IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 September 2022, accepted 27 September 2022, date of publication 30 September 2022,
date of current version 15 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3210950

RESEARCH ARTICLE

Empirical Measurement of Client Contribution for Federated Learning With Data Size Diversification

SUNG KUK SHYN[1], DONGHEE KIM[2], AND KWANGSU KIM[3], (Member, IEEE)
[1]Department of Artificial Intelligence, Sungkyunkwan Univeristy, Suwon, Gyonggi-do 16419, South Korea
[2]Department of Computer Science and Engineering, Sungkyunkwan Univeristy, Suwon, Gyonggi-do 16419, South Korea
[3]College of Computing and Informatics, Sungkyunkwan Univeristy, Suwon, Gyonggi-do 16419, South Korea

Corresponding author: Kwangsu Kim (kim.kwangsu@skku.edu)

This work was supported in part by the Healthcare AI Convergence Research & Development Program through the National IT Industry Promotion Agency of Korea (NIPA) funded by the Ministry of Science and ICT (MSIT) (50%) under Grant S0254-22-1006, and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korea Government (MSIT) (AI Graduate School Support Program, Sungkyunkwan University, 50%) under Grant 2019-0-00421.

ABSTRACT Client contribution evaluation is crucial in federated learning(FL) to effectively select influential clients. Contrary to data valuation in centralized settings, client contribution evaluation in FL faces a lack of data accessibility and consequently challenges stable quantification of the impact of data heterogeneity. To address this instability of client contribution evaluation, we introduce an empirical method, Federated Client Contribution Evaluation through Accuracy Approximation(FedCCEA), which exploits data size as a tool for client contribution evaluation. After several FL simulations, FedCCEA approximates the test accuracy using the sampled data size and extracts the client contribution from the trained accuracy approximator. In addition, FedCCEA grants *data size diversification*, which reduces the massive variation in accuracy resulting from game-theoretic strategies. Several experiments have shown that FedCCEA strengthens the robustness to diverse heterogeneous data environments and the practicality of partial participation.

INDEX TERMS Client contribution, client selection, data valuation, data heterogeneity, federated learning, incentive mechanism, shapley value.

I. INTRODUCTION

Federated Learning(FL) [1], [2], [3], [4] is an emerging field in distributed machine learning. It aggregates different models of clients in distributed systems without accessing data. Research on FL focuses on various approaches that attempt to reach a similar performance to centralized, optimal models.

In a data-centric approach, FL considers the data quality by measuring contribution at the client level. Client contribution is generally defined as the impact of a dataset for each client on the federated model performance. Measurement of client contribution is applied to two specific aspects of improving the federated model.

A. CLIENT SELECTION

Regarding the context of deep learning models, not all data have equal value [5]. Thus, preserving and discarding high- and low-quality data is a prerequisite to training a high-performing deep learning model [6]. Similarly, not all clients contribute equally to federated settings [7], [8], [9], [10]. Closely monitoring these clients and measuring the contribution of each client should be achieved to select influential clients and remove unneeded ones.

B. INCENTIVE ALLOCATION

Economically, client contribution is a suitable standard for allocating incentives fairly, while maximizing profit [11], [12], [13], [14], [15]. A proper incentive allocation with client contribution may motivate high-contributors to actively participate in FL, where the amount of high-quality data in each client affects the model accuracy. This incentive mechanism may facilitate the efficient management of revenues and costs in a business system with a highly-performing federated model by the central server (or coordinator).

Then the question is, *"how do we evaluate the client contribution in the FL setting?"* Unfortunately, a different

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

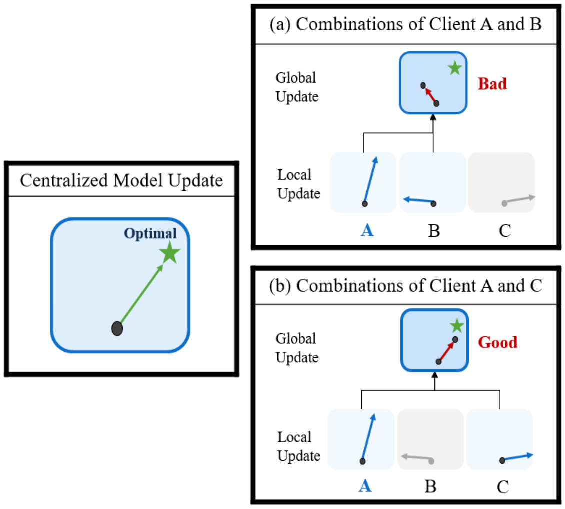


FIGURE 1. Examples of the combinatorial impact of client A in a single round with different combinations.

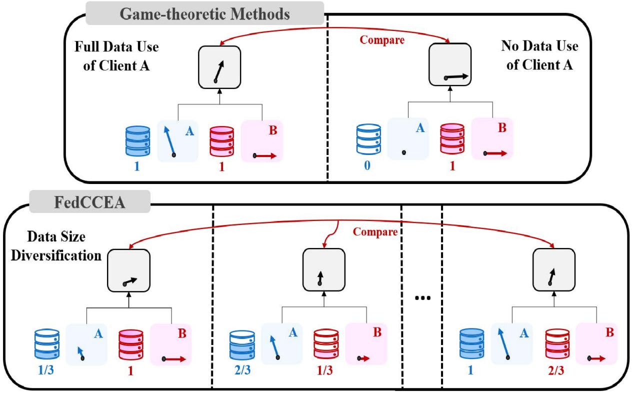


FIGURE 2. Data use cases of each contribution evaluation method while measuring contribution of client A. Regarding the game-theoretic methods, *full or no* data of client A are used to compare the global updates, including and excluding client A. On the contrary, FedCCEA enables several simulations with *data size diversification* to analyze the impact of client A on model performance, while considering various data size sets.

view from data valuation of centralized learning is required. In centralized data environments, the data value is measured based on data characteristics, such as the presence of samples with uncommon features [16], [17] and the presence of data corruptions [6], [18]. In contrast to centralized data environments, evaluating the contribution in FL environments with the actual data characteristics is impossible.

In particular, the central server cannot measure the exact impact of the data characteristics (or data heterogeneity) in FL environments. First, the blockage of accessing data restricts the server from analyzing the data heterogeneity of each local dataset [19]. This indicates that the server cannot directly seek and estimate the data distribution among the clients or the fraction of noisy data. Only local gradients, weights, and data sizes can be obtained from individual clients during the aggregation. Thus, the server can evaluate contribution only by this limited information.

Moreover, the impact of data distribution, noise, and data quantity is not as clear as in centralized settings because they strongly rely on combinations with other clients. As shown in Fig. 1, a single client can be a low contributor with some combinations ((a)) that update the global weight away from the optimal weight. On the contrary, it can also be a high contributor with different combinations ((b)) that update the global weight closer to the optimal weight. This double-faced combinatorial result causes an unstable impact of the data heterogeneity on the federated model performance.

From early explorations, Shapley Value [8], [20], a game-theoretic evaluation method, predicts the overall combinatorial impact of clients on performance by averaging the marginal test accuracy with all the possible client subsets including and excluding a client as shown in Fig. 2. Although it is a theoretically well-structured evaluation method, the client contribution measurement by Shapley Value faces challenges with extreme accuracy fluctuations of some combinations in heterogeneous data environments. These drastic combinatorial effects result in unstable client contribution estimates.[1]

To make a stable and precise quantification of the impact of data heterogeneity, we introduce a novel, empirical evaluation of client contributions using *data size*. Federated Client Contribution Evaluation through Accuracy Approximation, also known as FedCCEA, predicts the client contribution through a deep learning model named accuracy approximation model(AAM). Contrary to previous studies that only considered full or no data use, FedCCEA diversifies the proportion of data used in every round to stabilize client contribution measurement. This data size diversification may strengthen the robustness in any real-world decentralized setting and even allow the contribution measurement of partial participation with a free choice of data size. We demonstrate our strengths through experiments using three public image sets [21], [22], [23] and different data distribution settings.

