Collaborative Fairness in Federated Learning

Lingjuan Lyu[1*] , **Xinyi Xu**[1*] and **Qian Wang**[3*]
[1]Department of Computer Science, National University of Singapore, Singapore
[3]School of Cyber Science and Engineering, Wuhan University
Corresponding to: lyulj@comp.nus.edu.sg, xuxinyi@comp.nus.edu.sg, qianwang@whu.edu.cn.

Abstract

In current deep learning paradigms, local training or the *Standalone* framework tends to result in overfitting and thus poor generalizability. This problem can be addressed by *Distributed* or *Federated Learning* (FL) that leverages a parameter server to aggregate model updates from individual participants. However, most existing Distributed or FL frameworks have overlooked an important aspect of participation: collaborative fairness. In particular, all participants can receive the same or similar models, regardless of their contributions. To address this issue, we investigate the collaborative fairness in FL, and propose a novel *Collaborative Fair Federated Learning* (CFFL) framework which utilizes reputation to enforce participants to converge to different models, thus achieving fairness without compromising the predictive performance. Extensive experiments on benchmark datasets demonstrate that CFFL achieves high fairness, delivers comparable accuracy to the *Distributed* framework, and outperforms the *Standalone* framework. Our code is available on github.

1 Introduction

Training complex deep neural networks on large-scale datasets is computationally expensive so it may not be feasible by a single participant. Moreover, training a complex model on a limited local dataset may lead to poor generalizability. In this context, Federated Learning (FL) emerged as a promising paradigm, as it provides a way for multiple devices/participants to jointly train a model whiling keeping their datasets local. The objective of FL is to derive a global model with better generalizability by leveraging the local datasets from multiple participants [1].

However, most of the current FL paradigms [1–4] allow all participants to receive the same federated model in each communication round and in the end, regardless of their contributions. This is obviously unfair, because in practice not all participants contribute equally due to various reasons, such as the diverse quality/quantity of the data owned by different participants. Therefore, the data from some participants may lead to good model updates while updates from some other participants can even impair the model performance. Consider a motivating practical example: several banks may want to collaborate to build a credit score predictor for small and medium enterprises. However, larger banks with more data may be reluctant to train their local model based on high quality local data because sharing these high quality model parameters with smaller banks may potentially erode the market share of the larger bank [5]. Furthermore, as the paradigm cannot distinguish the participants with high contributions from the ones with relatively low contributions, it is vulnerable to free-riders. Hence, lacking collaborative fairness may hinder the formation and progress of a healthy FL ecosystem. Existing research on fairness mostly focuses on mitigating potential bias introduced to the model towards certain attributes [6, 7]. The problem of treating the FL participants fairly according to their contributions remains open [5].

For any proposed solution or framework to be practical, it is essential to achieve fairness *not* at the cost of model performance. In this work, we address this problem of treating FL participants fairly based on their contributions by proposing a *Collaborative Fair Federated Learning* (CFFL) framework. Unlike existing work such as [8] which requires external monetary incentives for good behaviour, CFFL makes fundamental changes to the learning process in FL so that the participants will receive models with performance commensurate with their contributions, instead of the same FL model. CFFL achieves collaborative fairness with a reputation mechanism, which evaluates the contributions of the participants in the learning process and iteratively updates their respective reputations. We highlight the practical relevance of our CFFL in horizontally federated learning (HFL) to businesses (H2B) [9], such as biomedical or financial institutions to whom collaborative fairness is very important.

Our work aims to achieve collaborative fairness in FL by adjusting the performance of the models allocated to each participant based on their contributions [10, 11]. Experiments on benchmark datasets demonstrate that CFFL achieves the highest fairness. In terms of utility, the accuracy of the most contributive participant in CFFL is comparable to that of the *Distributed* framework, and higher than that of the *Standalone* framework. In the following sections, we interchangeably use Distributed/Federated.

2 Related Work

In this section, we review the relevant literature on fairness in FL to position our research in relation to existing research.

One existing approach for promoting collaborative fairness among federated participants is based on incentive schemes [3]. In principle, participants shall receive payoffs commensurate with their contributions. Equal division is an example of egalitarian profit-sharing [12]. Under this scheme, the available total payoff at a given round is equally divided among all participants. Under the Individual profit-sharing scheme [12], each participant i's own contribution to the collective (assuming a singleton collective of i) is used to determine his share of the total payoff.

The Labour Union game [13] profit-sharing scheme determines a participant's share of the total payoff based on his marginal contribution to the utility of the collective formed by his predecessors. The Fair-value game scheme [13] is a marginal loss-based scheme. Under this scheme, a participant's share of the total payoff is determined according to the sequence of the participants leaving the collective. The Shapley game profit-sharing scheme [13] is also a marginal contribution-based scheme. Unlike the Labour Union game, Shapley game aims to eliminate the effect of the sequence in which the participants join the collective in order to obtain a fairer estimate of their marginal contributions to the collective. However, the complexity of this approach is exponential in the number of participants, making it prohibitively expensive for large-scaled FL in practice.

