Hierarchically Fair Federated Learning

Jingfeng Zhang[*1], Cheng Li[*1], Antonio Robles-Kelly[2], and Mohan Kankanhalli[1]

[1]School of Computing, National University of Singpoare, Singapore
[2]Deakin University, Australia
[1] {*j-zhang, licheng, mohan*}@*comp.nus.edu.sg*
[2] *antonio.robles-kelly@deakin.edu.au*

Abstract

When the federated learning is adopted among competitive agents with siloed datasets, agents are self-interested and participate only if they are fairly rewarded. To encourage the application of federated learning, this paper employs a management strategy, i.e., more contributions should lead to more rewards. We propose a novel hierarchically fair federated learning (HFFL) framework. Under this framework, agents are rewarded in proportion to their pre-negotiated contribution levels. HFFL+ extends this to incorporate heterogeneous models. Theoretical analysis and empirical evaluation on several datasets confirm the efficacy of our frameworks in upholding fairness and thus facilitating federated learning in the competitive settings.

1 Introduction

Traditional machine learning techniques require agents (e.g., mobile devices, terminals, companies, etc.) to upload their data to a central server. This approach not only increases communication between agents and the central server due to the data volume but also entails privacy risks during data transfer or due to a server breach [1]. This is an important concern since data protection regulations impose constraints on sharing of sensitive data.

Federated learning, a recent distributed and decentralized machine learning scheme [2] has attracted significant attention. In federated learning, agents maintain their data locally and collaboratively learn a global machine learning model that benefits all. Specifically, each agent sends parameters (or parameters update) of local models to the central server and receives the computed parameters of the global model from the central server. In this way, all agents can jointly train a global model without exposing their own data. This scheme has desirable properties such as privacy-preservation, efficient communication, and decentralized data storage.

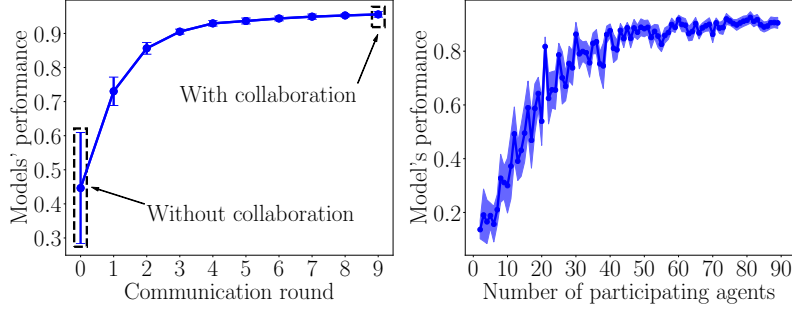[*]Equal Contributions. Preprint. Work in Progress.

Figure 1: *Left panel*: there are 30 agents (each having 1670 MNIST training data) participating in standard federated learning. At communication round 0, each agent learns a model locally. During communication rounds 1-9, they collaboratively share their data for federated learning. We report the median test accuracy and its standard deviation (error bar) over models' performance of 30 agents. *Right panel*: there are at most 90 agents (each having 550 MNIST training data) that participate in standard federated learning. We report the median test accuracy and its standard deviation (shaded color) over participating agents models' performance.

Federated learning could benefits all participating agents. As we can see in the left panel of Figure 1, at the end of communication round 0, each agent learns a model locally. Without federated learning, each agent can only learn from their local data. A model trained on local data only has poor generalization capability. Moreover, the performance of different agents' models has a large variance. This is because without federated learning, each agent has an incomplete and biased view of the global data distribution. At the end of every communication round 1 through 9, all agents send their model parameter updates to the central server and then receive the global model's parameters from the central server. The global model parameters are used to improve their own model locally. With federated learning, each agent gets significant (indirect) exposure to the global data distribution and as a result, the median test accuracy among agents increases from 45% to 95%. Moreover, each agent model's bias with respect to the global data distribution also substantially reduces. The right panel of Figure 1 illustrates the importance of the agents' participation. As more agents contribute their data and federally learn a global model, the overall performance of the global model improves. To conclude, federated learning benefits agents by improving agents' models.

The tacit assumption of federated learning is that all agents are willing to participate in the federated learning [2, 3, 4]. However, in real-world situations, this assumption may not necessarily hold. Agents with siloed datasets are typically under competition and self-interested so that the agents are reluctant to collaborate unless they get some reward which is fair [5, 6, 7, 8]. This necessitates the notion of fairness in federated learning.