This study provides three following main contributions:

1) To the best of our knowledge, this is the first *empirical* method that *allows the partial participation* of clients and *exploits data size sets* for a client contribution evaluation in FL.
2) We empirically measure client contribution through deep learning models with *diversified data size combinations* to make a stable contribution evaluation in any data setting.
3) We also conduct extensive experiments on three public image sets in real-world environments, such as non-IIDs and data corruptions. We empirically analyze the robustness to diverse heterogeneous data situations and the practicality of data size selection.

[1] The descriptions are mentioned in Section IV-B

TABLE 1. Summary of client contribution evaluation methods for federated learning.

Information Used	Keyword	Literature	Description
Local Weights/ Local Gradients	Leave-one-out	[14], [24]	Measure the marginal performance difference of a specific client's participation.
	Shapley Value	[8], [20], [25], [26]	Measure the weighted mean of the marginal performance difference of all possible subsets with a specific client's participation.
	Weight Difference	[27]	Use client contribution based on the directional difference of local weights/gradients for incentive allocation.
	DRL Models	[28], [29]	Empirically predict each client contribution using REINFORCE or DQN models with local weights/gradients.
Local Data Size	Data Quantity	[12]	Simply define a local data size as a total value of each local dataset for incentive allocation.
	FedCCEA	Ours	Empirically predict an averaged impact of each local dataset using deep learning models with diverse cases of data size.

II. RELATED WORKS

A. DATA VALUATION

Data Valuation, a phrase similar to Client Contribution Evaluation, has been widely studied recently to improve centralized machine learning models and to explain black-box predictions. The leave-one-out(LOO) method [30], [31] and the influence function [30], [31], [32] measured the counterfactual of a batch and verified whether the performance has changed owing to the batch. However, these perturbation-based methods performed poorly. For example, two identical but influential points do not value as high as they exist together.

Shapley Value [33], a classical concept in cooperative game theory, is on the rise in machine learning to tackle the poor performance of LOO. In contrast to LOO, Data Shapley [20] compared all possible training data combinations which a single datum is included and excluded. Moreover, several efficient methods to approximate the actual Shapley Value, such as Monte-Carlo SV and gradient-based SV [20], attempted to reduce the computational inefficiency that actual Data Shapley suffers. However, the issue of the high computational complexity of data valuation remains. Data Shapley costs $O(2^N)$ of computational complexity for data valuation, and Monte-Carlo SV costs $O(N \log N)$.

Subsequently, empirical methods of data valuation have been introduced as alternatives to theory-based data valuation. Data valuation using reinforcement learning(DVRL) [34] is a meta-learning framework that jointly learns the data value and trains the primary model using reinforcement learning. This method robustly approximates data values, even for low-quality datasets or other-domain samples. Moreover, it achieves high performance in machine learning tasks by removing parts of the low-valued ones.

B. CLIENT CONTRIBUTION EVALUATION FOR FL

In addition to a model-centric approach that focuses on FL optimization [35], [36], [37], [38], the client contribution evaluation in our study is a data-centric solution to the client-drift problem of *FedAvg* [2]. The server provides more credit to major clients and fewer credit to minor clients. Despite advances in data valuation methods in centralized machine learning, only a few techniques can be applied in federated environments owing to data blockages. Local gradients, local weights, and local data sizes are the only possible information that the server can use as a tool for client contribution evaluation. [2], [19]

In particular, LOO [14], [24] and Shapley Value [8] are applicable valuation methods in distributed systems that use local weights or gradients as a tool for client contribution evaluation. While these game-theoretic methods are time-consuming, a simple approximation for LOO [24] and Shapley Value [20], [25], [26], [39] makes the client contribution calculation feasible with a theoretical base.

Subsequently, evaluation methods using weight or gradient differences were introduced. RRAFL [27] considered the directional difference between the global model and local weight vectors, assuming that a lower angle contributes more. Empirically, F-RCCE [28] and FAVOR [29] applied REINFORCE and DQN models with local weights and gradients to find the best strategies for client selection to optimize the federated model regarding the measured client contribution.

On the other hand, the information of local data size has rarely been exploited for client contribution evaluation because a large amount of data does not clearly lead to a higher contribution in federated learning when data heterogeneity exists. Previously, the local data size was defined as a client contribution for simple construction of a DRL-based incentive mechanism [12] with strong assumptions. However, in addition to the local data size, quantification of the impact of data heterogeneity(e.g. data corruption and non-IID) is required to correctly measure the client contribution in any data environment.

III. PROPOSED METHOD

FedCCEA consists of two phases: simulator and evaluator. The simulator is the preparation step of the evaluation by simulating the FL procedures to obtain the inputs(*sampled data size*) and targets(*round-wise accuracy*) of the AAM in the evaluator. After all FL simulations are completed,

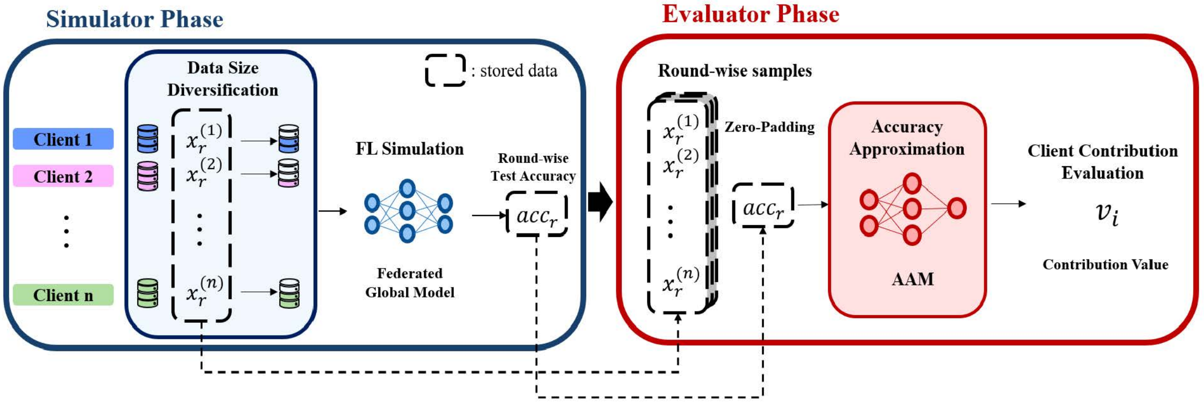


FIGURE 3. **Overall mechanism of FedCCEA. Through data size diversification, FL simulations, and testing in the Simulator Phase, *data size* and *round-wise test accuracy* sets are stored. Then, in the Evaluator Phase, these round-wise samples are reorganized by zero-padding and are used as inputs and targets of the Accuracy Approximation. We finally extract the client contribution value of client $i(v_i)$ from the AAM.**

the evaluator predicts the client contribution of each client through the AAM.

A. SIMULATOR

We denote $n \in \mathbb{N}$, $R \in \mathbb{N}$, and $S \in \mathbb{N}$ as the number of participating clients, rounds per FL simulation, and number of FL simulations, respectively. Moreover, we construct $\mathcal{D}^{(i)}$ of the training dataset for each client i and denote $\mid \mathcal{D}^{(i)} \mid$ as the total data size for each client. Nevertheless, $\mathcal{D}^{(i)}$ is not used during the testing step; the simulator uses a separate test set $\mathcal{D}^{t}$.[2] The main task of the simulator is to implement FL simulations to obtain a set of *sampled data size*($\mathbf{x}_{r,s}$) and *round-wise accuracy*($\text{acc}_{r,s}$). Therefore, we execute the simulator in three steps: data size diversification, a single FL iteration, and testing. The entire routine of these three steps is a single round of one FL simulation, and we repeat the R-rounds FL simulations S times as initially indicated.

