Incentive-Aware Autonomous Client Participation in Federated Learning

Miao Hu, *Member, IEEE*, Di Wu, *Senior Member, IEEE*, Yipeng Zhou, *Member, IEEE*, Xu Chen, and Min Chen, *Fellow, IEEE*

Abstract—Federated learning (FL) emerges as a promising paradigm to enable a federation of clients to train a machine learning model in a privacy-preserving manner. Most existing works assumed that the central parameter server (PS) determines the participation of clients implying that clients cannot make autonomous participation decisions. The above assumption is unrealistic because the participation in FL training may incur various cost and clients also have strong desire to be rewarded for participation. To address this problem, we design a novel autonomous client participation scheme to incentivize clients. Specifically, the PS provides a certain reward shared among participating clients for each training round. Clients decide whether to participate each FL training round or not based on their own utilities (i.e., reward minus cost). The process can be modeled as a minority game (MG) with incomplete information and clients end up in the minority side win after each training round because the reward of each participating client may not cover its cost if too many clients participate and vice verse. The challenge of autonomous participation schemes lies in lowering the *volatility* of participating clients in each round due to the lack of coordination among clients. Through solid analysis, we prove that: 1) The volatility of participating clients in each round is very high under the standard MG scheme. 2) The volatility of participating clients can be reduced significantly under the stochastic MG scheme. 3) A coalition based MG is proposed, which can further reduce the volatility in each round. By conducting extensive experiments in real settings, we demonstrate that the stochastic MG-based scheme outperforms other state-of-the-art algorithms in terms of utility and volatility, and the coalition MG-based client participation scheme can further boost the utility by 39%-48% and reduce the volatility by 51%–100%. Moreover, our algorithms can achieve almost the same model accuracy as that obtained by centralized client participation algorithms.

Index Terms—Client participation, federated learning, minority game, volatility

1 Introduction

Recent decade has witnessed the rapid growth of machine learning (ML) techniques and its applications in various fields. It is estimated that the global ML market will grow at a compound annual growth rate of over 40.1% in the period of 2020–2027 and reach a valuation of over US $76.8 billion [1]. More and more diverse ML tasks will be executed on resource-constrained mobile devices with specialized hardware (e.g., GPU, DSP) [2]. However, in the meanwhile, the barriers among different data owners and the issue of privacy leakage have impeded the further development of ML and thus received arising attentions in recent years.

As a promising solution, *Federated Learning (FL)* provides an efficient way to train machine learning models among decentralized clients while not requiring them to expose their private data [3], [4], [5]. Specifically, the FL training is conducted via multiple rounds of global iterations. A number of FL clients participate each round by uploading their computation results (e.g., parameters) to the centralized parameter server (PS). The PS applies the model aggregation algorithm (e.g., FedAvg [3], FedSGD [6]) on the information such as parameters submitted from clients, and then distribute aggregated parameters back to clients participating the next round. During the entire FL training, clients never expose their private data, and thus data privacy can be preserved.

Most of existing works (e.g., [7], [8]) assumed that the central PS can completely determine the participating client set in each round without considering the willingness of individual clients. In practice, participating FL training incurs various cost such as computation and energy for clients. For resource-constrained devices, e.g., mobile phones, they may be reluctant to participate FL training. Even if there is no resource constraint, participating clients such as business companies in the cross-silo FL [9], [10] also have strong desire to be rewarded after contributing their

- *Miao Hu, Di Wu, and Xu Chen are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China, and also with the Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China. E-mail: humiao@outlook.com, {wudi27, chenxu35}@mail.sysu.edu.cn.*
- *Yipeng Zhou is with the School of Computing, Faculty of Science and Engineering, Macquarie University, Macquarie Park, NSW 2109, Australia. E-mail: yipeng.zhou@mq.edu.au.*
- *Min Chen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China. E-mail: chenxu35@mail.sysu.edu.cn.*

Manuscript received 9 Mar. 2021; revised 11 Jan. 2022; accepted 18 Jan. 2022. Date of publication 4 Feb. 2022; date of current version 20 Apr. 2022.
This work was supported in part by the National Natural Science Foundation of China under Grants 62072486, U1911201, U2001209, 61972432, and 62176101, in part by the Science and Technology Planning Project of Guangdong Province under Grant 2021A0505110008, in part by the Natural Science Foundation of Guangdong under Grant 2021A1515011369, in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X355, in part by Pearl River Talent Recruitment Program under Grant 2017GC010465, and in part by Australia Research Council under Grant DE180100950.
(Corresponding author: Di Wu.)
Recommended for acceptance by R. Tolosana.
Digital Object Identifier no. 10.1109/TPDS.2022.3148113

computation results (or computing facilities) to the FL training. Therefore, an incentive mechanism to incentivize clients in FL is vital for the success of FL training. A common approach is to provide a certain amount of reward in each round that will be shared among participating clients [11], [12], [13], [14].

In this work, we address this key issue by designing an autonomous client participation scheme for FL training with the goal to properly solicit a certain number of clients to participate FL in each round. We consider a more realistic scenario, in which clients are uncooperative and with incomplete information. Each client knows its own utility (defined as reward minus cost), but does not know the utility information of other clients. We only assume that the amount of reward budget is limited in each round and the reward is shared by participating clients. The dilemma is that, the more clients participate in an FL training round, the less reward each participating client can receive. Thus, each client is more willing to participate when there are fewer participating clients, and vice verse. The above problem is inherently a kind of game [15]. This scenario is different from previous client participation problems (e.g., [11]), which assumed that clients are rational and with complete information.