Previous research on fairness in federated learning typically aims to optimize model performance across all agents, either maximizing the performance of the worst agent [3]

or uniforming the accuracy distribution over all agents [4]. However, both these approaches do not take into account the extent of an agent's contribution in the federated learning. More specifically, they tend to protect the weak agents (e.g., agents with small amounts of data) while neglecting the strong agents (e.g., agents with large amounts of data). As a result, the strong agents find it unfair on ignorance of their greater contribution. Consequently, strong agents may abort from the federated network sacrificing the small benefit or may choose to cooperate among themselves, excluding the weak ones. In order to obtain the full benefits of federated learning (see Figure 1), we need a fairness mechanism that is acceptable to all agents, weak or strong.

In this paper, we employ the management of reward schemes [9, 10] in federated learning in the sense that the agent who **contributes more** to the federated learning should be **rewarded more**. This fairness notion is applicable in many real-world situations. An illustrative example is that a company with a large amount of data would like not to cooperate with one that has a small amount of data if both receive the same reward (e.g., learn the same machine learning model). It usually exerts a lot of effort and resources to collect data, which is valuable. Thus, equal reward in this case is unfair. In order to facilitate federated learning among two companies, they should be rewarded differently based on their relative contribution of data. This fairness notion based on proportionality also finds support from social psychology [11], which advocates that an individual's outcomes (rewards) should match (be proportional to) his/her inputs (contributions). Similar ideas are also explored in game theory [12] and bandwidth allocation [13].

To apply this fairness notion to federated learning, two issues have to be addressed, i.e., **(a)** how to determine the extent of an agent's contribution to federated learning and **(b)** what is the proportionate reward that the agents should receive in order to achieve the fairness.

For the question **(a)**, data Shapley can perhaps be used to determine the extent of an agent's contribution since it is used to quantify data valuation [14, 15]. Specifically, the Shapley value of a datum computes the average of the marginal performance of this datum with any subsets of remaining data. The agent's contribution can then be measured by summing up the Shapley value of all data of that agent. For a pre-specified learning task, the Shapley value of an agent can be different for different chosen models – it is model-dependent [14]. Thus, data Shapley is not a consistent metric to agent's contribution for a pre-specified federated learning task. The self-interested agents may quit the collaboration due to perceived unfair treatment if the reward allocation is based on Shapley value. We elaborate this in Section 3.

Instead, we propose the use of *publicly verifiable factors* of agents to measure participating agent's contributions, such as the task-related data volume, data range, data collection cost, etc (details in Section 3). As long as agents should reach a consensus on the chosen publicly verifiable factors, they have to agree to be contractually bound [16] and commerce the federated learning. This approach circumvents the inconsistency issue of model-dependent methods such as data Shapley and influence functions [17].

For the second question **(b)** w.r.t. fair rewards, we propose a proportionate reward system i.e. agents who are deemed more valuable will receive more model updates. To achieve this, we first classify all agents into different levels based on their publicly verifiable factors. Agents at the same level are deemed to have the same contribution

and will be rewarded equally. Let us assume that low level agents contribute less and high level agents contribute more. Then we train multiple models at every level such that the high-level agents only contribute the roughly same amount of data as the low-level agents own when the low-level model is federally learned. On the other hand, high-level agents get access to the low-level models to federally learn their high-level models. Our theoretical analysis shows that a model with more training data can potentially have less generalization error. In such a federated learning framework, the agents at the same contribution level share the same model and the agents at a higher contribution level can share a better model, which aligns with the proportional fairness notion.

Our proposed method is called hierarchically fair federated learning (HFFL). Based on HFFL, we also design an improvement, namely, HFFL+. It allows different models (e.g., different structured deep neural networks) at different levels at the expense of training time which makes the framework is more flexible and capable. We run our algorithms on different datasets to test the fairness notion, i.e., agents in higher levels attain higher rewards.

Our main contributions in this paper are:

- We employ the reward schemes of the management in federated learning, i.e., more contribution leads to more reward.

- We propose a novel hierarchical federated learning framework to achieve proportional fairness so that it facilitate collaborations among agents.

- Empirical evaluation of our methods on four datasets, i.e., census dataset ADULT, vision datasets MNIST as well as Fashion MNIST and text dataset IMDB confirms our frameworks in upholding the fairness.

The rest of this paper is organized as follows. The related work is surveyed in Section 2. How to measure an agent's contribution is discussed in Section 3. The proposed federated learning framework and theoretical analysis are presented in Section 4. Experimental results are showed in Section 5 followed by the conclusion and future work in Section 6.

