SFFL: Self-Aware Fairness Federated Learning Framework for Heterogeneous Data Distributions

Jiale Zhang[a], Ye Li[b,*], Di Wu[c], Yanchao Zhao[b] and Shivakumara Palaiahnakote[d]

[a]School of Information Engineering, Yangzhou University, Yangzhou, 225009, China
[b]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China
[c]School of Mathematics, Physics and Computing, University of Southern Queensland, Toowoomba, 4350, Australia
[d]School of Science, Engineering and Environment, University of Salford, UK

ARTICLE INFO

Keywords:
Federated Learning
Fairness Machine Learning
Heterogeneous Distributions

ABSTRACT

Recent years have witnessed increasing development towards federated learning. However, federated learning has been proven to show biased predictions against certain demographic groups, such as sex or race, especially under heterogeneous data distributions. Training fair federated models under heterogeneous data distributions face the challenge of inherent unfair local training and bias propagation during aggregation and mismatch between local fairness and global fairness. Current fairness approaches for federated learning are struggling to balance fairness and privacy. More importantly, they neglect that the differences in update objectives between heterogeneous clients lead to difficulties in maintaining fair classification and learning among clients. To address these limitations, we propose a self-aware, fair federated learning framework, SFFL, which jointly improves fairness and performance under heterogeneous data distributions without the requirement for clients' sensitive information. Firstly, we present the FairEM method, which considers the heterogeneous distributions as a combination of multiple underlying distributions and decomposes the clients' training objects to the fair training objects on underlying distributions to alleviate the fairness and performance decrease caused by inconsistency update objects. Secondly, we introduce a self-aware aggregation method to mitigate the bias propagation across different component models without requiring sensitive statistics. Extensive evaluation results demonstrate the effectiveness of our proposed framework in achieving fairness and maintaining performance in heterogeneous data distributions.

1. Introduction

Federated learning [1], which is proposed as a machine learning framework with distributed privacy protection, has been increasingly applied throughout a wide range of the real-world, including in healthcare [2, 3], finance [4], recommendation systems [5, 6] and many more realms. Due to the distributed nature of federated learning, the clients may not only train a biased local model but also affect other clients during aggregation (i.e., bias propagation) [7, 8]. In detail, the local training for each client will encode the historical biases that exist in their local data to their model parameters and perpetuate such bias during decision-making, classification, or recommendation. Then, the biased local models will propagate and amplify their biased parameters to the global model through aggregation, which makes the global model unfair. Thus, achieving fairness in federated learning faces more challenges and has caused considerable attention in recent research.

Due to the privacy constraints in federated learning, where the local data are private, previous works on ML cannot extend to FL since those methods are under the assumption that the server has access to the entire dataset. In order to fulfill the requirement for fairness in federated learning, several works have investigated fairness issues, proposing that the local clients can share their local bias or some additional privacy information as a surrogate of the entire dataset to help the server formulate the global optimization objects, then clients can solve them on local to finally train a fair model [9, 10]. For example, Du et al. [11] propose a fairness-aware agnostic federated learning framework, AgnosticFair, to concur with the fairness challenge of unknown testing data. In AgnosticFair, the server collects the clients' bias and loss information to formulate a minimax optimization object to minimize the model bias under a given threshold. FPFL [12] extends the modified method of differential multipliers to empirical risk minimization with fairness constraints and introduces differential privacy to protect the clients' bias and the loss information. Besides, the requirement for group information of clients also limits the development of an FL-based recommendation system, F2MF [13] is a fairness-aware framework for recommendation through communicating group statistics during federated optimization and uses differential privacy techniques to avoid exposure of users' group information. Both of the above works perform well when the distributions of each client are similar, i.e., clients' data distributions follow the Independent and Identical Distributions (IID) since the update directions of clients are similar.

However, data is generally distributed in non-IID across clients, resulting in such approaches based on the global optimization objects not being applicable. The reason is that there exists a mismatch between client-level fairness and global fairness, and the global optimization methods in IID take a global perspective, which may not find a fair classification boundary for every client, then further results in the

jialezhang@yzu.edu.cn (J. Zhang); milesyli@163.com (Y. Li); di.wu@unisq.edu.au (D. Wu); yczhao@nuaa.edu.cn (Y. Zhao); S.Palaiahnakote@salford.ac.uk (S. Palaiahnakote)
ORCID(s):

model prediction fairly from the data calculated on global but unfairness on local. Therefore, addressing fairness issues in heterogeneous data distributions should make the model predict fairly on each client's local, which means we should leverage the client-level fairness to promote global fairness. Besides, the model is ultimately deployed on the client [14, 15] also indicates that client-level fairness should have been prioritized more over global fairness.

Recent studies on solving fairness issues in federated learning, from our investigation, only FCFL [14], FairFed [16], and GLocalFair [15] consider the client-level fairness problem. FCFL starts from the view of multi-objective constraint optimization to address the algorithmic disparity and performance inconsistency in heterogeneous federated learning. FairFed proposes to address the mismatch of local fairness and global fairness by combining the centralized debiasing algorithms locally and reweighting the aggregation weight on the server according to clients' local bias. Besides, GLocalFair follows a similar path to FairFed, combining local derbies and global fair aggregation methods. Specifically, GLocalFair leverages the Gini coefficient as a surrogate of privacy bias information, which addresses the privacy problem of previous works. However, existing research still faces some shortages:

Unbalance of privacy and fairness. Previous works did not balance privacy and fairness no matter the reweighting [15, 16] or the optimization methods [14]. On the one hand, FCFL and FairFed attempted to improve fairness through collecting additional information from clients, which may unintentionally leak the clients' privacy. On the other hand, GLocalFair introduces the Gini coefficient to move away from the requirement for privacy information, but it decreases the utility of the bias-based reweighting method since the Gini coefficient tends to be less sensitive within a certain fairness threshold.

Low-effective for non-IID. Recent works have not addressed the performance decrease introduced by heterogeneous data distributions. Approaches of FairFed and GLocalFair decrease the aggregation weight for highly biased clients but neglect the fact that less biased clients may not perform well for the prediction, resulting in the model losing usability. Besides, both methods focus on addressing the fairness issues on a powerful global model, however, a single model may not have the ability to accurately predict samples with similar features but have different labels. For example, medical centers in different regions may give different diagnoses when treating similar patients due to regional differences. As a result, a directly trained model may give wrong diagnoses to patients in areas with specific characteristics, causing unfair decisions.

In this paper, we focus on achieving group fairness in federated learning under heterogeneous distributions. Firstly, we propose a fairness-aware local training method: Fair-EM, which considers the heterogeneous distribution as the linear combination of underlying distributions with personalized weights and introduces the fairness constraints with adaptive budget adjustment to local training for clients with heterogeneous data distributions. Second, we present a self-aware aggregation method for fair aggregation, which reweighting clients' aggregate weight by measuring the updated distance to the global component model. The contributions of this paper can be summarized as follows:

- We propose SFFL, a **S**elf-aware **F**air **F**ederated **L**earning Framework to achieve client-level group fairness in heterogeneous data distributions with high effective and free for bias information.
- We present a fairness-aware local training method, Fair-EM, through incorporating the EM algorithm and dynamic updating of local fairness constraint achieves local fair training under heterogeneous distributions.
- We introduce a self-aware aggregation method by reweighting client aggregate weights of each component model according to distance to achieve fair aggregation.
- We conduct extensive experiments to demonstrate that the proposed framework can achieve fairness in federated learning under heterogeneous distributions while maintaining high performance.

This paper will be organized as follows. In Section 2, we review the important works on fairness machine learning. Then, we introduce the proposed SFFL framework in Section 3. After that, we present the Experiments on certain machine learning fairness datasets and analysis the results in Section 4. Finally, we conclude in Section 5 and point out future work directions.

2. Related works

2.1. Fairness Metrics

Group fairness essentially compares the results of the classification algorithm about two (or more) groups, which are defined according to sensitive groups, such as sex or race. Over time, many different metrics have been proposed to achieve fairness in machine learning in various aspects. In the following, we describe the most prominent definitions and measures of group fairness: statistical parity difference (also referred to as Demographic Parity), Equalized odds (EOD) and Equal opportunity (EOP).

Statistical Parity Difference (SPD) [17] is one of the earliest definitions of fairness, it ensures that the two sensitive groups have similar rates in positive predictions. The notion of SPD can be formulated as follows:

$$SPD = P[\hat{Y} = 1|S = 1] - P[\hat{Y} = 1|S = 0], \quad (1)$$

where $\hat{Y}$ represents the predictions and S is the sensitive group set with binary value. However, SPD is hard to introduce to federated learning, especially in non-IID settings, since it may consider an accurate classifier as unfair when there exists significant heterogeneity among sensitive groups [18, 19].

