key point is how to define the sensitive subspace. We first review how to learn such sensitive subspace A in ML. It is intuitive to conceive that sensitive attribute vectors such as race and gender should be included in A. That is not nearly enough, as other insensitive attributes may be relevant to sensitive attributes. For example, there is some relation between residential zip code and race. Naively ignoring implicit sensitive information will continue to result in bias in data. So in Mukherjee et al. (2020), researchers have proposed to learn the relation w_s through predicting sensitive attributes by other attributes. However, data are distributed among clients and cannot be shared in the standard FL setting. To solve this problem, we propose to learn the w_s in the FL context. The detailed steps are shown in Algorithm 2.

Algorithm 2 ***FirstFedAvg***. K clients indexed by k, training datasets $\{(\boldsymbol{x}_i'^k, \boldsymbol{x}_i^k[s])\}_{i=1}^{N_k}$ of size N_k, local minibatch size B, the number of local epochs E, global training round T, learning rate η, client fraction p.

Server:
initialize $\boldsymbol{\theta}_0$
for each round $t = 1, 2, \ldots, T$ **do**
  $m \leftarrow \lfloor p \cdot K \rfloor$
  $S_t \leftarrow$ (selected randomly set of m clients)
  for each client $k \in S_t$ **in parallel do**
    $\boldsymbol{\theta}_{t+1}^k \leftarrow \text{ClientUpdate}(k, \boldsymbol{\theta}_t)$
  $\boldsymbol{\theta}_{t+1} \leftarrow \sum_{k=1}^{K} p_k \boldsymbol{\theta}_{t+1}^k$

ClientUpdate$(k, \boldsymbol{\theta})$: // *Run on client k*
  $iter \leftarrow \lfloor N_k/B \rfloor$
  for each local epoch i from 1 to E **do**
    for each iteration j from 1 to $iter$ **do**
      $b \leftarrow$minibatch of size B sampled from $\{(\boldsymbol{x}'^k_i, \boldsymbol{x}_i^k[s])\}_{i=1}^{N_k}$ // *reconstructed dataset for learning* d_x
      $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla \ell(\boldsymbol{\theta}; b)$
  return $\boldsymbol{\theta}$

Specifically, the sensitive directions can be obtained as follows. As defined previously, there are K clients and each has a dataset D_k. Firstly, client k reconstruct a dataset $\{(\boldsymbol{x}_i'^k, \boldsymbol{x}_i^k[s])\}_{i=1}^{N_k}$ on the basis of the original dataset $\{\boldsymbol{x}_i^k, y_i^k\}_{i=1}^{N_k}$, where s is the coordinate of predefined sensitive attribute variable(s) S and $\boldsymbol{x}_i'^k$ is $\boldsymbol{x}_i^k[s] = 0$. The examples' true labels of the constructed dataset are the sensitive attribute values $\boldsymbol{x}_i^k[s]$. And the inputs $\boldsymbol{x}_i'^k$ are the same as the original inputs $\boldsymbol{x}_i$ but in which the sensitive attributes S are all 0, which means removing sensitive attributes of training data. Then we build a learning task on the reconstructed dataset with FedAvg McMahan et al. (2017), aiming to predict the sensitive attribute value of an example through other attributes. What we need is the model's weight w_S when the prediction model reaches the expected accuracy. Thus, the sensitive directions can be defined as $\{w_S, e_S\}$, where e_S is a unit vector with 1 in the sensitive attribute coordinate. This vector reflects the relationship between all attributes and sensitive attributes and their impact on sensitive attributes. Therefore, the differences between two original examples in the sensitive direction could be regarded as differences in sensitive attribute values. Two examples that differ only in the sensitive attribute value(such as gender, race, et al.) should be predicted to be the same by an individually fair model. That is, they are still viewed as similar samples. So the changes of the two examples along the sensitive directions should be ignored by the fair metric d_x.

4.2. Fair training process

As mentioned earlier, models that satisfy individual fairness will have similar outputs on similar inputs. In other words, models disobey individual fairness if their outputs on similar inputs are vastly different. Inspired by Yurochkin et al. (2020), we view such fairness violations as adversarial attacks. In adversarial attacks, an adversary would construct adversarial inputs that differ only slightly from real inputs in training data, just like similar pairwise inputs on the fair metric. And the performance of a vanilla model on those adversarial inputs and corresponding real inputs are disparate, which is analogous to the performance of an unfair model on similar inputs. So methods in defending against adversarial attack can be adopted to eliminate unfairness, that is, adversarial training. To train an individually fair model, we imagine an adversary trying to attack a model on its unfairness following (Yurochkin et al., 2020). A model that can defend against such an attack is considered fair. This is a variation of adversarial learning. In general adversarial learning, adversaries aim to attack models on their accuracy, and defenders focus on protecting against accuracy degradation. Instead, adversarial examples here are designed to elicit unfairness, and training based on those examples is designed to obtain a fairer model.

The same problem arises again due to the particular structure of FL. Adversarial training in centralized ML involves only one model, while there are several different local models in a training round in FL. Such problem had been studied in Shah et al. (2021). It turned out that adversarial training still works when distributionally executed on the client side with properly chosen local training epochs. Therefore, all the adversarial training procedures in our work are performed on the client side. Moreover, the choice of clients' local training epochs will be discussed in the next section.

In our hypothetical attack environment, the adversary collects a set of adversarial examples (we call them fair adversarial samples) similar to training examples, but model outputs on them are converse. For example, to attack an unfair resume screening system, the adversary may collect a stack of resumes and change the names on the resumes of Caucasian applicants to names more common among the African-American population. And if the system performs worse on the edited resumes, we treat this as a successful attack, which implies that the system is unfair.

To defend against such unfair attacks, we adopt an adversarial training method on the client side to collaboratively train an individually fair federated model. First of all, we generate a lot of fair adversarial examples $\boldsymbol{x}_i'$ on the selected clients' side, which are similar to their original local training examples $\boldsymbol{x}_i$(that is, $d_x(\boldsymbol{x}_i, \boldsymbol{x}_i')$ is small) but will make models predict wrongly. Next, we will locally train models to circumvent unfairness attacks to satisfy individual fairness before model aggregation. The procedure above is an instance of distributionally robust optimization(DRO). Thus, techniques taken from DRO can be applied here (Sinha et al., 2018).

