FairFed: Improving Fairness and Efficiency of Contribution Evaluation in Federated Learning via Cooperative Shapley Value

Yiqi Liu*, Shan Chang*, Ye Liu*, Bo Li†, Cong Wang‡
*Donghua University, China
†Hong Kong University of Science and Technology, Hong Kong
‡City University of Hong Kong, Hong Kong
{liuyiqi, liuye}@mail.dhu.edu.cn, changshan@dhu.edu.cn, bli@cs.ust.hk, congwang@cityu.edu.hk

IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621438

***Abstract*—The quality of federated learning (FL) is highly correlated with the number and quality of the participants involved. It is essential to design proper contribution evaluation mechanisms. Shapley Value (SV)-based techniques have been widely used to provide fair contribution evaluation. Existing approaches, however, do not support dynamic participants (e.g., joining and departure) and incur significant computation costs, making them difficult to apply in practice. Worse, participants may be incorrectly valued as negative contribution under the Non-IID data scenarios, further jeopardizing fairness. In this work, we propose FairFed to address the above challenges. First, given that each iteration is of equal importance, FairFed treats FL as Multiple Single-stage Cooperative Games, and evaluates participants by each iteration for effectively coping with dynamic participants and ensuring fairness across iterations. Second, we introduce Cooperative Shapley Value (CSV) to rectify negative values of participants to improving the fairness while preserving true negative values. Third, we prove if participants are Strategically Equivalent, the number of participant combinations can be sharply reduced from exponential to polynomial, thus significantly reducing the computational complexity of CSV. Experimental results show that FairFed achieves up to $25.3\times$ speedup and reduces deviations by three orders of magnitude to two state-of-the-art approximation approaches.**

***Index Terms*—federated learning, contribution evaluation, Shapley value, fairness**

I. Introduction

Federated learning (FL) [1] [2], a distributed machine learning (ML) framework, solves the ML model training problem in situations where training data is distributed across multiple parties who are unwilling to share private data with others. In this framework, individual data owners act as training workers to train local models based on their own data and collaboratively compute a global model, in an iterative way. Due to the heterogeneity (Non-IID: Non-Independent and Identical Distribution) of training data and hardware (computational, storage, bandwidth, etc.) resources held by the participating training workers, it is essential to evaluate the contributions of each training worker to the global model [6]. Designing an incentive mechanism based on contribution evaluation (CE) is a highly effective way to motivate participants to actively engage in federated learning tasks. This mechanism provides rewards that are appropriate to the contribution made, compensating for the computational, storage and transmission costs, as well as any potential privacy risks. It also provides a way to monetize data.

A successful contribution evaluation mechanism for federated learning should meet the following criteria: 1) it should be computationally efficient, especially in large-scale federated learning scenarios, such as cross-device federated learning, which can involve up to 10^{10} edge devices [2]; 2) it should also be fair, taking into account participants' training performance and the opportunities they have been given; and 3) since a different set of participants is chosen for each global model updating iteration, the evaluation mechanism should be able to accommodate participants who join and leave the training process dynamically.

Various data quality-based CE mechanisms have been proposed in FL [3] [4] [5], among which Shapley Value (SV)-based evaluation [7] has been the main focus of recent research. The SV, a classical concept from Cooperative Game Theory, tells how to quantify the contribution of individual participants within a group, which is the average marginal contribution of a participant considering all possible combinations of participants. The contribution of a participant in FL is commonly calculated according to the loss changes of the global model (on the validation dataset) when its data are involved [9] [10]. Some CE mechanisms treat FL as one *multi-stage cooperative game* and evaluate the contribution of participants to the well-trained final global model. To calculate the SV of participants, it is necessary to train global models on all possible data combinations of different participants separately. Then, the marginal contributions of each participant are computed based on the performance of those models on the validation dataset. However, this method leads to extremely high retraining overheads, which makes it completely impractical. To improve efficiency, several works utilize gradients in different iterations to conduct one-time updating on the initial global model so as to approximate the models on different combinations of the data sets [9] [10], which avoids model retraining by recording intermediate gradients. However, the computational complexity still increases exponentially with the number of participants. Furthermore, all CE mechanisms formulate FL as a multi-stage cooperative game that uses a fixed set of participants to complete the entire federated

Shan Chang is corresponding author.

training process and does not support participant dynamics.

Recently, some CE mechanisms have been developed that calculate SV for each iteration relevant to a different group of participants [11]–[13]. These single-round SV are then aggregated (e.g., by accumulation) to form global SV, which is considered as the overall contribution of the relevant participants. To further improve efficiency, sampling techniques such as Monte Carlo [27] and group testing [14] are adopted to approximate SV. However, while the computational cost can be limited by adjusting the sampling rate, a low rate can cause significant discrepancies from the exact SV. Besides, this approach still views FL as a multi-stage game, which is unfair as it is much more difficult for participants in the later iterations to improve the model. Thus, the late participants would get a smaller SV than those in the earlier iterations.

Additionally, Superadditivity, the basis of SV, suggests that two distinct groups of individuals training a model together will achieve a better result than if each group trained alone. Nevertheless, during federated training, the performance (e.g., loss function or accuracy) of a model which is trained by two participants is not necessarily superior to that of models trained individually. In other words, model performance-based SV does not meet Superadditivity, resulting in negative SV. It may not be a bad idea to assign negative SV to bad data. However, in the case of Non-IID data, which is common in FL, if participants are grouped according to their data distributions, the data distribution within the group can be drastically different from the global data distribution. This can lead to a significant increase in model loss and, consequently, to a negative and unreasonable SV, which can disrupt fairness [20]. As a result, to the best of our knowledge, there are no successful SV-based CE mechanisms.

In this work, we consider FL as *Multiple Single-stage Cooperative Games* to fully support dynamic joining and leaving of participants, and propose FairFed, a fair and efficient SV-based CE mechanism. The basic idea is that the contribution of participants should be assessed by iteration, and not be affected by the order of participation. In other words, a participant updating the global model with the same group of participants in different iterations has exactly the same contribution. In FairFed design, there are two challenges. First, to ensure fairness, Non-IID data contributing to the global model should not be given negative SV. Without accessing raw data, it is however difficult to distinguish Non-IID data from low-quality data. Second, to degrade the computational complexity of SV, it is necessary to reduce the number of combinations. However, it is extremely difficult to determine which combinations can be removed without causing significant approximation bias. Random sampling may fail to capture important combinations.