1) DATA SIZE DIVERSIFICATION

Considering the FL environment, each client freely selects the size of its local training data for the FL in each round. To observe all possible actions of clients, we expand the cases of data size selection by randomly selecting the proportion in the uniform distribution between zero and one: $\mathbf{p}_{r,s} = (p_{r,s}^{(1)}, p_{r,s}^{(2)}, \ldots, p_{r,s}^{(n)}) \sim \mathcal{U}(0, 1)$. We turn the proportion vector to a real data size vector $\mathbf{d}_{r,s}$ for use in a single FL iteration step. Therefore, to store the data size vector in a normalized term for evaluation, we calculate the standard data size $\mid \mathcal{D} \mid = \frac{\sum \mid \mathcal{D}^{(i)} \mid}{n}$ and determine the scaled data size vector $\mathbf{x}_{r,s}$:

$$\mathbf{d}_{r,s} = (d_{r,s}^{(1)}, d_{r,s}^{(2)}, \ldots, d_{r,s}^{(n)}) \quad \text{where } d_{r,s}^{(i)} = \mid \mathcal{D}^{(i)} \mid \times p_{r,s}^{(i)} \tag{1}$$

$$\mathbf{x}_{r,s} = \frac{\mathbf{d}_{r,s}}{\mid \mathcal{D} \mid} \tag{2}$$

Algorithm 1: FedCCEA - Procedure of the Simulator

```
Input: Number of clients $n$, number of rounds per FL
       simulation $R$, number of simulations $S$, training
       set $\mathcal{D}^{(i)}$ for each client $i = 1, 2, \ldots, n$, shared test
       set $\mathcal{D}^t$, and standard data size $\mid \mathcal{D} \mid = \frac{\sum \mid \mathcal{D}^{(i)} \mid}{n}$
Initialize empty list $E$;
for $s = 1, 2, \ldots, S$ do
    Initialize parameters of the global model $\theta_0^G$ ;
    for $r = 1, 2, \ldots, R$ do
        Sample a data size proportion vector $\mathbf{p}_{r,s}$
          $= (p_{r,s}^{(1)}, p_{r,s}^{(2)}, \ldots, p_{r,s}^{(n)}) \sim \mathcal{U}(0, 1)$;
        Derive a real data size vector $\mathbf{d}_{r,s}$ =
          $(d_{r,s}^{(1)}, d_{r,s}^{(2)}, \ldots, d_{r,s}^{(n)})$ where $d_{r,s}^{(i)} = \mid \mathcal{D}^{(i)} \mid \times p_{r,s}^{(i)}$;
        Derive a scaled data size vector $\mathbf{x}_{r,s} = \frac{\mathbf{d}_{r,s}}{|\mathcal{D}|}$;
        Clients Execute:
            Collect global parameters $\theta_{r-1}^G$;
            Update local models using $\mathbf{d}_{r,s}$ and obtain
              $\theta_r^{(i)}$ for each client $i$;
        Collect $\theta_r = [\theta_r^{(1)}, \ldots, \theta_r^{(n)}]$ from all clients;
        Implement FedAvg Algorithm and update $\theta_r^G$;
        Test the updated global model using $\mathcal{D}^t$ and
          obtain a round-wise test accuracy($\text{acc}_{r,s}$);
        Store ($\mathbf{x}_{r,s}$, $\text{acc}_{r,s}$) into $E$;
return list E
```

This scaled data size vector $\mathbf{x}_{r,s}$, also defined as *sampled data size*, accelerates the convergence of AAM and allows comparison of the data size between clients.

2) SINGLE FL ITERATION

The subsequent step is a one-round FL classification task. The global model is a neural network such as an MLP or a CNN, as hypothesized in this study. During this step, the central server renews the global model parameter θ_r^G based on

[2] The location of the test set depends on the federated system design: the server side or the client side. Testing the global model is held on the site where the test set is located.

the *sampled data size* of the clients. Each client updates their local model weights $\theta_r^{(i)}$ using only $d_{r,s}^{(i)}$ of their dataset, which is their actual size for training in this round. Thereafter, the central server aggregates the local model weights using the *FedAvg* algorithm. Based on the information obtained from clients, *FedAvg* is reformulated as follows:

$$\theta_r^G = \sum_{i=1}^{n} \frac{d_{r,s}^{(i)}}{\sum_j d_{r,s}^{(j)}} \times \theta_r^{(i)} \tag{3}$$

3) TESTING

After a single FL iteration, we evaluate the current performance of the federated model using a separate test set $\mathcal{D}^t$. The metric obtained in this step is acc_r, which is defined as the *round-wise accuracy* until round r.

These three steps are repeated until the final round(R), then a single FL simulation ends. The sets of *round-wise accuracies* obtained from several FL simulations(S) are used for accuracy approximation in the evaluator phase with the *sampled data size*.

B. EVALUATOR

The evaluator phase is independent of the simulator; nonetheless, it plays a substantial role in predicting client contribution from a learned accuracy approximation model. This phase begins after the simulator phase is completed to ensure that the evaluator can obtain all stored results from the simulator.

1) ACCURACY APPROXIMATION

The AAM is a regression model that predicts *round-wise accuracy* using the *sampled data size*. This model approximates the federated test accuracy of the current round r using the *sampled data size* sets until the current round. In this model, the *sampled data size* in the previous rounds also affects the *round-wise accuracy*. For instance, the accuracy in round r is also affected by the data size set in rounds 1 to $r-1$. Therefore, the *sampled data size set* can be analyzed as time series data.

We design the AAM in a combined framework of linear regressions and a time series model using several distinctive tools, forming $g : \mathbb{R}^{n\times R} \longrightarrow [0, 1]$, as shown in Fig. 4. This model allows the *sampled data size* to be considered sequentially by organizing **static-sized inputs with zero-padding** of future actions. In addition, **shared weights** enable the quantification of the averaged impact of the local dataset, considering a certain data size for each client among the overall rounds.

a: ZERO-PADDING

An input vector of the AAM, $\Psi_r \in \mathbb{R}^{n\times R}$, is constructed with the experienced *sampled data size* sets. Considering the static $n \times R$ shape, we list the *sampled data size* sets before the current round r. Then, we **zero-pad** the rest of the space. These are the subsequent actions after round r. For each sample, we assume that the federated model is continually trained until round R with clients not participating in FL after round r. Fig. 5 demonstrates the construction of zero-padded input vectors.