4.2.1. Fair adversarial samples generation

In order to generate synthetic samples similar to the original samples on client k, we intend to capture a distribution P^k, a distribution on space $\mathcal{Z}$, approximating its data-generating distribution P_0^k with respect to the fair metric d_x. In other words, samples sampling from P^k and P_0^k are similar under metric d_x. Wasserstein distance is a suitable metric for distribution and has been proven effective in Yurochkin et al. (2020). Following them, a fair Wasserstein distance between distribution P^k on space $\mathcal{Z}$ and the data-generating distribution P_0^k is defined as:

$$W_c(P^k, P_0^k) \triangleq \inf_{\gamma \in \Pi[P^k, P_0^k]} \int_{\mathcal{Z}\times\mathcal{Z}} c(\boldsymbol{z}_1, \boldsymbol{z}_2) d\gamma(\boldsymbol{z}_1, \boldsymbol{z}_2), \tag{5}$$

where transportation cost function $c : \mathcal{Z} \times \mathcal{Z} \to [0, +\infty)$ is

$$c((\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2)) \triangleq d_x^2(\boldsymbol{x}_1, \boldsymbol{x}_2) + \infty \cdot 1\{y_1 \neq y_2\}. \tag{6}$$

$c(\boldsymbol{z}_1, \boldsymbol{z}_2)$ denotes the cost consumed to change an original sample to a corresponding fair adversarial sample. This cost function transmits our intuition and definition of similar samples to the fair Wasserstein distance, i.e., if two distributions are relatively similar under the fair metric, the fair Wasserstein distance between them will also be smaller. On the contrary, a larger fair Wasserstein distance indicates a larger gap between the two distributions. Those samples generated by such distribution are no longer similar to training samples, so $W_c(P^k, P_0^k)$ needs to be limited to restrict the fair adversarial samples' search spaces.

Then we need to search for the distribution within a small fair Wasserstein distance ρ that will make the current model perform

worst, revealing the current model's unfairness. Hence the optimization problem that needs to be solved is:

$$\sup_{P^k: W_c(P^k, P_0^k) \le \rho} \mathbb{E}_{P^k}[\ell(\boldsymbol{\theta}; \boldsymbol{z})]. \tag{7}$$

Equation (7) formulates our objectives: (1) finding distributions around the data-generating distribution that are similar to the data-generating distribution, and (2) increasing the empirical risk of the model. Achieving these two objectives leads to a successful fair attack. The finally searched distribution P^k reveals that the model has different performance on similar samples, indicating that the model violates individual fairness. In the FL setting, the fair attack is performed simultaneously on the selected clients during one training round. Each selected client conducts an adversarial attack locally on the downloaded global model without knowledge of each other's information.

Since the optimization objective (7) is an infinite-dimensional optimization problem, we solve it with the help of its dual form. At the same time, since the data-generating distribution P_0^k is unknown, it can be replaced by the empirical data distribution P_n^k. Blanchet and Murthy (2019) proved that the dual form of objective (7) is:

$$\sup_{P^k: W_c(P^k, P_n^k) \le \rho} \mathbb{E}_{P^k}[\ell(\boldsymbol{\theta}; \boldsymbol{z})] = \inf_{\lambda \ge 0} \{\lambda \cdot \rho + \mathbb{E}_{P_n^k}[\phi_\lambda(\boldsymbol{\theta}; \boldsymbol{z})]\}. \tag{8}$$

$$\phi_\lambda(\boldsymbol{\theta}; \boldsymbol{z}) \triangleq \sup_{\boldsymbol{x} \in \mathcal{X}} \ell((\boldsymbol{x}, y_i), \boldsymbol{\theta}) - \lambda d_x(\boldsymbol{x}, \boldsymbol{x}_i). \tag{9}$$

Actually, Eq. (8) is a dual transform, making a supremum problem into an infimum problem. And the transformation (9) holds for any $\lambda \ge 0$. The right part of the formula (8) can also be interpreted from the perspective of Lagrangian relaxation (Sinha et al., 2018), in which $\lambda \ge 0$ is a fixed penalty parameter, and this will not break the Eq. (8) but make it more solvable. Therefore, the unsolvable problem is transformed into a univariate optimization problem so that it can be solved by stochastic optimization. Algorithm 3 describes the stochastic approximation algorithm to solving Eq. (8).

Algorithm 3 Stochastic gradient method to solve Eq. (8) on client k. start point λ_1, step size $\alpha > 0$, client k and its training dataset D_k.

repeat
  sample a batch $(\boldsymbol{x}_{t_1}, y_{t_1}), \ldots, (\boldsymbol{x}_{t_B}, y_{t_B}) \sim P_n^k$
  $\boldsymbol{x}_{t_b}^* \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{X}} (\ell(\boldsymbol{\theta}; (\boldsymbol{x}, y_{t_b}) - \lambda_t d_x(\boldsymbol{x}_{t_b}, \boldsymbol{x})), b \in [B]$
  $\lambda_{t+1} \leftarrow \max\{0, \lambda_t - \alpha \cdot (\rho - \frac{1}{B} \sum_{b=1}^{B} d_x(\boldsymbol{x}_{t_b}, \boldsymbol{x}_{t_b}^*))\}$
until converged

4.2.2. Fair training

As mentioned previously, we treat the task of training individually fair models as an adversarial training task on the client side. Therefore, models robust to sensitive perturbations that satisfy individual fairness are obtained through adversarial training. The above step has captured a distribution P^k and generated fair adversarial examples. And now, according to the general adversarial training process, clients then need to optimize the model based on the fair adversarial examples, i.e., the optimization objective of clients is a min-max problem:

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \sup_{P^k: W_c(P^k, P_n^k) \le \rho} \mathbb{E}_{P^k}[\ell(\boldsymbol{\theta}; \boldsymbol{z})] = \min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \inf_{\lambda \ge 0} \{\lambda \cdot \rho + \mathbb{E}_{P_n^k}[\phi_\lambda(\boldsymbol{\theta}; \boldsymbol{z})]\}. \tag{10}$$

This is a distributionally robust optimization problem that inherits some statistical properties of distributionally robust optimization. Optimizing the worst-case performance of the model on a collection of generated fair adversarial attack examples perturbed in sensitive directions minimizes Eq. (10) and ensures that the model performs well on all data distributions. In other words, the model will be fairer after eliminating all individual fairness violation cases. During one training round, the on-training clients in the training process initially perform adversarial attacks to simulate the worst-case scenario, and subsequently optimize their models locally to achieve distributional robustness. Notwithstanding the aggregation of different upload models, its utility remains effective until the end of the training process.