To address the above challenges, firstly, we start with Cooperative Shapley Value (CSV). This method does not assess the influence of a single participant on the global model, but instead calculates the average benefit that is derived from the collaboration of the participant and all other participant groups. Consequently, the irrational negative SVs caused by Non-IID data can be eliminated. CSV complies with three of the four axioms of traditional SV, namely *null player*, *symmetry*, and *additivity*. We normalize CSV to make the fourth axiom, *efficiency*, unnecessary. Secondly, we prove that participants with the same data distribution are *Strategically Equivalent*. This means that combinations with the same number of equivalent participants are also equivalent, and thus only need to be considered once in the CSV. To represent the data distribution of participants, we use gradients of local training and divide them into groups using bisection K-means. We evaluate the SVs of groups instead of individual participants, which reduces the computational cost from exponential to polynomial. For example, a group has *three* participants, i.e., $\{A, B, C\}$. If they are equivalent, the group can be seen as $\{A, A, A\}$. When calculating traditional SV, all *seven* combinations of three participants should be included. In contrast, only *three* combinations, i.e., $\{A, A, A\}$, $\{A, A\}$, $\{A\}$, are considered in CSV.

FairFed is tested on three image datasets (MNIST, CIFAR-10 and FMNIST). Our comparison of CSV and SV in terms of fairness reveals that both CSV and SV assign negative values to participants with bad data. However, CSV never assigns negative values to participants with Non-IID but high-quality data, whereas SV does. We assess the performance of FairFed by comparing it to exact SV as a baseline, as well as Carlo-based (TMC-Shapley [27]) and Random sampling (GT-Shapley [14]) schemes. The results show that FairFed can be up to $48.9\times$ faster than the traditional CSV calculation, and $8.3\times$ to $25.3\times$ faster than GT-shapley and TMC-shapley, respectively. Moreover, we employ three metrics, *Cosine Distance*, *Euclidean Distance* and *Maximum Difference*, to evaluate the approximation errors of FairFed compared to exact CSVs. FairFed's approximation error is $100\times$ to $1000\times$ lower than the other two methods. We also assess the influence of global data distribution and the number of participant groups on FairFed's performance. All three errors range from 10^{-2} to 10^{-4} for all distributions. As the Non-IID increases, the error decreases. The error diminishes slightly with more groups.

II. Related Work

A. SV-based Data Valuation

There has been some work on leveraging the SV to evaluate the quality of the training data in ML. Jia R *et al.* [14] discussed the efficient estimation of SV from a theoretical perspective. They propose a series of approximation methods that can be applied in cases where different assumptions are introduced on the characteristic functions. There is also some work dedicated to reducing the cost of calculating SV. Jia R *et al.* [14] proposed to use group testing to accelerate the calculation of SV. Castro J *et al.* [15] proposed to reduce the computational complexity of the SV to polynomials using Monte Carlo methods. But all these works require iterative re-training of the model. Applying them directly to FL introduces huge computational costs and communication costs.

B. SV-based Contribution Evaluation in FL

There are a few works in FL that use SV to evaluate participant contributions. These works can be broadly categorized into two types, the first of which treats the entire training process as a whole and entails forcing the participant to participate in the training from start to finish. Song T *et al.* [9] proposed to approximate the retrained model by the model reconstructed by the gradient. They propose One-Round Reconstruction based Algorithm and Multi-Rounds Reconstruction based Algorithm to avoid the huge cost associated with retraining the model. However, it still needs to evaluate the model exponentially. Liu Z *et al.* [8] proposed the Guided Truncation Gradient Shapley to further reduce the computational cost of the SV. They use the value of previously evaluated models as a guide to skip unnecessary evaluation of the value of some sub-models later. Wang J *et al.* [10] proposed DIGFL, which makes contribution evaluation not only applicable to horizontal FL, but also extendable to vertical FL. Also, DIGFL provides a reweight mechanism to accelerate model convergence during gradient aggregation. The second type of method would calculate the SV once per round. Wang T *et al.* [11] proposed the concept of federated SV. This concept fits more closely with the FL framework, as it does not mandate that the participant be involved in training all the time. And they experimentally verified that the federated SV can be used for Noisy Label Detection, Backdoor Attack Detection and Data Summarization. Fan Z *et al.* [12] continued the idea of federated SV by proposing a new approach called ”completed federated Shapley value”, a design that relies on a matrix consisting of all possible contributions from different subsets of participants, allowing two participants with the same local data to be evaluated the same way, further improving fairness. Dong L *et al.* [13] incorporated several acceleration methods to further improve the computational efficiency of the SV. But none of the above methods can assign a fair profit to participants who attend training late.

C. Other Applications of SV in FL

There is also some work dedicated to the use of SV for other tasks in FL. Tang Z *et al.* [16] uses SV for reweighting during gradient aggregation, which can improve accuracy during training. Nagalapatti L *et al.* [17] use SV to solve part of the FRCS problem, i.e., selecting participants with relevant data, detecting participants with data related to a specific target label, and correcting individual participant corrupted data samples. Yuan X *et al.* [18] devised a joint optimization scheme for participant selection and resource scheduling based on SV that maximizes learning energy efficiency.

III. System Model and Preliminaries

A. System Model

We consider a typical cross-device FL scenario where the participants involved in each round of training are often a small fraction of the total number of participants. The participants selected for different rounds are likely to be different.

There are n participants $N = \{c_1, c_2, \ldots, c_n\}$ and a server in a FL system, and each participant has its local private data $D_1, D_2, \ldots, D_n$. In total, there are T rounds of global iterations. For each iteration $t \in \{1, 2, \ldots, T\}$, the server randomly selects a group of participants C from all participants to participate in this training round. Each selected participant c_i uses its local private data to train based on the distributed global model $W^{(t)}$ and gets updated local model $W_i^{(t)}$ after local iterations. Participant c_i uploads its gradient $\Delta_i^{(t)}$ to the server:

$$\Delta_i^{(t)} = \ W_i^{(t)} - W^{(t)} \tag{1}$$

The server aggregates the local gradients uploaded by all participants by weighted averaging to get the global gradient $\Delta^{(t)}$:

$$\Delta^{(t)} = \frac{1}{n}\sum_{i=1}^{n} \Delta_i^{(t)} \tag{2}$$

Finally, the server uses it to update the global model:

$$W^{(t+1)} = W^{(t)} + \Delta^{(t)} \tag{3}$$

We assume there may exist participants holding bad (e.g., with incorrect labels) data maliciously or unintentionally, in FL. However, the majority of participants hold high-quality data. Data held by participants are Non-IID. There are several categories of data distributions however unknown to both the server and participants. Furthermore, there exists a small validation dataset that can be used for examining the performance of global models in each iteration.