For gradient-based FL approaches, the gradient information can be regarded as a useful source of data. However, in these cases, output agreement-based rewards are hard to apply as mutual information requires a multi-task setting which is usually not present in such cases. Thus, among these three categories of schemes, model accuracy is the most relevant way of designing rewards for FL. There are two emerging federated learning incentive schemes focused on model improvement.

Richardson *et al.* [14] proposed a scheme which pays for marginal improvements brought about by model updates. The sum of improvements might result in overestimation of contribution. Thus, the proposed approach also includes a model for correcting the overestimation issue. This scheme ensures that payment is proportional to model quality improvement, which means the budget for achieving a target model quality level is predictable. It also ensures that data owners who submit model updates early receive a higher reward. This in turn motivates them to participate even in early stages of the federated model training process.

Yu *et al.* [8] proposed a joint objective optimization-based approach that in addition to the contributions of the participants, takes costs and waiting time into account in order to achieve additional notions of fairness when distributing payoffs to the FL participants.

3 The CFFL Framework

3.1 Collaborative Fairness

Different from the existing approaches, our proposed framework provides an alternative paradigm, in which participants are allocated with different versions of the FL model with performance commensurate with their contributions. Under this context, we define collaborative fairness as follows.

Definition 1. *Collaborative fairness. In a federated system, a high-contribution participant should be rewarded with a better performing local model than a low-contribution participant. Mathematically, fairness can be quantified by the correlation coefficient between the contributions of participants and their respective final model accuracies.*

3.2 Fairness via Reputation

In our CFFL, we modify FL by allowing participants to download only the *allocated aggregated updates according to their reputations*. The server manages a reputation list for all participants, and updates it according to the quality of the uploaded gradients of each participant in each communication round. The upload rate – θ_u – denotes the proportion of parameters of which gradients are uploaded, i.e., if $\theta_u = 1$, gradients of all parameters are uploaded; if $\theta_u = 0.1$, gradients of only 10% the parameters are uploaded. We further denote the selected set of gradients as S, corresponding to θ_u gradients selected according to the "*largest values*" criterion: sort the gradients in $\Delta \boldsymbol{w}_j$ (by their magnitude), and upload θ_u of them, starting from the largest. Specifically, the server separately evaluates the validation accuracy of participant j by integrating j's uploaded gradients. In particular, if $\theta_u = 1$, $\Delta(\boldsymbol{w}_j)^S \triangleq \Delta \boldsymbol{w}_j$, the server can derive participant j's entire model $\boldsymbol{w}_j$, as all participants are initialized with the same parameters in the beginning. The server then computes the validation accuracy of participant j based on $\boldsymbol{w}_j$ as $vacc_j \leftarrow V(\boldsymbol{w}_j + \Delta(\boldsymbol{w}_j)^S)$, here V denotes the validation dataset. If $\theta_u \neq 1$, the server simply integrates participant j's uploaded gradients $\Delta(\boldsymbol{w}_j)^S$ into an auxiliary model $\boldsymbol{w}_g$ kept by the server to compute participant j's validation accuracy as $vacc_j \leftarrow V(\boldsymbol{w}_g + \Delta(\boldsymbol{w}_j)^S)$. Note $\boldsymbol{w}_g$ is an auxiliary model maintained by the server to aggregate gradients and calculate participants' reputations, and its parameters are *not* broadcast to individual participants as in the standard FL systems.

Then the server normalizes $vacc_j$ and passes the normalized $vacc_j$ through a $sinh(\alpha)$ function in Eq. (1) to calculate the reputation c_j of participant j in each communication round.

$$c_j = sinh(\alpha * x) \tag{1}$$

Here x is the normalized $vacc_j$, so the higher x, the more informative participant j's uploaded gradients are. $sinh(\alpha)$ is introduced as a *punishment function*, and α denotes the *punishment factor*, used to distinguish the reputations among participants based on how informative their uploaded gradients are. The server iteratively updates the reputation of each participant separately based on the calculated reputation in each round and its historical reputation. The high-contribution participant will be highly rated by the server, while the low-contribution participant can be detected and even isolated from the federated system, avoiding the low-contribution participants from dominating the whole system, or free-riding.

This computed reputation determines the number of aggregated gradients each participant will be allocated in the subsequent communication round. The higher the reputation of participant j, the more aggregated gradients will be allocated to participant j. The aggregated gradients refer to the collection of gradients from all participants, and are used here as a form of reward in each communication round. The detailed realization of CFFL is in Algorithm 1. In each communication round, each participant uploads θ_u fraction of clipped gradients to the server, and server updates the reputation based on the performance of these uploaded gradients on a validation set, and determines the number of aggregated updates to allocate to each participant. We adopt gradient clipping to reduce the impact of noise from abnormal examples/outliers.

