All Federated or Not: Optimizing Personal Model Performance in Cross-silo Federated Learning

1st Juan Li
Nanjing University of Aeronautics and Astronautics
China
juanli@nuaa.edu.cn

2st Yanmin Zhu
Shanghai Jiao Tong University
China
yzhu@sjtu.edu.cn

3st Jie Wu
Temple University
USA
jiewu@temple.edu

4st Weifan Wu
Nanjing University of Aeronautics and Astronautics
China
wwf0218@nuaa.edu.cn

5st Tianzi Zang
Nanjing University of Aeronautics and Astronautics
China
zangtianzi@nuaa.edu.cn

6st Liu Lu
Nanjing University of Aeronautics and Astronautics
China
luvluvlu@nuaa.edu.cn

***Abstract*—In cross-silo federated learning (FL), organizations cooperatively train a global model with their local data. The organizations, however, may be heterogeneous in terms of data distributions. In such cases, FL might produce a biased global model that is not optimal for each organization. Then each organization faces several fundamental questions: should I join FL or just remain alone? If joining FL, which organizations should I cooperate with? In this work, we formulate a coalition formation game in cross-silo FL to help organizations choose proper cooperators. We first build an estimation method to predict personal model performance for each organization before FL starts, and we treat performance improvement as individual utility. With estimated utilities, we design a distributed coalition formation algorithm to find stable coalition structures and optimize social welfare at the same time. Our simulations based on MNIST and FMNIST datasets show that the estimation model can predict the sign of the utility correctly with a probability of 0.9 and has an average relative error of 30%. With the above errors, the obtained coalition structure performs well from both perspectives of real social welfare and individual satisfaction.**

***Index Terms*—Federated Learning, Model Performance, Non-IID, Coalition Formation**

I. Introduction

Federated learning (FL) has become increasingly popular as a distributed machine learning framework. By FL, multiple data owners train machine learning models together, with data staying locally and only local models being transferred. As a result, data privacy is protected and network bandwidth is saved.

FL can be classified into two types: cross-device FL and cross-silo FL. In cross-device FL, an organization acts as the central server and the participants are usually their clients who are owners of smart devices. The global model is owned by the organization. In cross-silo FL, a third-party entity acts as the central server coordinating the training process. Organizations are participants performing local training. They own the global model and use it to serve their clients and make a profit. Recently, cross-silo FL has attracted much attention. For example, WeBank and Swiss Re have collaborated for data analysis in finance and insurance. Owkin cooperates with medical institutions for biomedical data analysis.

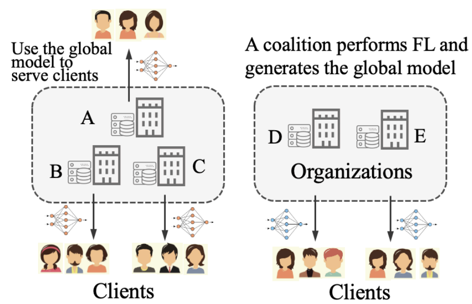


Fig. 1: Coalition formation in cross-silo federated learning.

In cross-silo FL, each organization cares most about its personal model performance, i.e., the performance of the global model on its personal data distribution. A better personal model performance means a higher potential profit when applying the model to serve clients. However, samples from different organizations are usually heterogeneous or non-IID. As a result, the global model may not suit everyone. For example, if a bank with small-loan business is federated with banks with large-loan business, the trained loan risk estimation model would perform poorly for this bank.

Due to such negative effects of non-IID data, all organizations being federated together is not always a good idea. We give an example illustrated by Table I based on a benchmark FL dataset MNIST. MNIST has handwritten digits from 0 to 9. The organizations have different distributions. Taking A for example, it has three classes. The data of each class is sampled

This work was supported by Natural Science Foundation of Jiangsu Province under Grant No. BK20220882, National Natural Science Foundation of China under Grant No. 62202224 and No. 62172313, China Postdoctoral Science Foundation under Grants No. 2022TQ0154 and No. 2023M741685, Dual Innovation Doctor Foundation of Jiangsu Province under Grant No. JSSCBS20220213. (Corresponding author: Tianzi Zang.)

TABLE I: Motivation example

ORG	Distribution	LAccu	FAccu	CAccu
A	{0, 1, 2}, 0.003	0.83	0.80	0.84
B	{1, 1, 3}, 0.001	0.75	0.79	0.82
C	{5, 6, 7}, 0.005	0.94	0.92	0.94
D	{6, 7, 8}, 0.001	0.73	0.91	0.93

from that class of MNIST according to a ratio of 0.003. When training a model alone, the local accuracy is $LAccu$. If all of them are federated together, the personal accuracy is $FAccu$. In the last case, A and B form a coalition, perform FL together, and share the global model. At the same time, C and D form a coalition. The resulted personal accuracy is $CAccu$. Compared with training a model alone, A and C have a lower accuracy when all organizations are federated. The last case is the best as the accuracies of all organizations are improved.

The above phenomenon raises some **fundamental questions** for each organization aiming to optimize its personal model performance. Should I federate with others? If yes, which organizations should I choose? Will they be interested in collaborating with me? Can we be arranged into stale coalitions and if so, what do these coalitions look like? To answer these questions, we study how to **arrange organizations into coalitions to improve their personal model performance**, as illustrated by Fig. 2.

Solving the coalition formation problem is nontrivial. **First**, the personal model performance is the optimization objective, but it can only be known after FL is done. There is a work deriving the expected error of personal model performance in linear regression problems [15]. However, it needs some impossible-to-know information including the error variance among all organizations. Some works study the relationship between the model performance and participants' contributions. Unfortunately, the values of contributions are also obtained after FL [13] [14] [10] [11]. **Secondly**, satisfying all organizations and optimizing the social welfare at the same time is difficult. An organization has its own utility. It would not follow a coalition formation decision not optimizing its utility even if this decision brings the largest social welfare. Even worse, the social welfare optimization problem is NP-hard. **Thirdly**, we estimate the personal model performance for each organization before FL and perform coalition formation according to estimation results. However, the estimation results have errors. Controlling the effect of the error on coalition formation is the last challenge we need to overcome.

Many works have explored the issue of non-IID data in FL. However, most of them deal with this issue by optimizing FL algorithms, e.g., increasing the weights of disadvantaged participants when aggregating local models [24] [25], limiting the deviation degree of each local model from the global model [19]–[23], fine-tuning the global model according to the local data set [26]. Distinctly from these works, we overcome the problem of non-IID data from the new perspective of data coalition formation. We directly avoid a significant non-IID degree before FL starts, rather than reducing its negative impact in the process of FL.

To overcome the challenges and limitations of existing works, we solve two problems: personal model performance estimation and coalition formation based on the estimated results. For the first problem, we find 8 critical factors influencing personal model performance by correlation analysis based on the benchmark FL dataset MNIST. Then we use neural networks to fit the personal model performance with these factors as inputs. The factors include non-IID degree, data volume, local performance trained by each organization alone, and so on. Calculating these factors needs some private information about organizations, such as their data distributions. We use differential privacy to protect them. And fortunately, it does not bring a much higher error to our estimation method. With the estimated results as inputs, we design distributed coalition formation algorithms based on the idea of better response dynamics. The algorithms converge to Nash-stable solutions or individually stable solutions depending on whether joining a coalition needs original members' permissions. By carefully designing the initial coalition structure, the social welfare of the obtained solution is very close to the optimal social welfare.