B. Cooperative Game (CG)

A CG consists of a binary group $< N, V >$, where N is the set containing n players involved in the game and $V : 2^N \rightarrow \mathbb{R}$ is a characteristic function $V\left(S\right)$ mapping a subset of players to real numbers. For any possible coalition (group) S belonging to N, $V\left(S\right)$ denotes the value created by all players in that group through cooperation.

C. Shapley Value

The SV measures the expectation of the marginal contribution that may result from a participant's participation in any group. The SV of player i is shown in (4), where $|S|$ denotes the number of players in the group S. We will use the same method to represent the number of elements of the set in the rest of the article.

$$\varphi_i\left(V\right) = \sum_{S \subseteq N} \frac{|S|!\left(n - |S| - 1\right)!}{n!}\left(V\left(S\right) - V\left(S \backslash i\right)\right) \tag{4}$$

It is required that the characteristic function V of SV satisfies *Superadditivity*, i.e., $V\left(i\right) + V\left(j\right) \leq V\left(i \cup j\right)$ [7].

Lloyd Shapley proved that the SV is the only method of contribution evaluation that satisfies the following four axioms.

1) Symmetry: If the marginal contribution of two players i, j to any group S is the same $V\left(S\ \cup \{i\}\right) = V\left(S \cup \{j\}\right)$, then their Shapely values are the same $\varphi_i\left(V\right) = \varphi_j\left(V\right)$.

2) Null player: If the player i's marginal contribution to any group S is zero $V(S\ \cup \{i\}) = V(S)$, then his Shapely value is zero.

3) Efficiency: The sum of the SV of all players equals the value of the grand group. $\sum_{i \in N} \varphi_i\left(V\right) = V\left(N\right)$

4) Additivity: The Shapely values can be summed for two different games V and W. $\varphi_i\left(V + W\right) = \varphi_i\left(V\right) + \varphi_i\left(W\right)$

D. Multiple Single-stage Cooperative Game Framework

We consider the procedure of FL as Multiple Single-stage Cooperative Games. This allows participants to join in or leave out federated training at any round. In iteration t, the server constructs the group models of all possible participant combinations from the local gradients uploaded by all participants in C. For a participant group $S \subseteq C$, the group model $W_S^{(t+1)}$ can be calculated by

$$W_S^{(t+1)} = W^{(t)} + \frac{1}{|S|}\sum_{i \in S} \Delta_i^{(t)}, S \subseteq C. \tag{5}$$

Next, the server calculates the corresponding V of this group S in iteration t as

$$V(S) = L(W^{(t)}) - L(W_S^{(t+1)}), \tag{6}$$

where L is the model loss on the validation set. $V(S)$ represents the performance change of the global model after being updated by using the group of local models relevant to C.

IV. Design of FairFed

A. Overview

We propose a fair and efficient CE mechanism, FairFed, in FL. The server runs FairFed in each iteration which includes three steps (as shown in Fig. 1) First, *Registration*, after the server selects participants in an iteration, it examines whether all selected participants have registered to the system or not. If not, the server multicasts the initial global model ($W_i^{(0)}$) to those registered participants, and they train the model using purely local data individually (e.g., participant 1 in the figure), and report the corresponding gradients to the server for registration. Second, the server records the gradient (as a vector) of each local model, then and applies Bisecting K-means clustering to divide the participants into two groups iteratively according to their registration information (gradients). Top-k significant dimensions are extracted and compared within group members to decide whether the partition should stop or not. Finally, the server calculates the CSV of each participant according to group information, and normalizes CSVs as the final contribution of participants in this iteration, in order to make the overall contribution of each iteration equal, ensuring cross-iteration fairness.

B. Cooperative Shapley Value

In the case of Non-IID participant data, there will even be indispensable participants who will be assigned negative SV. This is because, in the case of Non-IID participant data, each participant's data is only a small part of the global data distribution e.g., data distribution of the validation set. In the case of very different data distribution between the participant and the validation set, the loss on the validation set is very unstable. The loss value on the validation set of a local model trained by a single participant, or participants with data obeying the same distribution, is likely to be larger than the loss value of the original global model. This decline in performance dominates the calculation of the SV.

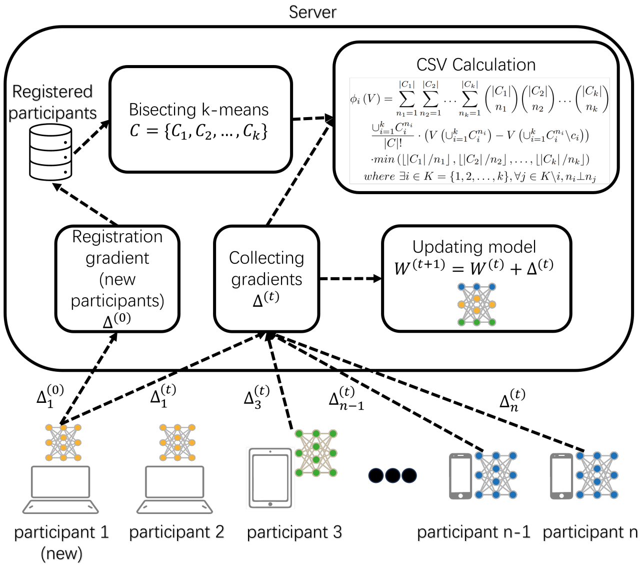
Server
Registered participants
Bisecting k-means
$C = \{C_1, C_2, \ldots, C_k\}$
CSV Calculation
$\phi_i\left(V\right) = \sum_{n_1=1}^{|C_1|}\sum_{n_2=1}^{|C_2|}\cdots\sum_{n_k=1}^{|C_k|}\binom{|C_1|}{n_1}\binom{|C_2|}{n_2}\cdots\binom{|C_k|}{n_k}$
$\frac{\cup_{i=1}^{k} C_i^{n_i}}{|C|!}\cdot\left(V\left(\cup_{i=1}^{k} C_i^{n_i}\right) - V\left(\cup_{i=1}^{k} C_i^{n_i}\backslash c_i\right)\right)$
$\cdot min\left(\lfloor|C_1|/n_1\rfloor, \lfloor|C_2|/n_2\rfloor, \ldots, \lfloor|C_k|/n_k\rfloor\right)$
$where\ \exists i \in K = \{1,2,\ldots,k\}, \forall j \in K\backslash i, n_i \perp n_j$
Registration gradient (new participants) $\Delta^{(0)}$
Collecting gradients $\Delta^{(t)}$
Updating model $W^{(t+1)} = W^{(t)} + \Delta^{(t)}$
$\Delta_1^{(0)}$
$\Delta_1^{(t)}$
$\Delta_3^{(t)}$
$\Delta_{n-1}^{(t)}$
$\Delta_n^{(t)}$
participant 1 (new)
participant 2
participant 3
participant n-1
participant n

Fig. 1. Procedure of FairFed in round t.