3.3 Quantification of Fairness

In this work, we quantify collaborative fairness via the correlation coefficient between participant contributions (X-axis: test accuracies of standalone models which characterize their individual learning capabilities on their own local datasets) and participant rewards (Y-axis: test accuracies of final models received by the participants).

Participants with higher standalone accuracies empirically contribute more. Therefore, the X-axis can be expressed by Equation 2, where $sacc_j$ denotes the standalone model accuracy of participant j:

$$\boldsymbol{x} = \{sacc_1, \cdots, sacc_n\} \tag{2}$$

Similarly, Y-axis can be expressed by Equation 3, where acc_j represents the final model accuracy of participant j:

$$\boldsymbol{y} = \{acc_1, \cdots, acc_n\} \tag{3}$$

As the Y-axis measures the respective model performance of different participants after collaboration, it is expected to be positively correlated with the X-axis for a good measure of fairness. Hence, we formally quantify collaborative fairness in Equation 4:

$$\rho_{X,Y} = \frac{COV(X,Y)}{D(X)D(Y)} \in [-1,1]$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \tag{4}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of $\boldsymbol{x}$ and $\boldsymbol{y}$, s_x and s_y are the corrected standard deviations. The range of fairness falls within [-1,1], with higher values implying good fairness. Conversely, negative coefficient implies poor fairness.

4 Experimental Evaluation

4.1 Datasets

We implement experiments on two benchmark datasets. The first is the MNIST dataset[1] for handwritten digit recognition, consisting of 60,000 training examples and 10,000 test examples. The second is the Adult Census dataset[2]. This dataset is commonly used to predict whether an individual makes over 50K dollars in a year (binary). There are total 48,843 records, we manually balance the dataset to have 11687 records over 50K and 11687 records under 50K, resulting in total 23374 records. We then conduct an 80-20 train-test split. For all datasets, we randomly choose 10% of training examples as the validation set.

Algorithm 1 Collaborative Fair Federated Learning

```
Input: reputable participant set $R$; auxiliary model $\boldsymbol{w_g}$ kept by server; local model $\boldsymbol{w}_j$; local model updates $\Delta\boldsymbol{w}_j$; upload rate $\theta_u$; validation set $V$; local epochs $E$; $c_j^o$: reputation of previous round; $D_j$: data owned by each participant; data shard vector $n = \{n_1, \cdots, n_{|R|}\}$; class shard vector $class = \{class_1, \cdots, class_{|R|}\}$.
Role: participant $j$
  if $j \in R$ then
    Runs SGD on local data by using current local model $\boldsymbol{w}_j$ and computes gradient vector: $\Delta\boldsymbol{w}_j \leftarrow \mathtt{SGD}(\boldsymbol{w}_j, D_j)$
    Clips gradient vector: $\Delta\boldsymbol{w}_j \leftarrow clip(\Delta\boldsymbol{w}_j)$
    Sends the selected gradients $\Delta(\boldsymbol{w}_j)^S$ of size $\theta_u * |\Delta\boldsymbol{w}_j|$ to the server, according to the "largest values" criterion;
    Downloads the allocated updates from the server, which is then integrated with all its local updates as: $\boldsymbol{w}'_j \leftarrow \boldsymbol{w}_j + \Delta\boldsymbol{w}_j + \Delta\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{j}} - \frac{n_j}{max(n)}\Delta(\boldsymbol{w}_j)^S$ (imbalanced data size) or $\boldsymbol{w}'_j \leftarrow \boldsymbol{w}_j + \Delta\boldsymbol{w}_j + \Delta\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{j}} - \frac{class_j}{max(class)}\Delta(\boldsymbol{w}_j)^S$ (imbalanced class number).
  end if

Role: Server
  Updates aggregation:
  if data size is imbalanced then
    $\Delta\boldsymbol{w}_g \leftarrow \sum_{j \in R}\Delta(\boldsymbol{w}_j)^S \times \frac{n_j}{sum(n)}$.
  end if
  if class number is imbalanced then
    $\Delta\boldsymbol{w}_g \leftarrow \sum_{j \in R}\Delta(\boldsymbol{w}_j)^S \times \frac{class_j}{max(class)}$.
  end if
  if $\theta_u = 1$ then
    for $j \in R$ do
      $vacc_j \leftarrow V(\boldsymbol{w}_j + \Delta(\boldsymbol{w}_j)^S)$.
      Updates local model of participant $j$ kept by the server: $\boldsymbol{w}'_j \leftarrow \boldsymbol{w}_j + \Delta(\boldsymbol{w}_j)^S$ for next round of reputation evaluation.
    end for
  else
    for $j \in R$ do
      $vacc_j \leftarrow V(\boldsymbol{w}_g + \Delta(\boldsymbol{w}_j)^S)$.
    end for
    Updates temp model maintained by server $\boldsymbol{w}'_g = \boldsymbol{w}_g + \Delta\boldsymbol{w}_g$ for next round of reputation evaluation.
  end if
  for $j \in R$ do
    $c_j \leftarrow sinh(\alpha * \frac{vacc_j}{\sum_{j \in R} vacc_j})$, $c'_j \leftarrow c_j^o * 0.5 + c_j * 0.5$
                  ?
  end for
  Reputation normalisation: $c'_j \leftarrow \frac{c'_j}{\sum_{j \in R} c'_j}$
  if ${c_j}' < c_{th}$ then
    $R \leftarrow R \setminus \{j\}$, repeat reputation normalisation.
  end if
  for $j \in R$ do
    if data size is imbalanced then
      $num_j \leftarrow \frac{c'_j}{max(c)} * \frac{n_j}{max(n)} * |\Delta\boldsymbol{w}_g|$
    end if
    if class number is imbalanced then
      $num_j \leftarrow \frac{c'_j}{max(c)} * \frac{class_j}{max(class)} * |\Delta\boldsymbol{w}_g|$
    end if
    Groups $num_j$ aggregated updates into $\Delta\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{j}}$ according to the "largest values" criterion, and allocates an adjusted version $\Delta\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{j}} - \frac{n_j}{max(n)}\Delta(\boldsymbol{w}_j)^S$ (imbalanced data size) or $\Delta\boldsymbol{w}_{\boldsymbol{g}}^{\boldsymbol{j}} - \frac{class_j}{max(class)}\Delta(\boldsymbol{w}_j)^S$ (imbalanced class number) to participant $j$.
  end for
```