Hardt et al. [20] proposed Equalized odds (EOD) and Equal opportunity (EOP) to conquer the limitations of SPD

under unbalanced data. EOD computes the difference between the false-positive rates (FPRs) and the difference between the true-positive rates (TPRs) of the two sensitive groups to mitigate the influence of unbalanced sensitive group distributions. This metric computes as follows:

$$\begin{aligned} EOD = \text{MAX}\{ & \\ & P[\hat{Y} = 1|S = 0, Y = 1] - [\hat{Y} = 1|S = 1, Y = 1], \\ & P[\hat{Y} = 1|S = 0, Y = 0] - P[\hat{Y} = 1|S = 1, Y = 0]\}. \end{aligned} \tag{2}$$

In contrast to EOD, EOP is more popular, which is a relaxed metric that considers a classifier to be fair when the difference between the TPRs of the two sensitive groups is close to zero, which can be formulated as:

$$\begin{aligned} EOP = & P[\hat{Y} = 1|S = 0, Y = 1] \\ & - [\hat{Y} = 1|S = 1, Y = 1]. \end{aligned} \tag{3}$$

In this paper, we consider achieving fairness in federated learning according to EOP since the heterogeneous data distributions lead to the SPD being unsuitable for FL, and EOP does not overly influence the model during training as EOD does, thus decreasing utility.

2.2. Fairness in Machine Learning

In order to achieve fairness in centralized machine learning, numerous mechanisms have been proposed, most of which can be classified into three categories: 1) Pre-Processing. Calmon [21] introduces a novel probabilistic formulation of data pre-processing for reducing bias. Biswas [22] leverage the causal method to reason about the fairness impact during the pre-processing stage. 2) In processing. Roh [23] converts the fairness problem into a bi-level optimization problem, introducing an outer optimizer to select minibatch size to keep fairness adaptively. Furthermore, traditional constraint optimization methods are still an efficient way to address the fairness problem [24, 25, 26]. 3) Post-Processing. Chiappa [27] provides a method that neglects the impact of unfair pathways based on counterfactual correction. Unfortunately, works in machine learning cannot directly be introduced into federated learning because not only does federated learning locally train biased models, but the aggregation process also leads to bias propagation.

2.3. Fairness in Federated Learning

There are two different notions of fairness in federated learning: Client Fairness and Group Fairness. The former pays attention to optimal client selection for training, providing a fair measurement of clients' contribution, or narrowing the performance difference between clients. The latter tries to make fair predictions among different sensitive groups, such as sex or race.

Client Fairness The distributed training of federated learning naturally introduces fairness problems, such as unfair client selection, mismatch between contribution and rewards, and inconsistent model performance. Several works have been done to address the above fairness issues. Huang et al. [28] introduces a C^2MAB-based method to estimate the model exchange time between each client and server. Then, they introduce RBCF-F, a fair client selection algorithm, to reduce bandwidth usage and improve model performance and convergence speed. FedCS [29] presents a client-selection framework in MEC federated learning, which enables operators to actively allocate resources among MEC clients and set deadlines for model downloading, updating, and uploading, thus accurate the efficiency of training. In addition to improving the overall training efficiency through fair client election, a fair incentive mechanism is also essential to client fairness. Gao et al. [30] propose a fair incentive mechanism for federated learning, FIFL, which provides a dynamic real-time worker assessment and rewards to encourage high-quality clients to join the training and prevent malicious participants. [31] proposes FLI, which adaptively adjusts clients' shares to mitigate the inequality between contributions and rewards fairly. Other studies on client fairness focus on the fairness problem caused by inconsistent model performance. Mohri et al. [32] address the client fairness problem by considering the FL problem as a minimax problem and optimizing the client with the worst performance. Q-FFL [33] achieves fair federated learning by assigning higher aggregate weights to clients with higher loss, reducing the performance difference of the model. Ditto reduces the performance difference between clients by introducing a regularization term in local training to make the local model converge to the optimal global model as much as possible [34]. Further, [35] introduces a regularized term to penalize the differences among clients to address the fairness problem of model accuracy.

Group Fairness Despite client fairness, group fairness in federated learning has attracted more and more attention. Abay et al. [36] explore the possibility of transferring the fairness approach in ML to FL and present several approaches for preliminary exploration. [11] notices the potential distribution shift between training data and agnostic test data and introduces a kernel-based fairness-aware reweighting method to avoid the global model producing the bias classification on unknown test data. In [12], authors introduce the modified method of differential multipliers with fairness constraints to federated learning to achieve fairness. Astral [37] proposes a self-corrective federated learning framework to reweight clients' aggregation weight according to the test accuracy and bias on the global test set. However, most of the above approaches are under the assumption that clients have similar distributions, which makes their application in more general heterogeneous scenarios. FCFL [14] proposes a gradient-based method, FCFL, to address both the group fairness problem and inconstant performance in federated learning, which formulate the bias classification and inconsistency performance as a multi-object optimization problem. Besides, reweighting-based methods have also been applied to achieve fair federated learning. Ezzeldin et al. [16] proposes FairFed, which allows clients to select the debias mechanism in machine learning and modify the weight of clients according to the biased information. Follow FairFed, Meerza [15] avoids the requirement for additional biased in-

formation in the weighting mechanism by introducing the Gini coefficient as a surrogate for biased information. However, it exhibits insensitivity under a certain bias threshold. Notwithstanding the efforts of existing approaches in different directions, these approaches still face one or more of the following problems: additional information, low-effective aggregation and utility. To overcome the shortages, we propose a self-aware fair federated learning framework to achieve fairness in heterogeneous federated learning.

3. The proposed framework

In this section, we will describe the proposed **S**elf-aware **F**airness **F**ederated **L**earning (SFFL) framework, which achieves client-level group fairness while maintaining utility. In the following, we first provide the problem formulation in our SFFL framework and then present the details.

3.1. Problem statement

We suppose there is a set of clients which can be denoted as $\mathcal{K}$ (where $|\mathcal{K}| = N$) participants in federated learning, and the overall goal is to train a model that can get a better estimation of their local data and perform fair classification. For a specific client $k \in \mathcal{K}$ holds a set of data points generated from its local distribution $\mathcal{D}_k$ over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ represent the input and output space, respectively. Generally speaking, it is appropriate to train a separate model (hypothesis) $h_t \in \mathcal{H}$ to fit their local distributions, which vary with other clients. Thus, the goal is then to solve the following optimization problems subject to the fairness constraints:

$$\begin{aligned} &\forall k \in \mathcal{K},\ \text{minimize} \mathcal{L}_{D_k}(h_k), \\ &s.t.\ Fair(h_k) \leq \epsilon, \end{aligned} \tag{4}$$

where $h_k : \mathcal{X} \to \Delta^{|\mathcal{Y}|}$ represents the separate model (hypothesis) of k-th client, and the $\Delta^{|D|}$is the unitary simplex of dimension D. $\mathcal{L}_{D_k}(h_k) = \mathbb{E}_{(\mathbf{x},y)\sim D_k}[l(h_k(\mathbf{x}), y))]$ is the true risk of model h_k under D_k and $l(\cdot) : \Delta^{|\mathcal{Y}|} \times \mathcal{Y} \to R^+$ indicate the loss function. $Fair(\cdot)$ is the fairness metric. In this paper, we consider EOP [20] as the fairness metric, and the sensitive attribution S is a binary attribute value of input space $\mathcal{X}$. Thus, the optimization problem can be rewritten as follows:

$$\begin{aligned} &\forall k \in \mathcal{K}, \text{minimize} \mathcal{L}_{D_k}(h_k), \\ &s.t.\ \text{EOP}(h_k) \leq \epsilon. \end{aligned} \tag{5}$$

3.2. Overview

In FedEM, the authors guarantee the performance consistency of models by treating the heterogeneous data distributions as the mixture of multiple underlying distributions. Inspired by FedEM, we first propose a fairness-aware local training method to train fair component models locally by incorporating the fairness constraints as a penalty term to the loss function. Then, we introduce a self-aware aggregation method that tunes the aggregation weights through measuring the distance of the clients' updates on the component models, further reducing the potential mismatch during aggregation. By jointly training the component models with fairness constraints and self-aware aggregation, each client can benefit from the knowledge distilled from other clients and keep their local model fair even if their distributions exit significant differences. The framework is presented in Fig. 1. In the following, we will describe the methodology in detail, i.e., fairness-aware local training and self-aware aggregation.