Algorithm 2: FedCCEA - Procedure of the Evaluator

```
Input: Number of global rounds $R$, number of
       simulations $S$, and the results from the simulator
       $E = \{(\mathbf{x}_{r,s}, \mathrm{acc}_{r,s})\}_{r=1,\ldots,R,s=1,\ldots,S}$,
Initialize parameters $\Omega$ of AAM;
for $s = 1, 2, \ldots, S$ do
    for $r = 1, 2, \ldots, R$ do
        Construct an input vector $\Psi_{r,s} \in \mathbb{R}^{n\times R}$;
        List all data size set $\mathbf{x}_{r,s}$ under round $r$ in $\Psi_{r,s}$
          and zero-pad for the rest;
Collect $\Psi = \{\Psi_{r,s}\}_{r=1,\ldots,R,s=1,\ldots,S}$ and
  $\mathrm{acc} = \{\mathrm{acc}_{r,s}\}_{r=1,\ldots,R,s=1,\ldots,S}$;
while until convergence do
    Using $\Psi$ inputs and acc targets, optimize $\Omega$ from
      AAM: $g(\Psi; \Omega)$;
Extract the shared weights($\omega$) from the Client
  Contribution Module;
return weight vector $\omega$
```

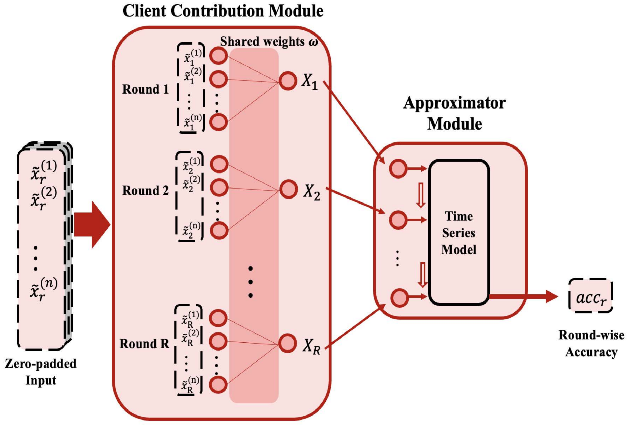


FIGURE 4. **Architecture of the AAM. This model aims to approximate round-wise accuracy(acc_r) with sampled data size($\tilde{x}_r^{(i)}$) of all clients(i) in all rounds(r) created from FL simulations.**

b: SHARED WEIGHTS

The focal point of the architecture is on the first layer of AAM, which contains **shared weights** $\omega \in \mathbb{R}^n$. Shared weights are widely used in CNN architectures as convolution filter shapes to extract the local features of image data. Similarly, the client contribution module is structured with round-wise shared weights to extract the averaged impact of the data heterogeneity for each client.

The inputs are split into R round sets, expressed as $\Psi_r = (\tilde{x}_r^{(1)}, \tilde{x}_r^{(2)}, \ldots, \tilde{x}_r^{(n)})$ for each set. The model initially designs a linear regression for each round set as $X_r(\Psi_r; \omega) = \tilde{x}_r^{(1)}\omega^{(1)} + \tilde{x}_r^{(2)}\omega^{(2)} + \ldots + \tilde{x}_r^{(n)}\omega^{(n)}$ using the shared parameter vector $\omega = (\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(n)})$. Moreover, X_r represents the

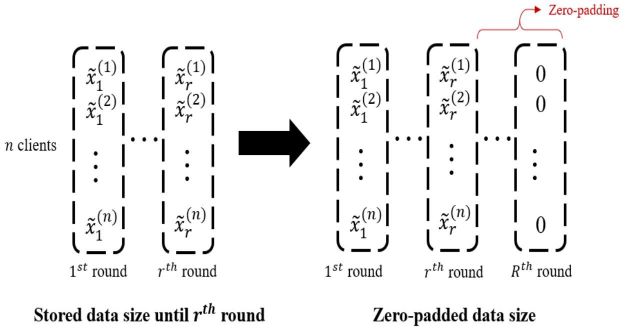

Zero-padding
n clients
$\tilde{x}_1^{(1)}$ $\tilde{x}_1^{(2)}$ $\tilde{x}_1^{(n)}$
$\tilde{x}_r^{(1)}$ $\tilde{x}_r^{(2)}$ $\tilde{x}_r^{(n)}$
0
1st round
rth round
Rth round
Stored data size until rth round
Zero-padded data size

FIGURE 5. Construction of input vector set(Ψ_r) for AAM. The elements of $\Psi_r \in \mathbb{R}^{n \times R}$ that are beyond round r are zero-padded.

round-wise latent impact of federated learning with all clients using Ψ_r in round r.

c: APPROXIMATOR MODULE

The remaining layers of the AAM are constituted as a many-to-one time series architecture, $f : X \longrightarrow [0, 1]$, which returns a single approximated accuracy. The concatenated vector $X = (X_1, X_2, \ldots, X_R)$, originating from the linear regressions $X_r(\Psi_r; \omega)$, is the input of these layers. While X indicates the latent impact of all clients in each round with a given data size set, the approximator module may be closely related to the round-wise impact with the given latent variables. Any type of sequence model is possible in this module; nonetheless, the model must perform well for the approximation of the test accuracy. For example, regarding task difficulty, we use a simple MLP [40] for MNIST and CIFAR-10 classification tasks and LSTM [41] frameworks for the EMNIST classification task. Hence, we formulate the corresponding optimization problem of the AAM as:

$$\min_{\Omega} \mathcal{L}(g(\Psi; \Omega), \text{acc}) = \min_{\omega, \Omega_{-\omega}} \mathcal{L}(f((X_1, \ldots, X_R); \Omega_{-\omega}), \text{acc}) \tag{4}$$

From Eq. 4, the latent values $(X_1, \ldots, X_R)$ are outputs of the client contribution module$(X_1(\Psi_1; \omega), \ldots, X_R(\Psi_R; \omega))$ with given data size samples(Ψ), where $\omega \in \Omega$ refers to the shared weight vector in the client contribution module. In addition, $f(X_1, \ldots, X_R)$ is the approximator module with the weight vector $\Omega_{-\omega} \in \Omega$ that predicts the round-wise accuracy(acc). We use the root mean squared error as the loss term of the AAM.

2) CLIENT CONTRIBUTION MEASUREMENT

Because quantifying the exact impact of data heterogeneity for each client is unfeasible in FL, we indirectly predict the latent impact of data heterogeneity in the client contribution module. The shared weight vector(ω) represents the importance of the data size set to the latent variable Xs. We interpret $\omega^{(i)}$ as the averaged impact of data heterogeneity for client i when $x^{(i)} = 1$. This index may indirectly include the combinatorial impact of the data distribution among clients and the noise fraction on federated model performance. In addition, the data size is an essential element that affects the performance of the federated model. Thus, the client contribution is formulated as the predicted latent impact of data heterogeneity($w^{(i)}$) with an extra weight of given data size($x^{(i)}$):

$$\text{Client Contribution Value}(v_i) = x^{(i)} \times \omega^{(i)}. \tag{5}$$

IV. EXPERIMENTS

In this section, we want to answer the following questions:

1) How does accuracy variation occur in the Shapley Value evaluation and how does FedCCEA address this problem?
2) Is FedCCEA evaluation accurate even in the strong non-IID and noisy environments?
3) Is FedCCEA evaluation accurate even with partial participation?

To answer each question, we design (i) an accuracy variation comparison, (ii) a client removal test, and (iii) a client removal test for partial participation. Moreover, we conduct additional client removal tests and experiments for complexity analysis with different numbers of clients.

A. BASIC EXPERIMENTAL SETTINGS

1) BASELINE EVALUATION METHODS

We answer the above questions and prove the strengths by comparing FedCCEA to the three baseline evaluation methods in recent studies.

- **RoundSV** [8], [39] is an approximation of Shapley Value in FL. We use the permutation-based RoundSV, utilizing Monte-Carlo sampling for SV approximation.
- **Fed-Influence in Accuracy(FIA)** [24] is a type of Fed-Influence measurement metric that simply measures the influence by investigating the effect of removing a client only. The actual FIA value can be obtained from the results of the leave-one-out test.
- **RRAFL** [27] measures the contribution of cosine similarity between the final global weight vector and current local weight vectors.

2) DATA DISTRIBUTION SETTINGS

We design diverse data distribution settings to answer these three questions. We diversify the degree of data heterogeneity based on the number of classes contained in each client and the presence of label noise. The detailed statistics of the data distribution settings are presented in Table 2. Specifically, we define the Earth Mover's Distance(ρ) [42], [43] between the distribution over classes on each client and the population distribution as the overall degree of IIDness.