Specifically, we model the autonomous client participation problem in the FL training as a minority game (MG). In the MG model, a client is said to be *in the minority side* and win the game if the client is on the side with higher utility. For FL training, a client is in the minority side if and only if the number of participating clients is below the maximum number of beneficial clients, which implies that a client can benefit from participating in the FL training process. Each client makes autonomous decision by optimizing its future utility with the goal of being in the minority side. Through analyzing the MG played by FL clients, we can derive that: 1) The number of participating clients under the standard MG model is highly volatile, which have negative impacts on FL model accuracy and convergence. 2) A stochastic MG-based client participation scheme is proposed, in which clients winning in the last round keep their decisions unchanged, while the rest probabilistically change their decisions. We prove that this scheme can considerably reduce the volatility of participating clients. 3) We further improve the stability of participating clients by proposing a coalition MG-based client participation algorithm, in which clients form multiple coalitions and clients in the same coalition make opposite decisions so as to maximize the long-term winning probability (i.e., staying at the minority side).

Overall, our main contributions in this paper can be summarized as below:

- To the best of our knowledge, we are the first to model the autonomous client participation process in FL as a minority game by enabling each client to make independent participating decisions with incomplete information.
- We develop a family of MG-based autonomous client participation algorithms, including the standard MG-based and stochastic MG-based algorithms. We prove that stochastic MG-based algorithms can significantly reduce the volatility of the number of participating clients.
- A coalition MG-based client participation algorithm is proposed to further lower the volatility, which allows multiple clients to form a coalition to play the MG game against others. The gain of improved winning probability has been theoretically derived.
- We conduct extensive experiments under realistic settings to evaluate our algorithms. The results indicate that the stochastic MG-based algorithm and the coalition MG-based algorithm can improve the utility (defined as reward minus cost) by 39%-48% and reduce the volatility by 51%-100% compared to other baselines. Moreover, the FL model accuracy achieved by our scheme is comparable to that achieved by centralized algorithms.

The remainder of this paper is organized as follows. Section 2 summarizes relevant works in recent years, and Section 3 presents the preliminary knowledge. Section 4 introduces the system model of autonomous client participation problem in FL. The standard MG-based client participation is proposed in Section 5, while more advanced stochastic MG-based algorithm and coalition MG-based algorithm are presented in Sections 6 and 7. The experimental results are discussed in Section 8. We conclude this work and envision our future work in Section 9.

2 Related Work

In this section, we introduce recent studies on client selection and incentive mechanism in federated learning, and related work of minority game.

2.1 Client Selection in Federated Learning

Federated learning (FL) has attracted tremendous attention (e.g., [3], [16], [17]) in the past years. In the FL training process, given the large number of clients, only a partial set of clients can participate in each FL training round [16].

Past studies [7], [8], [18], [19], [20] focused more on the problem of client selection from the perspective of the parameter server to speedup the FL training process. Yang *et al.* [18] proposed a client scheduling policy by considering the staleness of received parameters and instantaneous channel qualities to improve the efficiency of FL. Wadu *et al.* [19] proposed a joint client scheduling and resource allocation policy to minimize the loss of accuracy in federated learning under imperfect channel state information. Nishio *et al.* [7] performed federated learning by actively managing clients based on their resource conditions. Xu *et al.* [8] formulated a stochastic optimization problem for joint client selection and resource allocation under long-term client energy constraints, and developed a new client selection algorithm. Amiri *et al.* [20] designed scheduling and resource allocation policies to determine the subset of devices for transmission in each training round, and resources to be allocated among participating clients. However, the above algorithms were based on the assumption that all the FL clients are willing to participate in the FL training process, and the client selection was simply made by the parameter server itself.

In practical scenarios, clients are also sensitive to the cost incurred by participating in the FL training process. Moreover, clients are also independent and free to join or leave the FL training process. It is essential to design fully autonomous client participation decision algorithms, which are more suitable for practical FL systems. One thread of research is on the incentive mechanism design for FL clients [12], [13], [14]. Sarikaya *et al.* [11] modeled the interaction between FL clients and the parameter server as a Stackelberg game. Yu *et al.* [21], [22] proposed an FL incentivizer, which dynamically divides a given budget among clients in a federation by jointly maximizing the collective utility while minimizing the inequality among the clients, in terms of the received payoff and the waiting time for receiving payoffs. However, the formulated game was generally based on the assumption that clients are rational in a static setting. Most FL clients are personal devices such as mobile phones, which are inherently dynamic and highly autonomous. Therefore, without knowing the state of each client, it is difficult to determine the set of clients that should be involved in each iteration.

Recently, Zhan *et al.* [23] designed a deep reinforcement learning-based incentive mechanism to determine the optimal incentive strategy for the parameter server and the optimal training participation strategies for clients. However, the complexity of deep neural networks will increase with the number of clients in the system. Moreover, it is also opportunistic for clients to cooperate with each other to increase their rewards. In this work, we further exploit the opportunities of both competition and cooperation among clients, and propose a more efficient client participation decision scheme for federated learning.

2.2 Minority Game

In practical FL systems, a client has limited information of other clients in terms of available choices or conditions. For scenarios with incomplete information, *minority game (MG)* shows its efficiency on clients' distributed decision-making [24], [25]. Ranadheera *et al.* [26] applied MG to solve the distributed decision-making problems in wireless networks. Recent work [27] developed an MG-based distributed server activation mechanism for computation offloading in order to guarantee energy-efficient activation of servers. Furthermore, the asymmetric processing capacity between edge servers and clouds has been investigated by a variant of MG with arbitrary cut-offs [26], [27]. Hu *et al.* [28] proposed an MG-based computation offloading scheme, in which tasks with heterogeneous workloads are divided into subtasks and instructed to form into a set of groups.

The minority game can be applied to the design and analysis of client interaction in federated learning as well, however, this issue has not been explored to the authors' best knowledge.

3 Preliminary

In FL, multiple devices can jointly learn a common model through sharing their parameters (or gradients). As illustrated in Fig. 1, we consider an FL system that contains a parameter server (PS) and a set of N clients (e.g., smartphones) with datasets $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N$, where $K_i = |\mathcal{D}_i|$ is the maximum number of sample mini-batches in client i. In the tth training round, the ith data owner (i.e., client i) can contribute local model trained on its dataset to the PS.