- To the best of our knowledge, we are the first to study how to form coalitions to optimize each organization's personal model performance for general cross-silo FL problems.
- At the same time, we are the first to propose a personal model performance improvement estimation method which works before FL starts. This method can predict whether the improvement is positive or negative in about 90% cases. It has an average relative error of 30%.
- We design a coalition formation algorithm converging to stable coalition structures and close to the optimal one. Although we use the estimated utilities with errors, the obtained coalition structure performs well from both perspectives of real social welfare and individual satisfaction. 90% of organizations prefer our coalition structure to forming a grand coalition.

II. System Model and Problem Formulation

A. System Model

Consider a set N of organizations. Each organization i has a local sample set $W_i = \{(x, y)\}$. The vector $x = \{x^1, x^2, ..., x^m\}$ represents a specific value of input feature vector $X = \{X^1, X^2, ..., X^m\}$ and y is the value of label Y. The set $\hat{X}$ of most important features describing characteristics of organizations' clients is a subset of $X \cup \{Y\}$. For example, when hospitals train a disease prediction model together, characteristic vector $\hat{X}$ may include two features: the age of each patient and the label of whether having the target disease. The data distribution of organization i about $\hat{X}$ is $P_i(\hat{X}) = \{p_i(\hat{x})\}$. For each specific vector value $\hat{x}$, $p_i(\hat{x}) = n_i(\hat{x})/n_i$ where $n_i = |W_i|$ and $n_i(\hat{x})$ is the number of samples whose $\hat{X}$ is equal to $\hat{x}$. The local sample set W_i is divided into a training set D_i and a test set T_i, both of which follow distribution $P_i(\hat{X})$.

When some organizations form a data coalition $C \subseteq N$, they collaboratively perform federated learning and get a machine learning model $f(\theta)$ with θ as the parameter vector. We adopt the most widely used FL algorithm, FedAvg [2], whose aim is to train a model $f(\theta)$ minimizing the average loss $\sum_{i\in C}|D_i|l(\theta, D_i)/\sum_{i\in C}|D_i|$ of all organizations in coalition C. The notation $l(\theta, D_i)$ is the loss over dataset D_i given θ. In FedAvg, the training process goes on round by round. In each round, a central server first distributes the global model $f(\theta)$ to each organization. Then each organization i initializes its local parameters as θ and trains a local model $f(\theta_i)$ based on the local training set D_i. After all organizations submit local models, the central server updates the parameters θ of the global model as $\sum_{i\in C}|D_i|\theta_i/\sum_{i\in C}|D_i|$. This process continues until the global model converges.

Organizations in a coalition C care most about the performance of $f(\theta)$ on their personal data distributions $P_i(\hat{X})$, which is called the personal model performance $v_i(C)$. This is because organizations usually use $f(\theta)$ to serve their clients. Better model performance can bring more profit. Here, the model performance means accuracy, F1-score or other metrics measuring a model's prediction ability. Then $v_i(\{i\})$ is the model performance trained by the local set D_i alone. The utility $u_i(C)$ of organization i in coalition C is the personal model performance improvement formulated as follows.

Definition 1 (Utility of an Organization: Personal Model Performance Improvement). *For organization i, it's utility $u_i(C)$ is the difference between the personal model performance $v_i(C)$ obtained by joining coalition C and $v_i(\{i\})$ obtained by training the machine learning model alone.*

$$u_i(C) = v_i(C) - v_i(\{i\}), i \in C \tag{1}$$

B. Coalition Formation Framework in Cross-silo FL

To help organizations form proper coalitions, we propose a coalition formation framework which describes the interactions between the platform and organizations, as shown in Fig. 2. The platform can be the central server coordinating the federated learning process as mentioned in Section II-A. It can also be a new one only helping coordinate the coalition formation process.

In the framework, each organization first submits two pieces of information to the platform. The first is the number $n_i(\hat{x})$ of training samples of each $\hat{x}$. The second is the performance $v_i(\hat{x}, \{i\})$ of model $f(\theta_l)$ trained alone on each $\hat{x}$ in the local test set T_i, e.g., the accuracy of $f(\theta_l)$ on samples whose $\hat{X}$ is equal to $\hat{x}$ in T_i.

The information $n_i(\hat{x})$ may be sensitive and expose the privacy of an organization's clients. For example, the platform knowns $n_i(\hat{x})$ is 90 and 91 before and after a client comes to organization i, respectively. Then it knows that characteristic $\hat{X}$ of the client is $\hat{x}$. Fortunately, this kind of privacy exposure risk can be effectively reduced by the technique of differential privacy. Here, we adopt the Laplace mechanism in differential privacy which adds noise to $n_i(\hat{x})$ according to Laplace

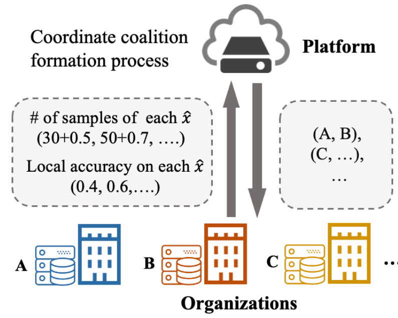


Fig. 2: Coalition formation framework.

distribution $L(0, \Delta f/\epsilon)$ [6]. Δf is the difference of $n_i(\hat{x})$ with and without the client. It is equal to 1 in our problem. ϵ is the privacy budget and we set it as 1.

After getting information from organizations, the platform solves the following two problems to help organizations form coalitions:

- **Estimate personal model performance improvement**: The platform estimates utility function $u_i(C)$ of each organization i according to $n_j(\hat{x})$ and $v_j(\hat{x}, \{j\})$ with $j \in C$. The estimated result is $u'_i(C)$.
- **Solve the coalition formation problem**: The platform solves the coalition formation problem based on estimated utilities $u'_i(C)$ by distributed algorithms to satisfy each organization.

At last, the platform publishes the found solution to organizations, and organizations perform FL collaboratively according to the coalition structure.

C. Coalition Formation Problem under Selfishness

The considered problem is how to divide organizations into proper coalitions so that each organization can be satisfied with its utility, i.e., personal model performance improvement. At the same time, we hope the social welfare can be as large as possible. Solving this problem helps encourage organizations to continuously participate into cross-silo FL, which is extremely important to cross-silo FL applications.

As each organization aims to optimize its own utility, we treat them as selfish players and formulate the considered problem as a coalition formation game $(N, \geq)$. The notation $N = \{1, ..., n\}$ is the set of organizations. In set $\geq = \{\geq_i |i \in N\}$, $\geq_i$ represents the preference of organization i over different coalitions. It is a binary and transitive relation determined by the utility of organization i, i.e., $C \geq_i C'$ if and only if $u_i(C) \geq u_i(C')$.