Consider the following situation, as shown in Table I. There are two participants i and j. They each have half of the global data distribution. They are jointly trained based on the near convergence model. The loss value of the existing model on the validation set is 100. The loss values of the local model trained by participants i and j alone on the validation set are 110 and 120, respectively. The loss of the global model trained by participants i and j together on the validation set is 95. In this case, participant j has a Shapely value of -2.5, which means that participant j makes a negative contribution. However, without the involvement of participant j, the loss of the model on the validation set does not shrink further. So, it is not fair to directly contribute to participant j as negative.

TABLE I
Calculation of SV and CSV

	$\emptyset$	i	j	$\{i, j\}$
loss	100	110	120	95
value	0	-10	-20	5
SV		7.5	-2.5	
CSV		25	15	

For a participant whose data distribution is different from the validation set, the original SV would assign them negative values even if he has high-quality data. It is very common for participants to hold Non-IID data in FL. We believe that it is unreasonable to consider the above-mentioned participants who hold high-quality data as negative contributors.

To solve the above problem, we no longer focus on local models trained by a single participant or participants who have data that follows the same distribution, and propose the CSV. We ignore the incremental loss values of these local models compared to the original global model on the validation set when calculating the SV. We only calculate the decrease in loss value for each participant when joining a group consisting of participants with a different data distribution X than that participant.

$$\phi_i(V) = \sum_{S\subseteq N} \frac{|S|!\,(n-|S|-1)!}{n!}\left(V(S) - V(S\backslash i)\right) \\ where\ \exists j \in S, \quad D_i \nsim X, D_j \sim X \tag{7}$$

We only distribute the benefits of model enhancements to participants with positive Sharpley values. Assuming that the set of participants with a positive SV is Φ^+, we compute the proportion of each participant's SV that accounts for the sum of the SV $\hat{\phi}(V)$.

$$\hat{\phi}_i(V) = \frac{\phi_i(V)}{\sum_{j\in\Phi^+}\phi_j(V)} \tag{8}$$

CSV reduces the bias against a participant's data quality caused by Non-IID participant data by only calculating the marginal contribution of the participant when working with others.

CSV obeys the following three axioms.

- Null player: For a participant i, if $V(S) = V(S\cup i)$ holds for any group S, then $\phi_i(V) = 0$.
- Symmetry: If $V(S\cup i) = V(S\cup j)$ is satisfied for any group S that does not include participant i and participant j, then $\phi_i(V) = \phi_j(V)$.
- Additivity: The SV of any two unrelated games can be summed. $\phi(V+U) = \phi(V) + \phi(U)$

Efficiency is the only axiom that the CSV does not satisfy. This caused the sum of the SV of all participants does not equal the value of the grand group. This means that we can't allocate benefits based on the value of the grand group and the exact value of each participant's SV. Because the Multiple Single-stage Cooperative Game framework we proposed would normalize the SV for each participant. So the problem can be solved by distributing the value of the grand group in proportion to the normalized SV.

While CSV has the ability to discriminate participants with Non-IID data and assign positive values, it can also discriminate participants with bad data and assign negative values.

Theorem 1. *For a participant i who holds bad data $D_i \sim X_i$, if SV assigns a negative value to this participant, then CSV will also assign a negative value to this participant.*

Proof. The difference between the calculation of SV and that of CSV is that SV also calculates the contribution of a group consisting of only one class of participants. We use $\varphi^{diff}(V)$ to represent this difference.

$$\varphi_i(V) = \phi_i(V) + \varphi_i^{diff}(V)$$

There are three situations that can cause $\varphi_i(V)$ to be less than 0.

$$\begin{cases} \phi_i(V) < 0,\ \varphi_i^{diff}(V) < 0 \\ \phi_i(V) < 0,\ \varphi_i^{diff}(V) > 0 \\ \phi_i(V) > 0,\ \varphi_i^{diff}(V) < 0 \end{cases}$$

For the first two cases, CSV and SV give negative values identically. And for the third case, this participant has made a positive contribution in working with most of the participants who are in the other distributions. So this kind of participant is not a participant with bad data. □

Consider the same example as before. After modification, participant i has a CSV of 25 while participant j has a CSV of 15. In FL, it makes more sense to use the CSV. Because it does not brutally treat participants holding Non-IID data as low-quality participants. It assigns a positive value to every indispensable participant unless that participant makes most of the group perform worse when he cooperates.

C. Accelerated Approach to Cooperative Shapley Value

In FL, most of the time spent on calculating the SV is the evaluation of different local models [8]. When we consider the characteristic function as a black box, we must enumerate all possible groups and evaluate them. This enumeration-based expression of the characteristic equation makes the computational complexity of the SV at the exponential level of the number of participating participants n i.e., $T(n) = \Theta(2^n)$. In FL, the number of participants is often very large. The exponential level of complexity is unacceptable. One way to effectively reduce the complexity of the SV is to use a compact representation of the characteristic function. Shrot *et al.* [19] propose a compact characteristic function representation based on participant types, named *Strategic Type*, in which if two participants have the same strategic power, then they belong to the same type.

Definition 1. *participant $i, j\ \in C$ are Strategically Equivalent if for any group S, such that $i, j \notin S : V(S\cup i) = V(S\cup j)$*

They proved that the Strategically Equivalence of participants is an equivalence relation. Accordingly, we put the participants with Non-IID data into different classes. Each class is a Strategic Type.