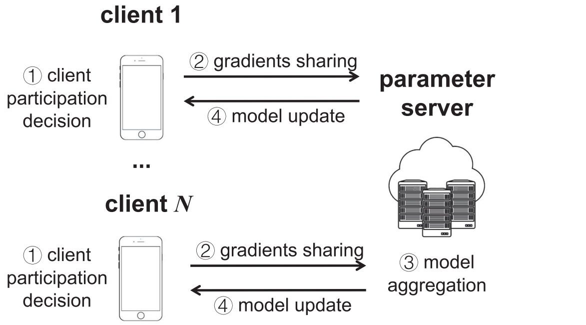


Fig. 1. The framework of federated learning with autonomous client participation.

In Fig. 1, we present an FL framework with autonomous client participation. Different from previous works that used the central PS to decide client participation, we describe the FL training process with autonomous client participation as below.

- *Step 1*: At the beginning of each training round, each client autonomously decides whether to conduct local model training on its own data or not.
- *Step 2*: If client i participates the tth training round, it will send the local computation results (e.g., gradients) to the PS.
- *Step 3*: After receiving the computation results (e.g., gradients) from all participating clients, the PS executes a model aggregation algorithm (e.g., FedAvg [3] or FedSGD [6]).
- *Step 4*: Next, the PS sends the updated global FL model to all clients.

The autonomous client participation continues until the PS claims that the FL training process ends.

Similar to [25], we can classify all clients in each round into two kinds, namely *cooperators* and *defectors*. A cooperator participates in the current FL training round. Let $x_i(t)$ denote the number of mini-batches contributed by client i at the tth training round. That is, if $x_i(t) > 0$, client i is a cooperator for the tth training round. Otherwise if $x_i(t) = 0$, client i is a defector who refuses to participate the tth training round.

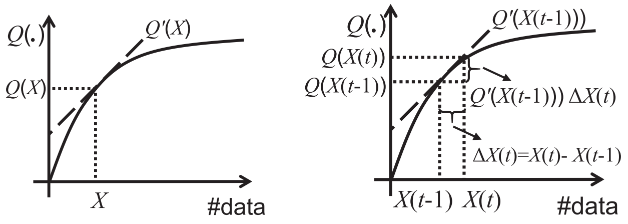
To quantify the relationship between model accuracy and data contribution from clients, we define $Q(X)$ as the model accuracy given the number of mini-batches, X. Specifically, $Q(\cdot)$ is a concave increasing function bounded by 1. The shape of the model accuracy function $Q(\cdot)$ is plotted in Fig. 2a for illustration. $Q(\cdot)$ is assumed to be concave increasing based on previous works, such as [29] (assuming that $Q(\cdot)$ is a logarithmic function) and [30] (assuming that $Q(\cdot)$ is a power law function). Our study is based on the concave increasing property of $Q(\cdot)$.

Based on the definition of $Q(\cdot)$ function, we then define the data contribution in the tth training round as below.

Definition 1 Data Contribution. *The model accuracy improvement $q(t)$ at the tth training round can be defined as:*

$$q(t) = Q(X(t)) - Q(X(t-1)), \tag{1}$$

where $X(t) = \sum_{k=1}^{t} \sum_{i=1}^{N} x_i(k)$ is the cumulative number of mini-batches contributed until round t.



(a) The general Q-function. (b) The Q-function approximation.

Fig. 2. The general Q-function and its approximation.

In previous works [3], [4], [5], [6], [7], [8], a certain number of clients are selected by the PS to participate each round of training with a pure random and uniform manner. However, it is difficult to guarantee that the volatility of the number of participating clients is low if they make independent and autonomous decisions. Basically, the volatility reflects the fluctuation of the number of participating clients. The objective of our work is to design autonomous client participation schemes that can achieve a low volatility of the number of participating clients in different training rounds.

4 System Model and Game Formulation

This section introduces the system model and analyzes the client participation decisions via a game theoretic approach.

4.1 Game Without Incentive From PS

We first consider a simple scenario in which the PS will not provide any additional reward to the FL system. FL clients are not volunteers (e.g., [21], [22], [23]), and rewards should be provided to incentivize clients to participate in FL model training. The incentive for FL clients to cooperate is to gain a common FL model that can be used by all clients. Thus, the reward of each client is related to the accuracy of the trained model.

Definition 2. FL public reward*, which is defined as the reward generated by performance improvement induced by a globally trained FL model. The public reward of each FL training round can be enjoyed by all participating clients and its value is dependent on the improvement of model accuracy. According to [22], [31], the FL public reward $R_i^{FL}(\cdot)$ allocated to client i can be quantified by a function of the total number of mini-batches as follows:*

$$R_i^{FL}(x_i(t), \mathbf{x}_{-i}(t)) = \gamma_{\text{client}}(X(t-1))q(t), \tag{2}$$

where $\mathbf{x}_{-i}(t)$ is the number of mini-batches contributed by clients except client i at the training round t, $X(t) = \sum_{k=1}^{t}\sum_{i=1}^{N} x_i(k)$, γ_{client} is the marginal per capital return (MPCR) [32] from the perspective of clients, which can be set at the client's discretion.

Unfortunately, the reward defined in Eq. (2) is hard to obtain in practical scenarios. Alternatively, we can approximate it by the following equation:

$$R_i^{FL}(x_i(t), \mathbf{x}_{-i}(t)) \approx \lambda_{\text{client}} \sum_{i=1}^{N} x_i(t), \tag{3}$$

where λ_{client} is the reward weight on the amount of data, which can be defined by $\gamma_{\text{client}}(X(t-1))Q'(X(t-1)) \triangleq \lambda_{\text{client}}$. The detailed approximation derivation of Eq. (3) can be found in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2022.3148113.