3.3. Client-side: Fair-EM algorithm

In order to achieve fairness in federated learning under heterogeneous data distributions, we propose Fair-EM, which considers the local distribution of clients as the mixture of multiple underlying independent distributions and trains a fair component model for each underlying distribution through incorporating the fairness constraints. Specifically, following the assumption in FedEM [38] that the local distribution of k-th client D_k can be represented as the mixture of M underlying distributions $\hat{D}_m$ with a set of weight $\pi_k = [\pi_{k,1}, \pi_{k,2}, \dots, \pi_{k,M}]$, i.e., $D_k = \sum_{m=1}^{M} \pi_{k,i} \cdot \hat{D}_m$. Under this assumption, we can train a component model for each underlying distribution. The predictor of k-th client can be illustrated as $h_k = \sum_{m=1}^{M} \pi_{k,i} \cdot h_{\theta_k}(\cdot)$, where $h_{\theta_k}(\cdot)$ represent the hypotheses parameterized by $\theta_k \in \mathbb{R}^d$. Then, let $l(\cdot, \cdot)$ be the log loss, and the optimization problem in Eq. 5 is converted to:

$$\begin{aligned} &\underset{\Theta,\Pi}{\text{minimize}} \mathop{\mathbb{E}}_{k\sim D_{\mathcal{K}}(\mathbf{x},y)\sim D_k} \left[l(h_k, \mathbf{x}, y)\right] \\ &= \underset{\Theta,\Pi}{\text{minimize}} \mathop{\mathbb{E}}_{k\sim D_{\mathcal{K}}(\mathbf{x},y)\sim D_k} \left[-\log p_k\left(\mathbf{x}, y \mid \Theta, \pi_k\right)\right], \\ &s.t.\ \text{EOP}(h_k) \leq \epsilon. \end{aligned} \tag{6}$$

For simplicity, let us first consider the problem in Eq. 6 without fairness constraints. FedEM introduces an EM-like algorithm that provides a promising solution for that, i.e., we can estimate the parameters $\{\Theta, \Pi_k\}$ through minimizing the empirical version of $l(h_k, \mathbf{x}, y)$ on k-th client. It can be summarized into two steps: Expectation and Maximization. In the Expectation step, k-th client updates the posterior probability of m-th underlying distribution on i-th sample $s_k^{(i)}$ according to the classification performances of m-th component model θ_m^t on i-th sample $s_k^{(i)}$ of client k and the mixture weight of m-th underlying distribution of the prior iteration which calculated by Eq. 7. Then, the Maximization step involves updating the mixture weights and the M component model parameters. In detail, the client updates the mixture weight of the underlying distribution $\hat{D}_m$ with $q_{k,m,i}^t$ to capture its prominence in D_k, which is represented as Eq. 8. Then, the client updates the parameters of the m-th component model via solving the empirical risk minimization problem weighted by $q_{k,m,i}^t$ defined in Eq. 9 to construct an unbiased estimate of the true risk over each underlying distribution $\hat{D}_m, m \in \mathcal{M}$.

Expectation-Step:

$$q_{k,m,i}^t \propto \pi_{k,m}^t \cdot \exp(-l(\theta_m^t, \mathbf{x}_k^{(i)}, y_k^{(i)})). \tag{7}$$

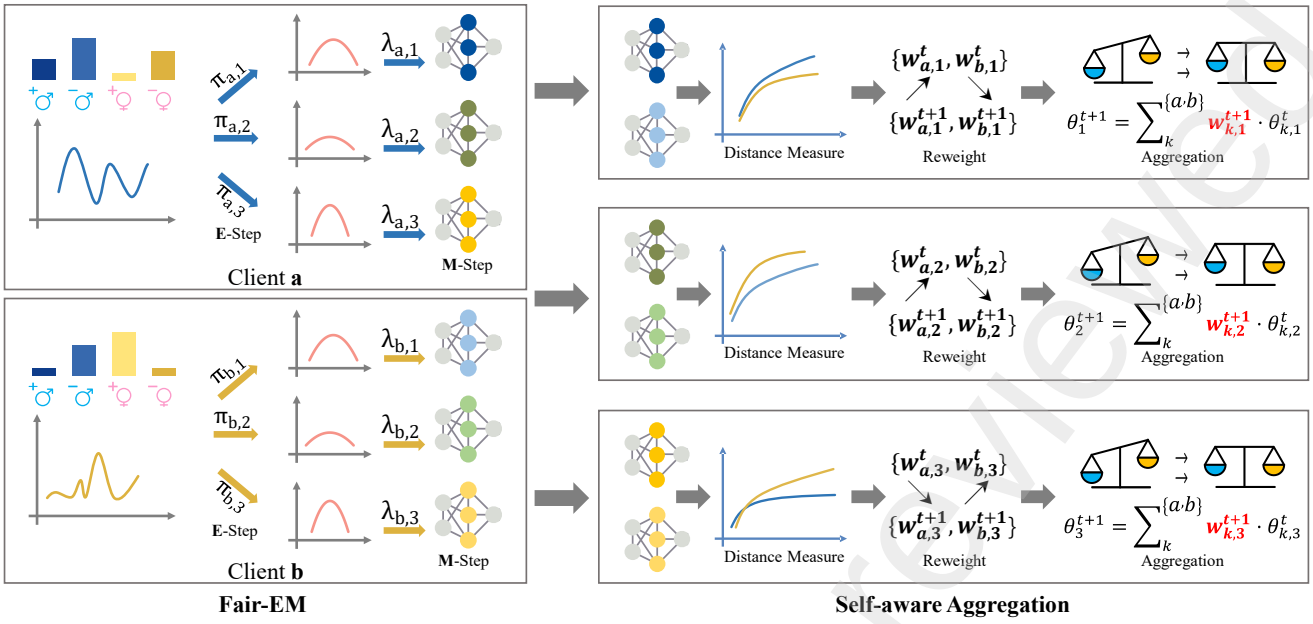


Figure 1: An overview of proposed SFFL framework. On client-side, client A utilizes the Fair-EM algorithm to train fair component models, which first updates the prominence of each underlying distribution of its distribution ($\pi_{a,i}$) and the corresponding fairness weight ($\lambda_{a,i}$), where $i \in \{1,2,3\}$, separately. Then, the client can train fair component models by solving a weighted empirical minimization problem. On server-side, the server employs a self-aware aggregation method to mitigate the fairness mismatch issues caused by inconsistent update objectives. It measures the distances of updates to estimate the prominence of the m-th underlying distribution of clients, thus reweighting the aggregation weights.

Maximization-Step:

$$\pi_{k,m}^{t} = \frac{\sum_{i=1}^{n_t} q_{k,m,i}^{t}}{n_t}, \tag{8}$$

$$\theta_m^t \in \arg\min_{\theta \in \mathbb{R}^d} \sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{n_t} q_{k,m}^{t} \cdot l\left(\theta_{k,m}^{t}, (\mathbf{x}, y)_k^{(i)}\right), \tag{9}$$

Note that the above-described process did not contain fairness constraints, which means it may expose potential biases in the data to produce unfair component models. In order to train a fair component model, we introduce the fairness constraint to the model training, i.e., solving the empirical risk minimization problem weighted by $q_{k,m,i}^{t}$ with the fairness constraint. Specifically, we start from the definition of the Equalized Opportunity (EOP), which requires the model to predict with similar true positive rates for the sensitive group ($S = 1$) and non-sensitive group ($S = 0$), i.e.,

$$\begin{aligned} EOP(h) &= P[\hat{Y} = 1 | S = 0, Y = 1] \\ &\quad - [\hat{Y} = 1 | S = 1, Y = 1] \\ &= \mathbb{E}_{\mathcal{X}|\mathcal{Y}=1}\left[\frac{P(S=1 \mid \mathbf{x}, Y=1)}{P(S=1 \mid Y=1)} \mathbb{1}_{h(\mathbf{x})>0}\right. \\ &\quad \left. - \frac{P(S=0 \mid \mathbf{x}, Y=1)}{P(S=0 \mid Y=1)} \mathbb{1}_{h(\mathbf{x})>0}\right] \\ &= \mathbb{E}_{\mathcal{X}|\mathcal{Y}=1}\left[\frac{P(S=1 \mid \mathbf{x}, Y=1)}{P(S=1 \mid Y=1)} \mathbb{1}_{h(\mathbf{x})>0}\right. \\ &\quad \left. + \frac{1 - P(S=1 \mid \mathbf{x}, Y=1)}{1 - P(S=1 \mid Y=1)} \mathbb{1}_{h(\mathbf{x})<0}\right] - 1. \end{aligned} \tag{10}$$