[1] http://yann.lecun.com/exdb/mnist/

[2] http://archive.ics.uci.edu/ml/datasets/Adult

4.2 Baselines

We demonstrate the effectiveness of our proposed CFFL framework through comparison with the following frameworks.

Standalone framework allows participants to train standalone models on local datasets without collaboration. This framework delivers minimum utility, because each participant is susceptible to falling into local optima when training alone. In addition, we remark that there is no concrete concept of collaborative fairness in the *Standalone* framework, because participants do not collaborate.

Distributed framework enables participants to train independently and concurrently, and share their gradients or model parameters to achieve a better global model. For comparison, we choose two representative *Distributed* baselines, including FedAvg [1] and DSSGD [15].

Furthermore, we investigate different upload rates $\theta_u = 0.1$ and $\theta_u = 1$ [15], where gradients are uploaded according to the "largest values" criterion when $\theta_u = 0.1$. The rationale behind introducing an upload rate less than 1 is to reduce overfitting and to save communication overhead.

4.3 Experimental Setup

Due to the fact that the data are often heterogeneous across participants both in terms of size and distribution, we investigate the following two realistic scenarios:

Imbalanced data size. To simulate data size heterogeneity, we follow a power law to randomly partition total {3000,6000,12000} MNIST examples among {5,10,20} participants respectively. Similarly, for Adult dataset, {4000,8000,12000} examples are randomly partitioned among {5,10,20} participants. In this way, each participant has a distinctly different number of examples, with the first participant has the least and the last participant has the most. We remark that the purpose of allocating on average 600 MNIST examples for each participant is to fairly compare with Shokri *et al.* [15], where each participant has a small number of 600 local examples to simulate data scarcity which necessitates collaboration.

Imbalanced class numbers. To examine data distribution heterogeneity, we vary the class numbers in the data of each participant, increasing from the first participant to the last. For this scenario, we only investigate the MNIST dataset. We distribute classes in a linspace manner, for example, participant-$\{1, 2, 3, 4, 5\}$ own {1,3,5,7,10} classes from MNIST dataset respectively. In more detail, for MNIST with total 10 classes and 5 participants, we simulate the first participant has data from only 1 class, while the last participant has data from 10 classes. We first partition the training dataset according to the labels, and then sample and assign subsets of training data with corresponding labels to the participants. Note that each participant still has the same number of examples, *i.e.*, 600 examples.

Model and Hyper-Parameters. For MNIST *Imbalanced data size* experiment, we use a two-layer fully connected neural network with 128 and 64 units respectively. The hyperparameters are: local epochs $E = 2$, local batch size $B = 16$, and local learning rate $lr = 0.15$ for $P = 5$ and $lr = 0.25$ for $P = \{10, 20\}$, with exponential decay of $\gamma = 0.977$, gradient clipping between $[-0.01, 0.01]$, with a total of 30 communication rounds. For MNIST *Imbalanced class numbers* experiment, the same neural network architecture is used. The hyperparameters are: local epochs $E = 1$, local batch size $B = 16$, and local learning rate $lr = 0.15$ for $P = \{5, 10, 20\}$, with exponential decay of $\gamma = 0.977$, gradient clipping between $[-0.01, 0.01]$, with a total of 50 communication rounds. For Adult, we use a single layer fully connected neural network with 32 units. The hyperparameters are: local epochs $E = 2$, local batch size $B = 16$, and local learning rate $lr = 0.03$ with exponential decay of $\gamma = 0.977$, gradient clipping between $[-0.01, 0.01]$, with a total of 30 communication rounds.