The solution of game $(N, \geq)$ is a coalition partition $\Pi = \{C_k\}_{k=1}^K$. $C_k \subseteq N$ are disjoint with each other and $\cup_{k=1}^K C_k = N$. We use $\Pi(i)$ to denote the index of the coalition including i, i.e., $i \in C_{\Pi(i)}$.

From the perspective of the platform, the optimal coalition partition is the one maximizing the overall utility of all organizations. At the same time, the utility of each organization should not be negative because an organization is usually an enterprise which needs to make a profit. Finding the optimal coalition partition is difficult because the related optimization problem (2) is NP-hard [30].

Definition 2 (**The Optimal Coalition Partition**). *The optimal coalition partition maximizes the social welfare, i.e., sum of utilities of all organizations, and satisfies the constraint that the utility of each organization is non-negative.*

$$\begin{aligned} & max_{\Pi} \sum_{i \in N} u_i(C_{\Pi(i)}) \\ & \quad s.t.\ u_i(C_{\Pi(i)}) \geq 0 \\ & \quad\quad u_i(C_{\Pi(i)}) = v_i(C_{\Pi(i)}) - v_i(\{i\}) \\ & \quad\quad \Pi(i) = k\ \ if\ \ i \in C_k \,\&\, C_k \in \Pi \end{aligned} \tag{2}$$

However, selfish organizations would not stay in the coalition specified by the optimal partition if they can improve their utilities by moving to other coalitions. Then desirable coalition structures should be those satisfying all organizations. For such a coalition partition, organizations would stay stably in the specified coalition and do not move.

Here, we make use of stable concepts in hedonic games to formulate desirable coalition structures in our problem. A coalition formation game is a hedonic game if each players utility is completely determined by its coalition and is independent of other coalitions [30]. In cross-silo FL, the personal model performance of an organization is totally determined by organizations in its coalition and has nothing to do with other coalitions. Therefore, our coalition formation game is a hedonic game.

In hedonic games, there are two types of stable concepts about individual deviation. The first is called the Nash stable partition defined by Definition 3. In a Nash stable partition, no player has an incentive to unilaterally change its coalition because it cannot get a higher utility.

Definition 3 (**Nash Stable Partition**). *A coalition partition Π is Nash stable if $C_{\Pi(i)} \geq_i C \cup \{i\}$ for any $i \in N$ and any $C \in \Pi \cup \{\emptyset\}$.*

The Nash stable partition has a default assumption that an organization can join a coalition as long as it wants. The Nash stable partition is strong and achieves the well-known Nash equilibrium. But it is actually difficult to be reached. According to our evaluation results, Nash stable partitions exist only in 50% cases.

To deal with non-existence of Nash stable partitions, we introduce another stable concept: the individually stable partition. It is adopted when joining a coalition needs the permissions of coalition members. According to Definition 4, in an individually stable partition Π, no organization would move to another coalition $C \in \Pi \cup \{\emptyset\}$ due to the following two reasons. First, some of them cannot get a higher utility, i.e., $u_i(C_{\Pi(i)}) \geq u_i(C \cup \{i\})$. Second, some organizations can get a higher utility by moving to target coalition C. But there exist several members of C not allowing it to join because their utilities would be decreased, i.e., $u_j(C \cup \{i\}) < u_j(C)$. Then, the partition would not change anymore and becomes stable. Individually stable partitions are easier to be achieved because it is harder for organizations to change their coalitions when entrance permissions are needed.

Definition 4 (**Individually Stable Partition**). *A coalition partition Π is individually stable if there does not exist $i \in N$ and coalition $C \in \Pi \cup \{\emptyset\}$ such that $C \cup \{i\} >_i C_{\Pi(i)}$, and $C \cup \{i\} \geq_j C$ for all $j \in C$.*

In this work, we hope to achieve Nash stability or individual stability and optimize the social welfare at that same time. Note that the real utility $u_i(C)$ is not available before FL. Therefore, we perform coalition formation based on the estimated utility $u'_i(C)$. A partition is stable if organizations cannot improve $u'_i(C)$ by changing coalitions. Fortunately, the coalition structure found based on estimated utilities works well according to the real social welfare and real individual satisfaction.

III. Personal Model Performance Improvement Estimation

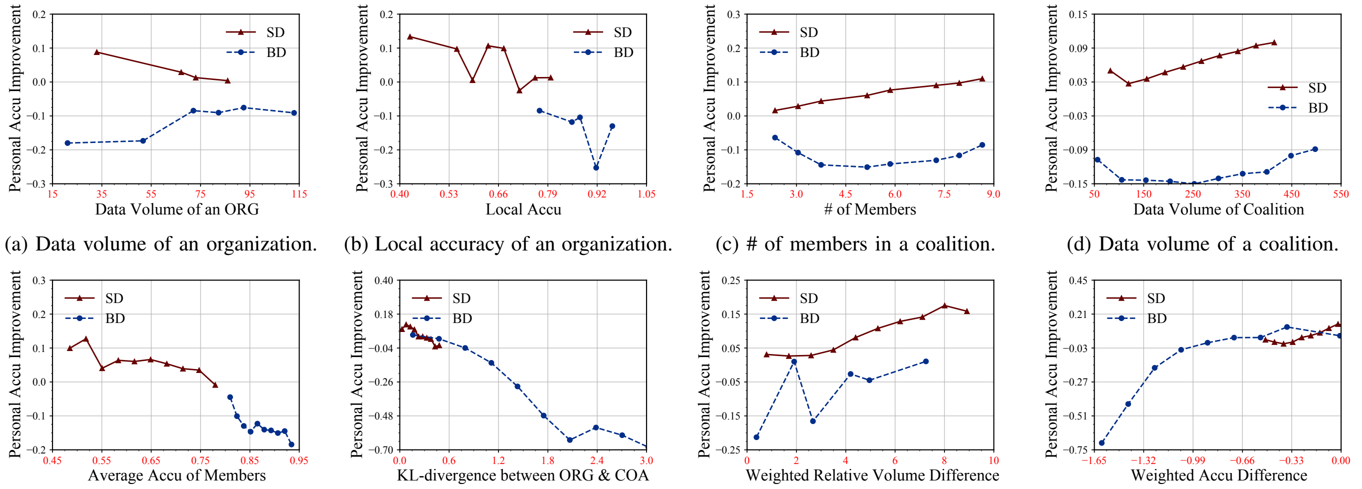
We find that utility $u_i(C)$ of each organization, i.e., its personal model performance improvement, has correlations with some factors such as non-IID degree, data volume, the local performance trained alone, and so on. We have verified such correlations about 17 factors and selected 8 critical factors by correlation analysis based on the experiments about a benchmark FL dataset MNIST [3]. We fit $u_i(C)$ by neural networks with these influence factors as inputs.