Therefore, the constraint optimization problem for k-th client of m-th component model with weight $q_{k,m}^{t}$ can be written as:

$$\begin{aligned} \theta_m^t &\in \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n_t} q_{k,m}^{t} \cdot l\left(\theta_{k,m}^{t}, (\mathbf{x}, y, s)_k^{(i)}\right), \\ &s.t.\ \mathbb{E}_{\mathcal{X}|\mathcal{Y}=1}\left[\frac{P(S=1 \mid \mathbf{x}, Y=1)}{P(S=1 \mid Y=1)} \mathbb{1}_{h(\mathbf{x})>0}\right. \\ &\left. + \frac{1 - P(S=1 \mid \mathbf{x}, Y=1)}{1 - P(S=1 \mid Y=1)} \mathbb{1}_{h(\mathbf{x})<0}\right] - 1 \leq \epsilon. \end{aligned} \tag{11}$$

However, solving the problem above is intractable to compute since the constraint contains the indicator functions $\mathbb{1}_{h(\mathbf{x})<0}$ and $\mathbb{1}_{h(\mathbf{x})>0}$. An alternative approach is substituting the indicator function with a surrogate function, thereby formulating the EOP as a convex constraint for direct integration into classification models [26]. In this paper, we follow [26] and set the surrogate function as a logistic function (noted as $f(\cdot)$), then the Eq. 11 is rewritten as:

$$\begin{aligned} \theta_m^t &\in \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n_t} q_{k,m}^{t} \cdot l\left(\theta_{k,m}^{t}, (\mathbf{x}, y, s)_k^{(i)}\right), \\ &s.t.\ \mathbb{E}_{\mathcal{X}|\mathcal{Y}=1}\left[\frac{P(S=1 \mid \mathbf{x}, Y=1)}{P(S=1 \mid Y=1)} f(h(\mathbf{x}))\right. \\ &\left. + \frac{1 - P(S=1 \mid \mathbf{x}, Y=1)}{1 - P(S=1 \mid Y=1)} f(h(\mathbf{x}))\right] - 1 \leq \epsilon. \end{aligned} \tag{12}$$

Then, the fairness constrain can be introduced as the penalty term into the loss function, and the optimization problem in Eq. 12 are reformulated as:

$$\begin{aligned}\theta_m^t &\in \arg\min_{\theta\in\mathbb{R}^d} \sum_{i=1}^{n_t} q_{k,m}^t \cdot \mathcal{L}\left(\theta_{k,m}^t, (\mathbf{x}, y, s)_k^{(i)}\right) \\ &= \arg\min_{\theta\in\mathbb{R}^d} \sum_{i=1}^{n_t} q_{k,m}^t \cdot \left(l_{k,m}^{acc} + \lambda l_{k,m}^{fairness}\right),\end{aligned} \tag{13}$$

where λ is the weight to balance the true risk and fairness. In centralized learning, λ can select from empirical experimental results. However, in federated learning, it is impossible to empirically set a suitable hyper-parameter for each client, especially under heterogeneous distributions. Intuitively, for a less-bias model, we can perform less modification to the classification boundary to guarantee it fair, and it should be higher in a high-bias model [13]. Following this inspiration, we can take the difference between the bias of the component model on D_k and the target value as the basis for the weight settings. Therefore, the update of hyper-parameter of m-th component model $\lambda_{k,m}^t$ is represented as:

$$\lambda_{k,m}^t = \lambda_{k,m}^{t-1} - (\epsilon - gap_{\theta_m^t, D_k}), \tag{14}$$

where $gap_{\{\theta_m, \hat{D}_m\}}$ is the EOP value of θ_m^t on D_m. Nevertheless, directly introduce gap_{θ_m, D_k} to update $\lambda_{k,m}^t$ is irrational, since the difference of EOP between continue iterations can be very large, especially in initial phase of training since the classification boundaries of models may be arbitrary. For this purpose, we introduce a parameter to control the update step size, which is similar to the learning rate mechanism in model training, to avoid the non-convergence of the model due to the over update. Thus, the hyper-parameter updates for m-th component model at t-th iteration are formulated as follows:

$$\lambda_{k,m}^t = \lambda_{k,m}^{t-1} - \eta_{k,m} \cdot (\epsilon - gap_{\theta_m^t, D_k}), \tag{15}$$

According to Fairness-aware local training, clients can train fair component model over their heterogeneous data distributions.

3.4. Server-side: Self-aware Aggregation

Recall that, FedAvg and FedEM utilize the regular aggregation method which aggregates the global model depending on the quantity of data held by the client, i.e.,

$$\begin{aligned}\theta^{t+1} &= \sum_{k=1}^{K} w_k^t \cdot \theta_k^t, \\ w_k^t &= \frac{n_k}{\sum_{i=1}^{K} n_i}.\end{aligned} \tag{16}$$

In general, considering only the amount of data in aggregation will bias the global component model towards clients with large amounts of data, ignoring the differences in the prominence of each client's underlying distribution. The client distributions vary in their prominence for the underlying distribution D_m corresponding to the m-th parameterized model, i.e., $\pi_{a,m} \neq \pi_{b,m}$. This allows their parameters to be updated with different objectives, which means they may update the parameters with different directions and sizes. If identical aggregation weights are allocated to the clients which π_m are not similar, then results in the model fails to converge to a fair classification boundary. In addition, since π_m for each client is iteratively updated during training, we cannot predict π_m in a priori way, and using fixed weights cannot solve the above problem. The key insight to address this problem is that the clients with high prominence on m-th underlying distribution should have similar objectives, which is similar to the fairness problem of FL in IID distribution.

Consequently, it is necessary to assign distinct aggregation weights according to their prominence of underlying distributions and allocate higher weights for the clients with higher π_m. Besides, considering the privacy requirements of federated learning, keeping the mixture weights locally within the client is essential, rather than uploading them to the server.

Following the intuitions, let us recall that of Fair-EM, each sample holds mixture weight $q_{k,m,i}^t$ represent the weight for m-th underlying distribution, which is the prominence of m-th underlying distribution, and update the local component model based on this weight, leads to different step size of model updates, i.e., samples with high weights updates more. Thus, the differences in model updates between clients can be used as a proxy for calculating mixture weights in the server.

According to this observation, we propose a self-aware aggregation method, which realizes the adjustment of component model aggregation weights through self-aware aggregation weight reweighting to mitigate the unfairness caused by aggregation weights. Specifically, we calculate the update size between the client component model of each round and the component model of the previous round as a reference for updating the aggregation weights, and ultimately achieve self-aware aggregation reweighting.

Next, we detail how this method reweights the aggregation weights in each round. We use the Euclidean norm to calculate the update distance of k-th client, $\mathrm{dis}_{\theta_m^t, \theta_{k,m}^t}$, which is a common way to calculate the differences between models:

$$\mathrm{dis}_{\theta_m^t, \theta_{k,m}^t} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}, \tag{17}$$

where, p_i represents the i-th weight of model θ_m^t, and q_i is the i-th weight in model $\theta_{k,m}^t$. Therefore, we have the average update distance in t-th iteration, which represent as:

$$\mathrm{dis}_{avg}^t = \frac{1}{|\mathcal{K}|} \sum_{k=1}^{\mathcal{K}} \mathrm{dis}_{\theta_m^t, \theta_{k,m}^t}. \tag{18}$$

Then, the aggregation weight assigned to each clients update follows:

$$w_m^{t+1} = \frac{\exp(\log(w_{k,m}^t) - \Delta_k)}{\sum_{k=1}^{\mathcal{K}} \exp(\log(w_{k,m}^t) - \Delta_k)} \tag{19}$$

Note that, $\Delta_k = (\mathrm{dis}_{avg}^t - \mathrm{dis}_{\theta_m^t, \theta_{k,m}^t})$ is the gap between the average distance and the update distance of k-th client.

Then, the server aggregates the component models according to the updated weights. The m-th component model for next iteration aggregates as follows:

$$\theta_m^{t+1} = \sum_{k=1}^{\mathcal{K}} w_m^{t+1} \cdot \theta_{k,m}^t. \tag{20}$$

The pseudo code of SFFL is provided in Algorithm. 1.