2 Related work

Fairness in federated learning. Fairness in machine learning is often defined as a notion of protecting against discrimination of some specific features in data, e.g., minorities. A number of prior studies has focused on addressing this feature-level fairness in models. Two commonly adopted strategies are per-processing sensitive features such as deletion or transformation [18] and modifying existing models to limit discrimination [19, 20]. Fairness has also been considered in resource division in multi-agent systems. In this context, the resource provided by the environment needs to fairly shared by all agents. Some typical work includes the maximin sharing policy [21], which improves the performance of the worst agent, and the fair-efficient policy which makes the variation of agents' utilities as small as possible [22].

In federated learning, existing work on fairness aims to ensure accuracy across all agents. For example, [3] proposed agnostic federated learning (AFL), which minimizes the maximal loss function of all agents with a consideration on the overall performance. [4] proposed q-Fair Federated Learning (q-FFL) to encourage a more uniform accuracy distribution across all agents, where q is a trade-off parameter between fairness and accuracy. They do not take the contribution of agents into consideration. Our notion of fairness is not to optimize the accuracy across all agents, but is to ensure that the agents which contribute more receive proportionately more reward.

Incentive design in federated learning. Our employed fairness is also a kind of an incentive to encourage participation of agents based on their resources. [17] used a similar incentive where the central server pays agents proportional to their data valuation. They employed influence functions for data valuation. Influence functions quantify how much the model's predictions would change if that datum was not used in the training process and is in fact the leave-out-one (LOO) method. The LOO method assigns zero value to the duplicate of a datum [14]. If two agents happened to have exactly the same data, then the extra data is deemed to be of zero value. Furthermore, the LOO method is also model-dependent, which has the same issue of data Shapley. In contrast, our work uses publicly verifiable factors of agents to identify agents' contributions, which are free from model dependency.

3 Identifying Contribution of Agents

Identifying agents' contribution is a key driver to the success of collaboration [23]. In this subsection, we first show why data Shapley is not a consistent metric of agents' contributions in federated learning. We then discuss economic and social factors that are used to measure different agents' contribution under the collaboration.

3.1 Data Shapley Is Not A Suitable Metric

For the same task, data valuation based on data Shapley is (1) model dependency, (2) negativity, and (3) evaluations metric dependency [14]. In this section, we verify points (1) and (2) and show why data Shapley is not a suitable measure of the contributions of the agents and hurdle their incentives to participate in the federated learning.

In machine learning, data Shapley is used to measure the value of each training datum towards the predictor performance. An agent's contribution is measured by summing the Shapley value of all data from the agent [14].

Given a learning algorithm $\mathcal{A}$ taking a training dataset $S \subseteq \mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ as input and returning a model, data Shapley is a metric to quantify the value of each training data towards the predictor performance. In supervised machine learning, the Shapley value ϕ_i of a datum $(\mathbf{x}_i, y_i)$ is calculated by

$$\phi_i = C \sum_{S \subseteq D - \{(\mathbf{x}_i, y_i)\}} \frac{V(S \cup \{(\mathbf{x}_i, y_i)\}) - V(S)}{\binom{n-1}{|S|}}$$

where C is a scaling constant and V is the performance score of the predictor trained on dataset S. $V(S)$ is in short for $V(\mathcal{A}(S))$.
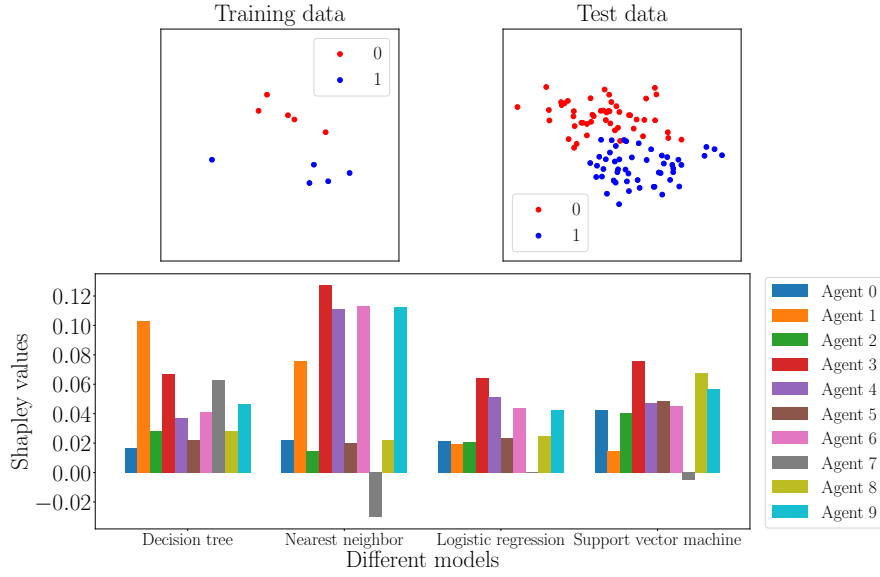
Figure 2: *Top*: training data and test data for calculating data Shapley. Each agent holds two data from training and uses the same test data. *Bottom*: Shapley values of different agents across different models for the binary classification.