Furthermore, to reduce the impact of different initializations and avoid non-convergence, we initialize the same model parameter $\boldsymbol{w}_0$ for all participants and the server $\boldsymbol{w}_g$. For all the experiments, we empirically set the reputation threshold via grid search as follows: $c_{th} = \frac{1}{|R|} * \frac{1}{3}$ for imbalanced data size, and $c_{th} = \frac{1}{|R|} * \frac{1}{6}$ for imbalanced class numbers, where $|R|$ is the number of participants with reputations higher than the threshold. For the punishment factor, we empirically choose $\alpha = 5$. Stochastic Gradient Descent(SGD) is used as the optimization technique throughout.

Communication Protocol. In standard FL, one global model is given to all participants, both during and at the end of the training process. Such a setup forbids the calculation of our definition of fairness via the pearson coefficient, when all participants have the same reward. To mitigate this, we follow [15] to adopt the round-robin communication protocol for DSSGD and FedAvg. In each communication round, participants upload *parameter updates* and download *parameters* in sequence, leading to models with insignificant performance differences, so that their test accuracies can be used for the calculation of fairness.

4.4 Experimental Results

Fairness comparison. Table 1 lists the calculated fairness of DSSGD, FedAvg and CFFL over MNIST and Adult under varying participant number settings from $\{5, 10, 20\}$, different pretraining status from $\{1, 0\}$, and different upload rates θ_u from $\{0.1, 1\}$. From the high values of fairness (some close to the theoretical limit of 1.0), we conclude that CFFL achieves good fairness, confirming the intuition behind our notion of fairness: the participants with higher contributions are rewarded with better-performing models. Moreover, pretrain=1 can lead to slightly higher fairness than pretrain=0. This is attributed to the individual pretraining of 5 epochs before collaborative learning starts, because the participants' models have already moved towards their respective model optimum. Note that pretraining is *only* conducted for CFFL. We also observe that DSSGD and FedAvg yield significantly lower fairness than our CFFL. This is expected since neither the communication protocol nor the learning algorithm incorporates the concept of fairness.

Accuracy comparison. Table 2 reports the corresponding accuracies on MNIST and Adult datasets of $\{5, 10, 20\}$ participants when $\theta_u = 0.1$. Here we report the best accuracy achieved among the participants, because CFFL enables

Table 1: Fairness [%] of DSSGD, FedAvg and CFFL under varying participant number settings (P-k), pretraining status and upload rate θ_u. .

Dataset	MNIST							Adult						
Framework	FedAvg	DSSGD		CFFL				FedAvg	DSSGD		CFFL			
Pretrain	NA	NA		1		0		NA	NA		1		0	
θ_u	NA	0.1	1	0.1	1	0.1	1	NA	0.1	1	0.1	1	0.1	1
P5	3.08	90.72	84.61	99.63	98.66	**99.76**	99.02	-3.33	15.61	35.71	98.50	97.75	98.44	**99.37**
P10	-50.47	-78.18	90.67	97.90	97.30	98.55	**98.74**	44.27	62.30	56.60	88.00	**93.07**	92.00	91.95
P20	60.41	-81.77	80.45	**99.23**	96.28	98.52	98.51	-34.32	60.30	58.01	**84.41**	82.46	80.56	79.52

Table 2: Maximum Accuracy [%] over MNIST and Adult of varying participant number settings, achieved by DSSGD, FedAvg, *Standalone* framework, and our CFFL ($\theta_u = 0.1$, where CFFL* denotes CFFL with pretraining).

Framework	MNIST			Adult		
	P5	P10	P20	P5	P10	P20
DSSGD	93.28	94.20	82.36	81.94	82.78	82.07
FedAvg	**93.62**	**95.32**	**96.26**	**82.58**	**83.14**	**83.16**
Standalone	90.30	90.88	90.64	81.93	82.31	82.07
CFFL	91.83	93.00	93.25	81.96	82.63	82.72
*CFFL**	91.85	92.85	93.34	81.89	82.63	82.63

participants to converge to different final models, so we expect the most contributive participant to receive a model with the highest accuracy comparable to both *Distributed* frameworks. For the *Standalone* framework, we show the accuracy of the same participant. It can be observed that CFFL obtains comparable accuracy to DSSGD and FedAvg, and always attains higher accuracy than the *Standalone* framework. For example, for MNIST with 20 participants, our CFFL (CFFL*) achieves 93.25 (93.34)% test accuracy, higher than the *Standalone* framework (90.64%), and slightly lower than FedAvg (96.26%). The observation that DSSGD achieves lowest accuracy in this setting can be attributed to its higher instability and fluctuations during training.