Recalling the Definition 2, the individual fairness defined by Dwork et al. (2012) requires that the predicted values of the model about two similar examples are similar. Whereas our individually fair model obtained by solving Eq. (10) would differ from it slightly. One fair adversarial example $\boldsymbol{x}$ is similar to the original example $\boldsymbol{x}_i$, but its predicted value $h(\boldsymbol{x})$ differs significantly from the true label y_i. Therefore when optimizing the model based on the fair adversarial example, the predicted value of the model on the fair adversarial example will be as close as possible to y_i, rather than to the predicted value $h(\boldsymbol{x}_i)$. Thus, our method could satisfy individual fairness while taking into account the accuracy of the final model. But overall, the goals of these two definitions of individual fairness are the same. These differences have been discussed in more detail in Yurochkin et al. (2020), in which they named it distributionally robustly fair.

We solve Eq. (10) based on the adversarial training algorithm proposed by Yurochkin et al. (2020) in each client side. Its whole training procedure is shown in Algorithm 4. One challenge here is to incorporate Algorithm 3 into the FL framework. Specifically, all clients share the same fair metric d_x and maintain a local variable λ_k. During the training procedure on clients, each client firstly samples a batch from its local training dataset. There is no requirement for batch size. Even when the batch size is 1, this algorithm still works well. Next, the adversarial samples that reveal the unfairness of the model will be found and then used to train the model. Finally, the algorithm updates λ_k based on adversarial and original samples. As we can see, there are no additional requirements for these processes, but the same as the FL procedure.

5. Experiments and analysis

In this section, we will apply our method to two datasets commonly evaluated in the fairness literature, including income prediction and recidivism prediction. All experiments are simulated on LEAF (Caldas et al., 2018), a benchmarking framework for FL setting.

5.1. Experimental setup

Evaluation metrics To assess individual fairness, we adopt consistency metrics as Yurochkin et al. (2020). As discussed in the previous section, an individually unfair model would discriminate two examples that are identical but differ only in sensitive attributes into different categories. So in the consistency metrics, we first replace the sensitive attribute value of each example in the data set with its opposite value (e.g., change the gender attribute of the example from male to female and female to male). And consistency value is the similarity of the model's predictions on the original and artificially modified datasets. Higher consistency implies that more similar examples could be treated similarly. In other words, we have trained an individually fairer model. Mathematically, for sensitive attribute S, the consistency value S-Cons. can be calculated by:

$$\begin{aligned} &\hat{Y}_0 = h(\boldsymbol{x}_{S=0}), \ \hat{Y}_1 = h(\boldsymbol{x}_{S=1}), \\ &\text{S-Con.} \triangleq \Pr\{\hat{Y}_0 = \hat{Y}_1\}, \end{aligned} \tag{11}$$

Algorithm 4 ***OurMethod***. K clients indexed by k, training datasets D_k of size N_k, local minibatch size B, the number of local epochs E, global training round T, learning rate η, client fraction p.

Server:
initialize $\boldsymbol{\theta}_0$
for each round $t = 1, 2, \ldots, T$ **do**
  $m \leftarrow \lfloor p \cdot K \rfloor$
  $S_t \leftarrow$ (selected randomly set of m clients)
  $w_S \leftarrow$ FirstFedAvg // *Algorithm* 2
  $d_x \leftarrow \{w_S, e_S\}$
  for each client $k \in S_t$ **in parallel do**
    $\boldsymbol{\theta}_{t+1}^k \leftarrow \text{ClientUpdate}(k, \boldsymbol{\theta}_t, d_x)$
  $\boldsymbol{\theta}_{t+1} \leftarrow \sum_{k=1}^{K} p_k \boldsymbol{\theta}_{t+1}^k$

ClientUpdate($k, \boldsymbol{\theta}, d_x$): // *Run on client k*
initialize λ_1^k
$iter \leftarrow \lfloor N_k / B \rfloor$
for each local epoch i from 1 to E **do**
  for each iteration j from 1 to $iter$ **do**
    $(\boldsymbol{x}_{t_1}, y_{t_1}), \ldots, (\boldsymbol{x}_{t_B}, y_{t_B}) \leftarrow$ minibatch of size B sampled from D_k
    $\boldsymbol{x}_{t_b}^* \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{X}} (\ell(\boldsymbol{\theta}; (\boldsymbol{x}, y_{t_b}) - \lambda_t^k d_x(\boldsymbol{x}_{t_b}, \boldsymbol{x})), b \in [B]$
    $\lambda_{t+1}^k \leftarrow \max\{0, \lambda_t^k - \alpha^k \cdot (\rho - \frac{1}{B} \sum_{b=1}^{B} d_x(\boldsymbol{x}_{t_b}, \boldsymbol{x}_{t_b}^*))\}$ // *Algorithm* 3
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla \ell(\boldsymbol{\theta}; \boldsymbol{x}_{t_b}^*)$
return $\boldsymbol{\theta}$

in which $h(\boldsymbol{x}_{S=0})$ represents input $\boldsymbol{x}$ with sensitive attribute S is 0. A consistency value equal to 1 indicates exact fairness. Meanwhile, we recorded some measures used in assessing group fairness for completeness. Following De-Arteaga et al. (2019), we compute the differences between the true positive rate (D_{TPR}) and the true negative rate (D_{TNP}) between different classes of a sensitive attribute S, and report the maximum value (GAP_{MAX}^S) and the root-mean-squared value (GAP_{RMS}^S) of them. The formula is shown below:

$$\begin{aligned} D_{TPR} &\triangleq |\Pr\{\hat{Y} = 1 \mid Y = 1, S = 1\} - \Pr\{\hat{Y} = 1 \mid Y = 1, S = 0\}|, \\ D_{TNR} &\triangleq |\Pr\{\hat{Y} = 0 \mid Y = 0, S = 1\} - \Pr\{\hat{Y} = 0 \mid Y = 0, S = 0\}|, \\ GAP_{MAX}^S &= \max(D_{TPR}, D_{TNR}), \\ GAP_{RMS}^S &= \sqrt{\tfrac{(D_{TPR}^2 + D_{TNR}^2)}{2}}. \end{aligned} \tag{12}$$