We conduct an experiment to illustrate Shapley value of agents, where there are 10 agents, each holding only one data point at the top left panel in Figure 2. They collaboratively learn a binary classification model. The performance score $V(S)$ is calculated based on the accuracy of the returned predictor $\mathcal{A}(S)$ on test data in the top right panel. The bottom panel in Figure 2 shows the Shapley value across different agents over different learning algorithms, i.e., decision tree, nearest neighbor, logistic regression and support vector machine. For the same classification task, we can see that the contribution of agents measured by data Shapley varies significantly over the learning algorithms. For example, Shapley value of Agent 4 is largest when $\mathcal{A}$ uses nearest neighbor, but it becomes relatively smaller when decision tree is used.

In addition, agent 7 even has a negative Shapley value when nearest neighbor and support vector machine is used. It is unrealistic to expect self-interested agents to participate if they get negative rewards and pay the cost of contributing data. Furthermore, data with negative Shapley value might be valuable and cannot be neglected. These data points might be outliers which impact negatively on the performance of the predictor, but they may represent the rarest of cases that should be recognized and classified [24]. For example, they could be critical in areas such as medical diagnosis, IoT sensors, fraud detection and intrusion detection.

To conclude, for a specific learning task (classification or regression), self-interested agents can find reward allocation based on data Shapley to be unfair and may quit from the collaboration. Thus, data Shapley is not a suitable metric to identify an agent's contribution.
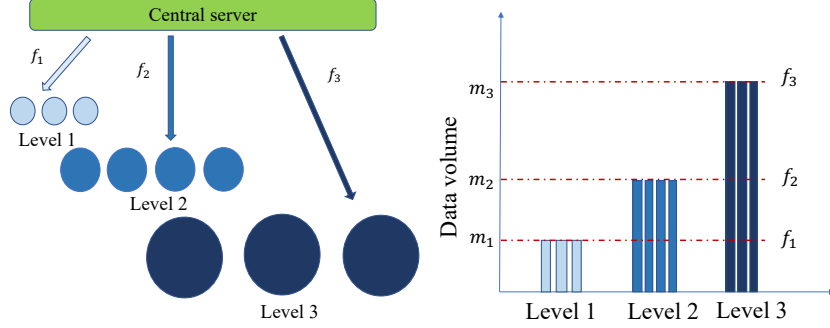
Figure 3: An illustrating example of hierarchically fair federated learning (HFFL). There are three levels ($L = 3$). The *left* figure presents the hierarchical structure, where agents are categorized into different levels for collaboration. The *right* figure shows how agents contribute their local data for federated learning. Agents at a lower level contribute less data and get a lesser reward (e.g., obtaining model f_1), and agent at a higher level contribute more data and get a better reward (e.g., obtaining model f_3).

3.2 Contribution Measures

Exactly identifying agents' contribution is often difficult in a federated network since multiple factors are involved such as agents' reputation, communication bandwidth, computation resource, data amount and quality, etc. Inspired by the rating mechanism in the finance industry such as Standard & Poor's which issues credit ratings with the levels from AAA to D for public and private companies, governmental entities and etc, we propose classifying all agents into multiple distinct contribution levels and then reward them based on their contribution levels.

We use the publicly verifiable factors of agents to classify them, such as data quality, data volume, cost of data collection, etc. Our strategy is more likely to be task-dependent rather than model-dependent. Agents should come to a consensus on which factors are relevant for the given task. It is also practically enforceable using contracts [16], and collaboration can be assured given fair rewards. This simple strategy is easy to understand. What is more, a trustworthy third party can also be employed to rate agents and classify them into different levels in practice.

4 Hierarchically Fair Federated Learning

In this section, we design a novel hierarchical federated learning framework, i.e., HFFL, that incorporates the fairness notion. Agents who contribute more to federated learning are rewarded more in this framework. The agents at the different contribution level thus receive the different number of model updates.

4.1 HFFL Algorithm

Once the participating agents come to a consensus about their contribution levels, they can collaborate using our HFFL algorithm. In HFFL, all agents hold their data locally without sharing data directly but they share the model parameters with the central server as FedAvg [2] does.