Individual model performance. To examine the impact of our CFFL on individual convergence, Figure 1 plots the test accuracy of each participant for the *Standalone* framework and CFFL with upload rate of $\{0.1, 1\}$ and with/without pretraining over MNIST across 30 (communication) rounds. It can be observed that our CFFL consistently delivers better accuracy than the standalone model of any participant. Importantly, it confirms that our CFFL enforces the participants to converge to different local models, which are still better than their standalone models without collaboration, thereby offering fairness and utility as claimed. We also observe slight fluctuations at the beginning of training. This can be attributed to the fact that participants are allocated with different aggregated updates from the server. As we can see from these figures, the convergence curves for CFFL (with pretrain) and CFFL (w/o pretrain) follow the similar trend, confirming that pretraining does not alter the overall convergence behaviour, but provides relatively better fairness in most cases.

For imbalanced class numbers, Figure 3 shows individual model accuracy in the *Standalone* framework and our CFFL. We see that all participants achieve higher accuracies in CFFL than their standalone counterparts. Similar to the scenario of imbalanced data size, all participants converge to different final models, but with more obvious accuracy gaps, resulting in higher fairness. Moreover, it takes longer for participants to converge when there are more participants in the system.

5 Discussions

Robustness to Free-riders. In an FL system, there may exist free-riders who aim to benefit from the global model, without really contributing. Typically, free-riders may pretend to be contributing by uploading random or noisy updates. In standard FL systems, there is no specific safeguard against this, so even free-riders can enjoy the system's global model at virtually no cost. Conversely, CFFL can automatically identify and isolate free-riders. This is because the empirical utility (on the validation set) of the random or noisy gradients is generally low. As collaborative training proceeds, the free-riders will receive gradually lower reputations, and eventually be isolated from the system when their reputations fall below the reputation threshold. Through our additional experiments (including 1 free rider who always uploads random values as gradients), we observe that our CFFL can always identify and isolate the free rider at the early stages of collaboration, without affecting both accuracy and convergence.

Choice of Reputation Threshold. Using a reputation threshold c_{th} allows the server to enforce a lower bound on the reputation. This can be used to detect and isolate the free-rider in the system. A key challenge lies in the selection of an appropriate threshold, as fairness and accuracy may be inversely affected. For example, too small c_{th} might allow low-contribution participant to sneak into the federated system without being detected and isolated. On the contrary, too large c_{th} might isolate too many participants in the system. In our experiments, we empirically find suitable values for different scenarios.

6 Conclusion and Future Work

This work initiates the research on collaborative fairness in federated learning (FL), and modifies FL to enforce participants to converge to different models. A novel collaborative fair federated learning framework named CFFL is proposed. Based on empirical individual model performance on a validation set, a reputation mechanism is introduced to mediate participant rewards across communication rounds. Experimental results demonstrate that CFFL achieves comparable accuracy to two *Distributed* frameworks, and always achieves better accuracy than the *Standalone* framework, confirming the effectiveness of CFFL in terms of both *fairness* and *utility*. A number of avenues for further work are appealing. In particular, we would like to study how to quantify fairness