Recall that the improvement of the model accuracy function Q in round t is determined by the sample mini-batches contributed by all clients. Thus, the intuition of the FL public reward is that the reward in round t is also dependent on the collective samples contributed by all clients for round t. In addition, the reward is the same for all clients, i.e., $R_i^{FL}(x_i(t), \mathbf{x}_{-i}(t)) = R_j^{FL}(x_j(t), \mathbf{x}_{-j}(t)), \forall i, j \in \mathbb{N}$, since all clients can obtain the common trained model with identical model accuracy. The key notations used in the paper are illustrated in Table 1.

The cost of each client is defined as below.

Definition 3. Cost of data contribution *Similar to most existing studies (e.g., [22], [33]), the cost is defined as a linear function of the number of mini-batches contributed by a client. Let $c_i(x_i(t))$ denote the cost of data contribution $x_i(t)$ by client i at the training round t. If client i contributes $x_i(t)$ in the tth FL training round, it turns out that*

$$c_i(x_i(t)) = \eta x_i(t), \tag{4}$$

where η is the cost coefficient that can be obtained from field measurements.

The cost can consist of communication cost and computation cost. The communication cost is mainly determined by the model size, i.e., the number of model parameters to be communicated. Thus, the communication cost can be fixed beforehand once the learning model is specified. In contrast, computation cost is mainly affected by the number of samples used for local training. Since the computation cost to process each sample is about the same, we employ a linear function to model the computation cost (similar to [22], [33]), i.e., the computation cost is a linear function of the number of used samples. More complicated cost function will be considered in our future work.

With the above defined reward and cost, we can formulate a general game model for FL client participation with incomplete information.

Definition 4. General Game. *A general game model for FL client participation with incomplete information can be formulated as a game $\Gamma = \langle \mathbb{N}, \mathbb{S}_i, U(x_i, \mathbf{x}_{-i})\rangle$, where:*

- *$\mathbb{N}$ is a finite set of N clients (i.e., the set of players)[1], indexed by i.*
- *$\mathbb{S}_i$ is the set of strategies for client i. Each client tactically chooses a strategy from a strategy set $\mathbb{S}_i = \{s_1, \ldots, s_k, \ldots, s_{K_i}\}$, where s_k is the strategy that s_k mini-batches are contributed to one FL model training round, $s_{K_i} > \cdots > s_1 = 0$, and K_i is the maximum number of mini-batches of client i.*
- *There is a cost c_k if contribution s_k is chosen, where $c_{K_i} > \cdots > c_1 = 0$. In particular, $s_1 = c_1 = 0$ indicates that client i does not participate in the FL training process.*

1. This work exchangeably uses notations of "client" and "player".

TABLE 1
Key Notations Used in the Paper

Notation	Description
N	The total number of clients (i.e., players) in the FL system
$\mathbb{N}$	the set of clients (i.e., players) in the FL system
$\mathcal{D}_i$	client i's dataset
K_i	the maximum number of mini-batches in client i, i.e., $K_i = \lvert\mathcal{D}_i\rvert$
$q(t)$	the clients' contribution (quality) at the tth training round
$Q(\cdot)$	the function mapping from the number of data to the training quality
$x_i(t)$	the number of mini-batches contributed from client i at the FL training round t
$\mathbf{x}_{-i}$	vector of the number of mini-batches contributed from clients except for client i
λ_{client}	the marginal per capital return (MPCR) from client's perspective
λ_{server}	the MPCR from server's perspective
$R_i^{FL}(x_i, \mathbf{x}_{-i})$	the FL public reward allocated to client i
$R_i^{PS}(x_i, \mathbf{x}_{-i})$	the PS incentive reward allocated to client i
C	the total number of credits allocated by the PS as the incentive reward in each training round
$c_i(x_i(t))$	the training cost for client i at the training round t
η	the cost coefficient that can be obtained from field measurements
Γ	the formulated game $\Gamma = \langle \mathbb{N}, \mathbb{S}_i, U(x_i, \mathbf{x}_{-i})\rangle$
$\mathbb{S}_i$	the set of strategies for client i, where $\mathbb{S}_i = \{s_1, \ldots, s_k, \ldots, s_{K_i}\}$
s_k	the strategy that the s_k times of mini-batches are contributed to one FL model training round
$U_i(x_i, \mathbf{x}_{-i})$	the utility function for client i defined in Eq. (5)
$R_i(x_i, \mathbf{x}_{-i})$	the rewards allocated to client i
ψ	the defined cut-off value in Definition 7
$a_i(t)$	the decision made by client i at the FL training round t
$N_+(t)$	the number of cooperators at the FL training round t
$N_-(t)$	the number of defectors at the FL training round t
M	the number of historical records (i.e., the memory length)
$A(t)$	the attendance on the collective sum of the difference in the actions of all players at a given time t
σ^2	the volatility on the attendance value that fluctuates around the mean attendance (i.e., cut-off value)
α	the defined ratio $\alpha = 2^M/N$, referred to as the training parameter or control parameter
α^*	the minimum value of the defined ratio α
p	the probability that the client will change its decision for the stochastic MG-based decision algorithm
$\boldsymbol{\pi}(0)$	the initial state $\boldsymbol{\pi}(0)$ for the stochastic MG-based decision algorithm
$\boldsymbol{\pi}(t)$	the state updated at the FL training round t for the stochastic MG-based decision algorithm
$\mathbf{P}$	the state transition matrix for the stochastic MG-based decision algorithm
$\pi^{\mathbf{s}}$	the steady state probability vector

- *The utility function for client i can be represented as:*

$$U_i(x_i, \mathbf{x}_{-i}) = R_i(x_i, \mathbf{x}_{-i}) - c_i(x_i). \tag{5}$$

The utility depends not only on client i's strategy x_i but also on strategies chosen by opponents $\mathbf{x}_{-i}$.

In the general game formulation, we consider a scenario with incomplete information, in which a client cannot observe the utility function (or even action space) of others.