In our HFFL, there are L contribution levels with each level having N_l ($l = 1, \cdots, L$) agents $A_{l1}, \cdots, A_{lN_l}$. We assume agents at a given level have the same amount of data (or have the same metric value that takes all data characteristics into account). Agents at level l has m_l amount of data. Agents at the same level are rewarded with the same machine learning model f_l. We provide an example of the HFFL structure in Figure 3. There are 3 agents ($N_1 = 3$) at level 1 with each agent holding m_1 amount of data locally, 4 agents ($N_2 = 4$) at level 2 with each agent holding m_2 amount of data, and 3 agents ($N_3 = 3$) at level 3 with each holding m_3 amount of data. Agents at a lower (higher) level normally contribute less (more) amount of data, thus $m_1 < m_2 < m_3$.

HFFL aims to federally learn different models at different contribution levels. To learn the model f_l at the level l, higher-level agents contribute the same amount of data as level-l agents have, while level-l and lower-level agents should contribute all the data they have. The data of high-level agents contributing to the lower level is randomly sampled (since only a subset is used). In our HFFL, we assume agents and the central server are benign so there is no cheating. We defer the adversarial settings to future work.

For example, in Figure 3, model f_1 is federally learned based on $((N_1 + N_2 + N_3) \cdot m_1)$ amount of data, which includes all data contributed by agents at level 1 and m_1 amount of data sampled from every agent at higher levels. Similarly, model f_2 is federally learned based on $(N_1 \cdot m_1 + (N_2 + N_3) \cdot m_2)$ amount of data, which includes all data from agents at level 1 and level 2 plus m_2 amount of data sampled from each agent at level 3. The model f_3 is federally learned based on $(N_1 \cdot m_1 + N_2 \cdot m_2 + N_3 \cdot m_3)$ amount of data. Our theoretical analysis in Section 4.2 shows that a model learned with more data has higher confidence to generalize better. Therefore, the agents at a higher contribution level have a better machine learning model. We assume that a better model implies a higher reward. The implementation details are in Algorithm 1.

Remark 1. Each agent holds their data locally without sharing directly. Without federated learning, each agent has access to very limited (its own) data. But when an agent participates in federated learning, it has (indirectly) access to other agents' data. Besides, in terms of knowing unknowns, it is strictly better off than when not participating.

In addition, an agent at a higher contribution level needs to contribute more amount of data but ends up obtaining a better model. Thus, HFFL encourages agents to collect and contribute more data in order to get promoted to a higher level.

Remark 2. Our HFFL framework is flexible. It allows an agent at a lower level (e.g., level $l = 1$) to get promoted to a higher level (e.g., level $l = 2$) as long as all agents at higher levels (i.e., $l \geq 2$) have no objections so that this agent could obtain a better model (i.e., f_2). The better model is trained on more data. But this agent will probably

Algorithm 1 Hierarchically fair federated learning (HFFL)

Input: Level number L, each level l having N_l agents from $A_{l1}, \cdots, A_{lN_l}$, each agent at level l contributes m_l amount of its local data, machine learning models with the same architecture $\mathcal{F}$, a central server $\mathcal{C}$, communication rounds T.

Output: Models $\{f_l\}_{l=1}^L$ for L levels.

1:   $\mathcal{C}$ initializes a model $f_0 \in \mathcal{F}$.
2:   **for** level $l = 1, 2, ..., L$ **do**
3:      $\mathcal{C}$ initializes a model for level l: $f_l \leftarrow f_{l-1}$.
4:      Denote all participating agents at level-l federated learning as an agent set $A = \{A_{ij}\}, i = l, \cdots, L, j = 1, \cdots, N_i$.
5:      **for** communication round $t = 1, 2, \ldots, T$ **do**
6:        $\mathcal{C}$ sends current parameters w_t^l of f_l to each agent in set A.
7:        **for** each agent A_{ij} in A **in parallel do**
8:          $w_t^{ij} \leftarrow \text{AgentUpdate}(i, j, w_t^l)$
           $\{* \ w_t^{ij} \leftarrow \eta \nabla \ell(w_t^l; S_{ij}^l)$, where S_{ij}^l is agent A_{ij} local data for level l, $|S_{ij}^l| = m_l$, η is the learning rate.$\}$
9:        **end for**
10:     $w_{t+1}^l \leftarrow \frac{1}{N_l + \cdots + N_L} \sum_{i=l}^{L} \sum_{j=1}^{N_i} w_t^{ij}$
       $\{*\mathcal{C}$ receives local model parameters w_t^{ij} of each agent A_{ij} in A and then updates f_l with w_{t+1}^l. $\}$
11:     **end for**
12:     Output federated learned model f_l for level l.
13: **end for**

---

need to somehow compensate the higher-level agents (e.g. pay money or promise more data in future) to get approval for such a promotion.