A. Critical Factors Influencing Utility

The dataset MNIST consists of handwritten digits uniformly distributed across 10 labels $0-9$. It has around 60000 training samples and 10000 test samples. The number of organizations (ORGs) is 9. The characteristic set $\hat{X} = \{Y\}$ as it is difficult to extract other characteristics from pixels. We have two settings. In the first setting, an organization only has three or four classes. Its distribution is listed in Table II as $P_i^e(Y)$, which is represented by a pair of (classes, ratio of data volume of each class to that class in MNIST). An organization has the totally same classes with someone and totally different classes with others. Therefore, this setting is called BD (Bigger data Difference). The second setting called SD (Smaller Difference) simulates a more realistic case where each organization has major and minor classes. The distribution is $P_i^r(Y)$. Major classes are listed in $P_i^r(Y)$ and all remaining classes are minor

TABLE II: Data distributions of ORGs

ORG	$P_i^e(y)$	$P_i^r(y)$
1	{0, 1, 2}, 0.001	{0, 1, 2}, 0.001
2	{0, 1, 2}, 0.003	{0, 1, 2}, 0.003
3	{0, 1, 2}, 0.005	{2, 3, 4}, 0.001
4	{3, 4, 5}, 0.001	{2, 3, 4}, 0.003
5	{3, 4, 5}, 0.003	{4, 5, 6}, 0.001
6	{3, 4, 5}, 0.005	{4, 5, 6}, 0.003
7	{6, 7, 8, 9}, 0.001	{6, 7, 8}, 0.001
8	{6, 7, 8, 9}, 0.003	{6, 7, 8}, 0.003
9	{6, 7, 8, 9}, 0.005	{7, 8, 9}, 0.001



(a) Data volume of an organization. (b) Local accuracy of an organization. (c) # of members in a coalition. (d) Data volume of a coalition. (e) Average accuracy of members in a coalition. (f) KL-divergence between ORG and COA. (g) Weighted relative volume difference between ORG and COA. (h) Weighted accuracy difference between ORG and COA.

Fig. 3: The correlation between the utility and each influence factor

classes whose data volume is 0.05% of that class in MNIST. An organization may have some same major classes and some different major classes with another organization.

The selected 8 factors influencing $u_i(C)$ can be divided into three categories: factors about organization i, coalition C and difference between organization i and the coalition C. We introduce how they influence $u_i(C)$ in Fig. 3. In each figure, the value of the factor is divided into 10 ranges, and we report average $u_i(C)$ corresponding to each range if there are some points falling in the range.

1) Factors about organization i: In this category, the first factor is the training data volume of organization i, which can be calculated according to submitted information $n_i(\hat{x})$ as $n_i = \sum_{\hat{x}} n_i(\hat{x})$. Utility $u_i(C)$ has opposite correlations with this factor as shown in Fig. 3a. In setting SD, the correlation is negative because larger data volume usually leads to better local accuracy and smaller improvement with respect to local accuracy. In setting BD, the correlation is positive because only when data volume is large enough, organization i would not be affected by other distributions. We can see that accuracy improvements are all negative in this setting.

The second factor is the local model performance obtained by training the model alone. We can calculate its value by $v_i(\{i\}) = \sum_{\hat{x}} n_i(\hat{x}) u_i(\hat{x}, \{i\})/n_i$. According to Fig. 3b, utility decreases with an increasing local accuracy. It means that when the local performance is already good, the improvement got by joining FL is limited. In setting SD, minor classes only have a small data volume and get low accuracy. Therefore, their overall accuracies are also low.

2) Factors about coalition C: The factors in this category are the number of members $|C|$, the total data volume $\sum_{i \in C} n_i$ and the average local accuracy of all members $\sum_{i \in C} v_i(\{i\})/|C|$. We show how they influence $u_i(C)$ in Fig. 3c-Fig. 3e. We can see that $u_i(C)$ has a significantly positive correlation with the first two factors and a negative correlation with the last one. In Fig. 3c and Fig. 3d, $u_i(C)$ under BD setting first decreases and then increases. This is because the non-IID degree in this setting is higher. When the number of members is not enough or the data volume is small, the benefit of FL cannot overcome the negative impact brought by non-IID phenomenon. Even in setting SD, $u_i(C)$ decreases at the beginning when the data volume is very small.

3) Factors about difference between organization i and coalition C: This category has three factors. The first is the KL-divergence $KL(P_i(\hat{X})||P_C(\hat{X}))$ measuring distribution difference. The second is the weighted relative volume difference with respect to each value $\hat{x}$ of variables $\hat{X}$, $\sum_{\hat{x}} v_i(\hat{x}, \{i\})(n_C(\hat{x}) - n_i(\hat{x}))/n_i(\hat{x})$. The third is the weighted accuracy difference calculated by equation (3). The accuracy difference is the difference between the local accuracy of organization i about $\hat{x}$ and the average local accuracy of all other organizations. The weight is the ratio of $n_i(\hat{x})$ to the total data volume of organization i.

$$\frac{\sum_{\hat{x}} n_i(\hat{x})(\sum_{j \in C} v_j(\hat{x}, \{j\}) - v_i(\hat{x}, \{i\}))}{\sum_{\hat{x}} n_i(\hat{x})} \tag{3}$$

Fig. 3f-Fig. 3h describe relationships between $u_i(C)$ and these factors. Fig. 3f shows that $u_i(C)$ decreases with a larger KL-divergence. The weighted relative volume difference and the weighted accuracy difference can explain well how FL benefits each participator. Only for these two factors, average $u_i(C)$ represented by each point increases to positive values under BD setting. The trend is more smooth when considering the weighted accuracy difference because this factor has a more direct relationship with $u_i(C)$.

B. Utility Fitting by Neutral Networks

We train neutral networks by available previous FL results to fit utility $u_i(C)$. This is because the considered 8 factors influence $u_i(C)$ collaboratively. Traditional fitting methods such as the method of least square error do not work as it is hard to choose a proper fitting function describing such a complex collaborative relationship involving many factors.

To reduce the burden of collecting previous FL results when our method is used in the real-world cross-silo FL system, we choose a simple network structure shown in Fig. 4. It has three fully connected layers with a tanh activation function following each layer. The input is the 8 factors $\{f_1, ..., f_8\}$.

Cold starting. We notice that our estimation method has a cold starting problem when there are no previous FL results of the considered machine learning problem to train the estimation model. Fortunately, our estimation model trained by a specific machine learning problem still works in similar learning problems. We verify this transferring ability by applying the estimation model trained by MNIST dataset to the learning problem about FMNIST. As shown in the evaluation, the relative estimation error is about 40%. The accuracy about sign prediction of $u_i(C)$ is about 80%. Coalition formation results based on this transferred network are still better than just forming a grand coalition all together.

IV. Design of Coalition Partition Algorithm

We design a distributed coalition partition algorithm, i.e., Alg. 1 based on the idea of best-response or better-response dynamics. It takes estimated utilities as inputs and converges to a Nash stable partition or an individually stable partition when it terminates. By setting the initial partition as the optimal one found by Alg. 2 based on dynamic programming, the social welfare achieved by Alg. 1 is close to the best social welfare.