For Experiment 1 in Section IV-B, we construct setting C.1 by assigning all data of the selected classes($|P| = 2$) to five clients($|C| = 5$). This may result in a high mean of ρ among clients. Moreover, 40% label noise[3] is injected into client

[3] e.g., change label '4' to '5'.

TABLE 2. Detailed statistics of data distribution settings. ($|C|$: number of clients, $|P|$: number of classes contained for each client, ρ: Earth Mover's Distance(Data Distribution), $|\mathcal{D}^{(i)}|/\sum|\mathcal{D}^{(i)}|$: scaled data size, ξ: presence of noise, $|C_{\xi}|$: number of noisy clients).

Experiments	Settings	IIDness	$\mid C \mid$	$\mid P \mid$	ρ $(mean)$	$\mid\mathcal{D}^{(i)}\mid/\sum\mid\mathcal{D}^{(i)}\mid$ $(mean)$	ξ	$\mid C_{\xi} \mid$
Experiment 1	**MNIST**							
	C.1	-	5	2	1.6	0.2	40% label	1
Experiment 2 & Experiment 3	**MNIST, CIFAR-10**							
	I.1	IID	20	10	0.00	0.05	40% label	4
	W.1	Weak non-IID	20	5	1.0	0.05	×	-
	W.2						40% label	4
	S.1	Strong non-IID	20	2	1.6	0.05	×	-
	S.2						40% label	4
	EMNIST							
	I.1	IID	20	62	0.00	0.05	40% label	4
	W.1	Weak non-IID	20	30	1.04	0.05	×	-
	W.2						40% label	4
	S.1	Strong non-IID	20	5	1.8	0.05	×	-
	S.2						40% label	4

A for more extreme data heterogeneity. This setting is used to empirically observe the unstable combinatorial effect of a client and the extreme accuracy variations of game-theoretic methods in non-IID.

For Experiments 2 and 3 in Sections IV-C and IV-D, 20 clients are constructed with a limited number of classes($|P|$) of the MNIST, EMNIST, and CIFAR-10 dataset. The settings with each client having all classes in an identical distribution are defined as IID. On the contrary, settings with each client having half or a few classes are defined as weak and strong non-IID. The mean of ρ increases as the degree of non-IID increases. Furthermore, for the weak and strong non-IID of each dataset, we assign 40% label noise(ξ) to four clients($|C_{\xi}| = 4$).[4]

3) FEDERATED LEARNING SIMULATION SETTINGS

Regarding the proper operation of FedCCEA, the configurations of federated learning simulations and a federated model should be set before the simulator phase. Three- and two-layer MLPs are constructed for the MNIST and EMNIST classification tasks, respectively, while two-layer CNNs are constructed for the CIFAR-10 classification task. In addition, we implement 100 simulations(S) for federated learning. Although 50 rounds are implemented for each simulation for the MNIST and EMNIST datasets, we implement 100 rounds for CIFAR-10 to improve the model. To reduce computational costs and enhance model performance in the EMNIST and CIFAR-10 classification task, we increase the initial learning rate and batch size compared with the MNIST classification task.

TABLE 3. Detailed configurations of a federated model simulation for each task. (lr: learning rate, B: batch size, L: local epochs, R: communication rounds, S: simulations).

Dataset	Model	lr	B	L	R	S
MNIST	3-layer MLP	0.001	32	3	50	100
EMNIST	2-layer MLP	0.05	256	1	50	100
CIFAR-10	2-layer CNN	0.1	512	1	100	100

4) CLIENT CONTRIBUTION INDEX

The evaluation methods extract client contribution values in different ranges, so we standardize these values into a unified index known as the *client contribution index*(CCI). CCI is newly measured by calculating the relative importance between clients, ranging from zero to one. The negative values outside the boundary are initially set to zero,[5] indicating that the client does not contribute to the federated model. By denoting v_i as the value of the client contribution measured in a given evaluation method, we calculate the CCIs as follows:

$$\text{Client Contribution Index(CCI)} = \frac{\tilde{v}_i}{\sum_j \tilde{v}_j}$$
$$\text{where } \tilde{v}_i = \begin{cases} 0 & \text{if } v_i \leq 0, \\ v_i & \text{otherwise.} \end{cases} \tag{6}$$

B. EXPERIMENT 1. ACCURACY VARIATION COMPARISON

Although the IIDness and noise proportion affect the client contribution of each client, the impact of these data heterogeneity elements differs based on the companions with which

[4] We do not conduct the strong non-IID setting for CIFAR-10, because the federated model is not well-performed to evaluate the client contributions. The model-centric approach(e.g. FL aggregation algorithms) is the primary approach to consider before the precise evaluation of client contribution.

[5] Despite valuing zero to negative contributors, we leave the rank of contribution between clients for the client removal experiment in Section IV-C.

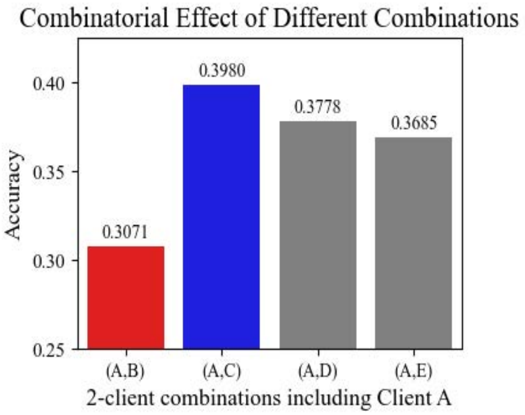


FIGURE 6. Empirical results of the combinatorial effect of the noisy client A in setting C.1.

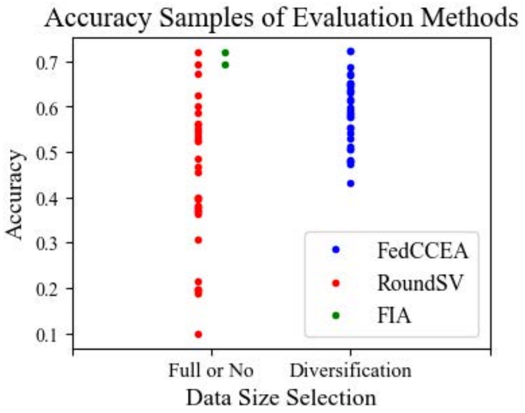
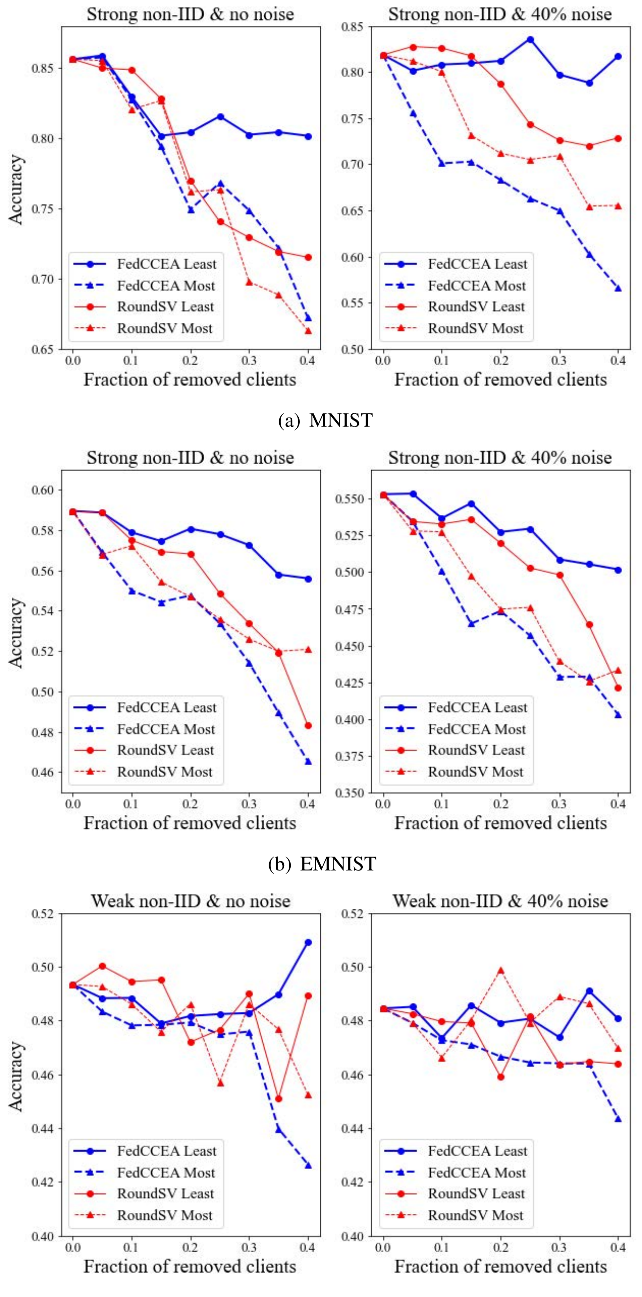