Remark 3. The HFFL framework can maintain the same training time as FedAvg [2] since HFFL finetunes low-level models when training a higher-level model. Note that HFFL has the flexibility to incorporate other different federated learning strategies, e.g., privacy preserving federated learning [25, 26] and robust federated learning [27].

HHFL+   In HFFL, agents at different levels have models with the same architecture. However, a more complicated model tends to overfit the small number of data. Agents in lower levels having less data probably prefer simpler machine learning models, while agents having more data probably prefer more complex models. To facilitate this, we design an improved version of HFFL, namely, HFFL+. It runs HFFL multiple times, each with different $\mathcal{F}$, i.e., models with different architectures. We then select the best-performing models for each level based on the models' test accuracy.

4.2   Theoretical Analysis

In this section, we theoretically justify (a) an agent that participates in federated learning has gain, (b) a higher level model can potentially have a less generalization error, which is aligned with our proportional fairness notion.

Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ ($|S| = m$) denote the training data sampled from an unknown distribution $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$. Suppose we have a finite model set $\mathcal{F}$, and the bounded loss function ℓ. For a specific level in HFFL, the federated learning algorithm is to utilize all available training data S to learn a model $f : \mathcal{X} \to \mathcal{Y}$ ($f \in \mathcal{F}$). The aim is to minimize generalization error $L_\mathcal{D}(f) := \mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}} \ell(f; \mathbf{x}_i, y_i)$ through minimizing the empirical error $L_S(f) := \frac{1}{m} \sum_{i=1}^m \ell(f; \mathbf{x}_i, y_i)$. We have the following lemma.

Lemma 1. *Given a training set S with m data, an error rate ϵ and a bounded loss ℓ within the range $[a, b]$, set $\delta = 2|\mathcal{F}| \exp(-2m\epsilon^2/(b-a)^2)$. Then,*

$$|L_\mathcal{D}(f) - L_S(f)| < \epsilon, \exists f \in \mathcal{F}$$

holds with probability $\geq 1 - \delta$.

Proof. To bound the difference between the empirical error $L_S(f)$ and the generalization error $L_\mathcal{D}(f)$, we use Hoeffding's inequality.
(**Hoeffding's inequality**) Let $\theta_1, \cdots, \theta_m$ be a sequence of $i.i.d.$ random variables and assume that for all i, $\mathbb{E}[\theta_i] = \mu$ and $\mathbb{P}[a \leq \theta_i \leq b] = 1$. Then, for any $\epsilon > 0$,

$$\mathcal{P}\left(\left|\frac{1}{m}\sum_{i=1}^m \theta_i - \mu\right| > \epsilon\right) \leq 2\exp(-2m\epsilon^2/(b-a)^2)$$

The proof of Hoeffding's inequality can be found in [28].

Since $\{(x_i, y_i)\}_{i=1}^m$ are $i.i.d$ sampled, the $\{\ell(f; x_i, y_i)\}_{i=1}^m$ are $i.i.d.$ Let us set $\theta_i = \ell(f; x_i, y_i)$. By applying Hoeffding's inequality, we obtain

$$\mathcal{P}\left(\{S :, |L_S(f) - L_\mathcal{D}(f)| > \epsilon\}\right) \leq 2\exp(-2m\epsilon^2/(b-a)^2).$$

We further apply union bound to yield

$$\mathcal{P}\left(\{S : \exists f \in \mathcal{F}, |L_S(f) - L_\mathcal{D}(f)| > \epsilon\}\right)$$
$$\leq \sum_{f \in \mathcal{F}} 2\exp(-2m\epsilon^2/(b-a)^2)$$
$$= 2|\mathcal{F}|\exp(-2m\epsilon^2/(b-a)^2).$$

Therefore, we prove Lemma 1 with $\delta = 2|\mathcal{F}| \exp(-2m\epsilon^2/(b-a)^2)$. $\qquad\square$

Note that in Lemma 1, we assume $\mathcal{F}$ is finite. For an infinite $\mathcal{F}$, we can obtain a similar error bound using Rademacher complexity [28].

In Lemma 1, for a fixed error rate ϵ, the probability $1 - \delta$ increases with the amount of data m. This indicates that the learning algorithm with more data could return a model that can achieve the desirable error rate of ϵ with a higher probability. Since an agent that participates in federated learning can obtain a model that learns data from other agents, the agent could obtain the model with less generalization error than the model that is learned only on its local data. Thus, an agent that participates in federated learning has the gain.