A. Finding Nash Stable Partition

The main idea of finding a Nash stable partition is based on best-response dynamics. In Alg. 1, lines 1-5 set an initial partition. We introduce the details of initialization later. After initialization, the algorithm goes on round by round. In each step of a round, one player i can change its coalition if the best coalition is not the current one. The best coalition is the one maximizing $u'_i(C)$. If a movement happens, the partition is updated at lines 12-14. The algorithm terminates when the number m of movements in a round is equal to 0.

B. Finding Individually Stable Partition

When finding an individually stable partition, commonly used best-response dynamics do not work. This is because the members of the best coalition may not accept player i. Therefore, we extend best-response dynamics to better-response dynamics. As shown in Alg. 1, when it is player i's turn to move, it lists coalitions according to $u'_i(C_k \cup \{i\})$ in a

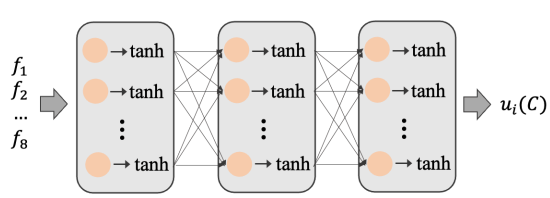


Fig. 4: Utility estimation model

Algorithm 1: Coalition Formation Algorithm

Input: N, $\{u'_i(C)\}_{C \subseteq N}$ sent to each $i \in N$, r
Output: Π
1: **if** $|N| \leq r$ **then**
2: $\quad \Pi = OptPar(N, \{u'_i(C)\}_{i \in N, C \subseteq N})$, $m = 1$
3: **else**
4: $\quad \Pi = \{C_i = \{i\}\}_{i \in N}$, $m = 1$
5: Send Π to each $i \in N$
6: **while** $m > 0$ **do**
7: $\quad m = 0$
8: $\quad$ **for** $i \in N$ **do**
9: $\quad\quad$ ———–If finding a Nash stable partition———–
10: $\quad\quad C_{k'} = argmax_{C_k \in \Pi \cup \{\emptyset\}} u'_i(C_k \cup \{i\})$
11: $\quad\quad$ **if** $k' \neq \Pi(i)$ **then**
12: $\quad\quad\quad m = m + 1$
13: $\quad\quad\quad C_{\Pi(i)} = C_{\Pi(i)} \backslash \{i\}$
14: $\quad\quad\quad C_{k'} = C_{k'} \cup \{i\}$
15: $\quad\quad$ ——–If finding an individually stable partition——–
16: $\quad\quad$ Sort $C_k \in \Pi \cup \{\emptyset\}$ according to $u'_i(C_k \cup \{i\})$ in a descending order as $C_{1_i}, C_{2_i}, ...$
17: $\quad\quad$ **for** each k_i **do**
18: $\quad\quad\quad$ **if** C_{k_i} is C_{Π_i} **then**
19: $\quad\quad\quad\quad$ Break
20: $\quad\quad\quad$ **else if** $\sum j \in C_{k_i} \mathbb{1}_{u'_j(C_{k_i} \cup \{i\}) \geq u'_j(C_{k_i})} / |C_{k_i}| >= \rho$ **then**
21: $\quad\quad\quad\quad m+ = 1$
22: $\quad\quad\quad\quad C_{\Pi(i)} = C_{\Pi(i)} \backslash \{i\}$
23: $\quad\quad\quad\quad C_{k_i} = C_{k_i} \cup \{i\}$

descending order. Then it moves to the first coalition C_{k_i} in the list satisfying the following two conditions. Condition 1 is that C_{k_i} is better than the current one. Second, the ratio of members accepting i in C_{k_i} is not smaller than threshold ρ. A member is willing to accept i when adding i into its coalition would not damage its utility, i.e., $u'_j(C_{k_i} \cup \{i\}) \geq u'_j(C_{k_i})$.

The value of ρ can be flexibly determined by the platform. If it is 1, player i can join a coalition successfully only when all members agree. Then Alg. 1 converges to an individually stable partition as proven in Proposition 1. It can also be set to another value, such as 0.5. Then a coalition accepts i if half of its members support i. This is in accordance with voting mechanisms.

C. Improve Social Welfare of Stable Partitions

Besides considering individual utilities, we hope to optimize the social welfare at the same time. Specifically, we hope Alg. 1 converges to the stable partition close to the optimal one if there are multiple stable solutions. Therefore, we set the initial coalition structure as the optimal structure. In the extreme case where the optimal partition satisfies Nash stability or individual stability, the algorithm would converge directly. However, the optimal partition is the solution to the NP-hard problem (2) whose computing time increases fast with the number $|N|$ of organizations. Therefore, when $|N|$ is larger than threshold r, we simply let each organization form a coalition in the initial partition. When $|N| < r$, we make use of dynamic programming to further reduce the computation burden, as shown in Alg. 2.

Alg. 2 compares the optimal partitions having different number l of coalitions and returns the partition with the highest social welfare among them. The social welfare of a partition is represented by s and the highest social welfare found so far is S.

The optimal partition with a specified l is calculated by Alg. 3 based on dynamic programming. The recursive structure of dynamic programming is defined by equation (4) and realized by lines 7-16 in Alg. 3. To calculate the optimal partition $\Pi^o_{N',l}$ with set N' and l, we determine and fix the first coalition C in $\Pi^o_{N',l}$ and find the optimal partition $\Pi^o_{N'\backslash C,l-1}$ whose organization set is $N'\backslash C$ and number of coalitions is $l-1$. We try every legal coalition C satisfying the three requirements R1, R2 and R3. Requirement R2 is that the size of C is not larger than $|N'|-(l-1)$ because the remaining organizations must be enough to form $l-1$ coalitions. Requirement R3 means that each member in C must have a non-negative utility. When the stop condition of recursion, $l=1$, is satisfied, the optimal partition only contains one coalition. It is returned if it is valid.

$$\begin{aligned}\Pi^o_{N',l} =& argmax_{C:R1-R3} s(\{C\} \cup \Pi^o_{N'\backslash C,l-1})\\ R1 :& C \subseteq N'\\ R2 :& |C| \leq |N'|-l+1\\ R3 :& u'_i(C) \geq 0,\ \forall i \in C\end{aligned} \tag{4}$$

D. Theoretical Analysis

We perform theoretical analysis in this subsection. Some related works about hedonic games analyze theoretically whether stable partitions exist and whether the stable partition is unique under the assumption that the utility function is simple. For example, the utility function is additively separable, symmetric or single-peaked [30] [31] [32].

Unfortunately, our utility $u'_i(C)$ has complex relationships with many factors such as data volume, local model performance and non-IID degree. It does not have those desirable properties. According to evaluation results, our coalition formation game has and only has one individually stable partition in almost all cases. For Nash stability, our game sometimes does not have stable partitions and sometimes has multiple stable partitions. The good news is that we can prove our algorithm converges to stable partitions if they exist.