4.2 Nash Equilibrium of General Game

In the following, we analyze the equilibrium achieved under the formulated game.

Given the reward and cost, the utility of client i can be transformed into the following form:

$$U_i(x_i, \mathbf{x}_{-i}) = \lambda_{\text{client}} \sum_{i=1}^{N} x_i - \eta x_i. \tag{6}$$

Assume that the game is played under standard game-theoretic assumptions, i.e., clients are rational in maximizing utility and selfish in that utility equals its own payoff. We can prove that clients can achieve a Nash equilibrium state. Intuitively, a Nash equilibrium is a stable strategy profile: no client can gain higher utility by changing its strategy if the strategies of other clients are unchanged. Formally, a strategy profile $\mathbf{x} = \{x_1, \ldots, x_N\}$ is a Nash equilibrium if, for all clients $\forall i$, x_i is a best response to $\mathbf{x}_{-i}$. Let x_i^* denote the optimal number of mini-batches contributed by client i. The Nash equilibrium can be achieved if and only if the following equations hold, i.e.,

$$U_i(x_i^*, \mathbf{x}_{-i}) \geq U_i(x_i, \mathbf{x}_{-i}), \; \forall i. \tag{7}$$

It is because that all of the clients simultaneously play best responses to each other's strategies in a Nash equilibrium.

Specifically, we have the following two propositions on the Nash equilibrium of the formulated game.

Proposition 1. *When $\lambda_{\text{client}} \geq \eta$, the Nash equilibrium can be achieved when each client contributes all its data to FL training. That is, the strategy profile $\mathbf{x}^* = \{K_1, \ldots, K_N\}$ is a Nash equilibrium.*

Proof. The proof can be found in Appendix B, available in the online supplemental material. □

Proposition 2. *When $\lambda_{\text{client}} < \eta$, the Nash equilibrium can be achieved when each client contributes nothing to FL training. That is, the strategy profile $\mathbf{x}^* = \{0, \ldots, 0\}$ is a Nash equilibrium.*

Proof. The proof can be found in Appendix C, available in the online supplemental material. □

Remark. From Propositions 1 and 2, we can see that there are two extreme cases. If the cost is small enough, i.e., $\lambda_{\text{client}} \geq \eta$, all clients autonomously participate the game, and the FL training can be conducted successfully. However, if $\lambda_{\text{client}} < \eta$, every client would like to be a free rider since contributing nothing is a dominant strategy, which brings the highest utility to the client regardless of others' actions. This implies that simple FL public reward is insufficient to guarantee sustainable federated learning. □

In the following discussion, we focus on the cases when $\lambda_{\text{client}} < \eta$. To solve the free-riding problem, we propose to introduce additional reward from the PS to recruit clients, which can be formulated as an MG.

4.3 Reward From PS

Definition 5. PS incentive reward, *which is defined as the reward allocated by the PS to each client based on its contribution. Similar to [22], [34], the PS incentive reward function can be defined as follows:*

$$R_i^{PS}(x_i(t), \mathbf{x}_{-i}(t)) = \frac{Cx_i(t)}{\sum_{i=1}^{N} x_i(t)}, \tag{8}$$

where C is the total amount of credits allocated by the PS as the reward in each training round. In practice, the credit value can be determined by jointly considering the training cost and the number of required participating clients per training round.

With the additional PS incentive reward, the total reward of each client should be updated by combining Eqs. (3) and (8), which gives:

$$R_i(x_i(t), \mathbf{x}_{-i}(t)) = \lambda_{\text{client}} \sum_{i=1}^{N} x_i(t) + \frac{Cx_i(t)}{\sum_{i=1}^{N} x_i(t)}. \tag{9}$$

As the participation recruitment budget C is limited, only a fraction of clients have incentives to participate in the FL training process when the received reward is higher than the incurred cost.

Discussion. A question arising here is that what is the motivation for the PS to contribute additional reward. If the role of the PS is simply to coordinate clients, the PS may not have the incentive to contribute reward. However, the PS may also benefit significantly from FL. For example, the PS of an online video system may target to train a recommender system via FL [35]. The recommender can definitely improve the service quality of video services and thus be desired by the PS as well.

Formally, the condition for the PS to trigger FL training under $\lambda_{\text{client}} < \eta$ can be analyzed as follows.

The utility function of the PS has two components:

- *Reward*. Similar to the definition of rewards for clients, the reward of the PS is $R_{\text{server}}(X^*(t)) = \lambda_{\text{server}} X^*(t)$. Here $X^*(t)$ denotes the optimal amount of mini-batches contributed to the FL model training until round t, and λ_{server} is the rewarding weight on the data contribution which can be set at the server's discretion.
- *Cost*. The cost in each FL training round is a fixed reward budget C.

The utility of the PS is $U_{\text{server}}(X^*(t)) = \lambda_{\text{server}} X^*(t) - C$. Following Eq. (9), we have the equation $\lambda_{\text{client}} X^*(t) + C - \eta X^*(t) = 0$, and thus $X^*(t) = \frac{C}{\eta - \lambda_{\text{client}}}$. By substituting it into the utility of the PS, we have

$$U_{\text{server}}(X^*(t)) = \frac{\lambda_{\text{server}} + \lambda_{\text{client}} - \eta}{\eta - \lambda_{\text{client}}} C. \tag{10}$$

Thus, the PS has the incentive to contribute reward as long as $\lambda_{\text{server}} + \lambda_{\text{client}} > \eta$. Note that $\lambda_{\text{server}} \geq 0$. If $\lambda_{\text{server}} = 0$, it implies that the PS has no incentive to provide additional reward to FL.

In the rest, we focus on the case that $\lambda_{\text{server}} + \lambda_{\text{client}} > \eta > \lambda_{\text{client}}$. Clients are solicited by the PS to participate FL training. They need to make independent and autonomous participation decisions in each round, and this problem can be formulated as an MG.