Algorithm 1: SFFL: Self-aware Fairness Federated Learning

Input: Underlying distributions Number M, clients $\mathcal{K}$, fairness threshold ϵ, hyper-parameters η, client dataset D, number of iterations T.
Output: $\theta_m, m \in M$, $\pi_k, k \in \mathcal{K}$.
Server initializes M component models.
Initialize π_k^0, $k \in \mathcal{K}$.
for *each iteration* $t \in [1, 2, \cdots, T]$ **do**
  // Client Executes
  for *each client* $k \in \mathcal{K}$ **do**
    EVENT: Received component models $\theta_m^t, m \in M$
    for *component* $m \in [1, 2, \cdots, M]$ **do**
      //Expectation step
      $q_{k,m,i}^t \propto \pi_{k,m}^t \cdot \exp(-l(\theta_m^t, \mathbf{x}_k^{(i)}, y_k^{(i)}))$;
      $\lambda_{k,m}^t = \lambda_{k,m}^{t-1} - \eta_{k,m} \cdot (\epsilon - gap_{\theta_m^t, D_k})$;
      //Maximization step
      $\pi_{k,m}^t = \frac{1}{n_t} \sum_{i=1}^{n_t} q_{k,m,i}^t$;
      $\theta_m \leftarrow \theta_m - \eta \cdot q_{m,i} \cdot \nabla_\theta (l_m^{acc} + \lambda_m l_m^{fairness})$;
    end
  end
  // Server Executes
  for *component* $m \in [1, 2, \cdots, M]$ **do**
    $\mathrm{dis}_{\theta_m^t, \theta_{k,m}^t} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$;
    $\mathrm{dis}_{avg}^t = \frac{1}{|\mathcal{K}|} \sum_{k=1}^{\mathcal{K}} \mathrm{dis}_{\theta_m^t, \theta_{k,m}^t}$
    $w_m^{t+1} = \frac{\exp(\log(w_{k,m}^t) - \Delta_k)}{\sum_{k=1}^{\mathcal{K}} \exp(\log(w_{k,m}^t) - \Delta_k)}$
    $\theta_m^{t+1} \leftarrow \sum_{k=1}^{\mathcal{K}} w_{m,k}^{t+1} \theta_{k,m}^t$;
  end
end
Return $\theta_m, m \in M$, $\pi_k, k \in \mathcal{K}$.

4. Experiments

4.1. Datasets

In the following, we evaluate SFFL's performance on the U.S. Census data, which consists of census data for the 50 U.S. states and Puerto Rico. Then, we consider three different tasks predefined by the folktables [39]: ACSEmployment, ACSIncome and ACSHealthInsurance. We consider the binary notion of *sex* (male and female) as the sensitive attribution. We leverage the data allocation method proposed by Hsu et al. [40], which is a synthesis method based on Dirichlet distribution through controlling the parameter α to achieve different heterogeneity of sensitive attributions in clients. We assign each state's data to clients to better reflect the potential heterogeneity between the data due to regional differences. We allocate nearly 4000 samples to each client consisting of 2400 training points and 1600 test points; the test distribution and training distribution are drawn from the same data distribution. For better understanding, we provide an example to illustrate the distribution of each participant for different heterogeneity levels, which is shown in Fig. 2. Note that some clients may have insufficient samples because some states do not have enough records for a specific attribute (e.g., Male positive in ACSIncome task), which is more noticeable when the α is small.

4.1.1. Baselines

We compare our methods with the classic federated learning algorithm FedAvg, a personalize federated learning framework FedEM, and two different state-of-the-art fair federated learning frameworks FCFL [14] and FairFed [16]. We provide a brief description of the baseline methods as follows:

FedAvg [1] is the distributed machine learning framework that trains clients' models locally and aggregates the global model according to the data size of each client.

FedEM [38] is a novel personalized federated learning framework based on multi-task learning. FedEM introduces the EM-like algorithm to update the mixture weights to fit the local distribution and update the component model based on these weights. Then, the component models are aggregated according to the data size of each client.

FCFL [14] is a state-of-the-art FL framework to achieve client fairness and performance inconsistency. FCFL considers the fairness problem and performance inconsistency as a multi-object constraint optimization problem and addresses this problem using a gradient-based optimization method.

FairFed [16] is a state-of-the-art fair federated learning framework based on the reweighting method, which utilizes the collection of fairness information uploaded by clients to reweight the aggregation weights to make the global model favor towards clients with high-fair clients as a way to achieve fairness in federation learning.

4.1.2. Implementation details

Parameters are carefully chosen to optimize the model for best performance. We implement the proposed method and the baseline methods based on PyTorch. Following common practices for fair federated learning, we train a logistic regression model as the component model for all tasks. The batch size during training is set to 128. The local learning rate is 0.01. The local fairness update step size in Eq. 15 is 0.1. We fixed the local epoch for all experiments as 2. The training iteration is set to 150. Client numbers are set to 20. The number of underlying distributions M is set to 3.

4.2. Accuracy and EOP

Table. 1 shows the performance of SFFL compared with four baseline federated learning baseline methods, and we

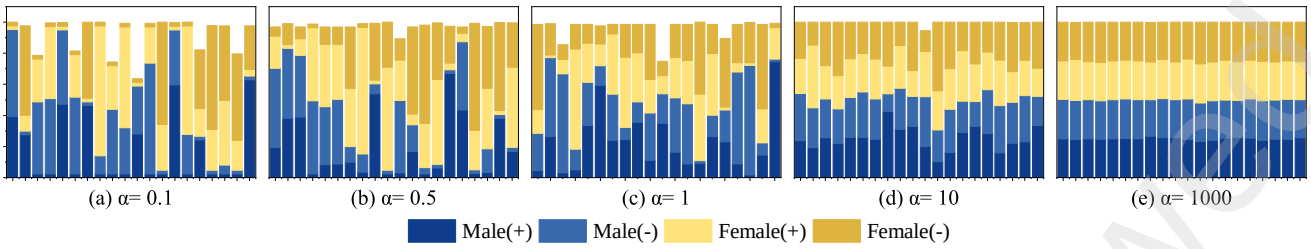


Figure 2: An example for data allocation. X-ares represent the clients, Y-ares are the percentages of data.

Table 1
Performance comparison between SFFL and baseline methods under five different heterogeneous levels. We test the performance of SFFL as well as other baselines five times and report the mean and standard deviation results to eliminate the effects of randomness. Note that the marker of "Acc(↑)" represents the accuracy, which higher is better, and "EOP(↓)" means EOP, which lower is better. Besides, we highlighted the best performance of each group's experiments in bold.