Lemma 1. *If there are participants $i, j \in C$, $\Delta_i^{(t)} = \Delta_j^{(t)}$. Then for any group S, such that $i, j \notin S$. We can obtain the following equation:*

$$W_{s\cup i}^{(t+1)} = W_{s\cup j}^{(t+1)} \tag{9}$$

Proof. With (5), we can obtain, for $x \in \{i, j\}$

$$W_{S\cup x}^{(t+1)} = W^{(t)} + \frac{1}{|S+1|}\sum_{l\in\{S\cup x\}}\Delta_l^{(t)} \tag{10}$$

We expand the right-hand side of the above equation to obtain:

$$W_{S\cup i}^{(t+1)} = W_{S\cup i}^{(t+1)} = W^{(t)} + \frac{1}{|S+1|}\Delta_i^{(t)} + \frac{1}{|S+1|}\sum_{l\in S}\Delta_l^{(t)} \tag{11}$$

□

Theorem 2. *If there are participants $i, j \ \in C$, $\Delta_i^{(t)} = \Delta_j^{(t)}$, then they are Strategically Equivalent.*

Proof. From (6), we can see that:

$$\begin{aligned} V\left(S \cup i\right) = L(W^{(t)}) - L(W_{S \cup i}^{(t+1)}) \\ V\left(S \cup j\right) = L(W^{(t)}) - L(W_{S \cup j}^{(t+1)}) \end{aligned} \tag{12}$$

Combined with Lemma 1, $V\left(S \cup i\right) = V\left(S \cup j\right)$ holds. □

Theorem 3. *In the cooperative game, if the number of types of participants is fixed at m, then the complexity of calculating the SV is $O\left(|C|^m\right)$.*

Proof. Assume that the division of the grand group C based on Strategic Type is $C = \{C_1, C_2, \ldots, C_m\}$, where

$$\forall C_i, C_j \ \subset C, C_i \cap C_j = \emptyset, \ \ \cup_{i=1}^{m} C_i = C \tag{13}$$

Because the participants within each class are Strategically Equivalent, their marginal contribution to any group is the same. Therefore, for a group, we do not need to be concerned about who is in it. We just need to focus on how many participants there are in each class in this group.

$$T\left(n\right) = \prod_{i=1}^{m} \left(|C_i| + 1\right) = O\left(|C|^m\right)$$

□

In FL, the number of local models to be evaluated can be further reduced.

Proposition 1. *Given two groups $S = \{S_1, S_2, \ldots, S_m\}$ and $T = \{T_1, T_2, \ldots, T_m\}$ that are clustered by gradient and contain the same m Strategic Types of participants. If there is a constant α such that*

$$\forall S_i \subset S, |S_i| = \alpha \left|T_i\right| \tag{14}$$

Then $V\left(S\right) = V\left(T\right)$.

Proof.

$$\begin{aligned} \forall S_i \subset S, W_{S_i}^{(t+1)} = W^{(t)} + \frac{1}{|S_i|} \sum_{j \in S_i} \Delta_j^{(t)} = \\ W^{(t)} + \frac{1}{|T_i|} \sum_{j \in T_i} \Delta_j^{(t)} = W_{T_i}^{(t+1)} \end{aligned} \tag{15}$$

Thus, $V\left(S\right) = L(W_S^{(t+1)}) = L(W_T^{(t+1)}) = V\left(T\right)$ holds. □

C_m^j denotes a sub group containing j participants of Strategic Type m. Based on proposition 1, given the grand group $C = \{C_1, C_2, \ldots, C_m\}$, the CSV is defined as ($\perp$ means co-prime),

$$\begin{aligned} \phi_i\left(V\right) = \sum_{n_1=1}^{|C_1|} \sum_{n_2=1}^{|C_2|} \cdots \sum_{n_m=1}^{|C_m|} \binom{|C_1|}{n_1} \binom{|C_2|}{n_2} \cdots \binom{|C_m|}{n_m} \\ \frac{\cup_{i=1}^{m} C_i^{n_i}}{|C|!} \cdot \left(V\left(\cup_{i=1}^{m} C_i^{n_i}\right) - V\left(\cup_{i=1}^{m} C_i^{n_i} \backslash c_i\right)\right) \\ \cdot min\left(\left\lfloor |C_1| / n_1 \right\rfloor, \left\lfloor |C_2| / n_2 \right\rfloor, \ldots, \left\lfloor |C_m| / n_m \right\rfloor\right) \\ where \ \exists i \in M = \{1, 2, \ldots, m\}, \forall j \in M \backslash i, n_i \perp n_j \end{aligned} \tag{16}$$

D. Gradient-based Clustering

We follow the same assumptions as Sattler *et al.* [21]. In the case of participants holding Non-IID data, participants can be divided into m classes. We emphasize that neither the server nor the participants are known about the value of m. The data of the participants within each class belong to the same distribution. They can be approximated as Strategically Equivalent. For any new participant that joins the training, it is necessary to train on the initial model and upload gradients for registration. All subsequent clustering uses the registered gradients.

We use the Bisecting K-means cluster algorithm to perform clustering based on the gradient uploaded by the participant. The difficulty of clustering is that the dimension of the gradient is extremely high. In the case of high-dimensional data, the distance of data points tends to be close to each other, so the distance relationship between data points will be weakened [22]. So we need to use the following data preprocessing methods. We use Top-*k* to reduce the dimensionality of the gradient. After experiments, we select the 5% of the gradient with the largest absolute value. It is general to most tasks. These values retain most of the information of the original gradient [23]. It can be seen from Table II, after selecting the 5% (i.e., 42 elements in one of our experimental setups) with the largest absolute value in the gradient, there are many more identical dimensions among participants whose data are from the same distribution.

The Bisecting K-means cluster algorithm we use is divided into the following two steps. First, for each cluster, the number of non-zero dimensions common to each gradient in this cluster is calculated. Second, if the common nonzero dimension (threshold) occupies more than 30% of the total nonzero dimension, The clustering process then stops and the cluster remains unchanged. Otherwise, we use the K-means algorithm to split this cluster into two classes.

TABLE II
NUMBER OF IDENTICAL DIMENSIONS

	min	max	average
Same distribution	42	32	38
Different distribution	5	2	3

V. Experimental Results

A. Settings

We set up a total of $n = 100$ participants. 10% of them are randomly selected for training in each round. A total of $T = 25$ rounds of training will be done. Each participant performs one local epoch per training round. We used the datasets MNIST, FMNIST and CIFAR10. For MNIST and FMNIST, we preprocessed the training set. We drew 5400 samples from each class to form the training set. So, the final training set contains 54,000 samples. The model we have chosen is LeNet-5 for MNIST and FMNIST and a CNN for CIFAR [24] [25] [26]. Both contain two convolutional layers and three fully connected layers. We use SGD as the optimizer with the learning rate setting to 0.01. The local batch size is

128. We set the number of participant classes m to 2, 3, and 4 respectively.

- Two Classes: participants are divided into two equal-size groups, where participants evenly select samples from classes 0-4 and 5-9, respectively.
- Three Classes: participants are divided into three equal-size groups, where participants evenly select samples from classes 0-3, 4-6 and 7-9, respectively.
- Four Classes: participants are first divided into two equal-size groups, then each group is separated into two subgroups containing 30 and 20 participants. Accordingly, the two 30-participant groups select samples from classes 0-2 and 3-5, respectively. The two 20-participant groups select samples from classes 6-7 and 8-9, respectively.