Proposition 1. *If Alg. 1 terminates, the found solution is a Nash stable partition or an individually stable partition when threshold ρ is 1.*

Proof. Assume that the found partition is Π when Alg. 1 terminates. Obviously, for each organization i, $u'_i(C_{\Pi(i)}) \geq u'_i(C \cup \{i\})$ for all $C \in \Pi \cup \{\emptyset\}$ because no one leaves the current coalition $C_{\Pi(i)}$. Therefore, Π satisfies Definition 3 and Alg. 1 terminates with a Nash stable partition.

When finding individually stable partitions with threshold $\rho = 1$, it terminates if for each organization i, one of the following two conditions is satisfied. First, $u_i(C_{\Pi(i)}) \geq u_i(C \cup \{i\})$ for all $C \in \Pi \cup \{\emptyset\}$. Second, in any coalition $C \in \Pi \cup \{\emptyset\}$ better than $C_{\Pi(i)}$, the utility of at least one member j is damaged if i joins C, i.e., $u_j(C \cup \{i\}) < u_j(C)$. Then, there does not exist $i \in N$ and a coalition $C \in \Pi \cup \{\emptyset\}$ such that $C \cup \{i\} >_i C_\Pi(i)$ and $C \cup \{i\} \geq_j C$ for all $j \in C$. Definition 4 is satisfied and Alg. 1 converges to an individually stable partition. □

Algorithm 2: Finding Optimal Partition: OptPar()

Input: N, $U = \{u'_i(C)\}_{i \in N, C \subseteq N}$
Output: Π
$S = 0, \Pi = \{\{i\}_{i \in N}\}$
for $l = 1, ..., N$ **do**
  $(s, \pi) = OptParL(l, N, U)$
  if $s \geq S$ **then**
    $S = s, \Pi = \pi$

Algorithm 3: Finding Optimal Partition with *l*: OptParL()

Input: l, N', $U = \{u'_i(C)\}_{i \in N, C \subseteq N}$
Output: S, Π
1: $\Pi = \emptyset, S = -1, u_i = -1, \forall i \in N$
2: **if** $l = 1$ **then**
3:   $u_i = u'_i(N'), \forall i \in N'$
4:   **if** $u_i \geq 0, \forall i \in N'$ **then**
5:     $\Pi = \{N'\}, S = \sum_{i \in N'} u_i$
6: **else**
7:   **for** $w = 1, ..., |N'|-(l-1)$ **do**
8:     **for** $C \subseteq N'$ with size w **do**
9:       $u_i = u'_i(C)$ for $i \in C$
10:       **if** $u_i \geq 0, \forall i \in C$ **then**
11:         $\pi = \{C\}$, $s = \sum_{i \in C} u_i$
12:       **else**
13:         continue
14:       $(\Delta s, \Delta \pi) = OptPart(l-1, N'\backslash C, U)$
15:       **if** $s + \Delta s \geq S$ & $\Delta s \neq -1$ **then**
16:         $S = s + \Delta s, \Pi = \pi \cup \Delta \pi$

V. Experimental Evaluation

A. Methodology and Settings

Our experiments are based on datasets MNIST [3] and FMNIST [4]. FMNIST has Zalando's article images with labels from 10 classes, such as dress, bag, shirt and so on. For each organization, the number of major classes is randomly selected in the range of [2,4]. The data volume of each class is a specific ratio of that class in the original dataset. The ratio is randomly taken among 0.1%, 0.3% or 0.5%. In the setting of Bigger Difference (BD), an organization only has major classes. In the setting of Medium Difference (MD), an organization can have other minor classes whose data volume is 0.05% of that class in MNIST or FMNIST. The privacy budget in Laplace mechanism is 1. Our utility estimation model is trained by FL results about MNIST. We also use this model to estimate utility in the learning problem about FMNIST to verify its transferring ability. All reported results are the average results of 10 experiments.

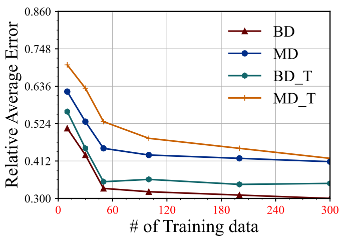


Fig. 5: Relative average estimation error vs. # of previous FL results as training data.

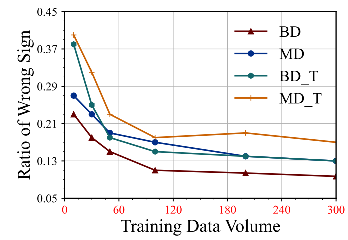


Fig. 6: Ratio of estimation with wrong sign vs. # of FL results.

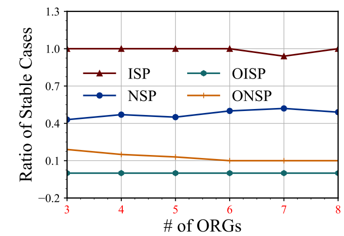


Fig. 7: Ratio of cases with stable partitions vs. # of ORGs.

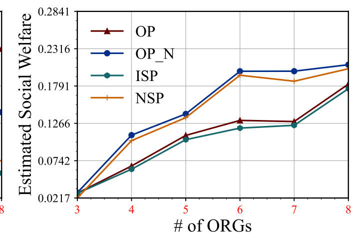


Fig. 8: Estimated social welfare vs. # of ORGs.

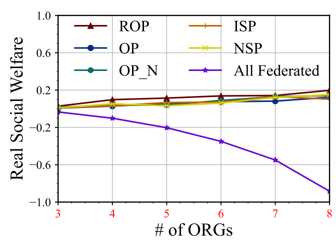


Fig. 9: Real social welfare vs. # of ORGs.

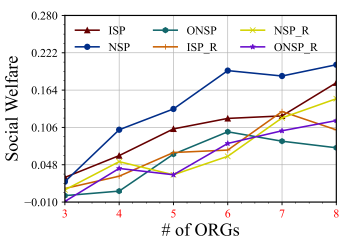


Fig. 10: Social welfare of stable partitions vs. # of ORGs.

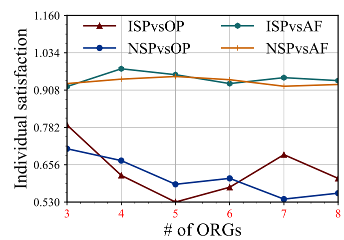


Fig. 11: Individual satisfaction vs # of ORGs.

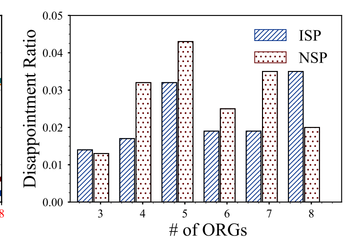


Fig. 12: Individual disappointment vs # of ORGs.

B. Personal Model Performance Improvement Estimation

We evaluate our method of estimating the personal model performance improvement in this subsection. We have two metrics. The first is the relative average estimation error $(\mathbb{E}[|u_i'(C)|] - \mathbb{E}[|u_i(C)|])/\mathbb{E}[|u_i(C)|]$. The second is the ratio of the cases where the estimated utility sign is wrong. This metric is critical to organization experience. If the platform tells an organization that you can get a positive utility as long as you stay in this coalition, but it gets a negative utility finally, this organization may never join FL anymore.