		FedAvg		FedEM		FCFL		FairFed		Ours	
		Acc(↑)	EOP(↓)	Acc(↑)	EOP(↓)	Acc(↑)	EOP(↓)	Acc(↑)	EOP(↓)	Acc(↑)	EOP(↓)
ACSEmployment	α=0.1	75.31 ± 1.32	9.28 ± 2.34	**79.46 ± 0.24**	5.37 ± 1.75	74.47 ± 0.66	4.26 ± 0.65	72.93 ± 3.01	5.40 ± 1.14	79.16 ± 0.55	**1.75 ± 1.83**
	α=0.5	78.32 ± 1.14	7.88 ± 1.98	78.98 ± 0.51	5.40 ± 0.79	73.75 ± 0.76	3.50 ± 0.39	73.73 ± 2.09	3.35 ± 0.93	**79.03 ± 0.43**	**1.58 ± 0.71**
	α=1000	78.53 ± 1.02	4.90 ± 1.93	79.09 ± 0.40	4.18 ± 0.48	74.30 ± 0.73	2.88 ± 0.63	73.97 ± 2.18	2.75 ± 0.79	**79.15 ± 0.39**	**1.77 ± 0.61**
	α=1000	79.16 ± 0.34	3.59 ± 0.78	**79.22 ± 0.31**	3.43 ± 0.38	75.58 ± 0.43	2.25 ± 0.54	74.40 ± 1.65	2.82 ± 1.41	79.21 ± 0.52	**1.20 ± 0.67**
	α=1000	79.18 ± 0.23	3.58 ± 0.89	**79.27 ± 0.34**	3.72 ± 0.37	74.45 ± 0.63	2.28 ± 0.53	73.92 ± 2.16	2.87 ± 1.17	79.27 ± 0.45	**1.57 ± 0.41**
ACSIncome	α=0.1	73.55 ± 1.26	16.35 ± 1.56	**79.49 ± 1.05**	12.55 ± 3.72	75.06 ± 0.40	5.37 ± 1.09	69.50 ± 1.97	5.93 ± 1.30	77.76 ± 0.87	**2.88 ± 1.75**
	α=0.5	75.85 ± 0.67	10.81 ± 1.80	**78.32 ± 0.67**	8.34 ± 2.49	75.09 ± 0.52	4.50 ± 0.98	69.97 ± 2.02	5.56 ± 1.53	77.73 ± 0.63	**2.59 ± 0.96**
	α=1000	76.12 ± 0.94	9.73 ± 2.47	**77.88 ± 0.67**	5.86 ± 2.50	75.27 ± 0.44	3.22 ± 0.71	70.13 ± 1.97	4.96 ± 1.01	77.76 ± 0.58	**2.17 ± 0.84**
	α=1000	76.54 ± 0.29	4.83 ± 2.51	77.74 ± 0.56	3.86 ± 0.76	75.36 ± 0.33	2.81 ± 0.45	70.63 ± 1.89	3.91 ± 0.90	**77.87 ± 0.43**	**2.16 ± 0.47**
	α=1000	76.38 ± 0.19	5.60 ± 0.65	**77.67 ± 0.53**	3.91 ± 0.86	75.30 ± 0.46	2.92 ± 0.46	71.43 ± 1.98	3.81 ± 0.91	77.66 ± 1.08	**2.13 ± 0.62**
ACSHealthInsurance	α=0.1	61.84 ± 4.32	11.89 ± 3.02	**70.13 ± 1.84**	11.64 ± 3.16	63.90 ± 2.11	6.09 ± 1.25	58.65 ± 2.10	6.86 ± 1.87	68.27 ± 1.39	**3.63 ± 1.17**
	α=0.5	66.10 ± 1.43	15.09 ± 4.07	**69.25 ± 1.01**	9.89 ± 1.85	64.02 ± 1.45	4.90 ± 1.30	57.96 ± 1.46	5.43 ± 1.82	67.83 ± 1.06	**2.57 ± 1.57**
	α=1000	66.90 ± 1.47	14.73 ± 3.09	**68.45 ± 1.37**	8.07 ± 3.50	64.40 ± 1.05	4.26 ± 0.84	59.08 ± 1.69	5.11 ± 1.50	67.83 ± 1.21	**2.64 ± 1.87**
	α=1000	65.27 ± 0.93	6.77 ± 2.36	67.78 ± 1.12	5.57 ± 0.69	64.57 ± 0.86	3.67 ± 0.85	59.32 ± 2.12	4.87 ± 2.07	**68.05 ± 1.15**	**2.57 ± 1.23**
	α=1000	64.53 ± 0.84	6.77 ± 1.43	67.97 ± 1.16	5.18 ± 0.60	64.49 ± 0.88	4.12 ± 1.34	59.33 ± 2.02	4.98 ± 2.49	**67.99 ± 1.20**	**2.85 ± 1.41**

follow the original settings in their papers. From the perspective of accuracy, the proposed SFFL only has a slight decrease to FedEM since the introduction of fairness constraints but still outperforms other baseline methods, especially the other two fair federated learning methods. Numerically, SFFL achieves an average accuracy improvement of 3.64% on the three tasks compared to FCFL and 7.31% for FairFed. Overall, introducing underlying distributions promotes knowledge sharing across clients, allowing them to learn from other clients even with completely different data distributions. Furthermore, the proposed Fair-EM can better fit the local distribution through weight updating, which improves the utility of each client.

For fairness benefits from the multi-component models, aggregation separately naturally decreases the mismatch of update directions, and SFFL achieves more fairness improvements than FCFL and FairFed. Specifically, compared to FairFed, SFFL has a maximum average fairness improvement of 2.60% over the three tasks, representing an increase of 56.89%. Concerning FCFL, the average fairness improvement is 1.76%, which is a growth of 53.68%. As to FedEM, the proposed Fair-EM algorithm and Self-aware aggregation method effectively address the fairness problem in FedEM with an average increase of 80.80% on three tasks.

Consequently, this experiment provides empirical evidence that the proposed SFFL framework can achieve fairness for each client while maintaining high utility.

4.3. Different Learners Numbers

An important assumption is that the client's heterogeneous distribution is the mixture of M underlying distributions. Therefore, the selection of M plays a vital role in SFFL. In this section, we test the performance of SFFL and FedEM with different numbers of underlying distributions. We present the results in Fig. 3. The conclusion on accuracy is similar to the conclusion in FedEM [38] where the accuracy may not increase too much with the increase of M ($M \geq 2$), and the improvement from $M = 1$ to $M = 2$ implies that the efficient of the mixture model. Besides, as described above, introducing fairness constraints slightly influences the accuracy of SFFL, but such a slight decrease is acceptable. However, different from the accuracy, the increase of M does affect the improvement of EOP in SFFL since the increase of underlying distribution better mitigates the mismatch during aggregation. There is also a certain degree of reduction in FedEM. Still, the lack of local fair training methods makes it difficult for the model to converge towards a fair direction, leading to a large gap between its performance and SFFL.

4.4. Different Client Numbers

The client number is a crucial parameter in federated learning, impacting federated learning models' efficiency of aggregation and fairness. In this case study, we extend the scaling factor $\mathcal{K}$ from 10 to 50 to better test the robustness of SFFL and several other baseline methods (FedEM, FCFL,

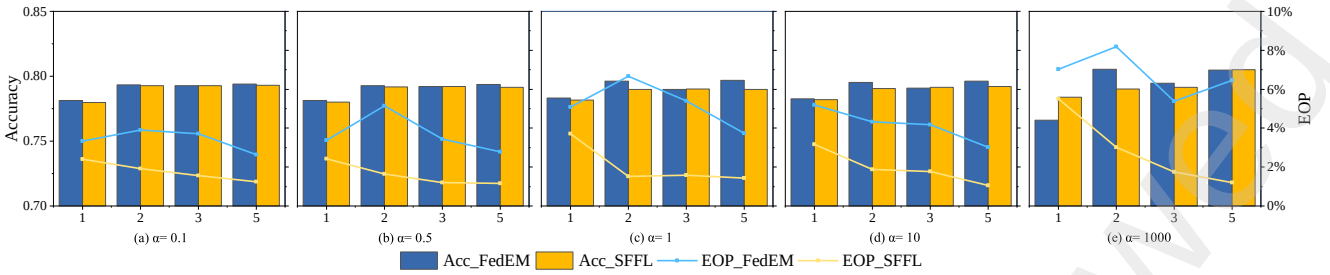


Figure 3: Different Learners Numbers under different α. X-axes are the number of underlying distributions, the left Y-axes represent the accuracy (bars), and the right Y-axes are EOP (lines).

and FairFed) under different client numbers, and the results are presented in Fig. 3. We can see from the figure above that both FedEM and SFFL receive remarkable performance on accuracy and have little decrease with the increase of $\mathcal{K}$. We consider this mainly because of the effectiveness of the EM algorithm since every client can contribute to the component models and learn from others. The EM algorithm can also find an appropriate mixture weight for each client according to the current component models, thus improving the performance under heterogeneous distributions.

For the EOP, due to the FedEM not considering the fairness problem, it has the worst performance, where clients embed bias into their local component models and propagate them to others. FCFL and FairFed perform better than FedEM since they take fairness into consideration. Still, they focus on solving the problem on a single global model, making it hard to find a balance to achieve fairness for both clients, and the trouble gets worse with the increase in client numbers. On the contrary, SFFL achieves the best performance in EOP, which, as described above, the Fair-EM and Self-aware aggregation address the problem of training a fair component model in heterogeneous environments. Although the increase of clients at low heterogeneity levels (α is low) slightly increases the mismatch, resulting in a slight rise in the fairness gap as well, the declination is still acceptable (numerically from 2.1% to 2.9 in the case of $\alpha = 0.1$).

4.5. Ablation study

To validate the effectiveness of our proposed components in SFFL, we conduct additional experiments as ablation studies. In detail, we decouple the Fair-EM method and Self-aware aggregation method and create three variations: FedEM (EM-like, FedAvg), Fair-EM Only (Fair-EM, FedAvg), and SFFL (Fair-EM, Self-aware aggregation). Note that we follow the setting above, and the task is ACSEmployment. The results are shown in Fig. 5, which indicate that both methods play vital roles in improving fairness but contribute slightly differ with the change of α. When α comes to 0.1, 0.5, and 1 (i.e., non-IID distribution), the unfairness is not only produced by the local training but also during the model aggregation. As described in Section.3.4, the mismatch between high-data clients and high-mixture-weight clients during aggregation results in the global component model aggregates in an unfair way. Thus, the Self-aware aggregation addresses the problem through reweighting according to the update distance, allowing high-weight clients to have higher aggregation weights, thus guaranteeing fairness when α is small. For $\alpha = 10$ and $\alpha = 1000$, the influences of mismatch are less than that of non-IID, and the major problem of bias converts to how to achieve fairness locally. Therefore, the Fair-EM can achieve more fairness improvements. In conclusion, the higher the degree of IID of the data distribution, the more outstanding contribution Fair-EM makes, and conversely, the greater importance of self-aware aggregation.