5 Standard MG-Based Client Participation Decision Algorithm

In this section, we first formulate the minority game, and clearly define the *state, action, transition, reward*, and *strategies* in the game. Then, we propose a standard MG-based client participation decision algorithm.

5.1 Minority Game Formulation

Minority game (MG) [25] is a powerful theory tool in modeling collective behaviors of clients when they have to compete for limited resources with incomplete information. MG can also be applied in the context of federated learning, considering the decentralized nature of FL.

In classic form of MG (e.g., the El Farol Bar problem [36]), players make their decisions on whether to attend a bar or not each night. Going to a bar is a enjoyable choice for a player only if the bar is not too crowded, otherwise, the player chooses to stay at home. Obviously, players should adjust their decision based on their expectations on what others choose, and these expectations are generated by the decisions of other players. In each round of MG, each player determines its action based on the historical and preference factors. After all players make their decisions, the action associated with less players is declared as the *minority side strategy*, and the players in the minority side win certain payoffs. The results are also broadcast to all players so that each player can update their information and adjust its action accordingly in the next round.

Next, we re-formulate the problem of FL client participation decision with minority game as follows:

Definition 6. Minority Game Model. *The properties of the minority game are given as below:*

- *There are N clients in the FL system.*
- *At each training round t, each client i makes a decision $a_i(t) \in \{+1, -1\}$. The action $a_i(t) = +1$ indicates that a client selects the "cooperated strategy", and $a_i(t) = -1$ indicates that a client selects the "defected strategy".*
- *The clients who are in the minority side, i.e., $a_i(t) = -\text{sign}(\sum_{i=1}^{N} a_i(t))$, win, and the others lose.*
- *No communications between players are allowed.*

In each FL training round, a client chooses either "*cooperated strategy*" or "*defected strategy*", and clients can be divided into two categories:

- *Minority*. A client is said to be in the minority side and win the game if the client is on the side with higher and non-negative utility.
- *Majority*. Conversely, clients in the majority side lose the game if the client is on the side with lower and negative utility.

There exists a threshold number of participating clients, below which participants can be in the minority side. On the contrary, if the number of participants is larger than the threshold, the participants are in the majority side. Thus, we define the threshold number of participating clients as the *cut-off value*, which can be formally defined as follows:

Definition 7. Cut-off value ψ *is defined such that, when the number of cooperated clients approaches ψ, the FL training process reaches the performance threshold. Thus, for a client to benefit from training, the number of FL training clients should not exceed the limit of ψ. As a result, being in the population minority (defined by ψ) is always desired.*

Let $N_+(t)$ denote the number of cooperators at the training round t, and $N_-(t)$ denote the number of defectors at the training round t. In total, we have $N = N_+(t) + N_-(t)$. The minority side can be formally defined as:

- When $N_+(t) \le \psi$, for cooperated clients, the allocated reward is greater than the training cost. That is, the clients choosing cooperated strategy belong to the minority side and win the game.
- Conversely, when $N_+(t) > \psi$, for cooperated clients, the training cost is greater than the allocated reward. That is, the clients choosing defected strategy belong to the minority side and win the game.

Each player has a given set of decision making strategies that help them select future actions. The action of player i at time t is shown by $a_i(t)$. The target of each client is to make the decision $a_i(t)$ to choose the winning side as possible. In other words, the client participation decision strategy needs to predict the winning action of the following training round(s), and choose the action $a_i(t)$ to make itself in the winning side. At the end of each round, players are informed of the winning action, which is then used as history data by players to improve their decision making in the next rounds.

5.2 Algorithm Design

5.2.1 State

In the context of FL, a client knows very limited information of others due to the privacy requirement. The winning side information will be broadcast by the PS to each client at the end of each training round. The winning side information is defined as "1" if the "cooperated strategy" is the minority side, and "-1" otherwise. It is also the only available information that each client can utilize.

In our model, the *state* of a client is defined based on its decision history and winning side information. Formally, we define the state as the past M winning choices from the client's perspective, with M being the number of historical records. In other words, a strategy is essentially a mapping of a history with M records to an action. In our model, there are two actions to select from ("*cooperated*" or "*defected*"), and the complete strategy space containing 2^{2^M} strategies is very huge with correlated and redundant strategies.

TABLE 2
An Example Strategy Table for a Client

The historical winning decisions			The decision at round t
-1	-1	-1	1
-1	1	-1	1
1	-1	-1	1
1	1	-1	1

Players using correlated strategies probably obtain identical decisions, and thus lower their chance to choose the minority side. Inspired by [37], [38], we shrink the strategy space with the following principle. In the original strategy space with size 2^{2^M}, there exists a subset of 2^M pairs of points and each pair of strategies is with the maximum irrelevant degree. They are called anti-correlated pairs in the sense that the two strategies of a pair always predict opposite actions. The reduced strategy space selects strategies according to the rule as below: one strategy is anti-correlated to the other one in the same pair. The reduced strategy space size is 2^{M+1}, which is much smaller than the size of the original strategy space 2^{2^M}. An example with two anti-correlated strategies is shown in Table 2, where a client has two strategies regardless of the historical winning information.

5.2.2 Action

Let $a_i(t)$ denote the action taken by a client in the tth FL training round with $a_i(t) \in \mathcal{A} = \{-1, +1\}$. For each client, the action is defined as the selected strategy, either "cooperated" or "defected". That is, $a_i(t) = +1$ indicates that a client selects the "cooperated strategy", and $a_i(t) = -1$ indicates that a client selects the "defected strategy".

In the minority game model, an important measure is the difference in the attendance of the two sides. The attendance is defined as below:

Definition 8. Attendance, *which is the collective sum of the difference in the actions of all players at a given time t. The attendance is formally defined as:*

$$A(t) = N_+(t) - N_-(t) = 2N_+(t) - N. \tag{11}$$

Equivalently, the attendance value can be obtained by aggregation operation, i.e.,

$$A(t) = \sum_{i=1}^{N} a_i(t). \tag{12}$$

The central quantity of interest is the difference in the crowd of the two sides.