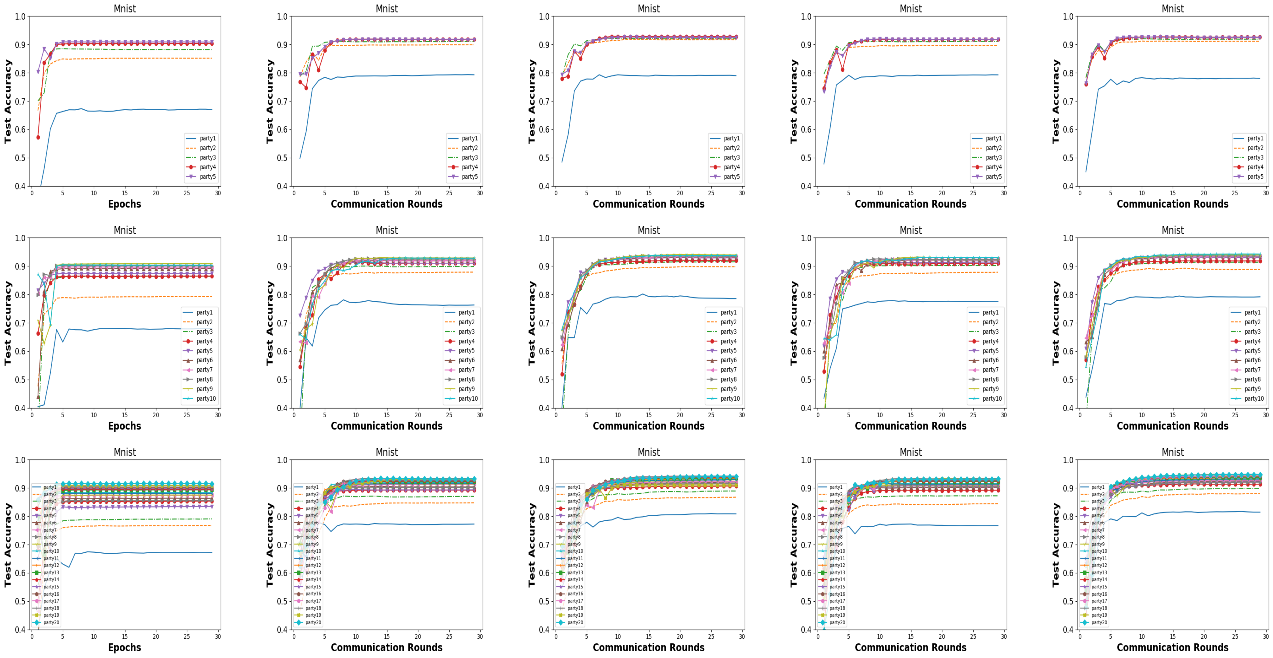

Figure 1: Individual convergence for MNIST using *Standalone* framework and our CFFL. The 3 rows correspond to $\{5, 10, 20\}$ participants, the 5 columns correspond to $\{$Standalone, CFFL with $\theta_u = 0.1$ and pretrain, CFFL with $\theta_u = 1$ and pretrain, CFFL with $\theta_u = 0.1$ but without pretrain, CFFL with $\theta_u = 1$ but without pretrain$\}$.

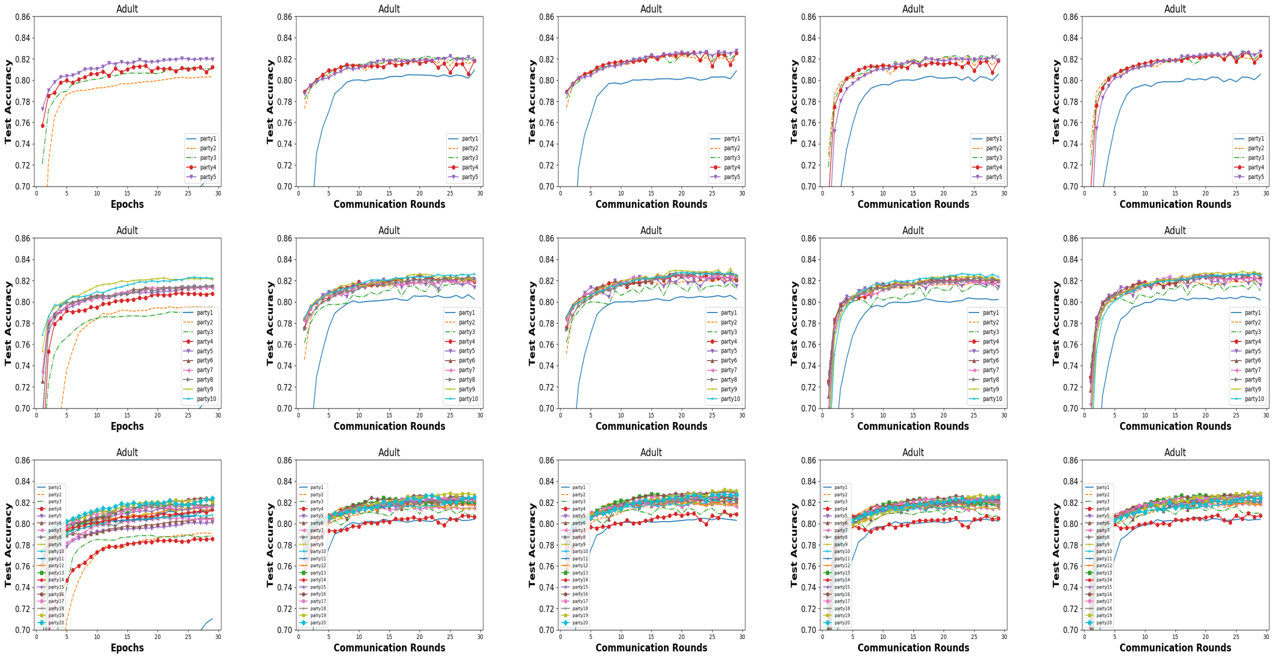

Figure 2: Individual convergence for Adult using *Standalone* framework and our CFFL. The 3 rows correspond to $\{5, 10, 20\}$ participants, the 5 columns correspond to $\{$Standalone, CFFL with $\theta_u = 0.1$ and pretrain, CFFL with $\theta_u = 1$ and pretrain, CFFL with $\theta_u = 0.1$ but without pretrain, CFFL with $\theta_u = 1$ but without pretrain$\}$.

in more complex settings, and apply our framework to various domains, such as financial, biomedical, speech, NLP, etc. Furthermore, we would like to systematically integrate robustness with fairness. It is expected that our system can find wide applications in real world.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial Intelligence and*