4.6. Hyper-parameter λ

In Section. 3.3, we introduce a hyper-parameter λ in Eq. 15 for controlling the step size of the fairness weight λ. In the previous experiments, we empirically set $\lambda = 0.1$. To evaluate the effect of different λ, we set four different λ using heterogeneity level $\alpha = 0.1$ to conduct a case study. The results are pictured in Fig. 6. Specifically, the performance under $\lambda = 0.5$ is the worst, where it has the lowest downward trend at the initial phase ($t \leq 45$) and shows significant fluctuation in the following training. As described before, we consider the reason that overly large modifications may make it difficult for the model to find an appropriate classification boundary in the initial phase. The results under $\lambda = 0.3$ are much better than that of $\lambda = 0.5$ but still exhibit slight fluctuation. However, it is also a problem when the λ goes too small since it loses the ability to adjust the fairness weight in time. The results come the best when $\lambda = 0.1$ since it does not change much in the initial phase and adjusts well when the bias appears.

4.7. Effectiveness of Self-aware aggregation

In Section. 3.4, we propose a self-aware aggregation to solve the mismatch of the update direction between clients with more data and clients with higher mixture weight during the aggregate of the component models. Self-aware aggregation aims to improve matching between aggregation weight and mixture weight without directly uploading mixture weight. To better illustrate the performance of the proposed method, we present the changing of aggregation weight and mixture weight for specific client, the results are provided in Fig. 7. It shows that our proposed method successfully captures the change of mixture weights and adjusts correspondingly. This also reflects that the proposed self-aware aggregation method effectively realizes our conception.

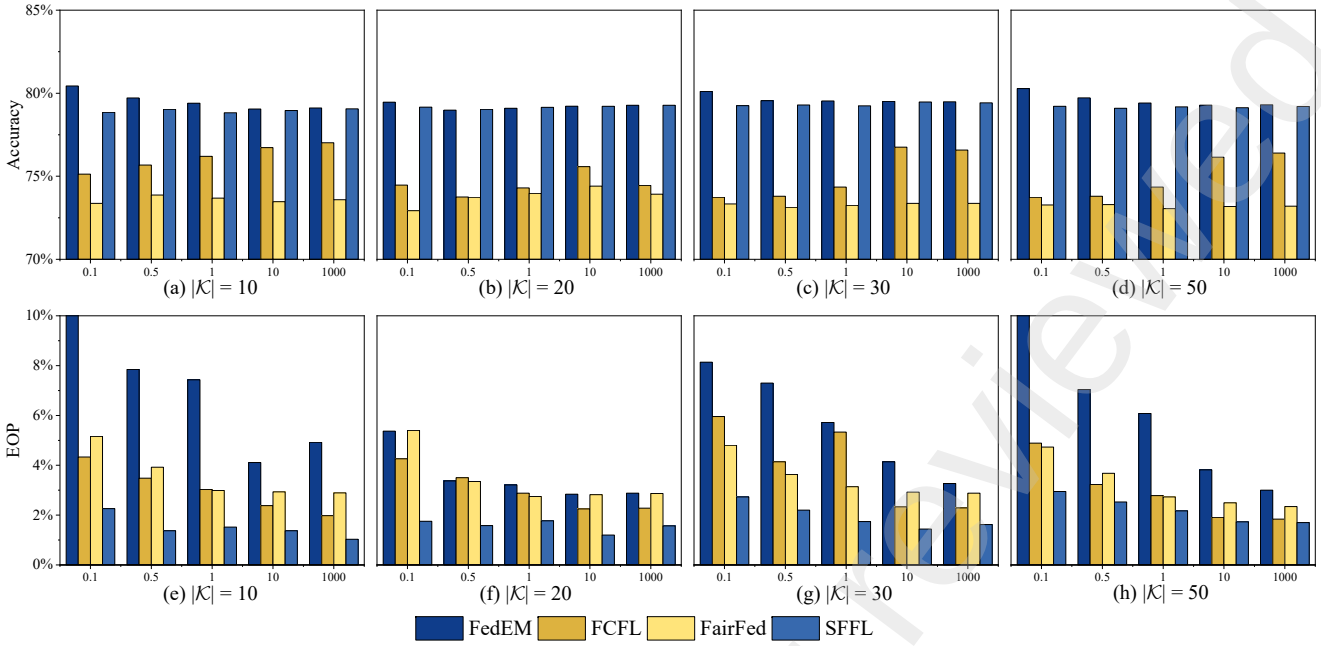
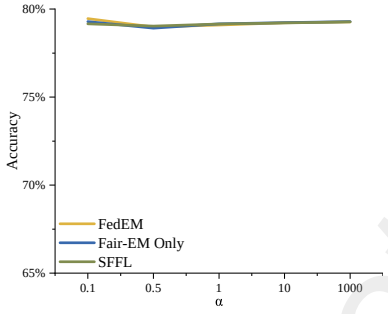
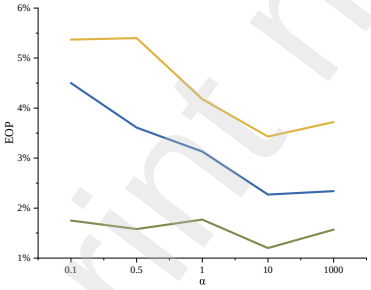


Figure 4: Performance of different client numbers. X-axes are α and the above Y-axes are Accuracy (higher is better) and the below Y-axes are EOP (lower is better).



(a) Accuracy



(b) EOP

Figure 5: Ablation studies for the effeteness of different component of the proposed framework. X-axes are α represent the different heterogeneous levels, and the Y-axis in (a) represents accuracy (higher is better) and EOP (lower is better) in (b).

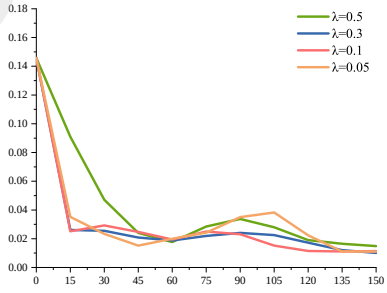


Figure 6: Performance with different hyper-parameter λ. (The X-axis represents iterations, and the Y-axis is EOP.)

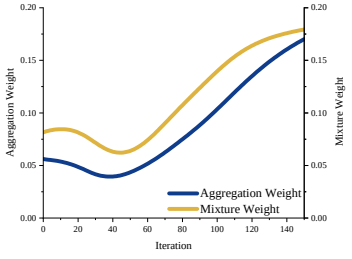


Figure 7: Trends of aggregation weights and mixture weights with training rounds.

5. Conclusion

In this paper, we propose the SFFL framework to address the fairness issues in federated learning under heterogeneous data distribution. Specifically, we first introduce a fairness local training method: Fair-EM, which first addresses the heterogeneous problem in FL and then introduces the fairness constraint as a penalty term. Then, we propose an adaptive weighting algorithm to train a local fair model under heterogeneous data distributions. Besides, we also propose a self-aware reweighting algorithm to adjust the aggregation

weights according to the update distance to alleviate both the performance decrease and the fairness mismatch. We provide sufficient experiments from different aspects and verify that the proposed framework outperforms the state-of-the-art fair federated learning framework in terms of accuracy and fairness under heterogeneous data. However, out proposed framework still has some weaknesses. The introduction of the Fair-EM algorithm requires clients to repeat the training locally for M times, and this may incur high computational and communication overheads, which may limit the applicability of our approach in large-scale scenarios. In the future, we will try to solve this problem and extend our framework in a lightweight way. Besides, we will extend our framework to cover more common fairness notions, such as equalized odds, by designing a mutual update method to balance the potential conflicts between TPR and FPR.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 62206238), the Natural Science Foundation of Jiangsu Province (No. BK20220562), the Natural Science Research Project of Universities in Jiangsu Province (No. 22KJB520010), and the China Postdoctoral Science Foundation (No. 2023M732985).