$\hat{Y}$ represents the predicted class. The smaller these two metrics entail that the model would treat different groups more equally and be more aligned with group fairness. The model is precisely fair to different groups when $D_{TPR} = 0$ and $D_{TNP} = 0$. We chose this group fairness metric because it not only captures the most commonly used notion Equalized Odds (Hardt et al., 2016) in fairness research but also provides more information about model performance on fairness. Besides, due to class imbalance in the dataset, we present balanced accuracy (B-acc.) instead of general accuracy. Its definition is:

$$\text{B-acc.} \triangleq \frac{\Pr\{\hat{Y} = 1 \mid Y = 1\} + \Pr\{\hat{Y} = 0 \mid Y = 0\}}{2}. \tag{13}$$

Comparison experiments With the model structure and hyperparameters unchanged, we apply the vanilla FL algorithm, FedAvg (McMahan et al., 2017), to all training tasks as baseline results. We also conduct experiments without fairness constraints on centralized ML in order to observe similarities and dissimilarities between ML and FL. As mentioned before, existing state-of-the-art techniques in the literature mainly consider fairness in ML or group fairness in FL. For this reason, we can only compare our method with group fairness-enhancing strategies in FL and post-processing techniques in ML that require no need to adjust the training data and no access to the training procedure. Therefore, we adopt three solutions as our baselines: (a) Project (Yurochkin et al., 2020). They eliminate the impact of sensitive attributes through projecting training data to the orthogonal complement space of sensitive subspace; (b) Local Reweighing (Abay et al., 2020). Reweighing (Kamiran and Calders, 2011) is a pre-processing technique used in ML that balances different groups by assigning weights calculated from group statistics information to the training data in that group. In local reweighing, all clients compute the weights of their data locally and privately before training without breaking privacy constraints; (c) Orthogonal Classifier (Xu et al., 2022). It is a new and simple but effective ML post-processing method. Given a well-trained model, it can find an orthogonal model that makes predictions regardless of certain attributes, such as sensitive attributes for fairness goals. Formally, the classifier orthogonalization procedure performs as follows:

$$\begin{aligned} w_1(\boldsymbol{x})_y &= \Pr(Y = y \mid S = \boldsymbol{x}[s]), \\ w_x(\boldsymbol{x})_y &= \Pr(Y = y \mid X = \boldsymbol{x}), \\ w_2(\boldsymbol{x})_y &= \Pr(Y = i \mid X' = \boldsymbol{x} \setminus \boldsymbol{x}[s]), \\ w_2(\boldsymbol{x})_y &= \Pr(Y = i) \frac{w_x(\boldsymbol{x})_i}{w_1(\boldsymbol{x})_i} \Big/ \sum_j \left(\Pr(Y = j) \frac{w_x(\boldsymbol{x})_j}{w_1(\boldsymbol{x})_j} \right). \end{aligned} \tag{14}$$

Here, $w_1(\boldsymbol{x})_y$ represents the correlation between sensitive attribute variable(s) S with label Y, and $w_x(\boldsymbol{x})_y$ is the well-trained model with all attributes X. X' is constructed by removing S from X, and $w_2(\boldsymbol{x})_y$ is the post-processed model that is orthogonal to $w_1(\boldsymbol{x})_y$, which can make decisions without being influenced by sensitive attribute(s).

5.2. Evaluation results on adult dataset

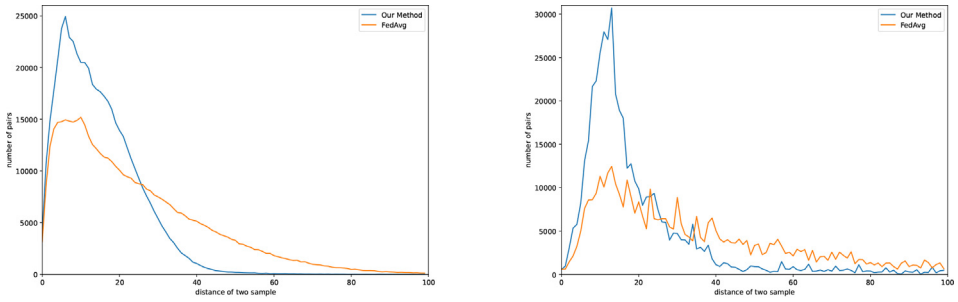
Adult (Dua and Graff, 2017) is a standard benchmark structured dataset in fairness research. The prediction task is to predict whether the income of an individual is over $50k per year or not based on attributes like gender and education. We set two binary attributes, gender (male or female) and race (Caucasian or non-Caucasian), as sensitive attributes. Therefore, the sensitive subspace is span$\{w_g, e_g, e_r\}$. And we measure individual fairness with relationship status consistency (S-cons.) and gender-race consistency (GR-cons.). The relationship status attribute (is_wife or is_husband) is not explicitly protected, but directly implied the gender. Such metrics could demonstrate the generalization properties of the fair metric.

Following the preprocessing of data in Yurochkin et al. (2020), we obtain 45,222 samples, each of which has 41 features. We divide 80% into training sets and distribute these training data to 5 clients. In each training round of the federated setting, the server randomly selects three clients, a total of 80 rounds. The global model is a neural network with one hidden layer. Selected clients train the global model on local data for 100 iterations. Also, due to class imbalance, each iteration will randomly sample 1000 samples, 500 from the positive category and 500 from the negative category.