FIGURE 7. Experiment 1. Accuracy samples used to measure the contribution of Client A in setting C.1.

the client collaborates. As shown in Fig. 6, noisy client A strongly supports the enhancement when collaborating with client C. It increases the accuracy by 9.09% compared to the combination with client B. Depending on which clients participate, data corruption and different data distributions, which are the main attacks of a centralized model, can significantly enhance the federated model.

Therefore, client contribution evaluations that only consider full data usage (including RoundSV and FIA) pose a critical challenge with extreme performance variations in non-IID distribution and data-corrupted environments. The actual Shapley Value and FIA are calculated as the marginal accuracy of the combinations, including a client and excluding a client. These estimates can detect the combinatorial influence of each client. Owing to the double-faced combinatorial impact, client A in Fig. 7 suffers from a wide variation in accuracy samples between the combinations with a standard deviation of 0.4883. However, the data size diversification of FedCCEA squeezes the range of accuracy samples with a standard deviation of 0.0754. The contribution of client A is measured more stably with FedCCEA than with the other methods in Setting C.1.



(a) MNIST

(b) EMNIST

(c) CIFAR-10

FIGURE 8. Experiment 2. Client removal test for (a)MNIST in strong non-IID settings(S.1, S.2), (b)EMNIST in strong non-IID settings(S.1, S.2), and (c)CIFAR-10 in weak non-IID settings(W.1, W.2). Ideally, the straight line should maintain high performance while the dashed line should drop dramatically after removing clients. For simplicity, FedCCEA and RoundSV are only shown, while the overall evaluation metrics are described in Table 4.

C. EXPERIMENT 2. CLIENT REMOVAL TEST

In a federated setting, a direct precision test of client contribution is challenging. There are no exact ground-truth values of client contribution in federated environments that can be precisely compared. In addition, the combinatorial impact distracts the measurement of ground-truth values owing to the unclear linearity with data heterogeneity.

TABLE 4. Evaluation metrics for experiment 2. (AbC: area between the Curves, AR : Accuracy Reversal) The higher AbC is the better evaluation, and the AR mark (×) represents a good evaluation method. The last column(Best) means the number of settings that achieve the best result among evaluation methods.

Methods		MNIST					EMNIST					CIFAR-10			Best
		I.1	W.1	W.2	S.1	S.2	I.1	W.1	W.2	S.1	S.2	I.1	W.1	W.2	
FedCCEA	AbC	0.0211	**0.1333**	**0.3314**	**0.3789**	**1.1443**	**0.0593**	0.3190	**0.3848**	**0.3732**	0.5189	0.0896	**0.1651**	**0.1241**	9
	AR	×	×	×	×	×	×	×	×	×	×	×	×	×	
RoundSV	AbC	0.0193	0.056	0.0084	0.1243	0.3964	0.0408	0.2645	0.0106	0.0421	0.2080	-0.1306	0.0556	-0.0742	0
	AR	×	×	√	√	×	×	×	×	√	√	√	√	√	
FIA	AbC	**0.0277**	0.1165	0.3240	0.1346	1.0995	0.0522	**0.3616**	0.3093	0.3313	**0.7308**	0.1132	0.0310	0.0739	3
	AR	×	×	×	√	×	×	×	×	×	×	×	√	√	
RRAFL	AbC	0.0232	-0.0166	0.2499	-0.3588	-0.4707	0.0567	0.2499	0.3796	0.1181	0.4680	**0.1162**	-0.0246	0.1238	1
	AR	×	√	×	√	√	×	×	×	×	×	×	×	×	

Alternatively, the client removal test [8], [24], [34] is commonly used for the precision testing of client contribution measurements. As shown in Fig. 8, we incrementally remove clients in descending and ascending order of CCIs and retrain the model. By correctly selecting clients to be removed, the model with removed low CCIs(straight line) consistently retain high test accuracy. In contrast, the performance of eliminating the highest contributors(dashed line) decreases substantially. The two possible evaluation metrics are as follows:

- **Accuracy Reversal(AR)**: Regarding any proportion of client removal, the accuracy of high-CCI removal should not exceed the accuracy of low-CCI removal within the same proportion. Moreover, AR should not exist for any type of dataset or setting.
- **Area between the Curves(AbC)**: If the client contribution is properly measured, the difference between the accuracy of the removed low-contributors($acc_{\text{low,frac}}$) and that of the removed high-contributors($acc_{\text{high,frac}}$) would be estimated to be high. Therefore, we measure AbC using the following equation:

$$\text{AbC} = \sum_{\text{frac}} acc_{\text{low,frac}} - acc_{\text{high,frac}} \tag{7}$$

Fig. 8 shows that FedCCEA makes a more precise evaluation of client contribution than RoundSV. While RoundSV experiences accuracy reversals in the MNIST and CIFAR-10 settings, FedCCEA produces the expected results of the correct client contribution measurement with no accuracy reversal.

Specifically, even though RoundSV achieves a clear gap between the ‘Least’ and the ‘Most’ in EMNIST setting(Fig. 8(b)), FedCCEA achieves a much wider gap than RoundSV that results in higher AbC than RoundSV. In addition, evaluating the client contributions for CIFAR-10 dataset(Fig. 8(c)) is challenging; however, FedCCEA achieves consistent results with a positive AbC and no AR, whereas RoundSV clearly experiences an accuracy reversal in both settings W.1 and W.2.

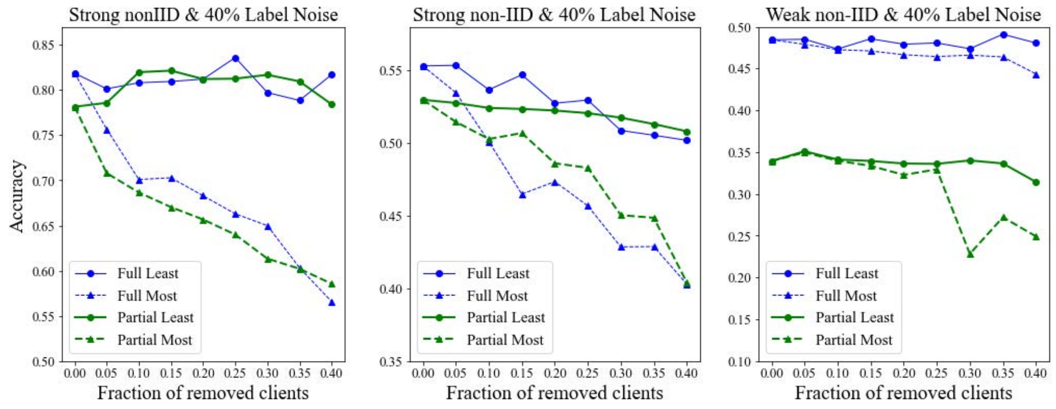
Table 4 presents the outstanding results of the proposed method. FedCCEA is the only method in which accuracy reversal does not exist in the data distribution settings that we construct, whereas the accuracy of the removed high-contributors exceeds the accuracy of the removed low-contributors in some cases for the other evaluation methods. Furthermore, FedCCEA obtain the highest AbCs in nine out of 13 data settings, whereas FIA and RRAFL obtain the highest AbCs in three and one setting, respectively. Overall, FedCCEA shows robust measurements in most of the heterogeneous data environments, whereas the other baseline methods fail to achieve robustness to specific data heterogeneity.