5.2.3 Transition

In each training round, a client chooses an action a, and transits to a new state s' from state s given the transition probability $\Pr\{s'|s,a\} = 1$ for the real state s' if a is the winning action, and $\Pr\{s'|s,a\} = 0$ otherwise. An example is

TABLE 3
An Example State Transition for a Client

The state at FL training round t			The winning decision	The state at FL training round $t+1$		
1	-1	1	1	-1	1	1
1	-1	1	-1	-1	1	-1

shown in Table 3, where the winning information of the past three training rounds are taken as the states. Note that the current state is $s = \{1, -1, 1\}$. The next state transits to $s' = \{-1, 1, 1\}$ if the winning decision is "cooperated". Otherwise, the next state transits to $s' = \{-1, 1, -1\}$ if the winning decision is "defected".

5.2.4 Reward

The reward of a client can be represented by a utility function, which characterizes the relative utility of a particular action. Let $U_{i,t}(s, a)$ denote the utility function of a corresponding action a in the tth training round. For a client in state s, after selecting a given action a, the corresponding utility function is updated according to the following rule:

$$U_{i,t}(s, a) = U_{i,t-1}(s, a) - a_i(t)\frac{A(t)}{N}, \tag{13}$$

where s is the current state of a client, i.e., the last M winning actions, and a denotes the action that a client may take. Eq. (13) indicates that the utility contains information about the accumulative experience from the history. The state-action strategy is rewarded (i.e., $U_{i,t}(s, a) - U_{i,t-1}(s, a) > 0$) when it correctly predicts the minority sign, that is, if $a_i(t) = -\text{sign } A(t)$, and penalized otherwise.

The states are updated during the course of the clients' interaction with the environment. Actions that lead to a higher reward are preferred.

5.2.5 Decision Strategy

Generally, a client will choose action a that can maximize the utility $U_{i,t}(s, a)$ based on the current state s, i.e.,

$$a_i^*(t) = \arg\max_a \quad U_{i,t}(s, a). \tag{14}$$

However, this might lead to the local optimality. Thus, we use both exploration and exploitation strategies in decision making. In our framework, the clients select the action by resorting to a probabilistic choice model, i.e.,

$$\Pr\{a_i(t) = a\} = \frac{e^{\beta U_{i,t}(s,a)}}{\sum_{a' \in \mathcal{A}} e^{\beta U_{i,t}(s,a')}}, \tag{15}$$

where β is the relative weight assigned by clients to the empirical evidence accumulated in U_t with respect to random idiosyncratic shocks. If $\beta \to \infty$, clients always play their best strategy according to the scores. When β decreases, clients take more opportunities to explore the unknown strategy. In Eq. (15), the probabilistic action decision making is adopted to avoid the trap of local optimality. This is a well-known choice model among economists, called the *Logit model* [39].

By utilizing the Logit model, for strategy exploitation, a client selects the action following Eq. (14). Meanwhile, it also explores other strategies by randomly selecting actions $a^*(t)$ with a probability following Eq. (15).

The standard MG-based client participation decision is illustrated in Algorithm 1, which contains four steps:

- *Decision initialization*. Before conducting the MG-based decision, the initial decision is randomly selected by each client. To improve the convergence rate to the optimality, each client can use a probability (i.e., ψ/N) to guide the possibility of joining the current round training.
- *Transaction*. After collecting the updated gradients, the parameter server will aggregate the results using method such as FedAvg [3]. Besides, the cooperators will be paid with the incentive regulations.
- *Utility update*. The cooperators receive rewards from the parameter server. Based on the historical reward information, each client will update the utility function following Eq. (13).
- *Decision iteration*. Based on the utility function, each client makes the decision for the training round $t+1$ following the principle defined in Eqs. (14) and (15). The client participation decision process continues until the training model converges.

Algorithm 1. Standard MG-Based Client Participation Decision

```
Given: N, a_i(t) ∈ {+1, −1}, C
% Step 1: decision initialization
for i = 1 : N do
    Randomly initialize decision for client i;
end for
repeat
    % Step 2: transaction
    The clients conduct local model update;
    PS calculates the attendance A following Eq. (11);
    PS sends rewards C/N_+(t) to each cooperator;
    % Step 3: utility update
    % minority side determination
    if A(t) > 0 then
        win(t) = −1
    else
    win(t) = +1
    end if
    PS broadcasts the winner information win(t);
    for i = 1 : N do
        % Update the utility function following Eq. (13);
        if a_i(t) ≠ win(t) then
            if (rand()%N) < abs(A) − 1 then
                a_i(t + 1) = (a_i(t) + 1)%2
            end if
        end if
    end for
    t = t + 1
    % Step 4: decision iteration
    Clients act following the probability in Eq. (15);
until PS claims that the training process ends.
```

5.2.6 Scalability

It is worth emphasizing that the communication overhead of our work is very small. In our formulated minority game

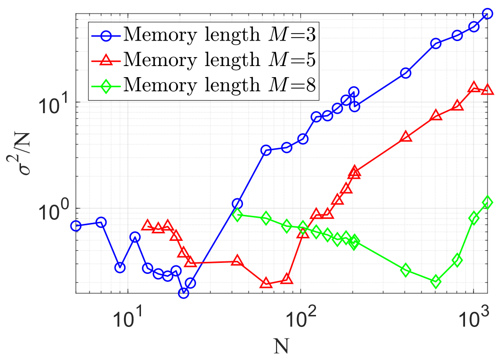


Fig. 3. The volatility per client versus the number of clients in the FL system.