On the other hand, in HFFL the higher-level model that can learn from more samples can have a less generalization error with a higher confidence according to Lemma 1. Thus, our proposed algorithms (HFFL and HFFL+) where a higher level agent contributing more data could receive more reward (i.e., a model with less generalization error) aligns with our proportional fairness notion.

5   Experiments

We conduct experiments with four datasets to validate HFFL and HFFL+ aligning with the fairness notion: More contribution levels has more rewards.

Dataset	$m_1\ (l=1)$	$m_2\ (l=2)$	$m_3\ (l=3)$
ADULT	200	500	2,000
MNIST	200	500	2,000
F-MNIST	400	1000	4000
IMDB	256	512	3584

Table 1: The amount of training data across agents at different levels. An agent at level l randomly samples m_l amount of data without replacement.

Dataset	model $\mathcal{F}_1$ (red)	model $\mathcal{F}_2$ (blue)	LR
ADULT	2-layer MLP	Logistic regression	0.01
MNIST	3-layer MLP	4-layer CNN	0.01
F-MNIST	4-layer MLP	4-layer CNN	0.003
IMDB	Bi-LSTM	LSTM	0.001

Table 2: The different models used in HFFL for different datasets. HFFL uses the same model across agents at different levels for the same dataset. LR is the learning rate (LR) of ADAM optimizer.

ADULT is a census dataset from the UCI Machine Learning Repository [29]. It is for the binary classification task predicting whether the personal income exceeds $\$50,000$ based on 14 attributes such as age, occupation, native country and so on. There are $31,561$ records of training data and $16,281$ records of test data.

MNIST [30] is handwritten digits dataset that has a training set of $60,000$ examples, and a test set of $10,000$ examples. It is for the classification task of recognizing 10 digits from 0 to 9.

Fashion MNIST (F-MNIST) dataset [31] is an MNIST-like dataset with clothing images of 10 classes instead of handwritten digits. It consists of a training set of 60,000 examples and a test set of 10,000 examples.

IMDB dataset [32] has 50K movie reviews for natural language processing for text analytics. It is a dataset for binary sentiment classification split into a training set of $25,000$ movie reviews and a test set of $25,000$.
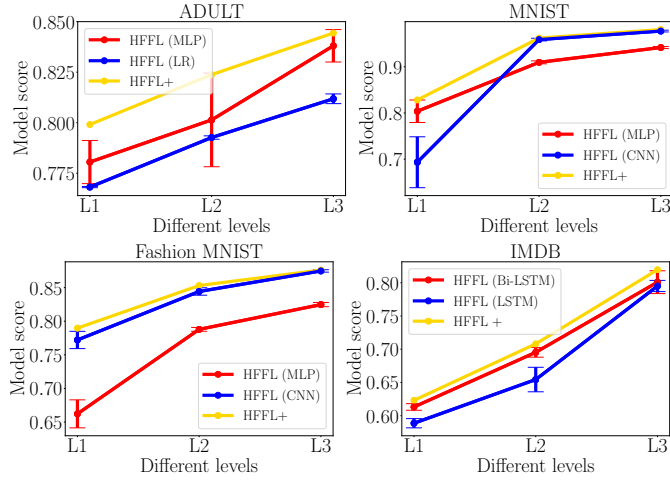
Figure 4: The model score of different levels in HFFL and HFFL+ on four datasets: ADULT, MNIST, Fashion MNIST and IMDB. Model score is represented by the test accuracy on the test data. For HFFL, we run 5 trials with different random seeds and report median model score and its variance plotted as error bar. For HFFL+, we return the best-scoring model for each level.

In all the experiments, we have three hierarchical levels ($L = 3$), with level 1 having 20 agents, level 2 having 10 agents and level 3 having 4 agents, i.e., $N_1 = 20$, $N_2 = 10$ and $N_3 = 4$. Each agent has its local data randomly drawn from training data without replacement (the training data allocation refers to Table 1).

In all experiments of HFFL, we set the number of communication rounds $T = 10$. In Figure 4, the model score is measured by test accuracy on test data. For HFFL we choose different model architectures (refer to Table 2). For HFFL +, only the best-scoring model is returned for each level. We compare models' qualities (measured by model score) across three levels obtained by HFFL and HFFL+.

5.1 Experimental Results and Analysis

In all panels of Figure 4, for HFFL (blue and red lines), a model with a higher score is returned for a higher contribution level. This is expected since high-level agents can exploit all data from lower-level agents while the low-level agents can gain from a limited amount of data (equal to what they have) from higher-level agents. The experimental results validate the fairness notion: More contribution leads to more reward.