D. EXPERIMENT 3. CLIENT REMOVAL TEST FOR PARTIAL PARTICIPATION

Another advantage of FedCCEA is that it can measure client contributions even if clients partially participate in FL using only a part of the local dataset. By verifying the same experiment as in Section IV-C, we demonstrate the precision of client contribution, even with the allowance of partial participation, and demonstrate its practicality. We (1) randomly assign the data size of each client in every round, (2) rank the CCIs in both descending and ascending orders, and (3) retrain the federated model by removing a given proportion of the highest and lowest contributors in every round. Finally, (4) we compare the results to the case of FedCCEA using the full data.[6]

As shown in Fig. 9, all datasets exhibit outstanding results in the partial use case. It does not lead to accuracy reversal(AR), and provides a similar or superior result for the area between the curves(AbC) compared to the case of full use. In addition to the settings in Fig. 9, we can also confirm that all other data distribution settings reach a similar superiority to the partial use cases. Thus, without reevaluating client contributions, FedCCEA can obtain precise results for partial participation. This widens the options for clients of data size selection in distributed systems.

[6] Partial participation of clients cannot be applied to the baseline methods because extra evaluations are required for every action. Therefore, we only consider FedCCEA results.

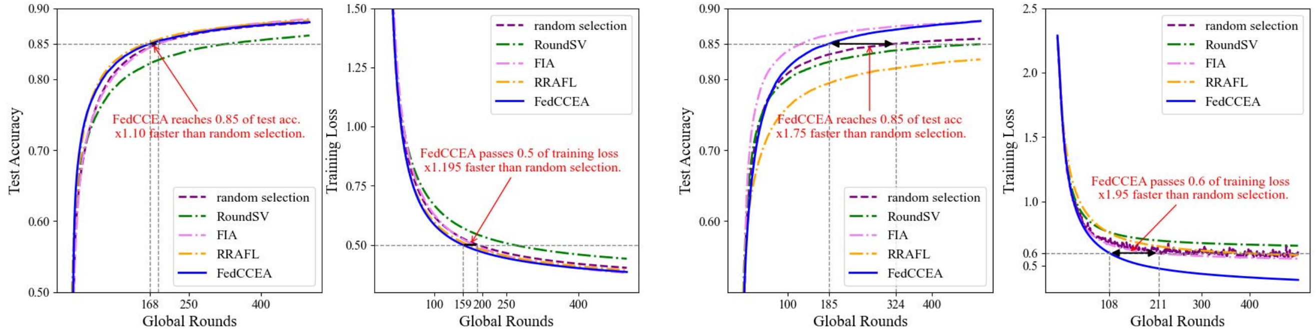


(a) MNIST S.2 setting

(b) EMNIST S.2 setting

(c) CIFAR-10 W.2 setting

FIGURE 9. Experiment 3. Client removal test for partial participation in (a)S.2 setting of MNIST, (b)S.2 setting of EMNIST, and (c)W.2 setting of CIFAR-10. Ideally, the straight line should maintain high performance while the dashed line should drop dramatically after removing clients.



(a) MNIST S.1 setting

(b) MNIST S.2 setting

FIGURE 10. Training loss and test accuracy performance for 4-client exclusion strategies in (a) S.1 setting and (b) S.2 setting. The difference of convergence speed seems to be trivial between FedCCEA-based client selection and contribution-based selection strategies. However, FedCCEA constantly reaches a convergence point faster than uniformly random selection in both settings.

E. FURTHER EXPERIMENTS

1) CONVERGENCE ANALYSIS FOR CLIENT SELECTION

From a client selection perspective, choosing influential clients and removing unnecessary ones are crucial challenges in federated learning with high data heterogeneity. [44], [45] Partial client participation in a highly heterogeneous data environment can result in slow convergence if inappropriate clients are selected for federated learning. On the other hand, efficient selection of influential clients can result in faster convergence and higher performance. Many studies have introduced client selection strategies for partial client participation using local losses [44], clustering through gradient diversity [46], and linear speedup [47].

With the same aggregation algorithm(*FedAvg*), contribution-based client selection strategies have a similar convergence rate to each other. However, when we measure the low- and high-contributors correctly, client selection or exclusion strategies with contributions can achieve a reward of fast convergence speed in a highly heterogeneous environment compared to uniformly random selection. As shown in Fig. 10, we exclude four low-contributors measured by each evaluation method and investigated the convergence point of both training loss and test accuracy. The MNIST S.1 and S.2 settings, which are strong non-IID environments with and without label noise, are used in this experiment.

As a result, we discover that the contribution-based client selection strategies have a trivial difference in convergence speed. However, we can claim that the FedCCEA-based client selection consistently achieves faster convergence than random selection in both heterogeneous settings. Furthermore, the gap in the convergence speed between FedCCEA and random selection becomes more substantial in S.2. setting(Fig. 10(b)) when FedCCEA correctly captured four poisoning attackers and excluded them from federated learning participation.

2) CLIENT REMOVAL TEST WITH DIFFERENT NUMBER OF CLIENTS

In addition, we implement client removal tests for FedCCEA with a different number of clients. In this experiment, we incrementally remove four clients for the 10-client FL and eight for the 20- and 50-client FL. Subsequently, we

TABLE 5. Client removal test with different numbers of clients. (AbC: Area between the Curves, AR : Accuracy Reversal) As a standard of a robust evaluation method, AbC should remain positive, and AR should be marked as (×).

Methods	Client Number	Metrics	IID	Weak Non-IID	Strong Non-IID
FedCCEA	10	AbC (avg.)	0.0021	0.0191	0.0715
		AR	×	×	×
	20	AbC (avg.)	0.0026	0.0414	0.1430
		AR	×	×	×
	50	AbC (avg.)	0.0084	0.0070	0.0172
		AR	×	×	×

TABLE 6. Experiment for complexity analysis with different number of clients(in seconds) (*S*: number of simulations, *N*: number of clients) Empirically, we design a federated setting with 20 samples for each client, 10 rounds, and only implement a single simulation(*S* = 1).

Methods	Complexity (S, N)	Client Number				
		5 $(\times 1)$	10 $(\times 2)$	20 $(\times 4)$	40 $(\times 8)$	80 $(\times 16)$
FedCCEA	$O(S)$	5.55s $(\times 1)$	5.43s $(\times 0.98)$	6.02s $(\times 1.08)$	5.82s $(\times 1.05)$	6.09s $(\times 1.10)$
RoundSV	$O(N \log N)$	5.71s $(\times 1)$	8.17s $(\times 1.43)$	8.32s $(\times 1.46)$	24.15s $(\times 4.23)$	77.63s $(\times 13.60)$
FIA	$O(N^2)$	59.88s $(\times 1)$	116.65s $(\times 1.95)$	229.72s $(\times 3.84)$	488.04s $(\times 8.15)$	1284.18s $(\times 21.44)$
RRAFL	$O(N)$	9.50s $(\times 1)$	10.17s $(\times 1.07)$	10.90s $(\times 1.15)$	12.87s $(\times 1.35)$	16.60s $(\times 1.75)$

compare the consequence of the average AbC and the presence of AR.