All the results are shown in Table 1, and the experiment results of Centralized ML are from Yurochkin et al. (2020). The baseline experiments (FedAvg and Centralized ML) reach the highest accuracy. However, it is quite apparent that naively training a model without fairness constraints will result in huge discrimination against gender and race from the perspective of both individuals and groups. On the contrary, our method improves consistency metrics considerably with little accuracy compromise, suggesting the model is more individually fair. Furthermore, the gap between different groups reduces significantly using our method. That is to say, improving individual fairness contributes to improving group fairness. Nevertheless, the group fairness improve-

Table 1
Adult: average results on 10 random 80%/20% train/test splits. The B-Acc., S-Con., and GR-cons. are better if they are higher, while all group fairness metrics are better if they are lower.

		Individual fairness		Group fairness			
Method	B-Acc.↑	S-Cons.↑	GR-cons.↑	$Gap^G_{RMS}\downarrow$	$Gap^G_{MAX}\downarrow$	$Gap^R_{RMS}\downarrow$	$Gap^R_{MAX}\downarrow$
Our Method	0.793	**0.928**	0.986	**0.066**	0.089	**0.059**	**0.073**
Project (Yurochkin et al., 2020)	0.817	0.863	**1**	0.117	0.155	0.067	0.082
Local Reweighing (Abay et al., 2020)	0.816	0.788	0.906	0.070	**0.085**	0.079	0.092
Orthogonal Classifier (Xu et al., 2022)	0.817	0.836	0.889	0.173	0.184	0.072	0.080
FedAvg (McMahan et al., 2017)	0.828	0.827	0.863	0.173	0.213	0.086	0.103
Centralized ML (Yurochkin et al., 2020)	**0.829**	0.848	0.865	0.179	0.216	0.089	0.105



(a) fair ratio of Adult (b) fair ratio of COMPAS

Fig. 1. Fair ratio distribution graph.

ment technique does not contribute to individual fairness as we compare the results of Local Reweighing and Orthogonal Classifier to FedAvg. The results of our method are even better than Local Reweighing. Though Project produces the best gender and race consistency result, our method outperforms it on the generalization of individually fair performance, which can be illustrated by relationship status consistency.

We plot a fair ratio distribution graph to elaborate on our method's performance in more detail. More specifically, we sample 1000 samples from dataset and calculate the $d_y(h(\boldsymbol{x}_i, \boldsymbol{x}_j))/d_x(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between each pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in the sampled data. For a individually fair model, lower $d_x(\boldsymbol{x}_i, \boldsymbol{x}_j)$ corresponds to lower $d_y(h(\boldsymbol{x}_i, \boldsymbol{x}_j))$ and higher $d_x(\boldsymbol{x}_i, \boldsymbol{x}_j)$ corresponds to higher $d_y(h(\boldsymbol{x}_i, \boldsymbol{x}_j))$. Consequently, the distribution of all $d_y(h(\boldsymbol{x}_i, \boldsymbol{x}_j))/d_x(\boldsymbol{x}_i, \boldsymbol{x}_j)$ should be more concentrated. And as shown in Fig. 1(a), our method produces a more concentrated distribution than FedAvg, meaning we have trained a fairer model on another metric.

Figure 2 reports the evaluation results of global model's performance on client side. There is only one client's performance because of space limitations, while all the results of clients are similar. The results of the global model on clients' local datasets are in line with the whole test set. Thus, such a model can be applied directly to clients to improve individual fairness.

Overall, this experiment provides empirical evidence that our method trains individually fair classifiers while still maintaining high accuracy.

5.3. Evaluation results on COMPAS dataset

COMPAS is another common standard dataset in the fairness literature. This task of COMPAS is to predict whether a criminal defendant would recidivate within two years (Angwin et al., 2016) through their age and other attributes. We still consider race (Caucasian or not-Caucasian) and gender (binary) as protected attributes and use them to learn the sensitive subspace span$\{w_r, e_g, e_r\}$ similarly to previous experiments. The individual fairness measures are gender consistency (G-cons.) and race consistency (R-cons.).

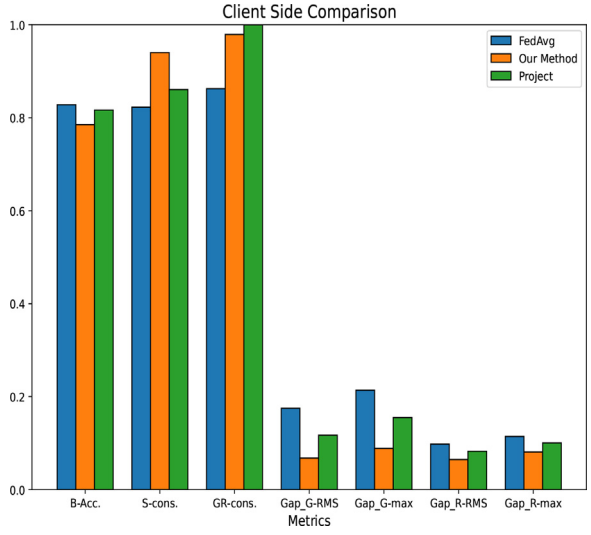


Fig. 2. The final model performed well on one client side's data too.

Following the preprocessing in Vargo et al. (2021), we obtain 5278 samples, each of which has seven features. We divided 80% into training sets and distributed these training data to 3 clients due to the small dataset size. In each training round, the server randomly selects two clients, totaling 80 rounds. The global model is a neural network with one hidden layer. Selected clients train the global model on local data for 100 iterations. Each iteration will randomly sample 400 samples, where 200 are from the positive and 200 are from the negative classes.

The results are reported in Table 2. Similarly, simply applying FedAvg to COMPAS leads to undesired biases towards gender and race. With our method, the trained model performs significantly better both on individual and group fairness without a noticeable decrease in accuracy. In addition, the fair ratio distribution graph of COMPAS (Fig. 1(b)) again demonstrates that our method could train an individually fairer model.