Our HFFL+ is the advanced version of HFFL. It runs HFFL multiple times employing models with different architectures. HFFL+ returns the best-scoring model for each level. As shown in all panels of Figure 4, the yellow line (HFFL+) is always higher than red and blue lines (HFFL). Thus, different from HFFL, HFFL+ could pick different types of model for different levels. For example, in the MNIST experiment (second left panel in Figure 4), agents at level 1 have less amount of data and there-

fore prefer a simpler model, e.g., MLP, and agents at higher levels (2 or 3) have more data and prefer a more complicated model that can benefit from having more data, e.g., CNN. HFFL+ returns the best-scutring MLP at level 1 and returns the best-scoring CNN at level 2 and level 3.

6   Conclusion and Future Work

In this paper, we propose a novel federated learning framework, HFFL, to achieve fairness among self-interested agents by rewarding agents hierachical model updates based on their contribution levels, thereby facilitating federated learning. We first identify agents' contributions based on publicly verifiable factors such as data quality, data volume, etc. Then, we develop the hierarchical federated learning framework, HFFL, that upholds the fairness notion. Experimental results indicate the efficacy of our proposed methods. Future potential research includes (a) how to handle dishonest or heterogeneous agents in HFFL, (b) how to make HFFL+ computationally efficient, e.g., by leveraging transfer learning, (c) comprehensively quantify all publicly verifiable data factors, and (d) introduce differential privacy into HFFL to thwart inference attacks on models.

References

[1] Jason Andress. *The Basics of Information Security (Second Edition)*. Syngress, 2014.

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[3] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *ICML*, volume 97, pages 4615–4625, 2019.

[4] Tian Li, Maziar Sanjabi, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[5] 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.

[6] H. Kim, J. Park, M. Bennis, and S. Kim. Blockchained on-device federated learning. *IEEE Communications Letters*, pages 1–1, 2019.

[7] Jiawen Kang, Zehui Xiong, Dusit Niyato, Han Yu, Ying-Chang Liang, and Dong In Kim. Incentive design for efficient federated learning in mobile networks: A contract theory approach. *arXiv preprint arXiv:1905.07479*, 2019.

[8] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong. Federated learning over wireless networks: Optimization model design and analysis. In *INFOCOM*, 2019.

[9] John Bratton and Jeff Gold. *Human resource management: theory and practice*. Palgrave, 2017.

[10] Frederick Herzberg et al. One more time: How do you motivate employees, 1968.

[11] Kjell Y. Tornblom and Dan R. Jonsson. Subrules of the equality and contribution principles: Their perceived fairness in distribution and retribution. *Social Psychology Quarterly*, 48(3):249–261, 1985.

[12] Matthew Rabin. Incorporating fairness into game theory and economics. *The American Economic Review*, 83(5):1281–1302, 1993.

[13] Li Li, Martin Pal, and Yang Richard Yang. Proportional fairness in multi-rate wireless lans. In *INFOCOM*, pages 1004–1012, 2008.

[14] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *ICML*, volume 97, 2019.

[15] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. Measure contribution of participants in federated learning. *arXiv preprint arXiv:1909.08525*, 2019.

[16] Bengt Holmstrom and Paul Milgrom. Multitask principal-agent analyses: Incentive contracts, asset ownership, and job design. *JL Econ. & Org.*, 7:24, 1991.

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

[18] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *KDD*, pages 259–268, 2015.

[19] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, 2017.

[20] Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. Satisfying real-world goals with dataset constraints. In *NeurIPS*, pages 2415–2423, 2016.

[21] Chongjie Zhang and Julie A Shah. Fairness in multi-agent sequential decision-making. In *NeurIPS*, pages 2636–2644. 2014.

[22] Jiechuan Jiang and Zongqing Lu. Learning fairness in multi-agent systems. In *NeurIPS*, pages 13854–13865, 2019.

[23] Joniarto Parung and Umit S Bititci. A metric for collaborative networks. *Business Process Management Journal*, 14(5):654–674, 2008.

[24] Lincy Mathews and Seetha Hari. Learning from imbalanced data. In *Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics*, pages 403–414. IGI Global, 2019.

[25] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE Symposium on Security and Privacy*, pages 739–753, 2019.

[26] Naman Agarwal, Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed SGD. In *NeurIPS 2018*, 2018.

[27] Meng Hao, Hongwei Li, Guowen Xu, Sen Liu, and Haomiao Yang. Towards efficient and privacy-preserving federated deep learning. In *ICC 2019*, pages 1–6, 2019.

[28] Shalev-Shwartz Shai and Ben-David Shai. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014.

[29] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[30] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]*, 2, 2010.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[32] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL: Human Language Technologies*, 2011.