Clearly, as the number of clients participating in federated learning increases, the contribution of each client becomes more difficult to evaluate. The increased number of client combinations causes more extreme accuracy variations and makes the evaluation unstable. However, as shown in Table 5, FedCCEA provides robust results for evaluation metrics with positive averaged AbC and no AR for any number of clients.

3) COMPLEXITY ANALYSIS

FedCCEA needs to repeat numerous simulations(S) of federated learning to obtain sufficient samples for an accuracy approximation in the evaluator phase. Thus, FedCCEA requires $O(S)$ complexity for contribution evaluation. In contrast, the evaluation complexity of other baseline methods is highly dependent on the number of clients participating in FL. RoundSV costs $O(N \log N)$ [8], FIA costs $O(N^2)$ [24], and RRAFL costs $O(N)$.

The empirical experiment for the complexity analysis in Table 6 shows that RoundSV, FIA, and RRAFL incur a massive cost of contribution evaluation when the number of clients increases. In contrast, the time cost of FedCCEA remains nearly constant. Therefore, when the number of clients in each round is large, the evaluation of FedCCEA is remarkably faster than that of the other baselines. However, when the number of clients is small, FedCCEA, with numerous FL simulations(S), evaluates the client contribution much slower than the other baseline methods.

V. CONCLUSION

In addition to a model-centric approach that focuses on FL optimization, client contribution evaluation is another effective approach for improving the FL performance. In this study, we proposed FedCCEA, an empirical measurement of client contributions, without accessing local datasets. We built an accuracy approximation model to distinctively exploit the data size for accuracy approximation and to extract stable client contributions by considering a given data size. Contrary to other evaluation methods, the data size diversification of FedCCEA loosens the accuracy variation of FL simulations and strengthens the robustness to diverse data settings and practicality for partial participation.

REFERENCES

[1] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. 2017, pp. 1273–1282.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, and R. G. D'Oliveira, "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*.

[5] M. Toneva, A. Sordoni, R. T. des Combes, A. Trischler, Y. Bengio, and G. J. Gordon, "An empirical study of example forgetting during deep neural network learning," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19.

[6] H. Ferdowsi, S. Jagannathan, and M. Zawodniok, "An online outlier identification and removal scheme for improving fault detection performance," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 908–919, May 2013.

[7] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[8] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song, "A principled approach to data valuation for federated learning," in *Federated Learning*. Springer, 2020, pp. 153–167.

[9] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," 2020, *arXiv:2002.00211*.

[10] W. Zhang, T. Zhou, Q. Lu, X. Wang, C. Zhu, H. Sun, Z. Wang, S. K. Lo, and F.-Y. Wang, "Dynamic-fusion-based federated learning for COVID-19 detection," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15884–15891, Nov. 2021.

[11] K. L. Ng, Z. Chen, Z. Liu, H. Yu, Y. Liu, and Q. Yang, "A multi-player game for studying federated learning incentive schemes," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 5179–5281.

[12] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[13] L. Lyu, X. Xu, Q. Wang, and H. Yu, "Collaborative fairness in federated learning," in *Federated Learning*. Springer, 2020, pp. 189–204.

[14] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 2597–2604.

[15] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu, "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2020.

[16] H. Touvron, A. Vedaldi, M. Douze, and H. Jegou, "Fixing the train-test resolution discrepancy," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8252–8262.

[17] J. Ngiam, D. Peng, V. Vasudevan, S. Kornblith, Q. V. Le, and R. Pang, "Domain adaptive transfer learning with specialist models," 2018, *arXiv:1811.07056*.

[18] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2013.

[19] P. Vepakomma, T. Swedish, R. Raskar, O. Gupta, and A. Dubey, "No peek: A survey of private distributed deep learning," 2018, *arXiv:1812.03288*.

[20] A. Ghorbani and J. Zou, "Data Shapley: Equitable valuation of data for machine learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2242–2251.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[22] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2921–2926.

[23] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[24] Y. Xue, C. Niu, Z. Zheng, S. Tang, C. Lyu, F. Wu, and G. Chen, "Toward understanding the influence of individual clients in federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, 2021, pp. 10560–10567.

[25] T. Song, Y. Tong, and S. Wei, "Profit allocation for federated learning," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 2577–2586.

[26] S. Maleki, L. Tran-Thanh, G. Hines, T. Rahwan, and A. Rogers, "Bounding the estimation error of sampling-based Shapley value approximation," 2013, *arXiv:1306.4265*.

[27] J. Zhang, Y. Wu, and R. Pan, "Incentive mechanism for horizontal federated learning based on reputation and reverse auction," in *Proc. Web Conf.*, 2021, pp. 947–956.

[28] J. Zhao, X. Zhu, J. Wang, and J. Xiao, "Efficient client contribution evaluation for horizontal federated learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3060–3064.

[29] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 1698–1707.

[30] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.

[31] R. D. Cook, "Detection of influential observation in linear regression," *Technometrics*, vol. 19, no. 1, pp. 15–18, Feb. 1977.

[32] A. Richardson, A. Filos-Ratsikas, and B. Faltings, "Rewarding high-quality data via influence functions," 2019, *arXiv:1908.11598*.

[33] L. S. Shapley, "A value for n-person games," in *Classics Game in Theory*, vol. 69. 1997.

[34] J. Yoon, S. Arik, and T. Pfister, "Data valuation using reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10842–10851.

[35] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–16.

[36] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 1–22.

[37] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. Vincent Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," 2020, *arXiv:2007.07481*.

[38] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1–12.

[39] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos, "Towards efficient data valuation based on the Shapley value," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1167–1176.

[40] V. F. Barabanov, O. J. Kravets, I. N. Kryuchkova, O. Y. Makarov, A. K. Pogodayev, and O. N. Choporov, "Discrete processes dynamics neural network simulation based on multivariate time series analysis with significant factors delayed influence consideration," *World Appl. Sci. J.*, vol. 23, no. 9, pp. 1239–1244, 2013.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FEDAVG on non-IID data," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–26.

[43] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3557–3568.

[44] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," 2020, *arXiv:2010.01243*.

[45] Y. Ruan, X. Zhang, S.-C. Liang, and C. Joe-Wong, "Towards flexible device participation in federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 3403–3411.

[46] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning via submodular maximization," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–18.

[47] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-IID federated learning," 2021, *arXiv:2101.11203*.

SUNG KUK SHYN received the B.S. degree in economics and science in engineering from Sungkyunkwan University, in 2021, where he is currently pursuing the M.S. degree with the Department of Artificial Intelligence. His research interests include federated learning, time series forecasting, AI applications, and explainable AI.

DONGHEE KIM received the M.S. degree in electric and computer engineering from Sungkyunkwan University, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. He served as the AI Assistant Research Engineer at Ntels Company Ltd. He is also working at Hippo T&C Inc., as an AI Research Engineer. His research interests include federated learning, AI applications, and self-supervised learning.

KWANGSU KIM (Member, IEEE) received the Ph.D. degree in computer science from the University of Southern California, in 2007. He worked as the Director-General of the Ministry of Science and ICT, South Korea. He is currently a Professor with the College of Computing and Informatics, Sungkyunkwan University. He is also the Director of the Sungkyun AI Research Institute. His research interests include computer vision, domain adaptation, federated learning, AI applications, and explainable AI.

• • •