In addition to the above three non-overlapped partitions (NOP), we also use Dirichlet distribution for data partitioning [28]. For each of the three different numbers of participant types described above, we used the coefficient $\alpha = 0.1, 0.3, 0.5$ for the partition. The coefficient α indicates how ”Non-IID” the data are divided, with smaller values being closer to non-overlapped partitioning and larger values being closer to ”IID”.

B. Comparison Method

1) Precise CSV: This method refers to the direct use of gradients to reconstruct all sub-models for evaluation and is used as ground truth. Unless explicitly named, this method refers to the CSV calculated using (16), rather than the original SV calculated using (4).

2) TMC-shapley [27]: A Monte Carlo-based method, which randomly samples a permutation π from all $n!$ permutations of participants, and then calculates the marginal contribution of each participant in this permutation when he joins a sub-group of participants who preceded him. Also, it truncates unnecessary sub-model evaluations.

3) GT-shapley [14]: This method randomly samples multiple sub-groups and evaluates their value. It estimates the difference in SV between participants rather than the SV per participant. Then it calculates the SV by solving a feasibility problem. Because one constraint of the feasible problem is constituted by the efficiency of the SV. In the experiment, it approximates the original SV calculated using (4).

C. Metrics

1. Time: The average time to calculate the CSV for each training round throughout the training process.

2. Cosine Distance (CD): In the t-th global iteration, we calculate the CD between the approximate $BV^{(t)}$ and the real $BV^{(t)*}$according to the following equation:

$$CD^{(t)} = 1 - \frac{BV^{(t)} \cdot BV^{(t)*}}{\left|BV^{(t)}\right| \times \left|BV^{(t)*}\right|}. \tag{17}$$

Then calculate the average value during T training rounds.

3. Euclidean Distance (ED): calculated using

$$(\sum_{i=1}^{n}(BV_i^{(t)*} - BV_i^{(t)})^2)^{1/2} \tag{18}$$

4. Maximum Difference (MD): calculated using

$$max\left(\left|BV_1^{(t)*} - BV_1^{(t)}\right|, \ldots, \left|BV_n^{(t)*} - BV_n^{(t)}\right|\right) \tag{19}$$

D. Parameter Sensitivity on Clustering

The accuracy of clustering directly determines the efficiency and accuracy of the FairFed approximation computation. If the number of participant types obtained from clustering is more than the actual number of participant types (i.e., the same types of participants are divided into two different classes), the time overhead of the approximation computation becomes large. If the number of participant types obtained from clustering is less than the actual number of participant types (i.e., participants in different classes are classified into one class), the accuracy of the approximation calculation becomes smaller. There are two hyperparameters that affect the accuracy of the clustering, which are the threshold (percentage of same non-zero dimension) for determining whether the clustering process is stopped or not and the value of Top-*k*.

In this experiment, we explore the effect of these two hyperparameters on clustering accuracy separately. We've experimented with all the settings a hundred times. For testing the effect of the threshold, the Top-*k* value was chosen to be 0.05. and for testing the effect of Top-*k*, the threshold value was chosen to be 0.5. these two values were used for all other experiments. Tables III and IV show the effects of these two hyperparameters, respectively. The check mark in the table indicates that the clustering was correct one hundred times, and the cross indicates that the clustering failed completely without any pattern. The binary x,+/-n in the table indicates that there were x% of experimental clustering results with n more/less classes than the actual value.

1) Threshold: From Table III, it can be seen that clustering is always correct on any dataset when the clustering thresholds are 0.3, 0.5, and 0.7. When the clustering threshold is 0.9, sometimes the values obtained from clustering are a few classes more than the true values. This is because these gradients are clustered when only 90% of the non-zero dimensions are the same, an overly stringent condition. For similar reasons, with a threshold of 0.1, the values obtained from clustering are sometimes smaller than the actual values.

2) Top-k: As can be seen in Table IV, the clustering is perfectly correct when Top-*k* is taken to 0.05 or 0.10. While smaller or larger values are sometimes not able to get the correct clustering. This is because either the information used for clustering is too little or the information used for clustering is too redundant masking the valid information.

E. Fairness

In this experiment, we will verify the fairness of CSV. We will test whether CSV can distinguish and assign positive values to participants holding Non-IID but high-quality data, while assigning negative values to participants holding bad data. We will use non-overlapped partitioning as an experimental setup, where any class of participants is indispensable.

TABLE III
CORRECTNESS OF CLUSTERING FOR DIFFERENT THRESHOLDS

	MNIST					CIFAR10					FMNIST				
	0.1	**0.3**	**0.5**	**0.7**	**0.9**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
2 classes	✔	✔	✔	✔	✘	✘	✔	✔	✔	28%,+1 24%,+2	✘	✔	✔	✔	48%,+1 4%,+2
3 classes	44%,-1	✔	✔	✔	36%,+1 4%,+2	✔	✔	✔	✔	✔	40%,-1	✔	✔	✔	✔
4 classes	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

TABLE IV
CORRECTNESS OF CLUSTERING FOR DIFFERENT TOP-k VALUES

	MNIST					CIFAR10					FMNIST				
	0.01	**0.05**	**0.10**	**0.20**	**0.40**	**0.01**	**0.05**	**0.10**	**0.20**	**0.40**	**0.01**	**0.05**	**0.10**	**0.20**	**0.40**
2 classes	16%,+1	✔	✔	✔	✘	✔	✔	✔	✘	✘	✔	✔	✔	✘	✘
3 classes	8%,+1	✔	✔	✔	✘	✔	✔	✔	✔	✘	8%,+1	✔	✔	✔	✘
4 classes	✔	✔	✔	✔	✘	✔	✔	✔	✔	✘	✔	✔	✔	✔	✘

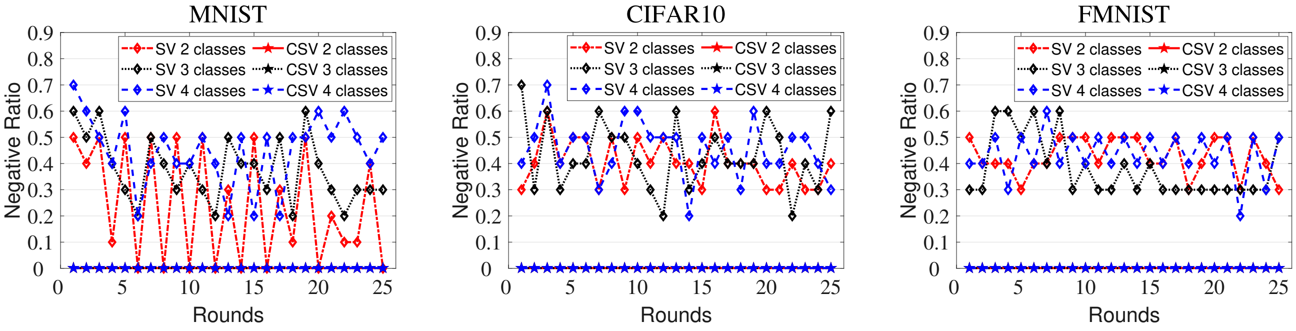

Fig. 2. Proportion of participants incorrectly assigned negative values in each round throughout the training.