We show the evaluation results in Fig. 5 and Fig. 6. Fig. 5 describes the relative average estimation error under four settings, BD, MD, BD_T, and MD_T. In the settings of BD_T and MD_T, we transfer the estimation model trained by previous FL results about MNIST dataset to FL about FMNIST dataset. The error under setting BD converges to 30%. This is the smallest among four settings because the number of intersected classes among organizations is smaller and the pattern of mutual influence is relatively simple. The evaluation results under settings BD_T and MD_T show that our model has the ability of transferring to similar FL problems and can be used to deal with cold-staring scenarios. Another good news is that our estimation method does not need much previous FL experience as the estimation error starts to be stable when the number of previous FL results is about 60.

Fig. 6 shows the ratio of estimation results with wrong signs under the four settings. The performance of our estimation model is satisfactory. The ratio of wrong signs decreases to 10% when the number of previous FL results is about 120 under setting BD. Even when being transferred to similar FL problems about FMNIST, the wrong-sign ratio does not exceed 20%.

C. Coalition Formation

In this subsection, we show three types of evaluation results about coalition formation. The first is the convergence performance, i.e., whether the algorithm can converge to stable partitions. The second is social satisfaction including estimated social welfare and actual social welfare obtained after FL is performed. The last is individual satisfaction measuring individual utility improvement compared with the optimal partitions and cases where all organizations are federated together.

Convergence performance. Fig. 7 shows the probabilities of at least one stable solution existing and multiple stable solutions existing, respectively. From Fig. 7, Individually Stable Partitions (ISP) almost always exist and we can find them. Nash Stable Partitions (NSP) only exist in about 50% cases. This is because it is much easier for organizations to change its coalition in the Nash stable setting. The ratio of having Other Nash Stable Partitions (ONSP) is as low as 10% when the number of organizations is larger than 6. What's more, there is only one ISP in most cases according to the line for Other Individually Stable Partitions (OISP). In conclusion, it is better to pursue ISP as there exists one and only one ISP in most cases.

Social satisfaction. Fig. 8 shows the social welfare calculated based on the estimated individual utilities. We compare several types of partitions: the Individually Stable Partitions (ISP) found by Alg. 1, the Nash Stable Partitions (NSP) found by Alg. 1, the Optimal Partitions (OP) found by Alg. 2. The line for OP_N is the average social welfare of OPs when Nash stable partitions exist. As shown in Fig. 8, the estimated social welfare increases with more organizations (ORGs). This is because the optimization space is larger when there are more ORGs. In addition, ISP performs almost the same as OP, and NSP performs almost the same as OP_N because we set OP as the initial partition in Alg. 1. NSP usually exists in cases where cooperation is better than training alone because the social welfare for OP_N and NSP is much higher than OP and ISP.

Fig. 9 shows the actual social welfare obtained after FL is done when different partitions are adopted. ROP means the Optimal Partition found according to the Real individual utilities. The line for "All Federated" is the social welfare when all organizations form one coalition. Its average social welfare is negative, which means that in cross-silo FL, letting all organizations be federated together is not a good idea. The social welfares of OP and ISP are close to ROP, which represents that the loss of social welfare caused by utility estimation error can be accepted.

Fig. 10 plots the estimated and real social welfares when multiple stable partitions exist. The lines for ISP, OISP, ISP_R (Real social welfare of ISP) and OISP_R (Real social welfare of OISP) consider the cases where multiple individually stable partitions are found. Similarly, the lines for NSP, ONSP, NSP_R and ONSP_R consider the cases where multiple Nash stable partitions exist. As shown in the figure, ISP is better than OISP according to both the estimated and actual social welfares. The same as NSP and ONSP. Then we can say that by setting the initial partition of Alg 1 as the optimal partition, the obtained stable solutions are more close to the optimal one than other stable solutions.

Individual satisfaction. We define two metrics, individual satisfaction and individual disappointment to measure whether organizations prefer our coalition formation scheme. Individual satisfaction of partition Π_1 compared to Π_2 is defined as $S_{\Pi_1,\Pi_2} = |\{i|u_i(C_{\Pi_1(i)}) \geq u_i(C_{\Pi_2(i)})\}|/|N|$. It is the ratio of organizations whose utilities are increased in Π_1 compared to Π_2. Individual disappointment measures whether those organizations expecting positive utilities really get positive utilities. If an organization is told that it can get a positive utility as long as it follows the stable solution, but it gets a negative utility after FL is done, the organization would be disappointed.

Fig. 11 and Fig. 12 show individual satisfaction and individual disappointment, respectively. According to Fig. 11, ISP and NSP found by our algorithms are better than both OP and AF (All Federated together) because corresponding individual satisfaction is larger than 0.5, which means more than half of organizations prefer our coalition partition scheme than OP and AF. From Fig. 12, the ratio of disappointed organizations is less than 5%. Then most organizations who expect positive utilities would really get positive utilities.

VI. Related Work

A. Participant Selection in FL

Most related works consider cross-device scenarios because the first emerging and popular FL application is in cross-device scenarios, i.e., Gboard training a next word prediction model by coordinating many smartphones through FL. These works usually dynamically select devices having strong computing ability, fast communication speed or high data contribution in each FL round [5], [7]–[12], [16]. They aim to decrease the total training time or improve the model performance. Whether a device is good can be measured by its performance in the last round, e.g, the used time to return the local model and the accuracy contribution. These works cannot be applied to our problem where members of a coalition need to be determined before the first round. They need information in previous rounds to select participants in the next round.

In cross-silo FL, existing works usually assume that all organizations are federated together because the common assumption is that more participants lead to better model performance. And organizations have no limit about computing ability and communication speed. They would not affect the training time. Related works study how many training samples or how many computing resources each contribute should contribute to minimize the training cost and optimize the payment minus the cost [17], [18].

There is a work similar to our paper [15]. However, they only consider the learning problems of average value prediction or linear regression. They derive the expected error of the global model on each organization's distribution according to the error variance among organizations. However, error variance is impossible to be known. Moreover, the derived error is an expectation. Moreover, they consider simple FL where the global model is just the average of local models trained alone.

B. Model Performance Optimization in Non-IID Scenario

When data distributions of different participants are heterogeneous, the convergence of FL process would be slow. And some participants may obtain poor model performance. Most works optimize model performance in non-IID scenarios by improving FL training algorithms. For example, some limit the deviation degree of each local model from the global model [19]–[23]. Some increase the weights of disadvantaged participants when aggregating local models to improve fairness [24], [25]. Some other works generate personal models for each participant by fine-tuning the global model or multi-task training algorithms [26], [27].

These works cannot solve our problem. They try to reduce the negative impact of non-IID data in the process of FL. However, we hope to avoid a significant non-IID degree before FL starts by carefully designing a proper coalition structure.