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, FedAvg: Communication-Efficient Learning of Deep Networks from Decentralized Data, in: International Conference on Artificial Intelligence and Statistics, 2017, pp. 1273–1282.

[2] F. Zhang, Z. Shuai, K. Kuang, F. Wu, Y. Zhuang, J. Xiao, FedUFO: Unified fair federated learning for digital healthcare, Patterns 5 (1) (2024) 100907. doi:10.1016/j.patter.2023.100907.

[3] Z. Liu, Y. Chen, Y. Zhao, H. Yu, Y. Liu, R. Bao, J. Jiang, Z. Nie, Q. Xu, Q. Yang, Contribution-aware federated learning for smart healthcare, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 12396–12404.

[4] Z. Wang, J. Xiao, L. Wang, J. Yao, A novel federated learning approach with knowledge transfer for credit scoring, Decision Support Systems 177 (2024) 114084.

[5] D. Chai, L. Wang, K. Chen, Q. Yang, Secure federated matrix factorization, IEEE Intelligent Systems 36 (5) (2020) 11–20.

[6] C. Zhang, G. Long, T. Zhou, P. Yan, Z. Zhang, C. Zhang, B. Yang, Dual personalization on federated recommendation, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023, pp. 4558–4566.

[7] H. Chang, R. Shokri, Bias Propagation in Federated Learning, in: International Conference on Learning Representations, 2023.

[8] Z. Zhou, L. Chu, C. Liu, L. Wang, J. Pei, Y. Zhang, Towards Fair Federated Learning, in: ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual Event Singapore, 2021, pp. 4100–4101. doi:10.1145/3447548.3470814.

[9] D. Y. Zhang, Z. Kou, D. Wang, FairFL: A Fair Federated Learning Approach to Reducing Demographic Bias in Privacy-Sensitive Classification Models, in: 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 1051–1060.

[10] L. Chu, L. Wang, Y. Dong, J. Pei, Z. Zhou, Y. Zhang, FedFair: Training Fair Models In Cross-Silo Federated Learning (2021). arXiv:2109.05662, doi:10.48550/arXiv.2109.05662.

[11] W. Du, D. Xu, X. Wu, H. Tong, AgnosticFair: Fairness-aware Agnostic Federated Learning, in: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), Proceedings, 2021, pp. 181–189. doi:10.1137/1.9781611976700.21.

[12] B. Rodríguez-Gálvez, F. Granqvist, R. van Dalen, M. Seigel, FPFL: Enforcing fairness in private federated learning via the modified method of differential multipliers, in: NeurIPS 2021 Workshop Privacy in Machine Learning, 2022.

[13] S. Liu, Y. Ge, S. Xu, Y. Zhang, A. Marian, F2MF: Fairness-aware Federated Matrix Factorization, in: Proceedings of the 16th ACM Conference on Recommender Systems, 2022, pp. 168–178.

[14] S. Cui, W. Pan, J. Liang, C. Zhang, F. Wang, FCFL: Addressing Algorithmic Disparity and Performance Inconsistency in Federated Learning, in: NeurIPS, Vol. 34, 2021, pp. 26091–26102.

[15] S. I. A. Meerza, L. Liu, J. Zhang, J. Liu, GLOCALFAIR: Jointly Improving Global and Local Group Fairness in Federated Learning (2024). arXiv:2401.03562.

[16] Y. H. Ezzeldin, S. Yan, C. He, E. Ferrara, A. S. Avestimehr, FairFed:Enabling Group Fairness in Federated Learning, Proceedings of the AAAI Conference on Artificial Intelligence 37 (6) (2023) 7494–7502.

[17] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Innovations in Theoretical Computer Science Conference, ITCS '12, New York, NY, USA, 2012, pp. 214–226.

[18] S. Caton, C. Haas, Fairness in Machine Learning: A Survey, ACM Computing Surveys 56 (7) (2024) 1–38.

[19] D. Pessach, E. Shmueli, A Review on Fairness in Machine Learning, ACM Computing Surveys 55 (3) (2023) 1–44. doi:10.1145/3494672.

[20] M. Hardt, E. Price, E. Price, N. Srebro, Equality of Opportunity in Supervised Learning, in: Advances in Neural Information Processing Systems, Vol. 29, 2016.

[21] F. P. Calmon, D. Wei, K. N. Ramamurthy, K. R. Varshney, Optimized Data Pre-Processing for Discrimination Prevention (2017). arXiv:1704.03354.

[22] S. Biswas, H. Rajan, FairPreprocessing: Towards Understanding Compositional Fairness of Data Transformers in Machine Learning Pipeline, in: The ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2021, pp. 981–993.

[23] Y. Roh, K. Lee, S. E. Whang, C. Suh, Fairbatch: Batch selection for model fairness, in: International Conference on Learning Representations, 2021.

[24] M. B. Zafar, I. Valera, M. Gomez Rodriguez, K. P. Gummadi, Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment, in: International Conference on World Wide Web, Perth Australia, 2017, pp. 1171–1180.

[25] M. B. Zafar, I. Valera, M. G. Rogriguez, K. P. Gummadi, Fairness Constraints: Mechanisms for Fair Classification, in: International Conference on Artificial Intelligence and Statistics, 2017, pp. 962–970.

[26] Y. Wu, L. Zhang, X. Wu, On Convexity and Bounds of Fairness-aware Classification, in: The World Wide Web Conference, 2019, pp. 3356–3362.

[27] S. Chiappa, Path-Specific Counterfactual Fairness, Proceedings of the AAAI Conference on Artificial Intelligence 33 (01) (2019) 7801–7808. doi:10.1609/aaai.v33i01.33017801.

[28] T. Huang, W. Lin, W. Wu, L. He, K. Li, A. Zomaya, RBCS-F: An Efficiency-boosting Client Selection Scheme for Federated Learning with Fairness Guarantee, IEEE Transactions on Parallel and Distributed Systems (2020) 1–1.

[29] T. Nishio, R. Yonetani, FedCS: Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge, in: IEEE International Conference on Communications (ICC), 2019, pp. 1–7. arXiv:1804.08333, doi:10.1109/ICC.2019.8761315.

[30] L. Gao, L. Li, Y. Chen, W. Zheng, C. Xu, M. Xu, FIFL: A Fair Incentive Mechanism for Federated Learning, in: International Conference on Parallel Processing, Lemont IL USA, 2021, pp. 1–10. doi:10.1145/3472456.3472469.

[31] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, Q. Yang, FLI:A Fairness-aware Incentive Scheme for Federated

Learning, in: AIES '20: AAAI/ACM Conference on AI, Ethics, and Society, New York NY USA, 2020, pp. 393–399. doi:10.1145/3375627.3375840.

[32] M. Mohri, G. Sivek, A. T. Suresh, Agnosticfl: Agnostic federated learning, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 4615–4625.

[33] T. Li, M. Sanjabi, A. Beirami, V. Smith, Q-FFL: Fair Resource Allocation in Federated Learning, in: International Conference on Learning Representations, 2020.

[34] T. Li, S. Hu, A. Beirami, V. Smith, Ditto: Fair and Robust Federated Learning Through Personalization, in: International Conference on Machine Learning, 2021, pp. 6357–6368.

[35] X. Yue, M. Nouiehed, R. Al Kontar, GIFAIR-FL: A Framework for Group and Individual Fairness in Federated Learning, INFORMS Journal on Data Science 2 (1) (2023) 10–23.

[36] A. Abay, Y. Zhou, N. Baracaldo, S. Rajamoni, E. Chuba, H. Ludwig, Mitigating Bias in Federated Learning (2020). arXiv:2012.02447.

[37] Y. Djebrouni, N. Benarba, O. Touat, P. De Rosa, S. Bouchenak, A. Bonifati, P. Felber, V. Marangozova, V. Schiavoni, Astral: Bias Mitigation in Federated Learning for Edge Computing, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 7 (4) (2023) 1–35. doi:10.1145/3631455.

[38] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, R. Vidal, Fedem: Federated multi-task learning under a mixture of distributions, in: Advances in Neural Information Processing Systems, 2021, pp. 15434–15447.

[39] F. Ding, M. Hardt, J. Miller, L. Schmidt, Folktables: Retiring Adult: New Datasets for Fair Machine Learning, in: Advances in Neural Information Processing Systems, Vol. 34, 2021, pp. 6478–6490.

[40] T.-M. H. Hsu, H. Qi, M. Brown, Measuring the effects of non-identical data distribution for federated visual classification, arXiv preprint arXiv:1909.06335 (2019).