This is because the accuracy of the global model cannot be improved more if any class of participants is absent.

Fig. 2 indicates that if the original SV is used, then almost half of the participants are given negative values in each round during training, regardless of the dataset and number of participant classes. FairFed, on the other hand, has always given positive value to all participants.

To test whether CSV can discriminate against participants holding bad data, we randomly replace the labels of two participants with the wrong labels in each round of training. Both SV and CSV can find participants holding bad data and assign negative values in each round of training. We have not presented them graphically.

F. Accuracy and Time Overhead

In this experiment, we compare the clustering approximation in FairFed with TMC and GT to verify whether our method has higher approximation accuracy and smaller time overhead.

We use an experimental setup that divides participants into two classes by non-overlapped partitioning. Table V shows that FairFed achieves higher approximate accuracy in less time. We conducted experiments on all three datasets. The time overhead of the FairFed approximation computation on any given dataset is 1-2 orders of magnitude smaller compared to the other two approximation methods. And compared to the exact calculation, FairFed achieves a speedup of 40× or more on all datasets. In terms of approximation accuracy, the cosine distance between the approximation and the exact value calculated by FairFed is 2-3 orders of magnitude smaller than the other two methods. Besides, both Euclidean Distance and Maximum Difference were also significantly better than the other two methods. Compared between the three methods, our method achieves the highest approximation accuracy in the shortest time on any dataset.

G. Performance on Different Data Distributions

In this experiment, we will test the effect of different numbers of participant types and the degree of Non-IID of the data distribution on the accuracy and time overhead of the FairFed approximation, respectively.

To test the effect of the number of participant types on FairFed, we set the number of participant types by non-overlapped partitioning to 2, 3, and 4, respectively. Fig. 3 shows that the computation time of FairFed increases with the number of participant types. Also, the approximation accuracy increases with the number of participant types. For a clearer and more concise picture, we logarithmically processed all the data. For both time and accuracy, smaller values represent less time overhead or higher accuracy.

To test the effect of the degree of data Non-IID on FairFed, we used the following experimental setup. For all three datasets, we classified participants into two classes. We use non-overlapped partitioning as well as a Dirichlet distribution

TABLE V
APPROXIMATION ERROR AND TIME COST OF DIFFERENT ALGORITHMS

	MNIST			CIFAR10			FMNIST		
	FairFed	**TMC**	**GT**	**FairFed**	**TMC**	**GT**	**FairFed**	**TMC**	**GT**
Time(s)	$6.47 * 10^2$	$1.64 * 10^4$	$5.42 * 10^3$	$8.29 * 10^2$	$1.80 * 10^4$	$7.10 * 10^3$	$6.14 * 10^2$	$1.33 * 10^4$	$5.65 * 10^3$
CD	$5.61 * 10^{-4}$	$3.46 * 10^{-2}$	$6.27 * 10^{-1}$	$5.89 * 10^{-4}$	$3.88 * 10^{-2}$	$8.16 * 10^{-1}$	$7.28 * 10^{-4}$	$2.94 * 10^{-2}$	$6.86 * 10^{-1}$
ED	$1.17 * 10^{-2}$	$8.89 * 10^{-2}$	1.14	$9.47 * 10^{-3}$	$9.30 * 10^{-2}$	1.77	$1.50 * 10^{-2}$	$9.27 * 10^{-2}$	1.05
MD	$7.40 * 10^{-3}$	$5.43 * 10^{-2}$	$7.02 * 10^{-1}$	$5.75 * 10^{-3}$	$5.84 * 10^{-2}$	1.21	$8.71 * 10^{-3}$	$6.14 * 10^{-2}$	$5.68 * 10^{-1}$

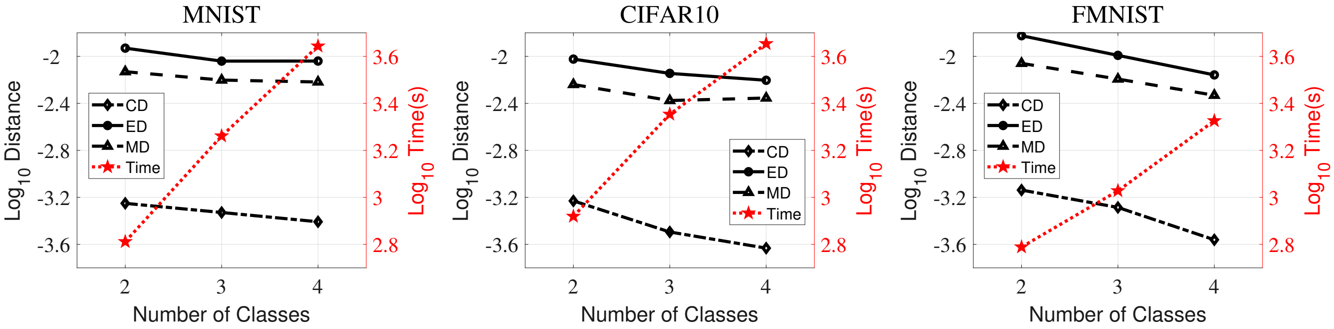

Fig. 3. Approximation error and time cost of CSV for different numbers of participant types.