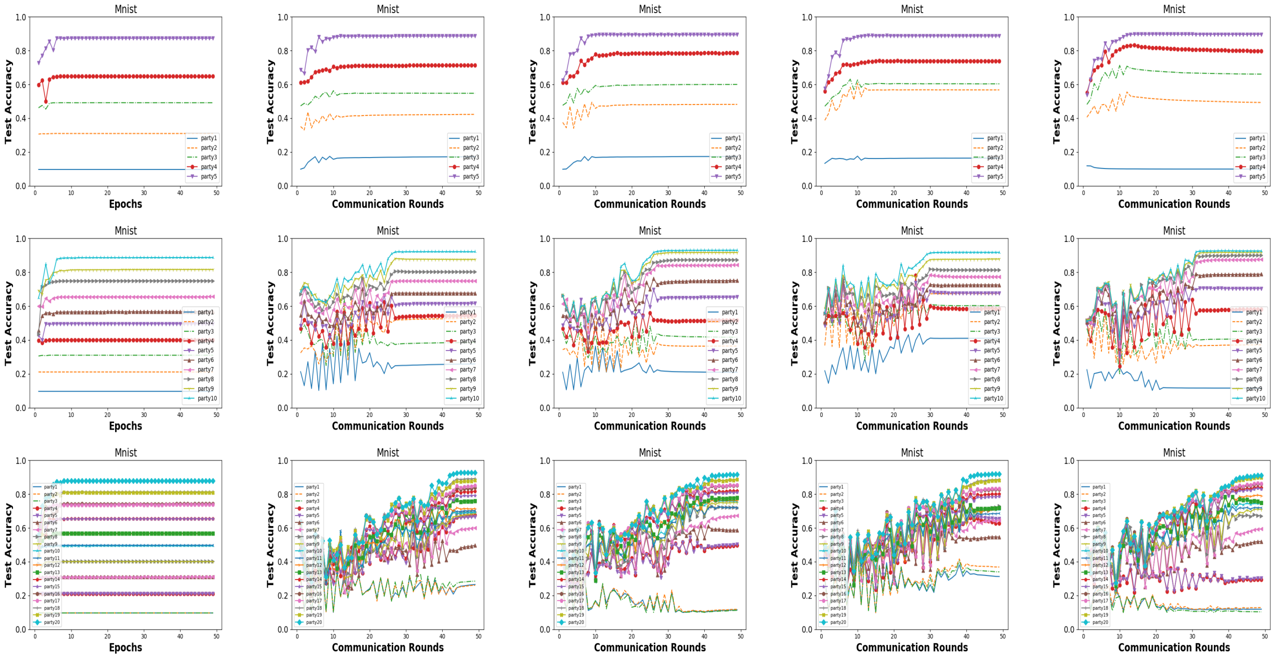

Figure 3: Individual model accuracy for MNIST class imbalanced scenario, where classes are distributed in a linspace manner (for example, participant-$\{1, 2, 3, 4, 5\}$ own $\{1,3,5,7,10\}$ classes respectively). 5 columns correspond to {Standalone, CFFL $\theta_u = 0.1$ w pretrain, CFFL $\theta_u = 1$ w pretrain, CFFL $\theta_u = 0.1$ w/o pretrain, CFFL $\theta_u = 1$ w/o pretrain}.

Statistics, 2017, pp. 1273–1282.

[2] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al., "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[3] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[4] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith, "Federated learning: Challenges, methods, and future directions," *CoRR, arXiv:1908.07873*, 2019.

[5] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu, *Federated Learning*, Morgan & Claypool Publishers, 2019.

[6] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern, "On the compatibility of privacy and fairness," 2019.

[7] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman, "Differentially private fair learning," *arXiv preprint arXiv:1812.02696*, 2018.

[8] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang, "A fairness-aware incentive scheme for federated learning," in *Proceedings of the 3rd AAAI/ACM Conference on AI, Ethics, and Society (AIES-20)*, 2020, pp. 393–399.

[9] Lingjuan Lyu, Han Yu, and Qiang Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.

[10] Lingjuan Lyu, Jiangshan Yu, Karthik Nandakumar, Yitong Li, Xingjun Ma, Jiong Jin, Han Yu, and Kee Siong Ng, "Towards fair and privacy-preserving federated deep models," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2524–2541, 2020.

[11] Lingjuan Lyu, Yitong Li, Karthik Nandakumar, Jiangshan Yu, and Xingjun Ma, "How to democratise and protect ai: Fair and differentially private decentralised deep learning," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[12] Shuo Yang, Fan Wu, Shaojie Tang, Xiaofeng Gao, Bo Yang, and Guihai Chen, "On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832–847, 2017.

[13] Sreenivas Gollapudi, Kostas Kollias, Debmalya Panigrahi, and Venetia Pliatsika, "Profit sharing and efficiency in utility games," in *ESA*, 2017, pp. 1–16.

[14] Adam Richardson, Aris Filos-Ratsikas, and Boi Faltings, "Rewarding high-quality data via influence functions," *arXiv preprint arXiv:1908.11598*, 2019.

[15] Reza Shokri and Vitaly Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1310–1321.