Table 2
COMPAS: average results on 10 random 80%/20% train/test splits. The B-Acc., S-Con., and GR-cons. are better if they are higher, while all group fairness metrics are better if they are lower.

		Individual fairness		Group fairness			
Method	B-Acc.↑	G-cons.↑	R-cons.↑	Gap^G_{RMS} ↓	Gap^G_{MAX} ↓	Gap^R_{RMS} ↓	Gap^R_{MAX} ↓
Our Method	0.615	**0.984**	0.979	**0.092**	**0.117**	0.139	0.167
Project (Yurochkin et al., 2020)	0.530	0.874	**1**	0.207	0.232	0.061	0.080
Local Reweighing (Abay et al., 2020)	0.662	0.815	0.827	0.269	0.298	**0.048**	**0.061**
Orthogonal Classifier (Xu et al., 2022)	0.673	0.808	0.964	0.306	0.341	0.202	0.236
FedAvg (McMahan et al., 2017)	0.675	0.826	0.961	0.293	0.327	0.244	0.279
Centralized ML (Vargo et al., 2021)	**0.682**	0.841	0.908	0.246	0.282	0.228	0.258

Table 3
Bank: average results on 10 random 80%/20% train/test splits. The Acc., L-Con. are better if they are higher, while all group fairness metrics are better if they are lower.

		Individual fairness	Group fairness	
Method	Acc.↑	L-Cons.↑	Gap^L_{RMS} ↓	Gap^L_{MAX} ↓
Our Method	0.883	**1.000**	**0.000**	**0.000**
Project (Yurochkin et al., 2020)	0.894	**1.000**	0.162	0.228
Local Reweighing (Abay et al., 2020)	0.900	0.994	0.059	0.082
Orthogonal Classifier (Xu et al., 2022)	0.898	0.976	0.043	0.055
FedAvg (McMahan et al., 2017)	0.899	0.984	0.058	0.080
Centralized ML (Yurochkin et al., 2020)	**0.901**	0.979	0.052	0.071

5.4. Evaluation results on bank dataset

The UCI Bank Marketing dataset (Moro et al., 2014) is a collection of data related to a direct marketing campaign conducted by a Portuguese banking institution, which aims to predict if the client will subscribe to a term deposit. We take whether there is a personal loan as a sensitive attribute and then learn the sensitive subspace span $\{w_l, e_l\}$, and take loan consistency (L-cons.) as individual fairness performance.

Then, we follow Zeng et al. (2021)'s step to preprocess Bank to 45,211 samples with 35 variables and split Bank to 5 clients after dividing 80% into training sets. The global model is a neural network with one hidden layer. During one training round, the server randomly selects three clients, a total of 40 rounds, and selected clients train the global model on local data for 100 iterations, each with 128 samples.

We present the results in Table 3. It is clear that the data confirm the validity of our approach again. While maintaining high accuracy, our method drastically enhances individual fairness as well as group fairness. One point that needs to be mentioned is that we use accuracy here instead of balanced accuracy. This is because classes are far too unbalanced, resulting in low balanced accuracy. Using the accuracy metrics does not impact either.

5.5. The impact of local epoch E

To further investigate the impact of different hyper-parameter choices on the experimental results, we select various local training epochs E and the number of selected clients m. With other hyper-parameters fixed, we evaluate our method with different E. For simplicity, only consistency metrics and accuracy will be shown, and other metrics results show similar patterns. One epoch is one batch here instead of all training data. As shown in Table 4, a larger local training epoch leads to better results in our method, while there is little influence in FedAvg (Baseline). When E is 20, the model is far from optimal in our method but has converged in FedAvg. That suggests that the fair adversarial attack in our method may have a relatively high impact on the model convergence speed. Furthermore, as E gets larger, the model can be trained more adequately and perform better on fairness metrics. It is notable that S-con. decreases slightly when E is 160. From this, we can deduce that too much local fair training may be harmful to the generalization of individual fairness.

Table 4
different results with different local epoch E. The B-Acc., S-Con., and GR-cons. are better if they are higher.

	Our method			FedAvg		
E	B-Acc.↑	S-con.↑	GR-con.↑	B-Acc.↑	S-con.↑	GR-con.↑
20	0.754	0.959	0.969	0.825	0.806	0.877
60	0.782	0.952	0.983	0.827	0.810	0.861
100	0.793	0.928	0.986	0.828	0.827	0.863
160	0.800	0.896	0.987	0.829	0.830	0.863
200	0.803	0.879	0.988	0.828	0.833	0.864

5.6. The impact of selected clients m

In the experiment Adult, there are five clients in all. We evaluate our method by choosing 1 to 5 clients in one training round respectively. All results on different numbers of clients m selected for each training round are presented in Table 5. It is clear that different m lead to negligible differences. This is probably due to the fact that the data between different clients are independently and identically distributed(iid). There needs to be more research done on non-iid data.

5.7. The impact on privacy

Most works aiming to improve fairness have ignored assessing their impact on privacy. For example, are models trained by

Table 5
different results with different m. The B-Acc., S-Con., and GR-cons. are better if they are higher.

	Our method			Baseline/FedAvg		
m	B-Acc.↑	S-con.↑	GR-con.↑	B-Acc.↑	S-con.↑	GR-con.↑
1	0.793	0.926	0.986	0.828	0.824	0.865
2	0.792	0.927	0.986	0.828	0.827	0.862
3	0.793	0.928	0.986	0.828	0.827	0.863
4	0.793	0.927	0.986	0.828	0.829	0.864
5	0.793	0.926	0.986	0.828	0.828	0.864