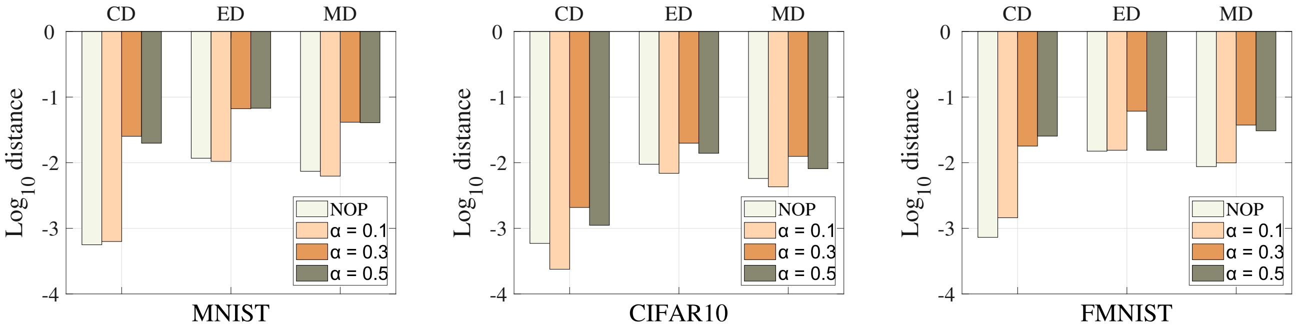

Fig. 4. Approximation to CSV for different distributions.

with parameter $\alpha = 0.1, 0.3, 0.5$ for data partitioning, respectively. Because the closer the parameter a is to zero, the higher the degree of Non-IID is and the closer it is to non-overlapped partitioning, the degree of Non-IID gradually decreases for the four division methods mentioned above. Fig. 4 shows that the approximation accuracy decreases after the decrease in the degree of Non-IID (Because the time overhead does not vary with the degree of Non-IID, it is not shown.). However, FairFed still maintains a high approximation accuracy while greatly speeding up the computation process. This data set was also logarithmically processed, so the longer bar indicates a higher approximate accuracy.

VI. CONCLUSION

In this work, we introduce a CSV-based CE mechanism - FairFed with the goal of improving fairness and efficiency, as well as accommodating participants who join and leave the FL process dynamically. We propose the Cooperative Shapley value to rectify the negative values of participants in conventional Shapley value-based schemes under the Non-IID data scenarios. We prove that CSV satisfies the three axioms of traditional SV, and conduct normalization on CSV to guarantee cross-iteration fairness. We use Strategically Equivalence among participants as a guidance to remove redundant combinations when computing CSV, which sharply reduces the computation complexity from exponential to polynomial. Experimental results demonstrate that FairFed outperforms two state-of-the-art approaches in terms of both time overhead and deviations.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61972081), the Natural Science Foundation of Shanghai (Grant No.22ZR1400200), the Fundamental Research Funds for the Central Universities (No. 2232023Y-01), a RGC RIF grant under the contract R6021-20, RGC CRF grants under the contracts C7004-22G and C1029-22G, RGC GRF grants under the contracts 16209120, 16200221 and 16207922, HK RGC under Grants R6021-20F, R1012-21, RFS2122-1S04, C2004-21G, C1029-22G, and N_CityU139/21.

References

[1] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282.

[2] Kairouz P, McMahan H B, Avent B, et al. Advances and open problems in federated learning[J]. Foundations and Trends® in Machine Learning, 2021, 14(1–2): 1-210

[3] Pandey S R, Tran N H, Bennis M, et al. A crowdsourcing framework for on-device federated learning[J]. IEEE Transactions on Wireless Communications, 2020, 19(5): 3241-3256.

[4] Lyu L, Yu J, Nandakumar K, et al. Towards fair and privacy-preserving federated deep models[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(11): 2524-2541.

[5] Xu L, Chen J, Chang S, et al. Toward Quality-aware Data Valuation in Learning Algorithms: Practices, Challenges, and Beyond[J]. IEEE Network, 2023.

[6] Zeng R, Zeng C, Wang X, et al. A comprehensive survey of incentive mechanism for federated learning[J]. preprint arXiv:2106.15406, 2021.

[7] L. S. Shapley, “A value for n-person games,” Annals of Mathematical Studies, vol. 28, pp. 307–317, 1953.

[8] Liu Z, Chen Y, Yu H, et al. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2022, 13(4): 1-21.

[9] Song T, Tong Y, Wei S. Profit allocation for federated learning[C]//2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019: 2577-2586.

[10] Wang J, Zhang L, Li A, et al. Efficient participant contribution evaluation for horizontal and vertical federated learning[C]//2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE, 2022: 911-923.

[11] Wang T, Rausch J, Zhang C, et al. A principled approach to data valuation for federated learning[M]//Federated Learning. Springer, Cham, 2020: 153-167.

[12] Fan Z, Fang H, Zhou Z, et al. Improving fairness for data valuation in horizontal federated learning[C]//2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE, 2022: 2440-2453.

[13] Dong L, Liu Z, Zhang K, et al. Affordable federated edge learning framework via efficient Shapley value estimation[J]. Future Generation Computer Systems, 2023, 147: 339-349.

[14] Jia R, Dao D, Wang B, et al. Towards efficient data valuation based on the shapley value[C]//The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 2019: 1167-1176.

[15] Castro J, Gómez D, Tejada J. Polynomial calculation of the Shapley value based on sampling[J]. Computers & Operations Research, 2009, 36(5): 1726-1730.

[16] Tang Z, Shao F, Chen L, et al. Optimizing federated learning on non-IID data using local Shapley value[C]//Artificial Intelligence: First CAAI International Conference, CICAI 2021, Hangzhou, China, June 5–6, 2021, Proceedings, Part II 1. Springer International Publishing, 2021: 164-175.

[17] Nagalapatti L, Narayanam R. Game of gradients: Mitigating irrelevant clients in federated learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(10): 9046-9054.

[18] Yuan X, Zhang K, Zhang Y. Selective Federated Learning for Mobile Edge Intelligence[C]//2021 13th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2021: 1-6.

[19] Shrot T, Aumann Y, Kraus S. On agent types in coalition formation problems[C]//Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Vol. 1, 2010: 757-764.

[20] Zhao Y, Li M, Lai L, et al. Federated learning with non-iid data[J]. arXiv preprint arXiv:1806.00582, 2018.

[21] Sattler F, Müller K R, Samek W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints[J]. IEEE TNNLS, 2020, 32(8): 3710-3722.

[22] Giraud C. Introduction to high-dimensional statistics[M]. Chapman and Hall/CRC, 2021.

[23] Shi S, Zhou X, Song S, et al. Towards scalable distributed training of deep learning on public cloud clusters[J]. Proceedings of Machine Learning and Systems, 2021, 3: 401-412.

[24] Deng L. The mnist database of handwritten digit images for machine learning research [best of the web][J]. IEEE signal processing magazine, 2012, 29(6): 141-142.

[25] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms[J]. arXiv preprint arXiv:1708.07747, 2017.

[26] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[27] Ghorbani A, Zou J. Data shapley: Equitable valuation of data for machine learning[C]//International conference on machine learning. PMLR, 2019: 2242-2251.

[28] Yurochkin M, Agarwal M, Ghosh S, et al. Bayesian nonparametric federated learning of neural networks[C]//International conference on machine learning. PMLR, 2019: 7252-7261.