VII. Conclusion

In this paper, we solve a coalition formation problem in cross-silo federated learning to optimize the personal model performance for each organization. We first make use of previous FL results to train a neural network which estimates the utility function of each organization for the next time of FL. Based on the estimation results, we help organizations form stable coalitions by a distributed algorithm. The found stable coalition structure is close to the optimal one. The solution performs well with respect to both real social welfare and individual satisfaction.

REFERENCES

[1] A. Durrant, M. Markovic, D. Matthews, D. May, J. Enright, G. Leontidis, "The role of cross-silo federated learning in facilitating data sharing in the agri-food sector," *Computers and Electronics in Agriculture*, vol. 193, 106648, 2022.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, Communication-efficient learning of deep networks from decentralized data, *Proc. of Intl Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2017.

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

[4] https://github.com/zalandoresearch/fashion-mnist

[5] B. Luo, W. Xiao, S. Wang, J. Huang, L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," *Proc. of 2021 IEEE Conference on Computer Communications(INFOCOM)*, pp. 1739-1748, 2022.

[6] C. Dwork, F. McSherry, K. Nissim, A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," *in Theory of cryptography conference*, pp. 265-284, 2006.

[7] S. Liu, G. Yu, R. Yin et al., "Joint Model Pruning and Device Selection for Communication-Efficient Federated Edge Learning," *IEEE Transactions on Communications(ToC)*, vol. 70, no. 1, pp. 231-244, 2021.

[8] S. AbdulRahman, H. Tout, A. Mourad et al., "FedMCCS: Multicriteria client selection model for optimal IoT federated learning," *IEEE Internet of Things Journal(IoTJ),* vol. 8, no. 6,pp. 4723-4735, 2020.

[9] J. Shin, Y. Li, Y. Liu et al., "Sample Selection with Deadline Control for Efficient Federated Learning on Heterogeneous Clients," *arXiv preprint arXiv:2201.01601*, 2022.

[10] Y. Deng, F. Lyu, J. Ren et al., "AUCTION: Automated and Quality-Aware Client Selection Framework for Efficient Federated Learning," *IEEE Transactions on Parallel and Distributed Systems(TPDS)*, vol. 33, no. 8, pp. 1996-2009, 2022.

[11] J. Shin, Y. Li, Y. Liu et al., "Sample Selection with Deadline Control for Efficient Federated Learning on Heterogeneous Clients," *arXiv preprint arXiv:2201.01601*, 2022.

[12] T. Huang, W. Lin, W. Wu et al., "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Transactions on Parallel and Distributed Systems(TPDS)*, vol. 32, no. 7, pp. 1552-1564, 2020.

[13] M. Cong, H. Yu, X. Weng et al., "A VCG-based Fair Incentive Mechanism for Federated Learning," *arXiv preprint arXiv:2008.06680*, 2020.

[14] Y. Deng, F. Lyu, J. Ren et al., "Fair: Quality-aware federated learning with precise user incentive and model aggregation," *Proc. of 2021 IEEE Conference on Computer Communications(INFOCOM)*, 2021.

[15] K. Donahue, J. Kleinberg, "Model-sharing Games: Analyzing Federated Learning Under Voluntary Participation," *Proc. of Thirty-Fifth Conference on Artificial Intelligence(AAAI)*, pp. 53035311, 2021.

[16] J. S. Ng, W. Y. B. Lim, Z. Xiong, X. Cao et al., "Reputation-aware hedonic coalition formation for efficient serverless hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 33, no. 11, pp. 2675-2686, 2022.

[17] M. Tang, V. W.S. Wong, "An Incentive Mechanism for Cross-Silo Federated Learning: A Public Goods Perspective," *Proc. of 2021 IEEE Conference on Computer Communications(INFOCOM)*, 2021.

[18] A. Blum, N. Haghtalab, R. L. Phillips et al., "One for One, or All for All: Equilibria and Optimality of Collaboration in Federated Learning," *Proc. of the 38th International Conference on Machine Learning(ICML)*, pp. 1005-1014, 2021.

[19] H. Huang, F. Shang, Y. Liu et al., "Behavior mimics distribution: Combining individual and group behaviors for federated learning," *arXiv preprint arXiv:2106.12300*, 2021.

[20] Y. Xue, C. Niu, Z. Zheng et al., "Toward Understanding the Influence of Individual Clients in Federated Learning," *Proc. of Thirty-Fifth Conference on Artificial Intelligence(AAAI)*, pp. 1056010567, 2021.

[21] H. Lv, Z. Zheng, T. Luo et al., "Data-Free Evaluation of User Contributions in Federated Learning," *Proc. of 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, pp. 1-8, 2021.

[22] T. Nishio, R. Shinkuma, N. B. Mandayam, "Estimation of individual device contributions for incentivizing federated learning," *Proc. of 2020 IEEE Globecom Workshops*, 2020.

[23] L. Nagalapatti, R. Narayanam, "Game of gradients: Mitigating irrelevant clients in federated learning," *Proc. of Thirty-Fifth Conference on Artificial Intelligence(AAAI),* pp. 9046-9054, 2021.

[24] H. Zeng, T. Zhou, Y. Guo et al., "FedCav: Contribution-aware Model Aggregation on Distributed Heterogeneous Data in Federated Learning," *Proc. of 50th International Conference on Parallel Processing(ICPP)*, 2021.

[25] L. Wang, S. Xu, X. Wang et al., "Addressing class imbalance in federated learning", *Proc. of the AAAI Conference on Artificial Intelligence(AAAI)*, vol. 35, no. 11, pp. 10165-10173, 2021.

[26] Y. Huang, L. Chu, Z. Zhou et al., "Personalized cross-silo federated learning on non-iid data,", *Proc. of the AAAI Conference on Artificial Intelligence(AAAI),* vol. 35, no. 9, pp. 7865-7873, 2021.

[27] J. Mills, J. Hu, G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 33, no. 3, pp. 630-641, 2022.

[28] S. Chen, B. Li, "Towards Optimal Multi-Modal Federated Learning on Non-IID Data with Hierarchical Gradient Blending," *Proc. of 2022 IEEE Conference on Computer Communications(INFOCOM)*, pp. 1469-1478, 2022.

[29] C. He, E. Ceyani, K. Balasubramanian, M. Annavaram, S. Avestimehr, "SpreadGNN: Decentralized Multi-Task Federated Learning for Graph Neural Networks on Molecular Data", *Proc. of the AAAI Conference on Artificial Intelligence(AAAI),* vol. 36, no. 6, pp. 7865-7873, 2021.

[30] A. Bogomolnaia, "The Stability of Hedonic Coalition Structures", *Games and Economic Behavior,* vol. 38, pp. 201230, 2002.

[31] I. Jang, H. Shin, A. Tsourdos, "Anonymous Hedonic Game for Task Allocation in a Large-Scale Multiple Agent System", *IEEE TRANSACTIONS ON ROBOTICS,* vol. 34, no. 6, pp. 1534-1548, 2018.

[32] T. Genin, S. Aknine, "Coalition Formation Strategies for Multiagent Hedonic Games", *Proc. of 22nd International Conference on Tools with Artificial Intelligence,* vol. 1, pp. 465-472, 2010.