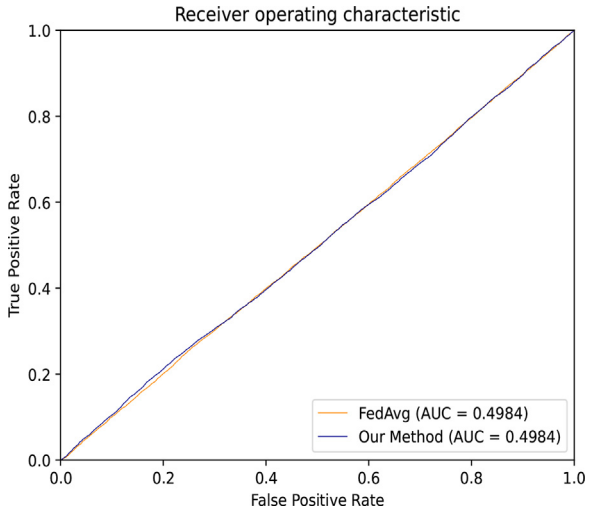


Fig. 3. The receiver operating characteristic and the area under the curve of membership inference attack.

fair algorithm more vulnerable to membership inference attacks than other vanilla models? It is significant and insightful to explore the impact of fairness methods on data privacy deeply. Chang and Shokri (2021) have investigated this topic and concluded that many fair algorithms would affect the privacy of certain data. To find out our method's impact on data privacy, we have executed membership inference attacks on the dataset Adult as an example. In our hypothesis, the attack can access all data and information about the model. Since, with the most powerful adversary, we are able to be aware of our maximum privacy leakage. At first, we have implemented an attack based on metrics (Yeom et al., 2018). More specifically, we collected the trained model's loss on all training and testing data. Usually, the attacker takes the average of the loss of the testing data as a threshold. And for any query sample, if the model's loss on it is less than the threshold, the sample is considered a member. Rather than choose a threshold, it is more accurate to calculate the area under the receiver operating characteristic curve since we have all the loss information. The larger the area, the more likely to infer whether data belongs to the training set, which also implies a greater risk of data privacy. As illustrated in Fig. 3, there is little change in the probability that the model suffers a successful membership inference attack before and after applying the proposed algorithm. We also have conducted another membership inference attack following the classic attacks in Shokri et al. (2017). They used a shadow model because they assumed the attacker could not access the trained model. Instead, we use the trained model as a "shadow model" to simulate the most potent attacker. It draws the same conclusion as above because the attacker's prediction accuracy is around 0.5 with or without the fair algorithm. Those have proved the optimization of our fairness algorithm has no impact on the privacy of the training data.

6. Conclusion

Previous research on bias mitigating has only concentrated on group fairness in FL or centralized ML. To the best of our knowledge, this is the first paper to investigate how to train an individually fair FL model. In this paper, we presented an adversarial training approach to train FL models that satisfy individual fairness and trained it through DRO on the client side. We have evaluated the proposed method on two real-world datasets, and the evaluation results show that it significantly enhances FL model fairness while maintaining accuracy. In addition, we demonstrated that the trained model could guarantee fairness on global and local models, allowing the model to be deployed directly to clients. In the future, we will extend our method to image datasets in an effort to reduce bias in image recognition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Jie Li: Conceptualization, Methodology, Writing – original draft. **Tianqing Zhu:** Writing – review & editing. **Wei Ren:** Data curation, Visualization. **Kim-Kwang Raymond:** Investigation.

Data availability

Data will be made available on request.

References

Abay, A., Zhou, Y., Baracaldo, N., Rajamoni, S., Chuba, E., Ludwig, H., 2020. Mitigating bias in federated learning. CoRR abs/2012.02447.

Angwin, J., Larson, J., Mattu, S., Kirchner, L., 2016. Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And it's Biased Against Blacks. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Blanchet, J.H., Murthy, K.R.A., 2019. Quantifying distributional model risk via optimal transport. Math. Oper. Res. 44 (2), 565–600.

Caldas, S., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., Talwalkar, A., 2018. LEAF: a benchmark for federated settings. CoRR abs/1812.01097.

Chang, H., Shokri, R., 2021. On the privacy risks of algorithmic fairness. In: IEEE European Symposium on Security and Privacy, EuroS&P 2021, Vienna, Austria, September 6–10, 2021. IEEE, pp. 292–303.

Cui, S., Pan, W., Liang, J., Zhang, C., Wang, F., 2021. Addressing algorithmic disparity and performance inconsistency in federated learning. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6–14, 2021, Virtual, pp. 26091–26102.

De-Arteaga, M., Romanov, A., Wallach, H.M., Chayes, J.T., Borgs, C., Chouldechova, A., Geyik, S.C., Kenthapadi, K., Kalai, A.T., 2019. Bias in bios: a case study of semantic representation bias in a high-stakes setting. In: danah boyd, Morgenstern, J.H. (Eds.), Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29–31, 2019, pp. 120–128.

Du, W., Xu, D., Wu, X., Tong, H., 2021. Fairness-aware agnostic federated learning. In: Demeniconi, C., Davidson, I. (Eds.), Proceedings of the 2021 SIAM International Conference on Data Mining, SDM 2021, Virtual Event, April 29, - May 1, 2021, pp. 181–189.

Dua, D., Graff, C., 2017. UCI machine learning repository. http://archive.ics.uci.edu/ml.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.S., 2012. Fairness through awareness. In: Goldwasser, S. (Ed.), Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8–10, 2012, pp. 214–226.

Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings.

Hardt, M., Price, E., Srebro, N., 2016. Equality of opportunity in supervised learning. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5–10, 2016, Barcelona, Spain, pp. 3315–3323.

Hébert-Johnson, Ú., Kim, M.P., Reingold, O., Rothblum, G.N., 2018. Multicalibration: calibration for the (computationally-identifiable) masses. In: Dy, J.G., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018. In: Proceedings of Machine Learning Research, vol. 80, pp. 1944–1953.

Ilvento, C., 2020. Metric learning for individual fairness. In: Roth, A. (Ed.), 1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1–3, 2020, Harvard University, Cambridge, MA, USA (Virtual Conference). In: LIPIcs, vol. 156, pp. 2:1–2:11.

Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K.A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Eichner, H., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W.,

Stich, S.U., Sun, Z., Suresh, A.T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H., Zhao, S., 2021. Advances and open problems in federated learning. Found. Trends Mach. Learn. 14 (1–2), 1–210.
Kamiran, F., Calders, T., 2011. Data preprocessing techniques for classification without discrimination. Knowl. Inf. Syst. 33 (1), 1–33.
Kearns, M.J., Neel, S., Roth, A., Wu, Z.S., 2018. Preventing fairness gerrymandering: auditing and learning for subgroup fairness. In: Dy, J.G., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018. In: Proceedings of Machine Learning Research, vol. 80, pp. 2569–2577.
Kearns, M.J., Neel, S., Roth, A., Wu, Z.S., 2019. An empirical study of rich subgroup fairness for machine learning. In: danah boyd, Morgenstern, J.H. (Eds.), Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29–31, 2019, pp. 100–109.
Kim, M.P., Reingold, O., Rothblum, G.N., 2018. Fairness through computationally-bounded awareness. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada, pp. 4847–4857.
Kleinberg, J.M., Mullainathan, S., Raghavan, M., 2017. Inherent trade-offs in the fair determination of risk scores. In: Papadimitriou, C.H. (Ed.), 8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9–11, 2017, Berkeley, CA, USA. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 43:1–43:23.
Li, T., Sanjabi, M., Beirami, A., Smith, V., 2020. Fair resource allocation in federated learning. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020.
Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30, - May 3, 2018, Conference Track Proceedings.
McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data. In: Singh, A., Zhu, X.J. (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20–22 April 2017, Fort Lauderdale, FL, USA. In: Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282.
Mohri, M., Sivek, G., Suresh, A.T., 2019. Agnostic federated learning. In: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA. In: Proceedings of Machine Learning Research, vol. 97, pp. 4615–4625.
Moro, S., Cortez, P., Rita, P., 2014. A data-driven approach to predict the success of bank telemarketing. Decis. Support Syst. 62, 22–31.
Mukherjee, D., Yurochkin, M., Banerjee, M., Sun, Y., 2020. Two simple ways to learn individual fairness metrics from data. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13–18 July 2020, Virtual Event. In: Proceedings of Machine Learning Research, vol. 119, pp. 7097–7107.
Shah, D., Dube, P., Chakraborty, S., Verma, A., 2021. Adversarial training in communication constrained federated learning. CoRR abs/2103.01319.
Sharifi-Malvajerdi, S., Kearns, M.J., Roth, A., 2019. Average individual fairness: algorithms, generalization and experiments. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada, pp. 8240–8249.
Shokri, R., Stronati, M., Song, C., Shmatikov, V., 2017. Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22–26, 2017. IEEE Computer Society, pp. 3–18. doi:10.1109/SP.2017.41.
Sinha, A., Namkoong, H., Duchi, J.C., 2018. Certifying some distributional robustness with principled adversarial training. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30, - May 3, 2018, Conference Track Proceedings.
Suresh, H., Guttag, J. V., 2019. A framework for understanding unintended consequences of machine learning. CoRR abs/1901.10002.
Vargo, A., Zhang, F., Yurochkin, M., Sun, Y., 2021. Individually fair gradient boosting. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021.
Xu, Y., He, H., Shen, T., Jaakkola, T.S., 2022. Controlling directions orthogonal to a classifier. In: The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022.
Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D., Beaufays, F., 2018. Applied federated learning: improving google keyboard query suggestions. CoRR abs/1812.02903.
Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S., 2018. Privacy risk in machine learning: analyzing the connection to overfitting. In: 31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9–12, 2018. IEEE Computer Society, pp. 268–282. doi:10.1109/CSF.2018.00027.
Yurochkin, M., Bower, A., Sun, Y., 2020. Training individually fair ML models with sensitive subspace robustness. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020.
Yurochkin, M., Sun, Y., 2021. Sensei: sensitive set invariance for enforcing individual fairness. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021.
Zeng, Y., Chen, H., Lee, K., 2021. Improving fairness via federated learning. CoRR abs/2110.15545https://arxiv.org/abs/2110.15545.
Zhang, D.Y., Kou, Z., Wang, D., 2020. Fairfl: a fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In: Wu, X., Jermaine, C., Xiong, L., Hu, X., Kotevska, O., Lu, S., Xu, W., Aluru, S., Zhai, C., Al-Masri, E., Chen, Z., Saltz, J. (Eds.), 2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10–13, 2020, pp. 1051–1060.

Jie Li is currently an undergraduate student at China University of Geosciences, Wuhan, China. Her research interests include privacy preserving, AI security and privacy, and network security.

Tianqing Zhu received her B.Eng. degree and her M.Eng. degree from Wuhan University, China, in 2000 and 2004, respectively. She also holds a Ph.D. in computer science from Deakin University, Australia (2014). She is currently an associate professor at University of Technology Sydney, Australia. Her research interests include privacy preserving, AI security and privacy, and network security.

Wei Ren currently is a Professor at the School of Computer Science, China University of Geosciences (Wuhan), China. He was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, USA in 2007 and 2008, the School of Computer Science, University of Nevada Las Vegas, USA in 2006 and 2007, and the Department of Computer Science, The Hong Kong University of Science and Technology, in 2004 and 2005. He obtained his Ph.D. degree in Computer Science from Huazhong University of Science and Technology, China. He has published more than 70 refereed papers, 1 monograph, and 4 textbooks. He has obtained 10 patents and 5 innovation awards. He is a senior member of the China Computer Federation and a member of IEEE.

Kim-Kwang Raymond Choo received his Ph.D. in Information Security in 2006 from the Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). In 2016, he was named the Cybersecurity Educator of the Year - APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn), and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of the 2019 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing (Middle Career Researcher), 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, Outstanding Associate Editor of 2018 for IEEE Access, British Computer Society's 2019 Wilkes Award Runnerup, 2019 EURASIP Journal on Wireless Communications and Networking (JWCN) Best Paper Award, Korea Information Processing Society's Journal of Information Processing Systems (JIPS) Survey Paper Award (Gold) 2019, IEEE Blockchain 2019 Outstanding Paper Award, IEEE TrustCom 2018 Best Paper Award, ESORICS 2015 Best Research Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is also a Fellow of the Australian Computer Society, and Co-Chair of IEEE Multimedia Communications Technical Committee's Digital Rights Management for Multimedia Interest Group.