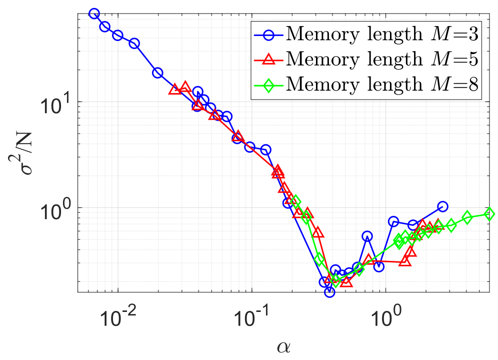


Fig. 4. The volatility per client versus the training parameter α in the FL system.

model, it is only necessary to use M-bit size memory to record the historical winning side information. It is independent of the network size. Thus, the algorithm complexity remains unchanged as the network size grows. Therefore, our proposed model can guarantee a good system scalability even for very large networks with a large population of clients.

5.3 Observations

In the following, we discuss the properties of the standard MG-based client participation decision policy, including volatility and phase transition.

5.3.1 Volatility

Basically, the attendance value never settles but fluctuates around the mean attendance (i.e., cut-off value). The fluctuation around the mean attendance is known as *volatility*, denoted by σ^2. We focus on the cooperative properties of the system in the stationary state. Symmetry arguments suggest that none of the two strategies ("cooperated" or "defected") will be systematically the minority one. This means that $A(t)$ will fluctuate around $E_t\{A\}$. The fluctuation is a measure of the quality of cooperation in the game.

The size of fluctuations of $A(t)$ indicates a remarkable non-trivial behaviour. The volatility is defined as [25]:

$$\sigma^2 = \mathrm{E}_t\left(\sum_{i=1}^{N} a_i(t)\right)^2 = \mathrm{E}_t\{A^2\} - (\mathrm{E}_t\{A\})^2. \tag{16}$$

Volatility is an inverse measure of the system performance. When the fluctuation is smaller, it implies that the size of the minority (i.e., the number of winners) is larger. Hence, a smaller volatility corresponds to a higher clients' satisfaction level along with better resource utilization.

Fig. 3 illustrates the volatility per client versus the number of clients in the federated learning system. Let M denote the number of historical records, also named as the *memory length*. From these curves, we have three key observations. *First*, regardless of the value of M, the volatility first ascends and then descends with the increase of the number of clients N. This might be due to that, when the number of clients is small compared to the number of possible historical records, the outcome is seemingly random. When the historical information is too complex (e.g., M is too large), the learning function over-fits the fluctuations and thus the volatility is relatively high. *Second*, with the increase of M, we find a graphic right shifting of the curves. The reason is that, it needs more clients to process the more complex historical information with the increase of memory length. *Third*, the number of clients that can achieve the minimum value of volatility is linear with the exponential function of memory length, i.e., $N^* \sim e^M$.

5.3.2 Phase Transition

From the above observations and the minority game theory, volatility depends on the ratio $\alpha = 2^M/N$ as the training parameter or control parameter [25]. Based on these results, Fig. 4 illustrates the volatility per client versus the training parameter α. According to the variation of the global efficiency, the game can be divided into two phases by the minimum value of α (denoted by α^*).

- When $\alpha < \alpha^*$, for a small M, the number of strategies is smaller than the number of clients N. Thus many clients could use the same strategy, leading them to make the same decision. This creates a herding effect.
- Once $\alpha > \alpha^*$, the value of M is large enough to make the strategy space larger than the number of clients N, so that the probability of any two clients using identical strategies diminishes. Note that α^* corresponds to the minimum volatility, indicating the system's ability to a state where the resource utilization can be maximized.

6 Stochastic MG-Based Client Participation Decision Algorithm

In the previous section, we point out the volatility problem existing in the standard MG-based client participation decision algorithm. To tackle the volatility problem, we extend the standard minority game by adding some randomness: winners of the last game stick to their choice, and losers individually change their decision with probability p [40]. The process is as follows:

- If a client i is successful in a given round, it will make the same decision in the next round: $a_i(t+1) = a_i(t)$.
- Otherwise, the client will change its action with a probability $p = \Pr\{a_i(t+1) = -a_i(t)\}$.

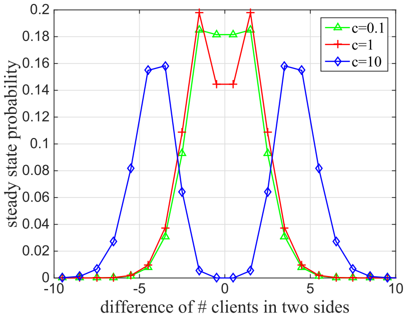


Fig. 5. The steady state probability versus the difference of the number of clients in two sides.

This stochastic principle is a reasonable way of behaving in the absence of complete information. It is related to the Johnson's model [40], but different in decisive details. Following the stochastic minority game principle, there are $(N+|A|)/2$ losers and $|A|$ is at most in the order of N. The average number of changes is $p(N+|A|)/2 \sim pN$.

This is evidently a stochastic one-step process and can be handled well by existing tools for Markov processes.

Instead of using the whole set $\{a_i(t)\}_{i=1}^N$ of time dependent random variables we consider the stochastic process:

$$K(t) = \frac{1}{2}\sum_{i=1}^{N} a_i(t). \tag{17}$$

The possible values k that $K(t)$ can take are from $-N/2$ to $N/2$. Then, the probabilities

$$\pi_k(t) = \Pr\{K(t) = k\} \tag{18}$$

together with the transition probabilities

$$p_{lk} = \Pr\{K(t+1) = k | K(t) = l\} \tag{19}$$

are the basic quantities to describe the system. To shorten the notation, we consider the probabilities $\pi_k(t)$ as components of the state vector $\pi(t) = \{\pi_{-N/2}, \ldots, \pi_{N/2}\}$. The number of players in the majority in the tth round is $N = 2 + |K(t)|$. Since players perform independent Bernoulli trials, the transition probability is given by the binomial distribution

$$p_{lk} = \begin{cases} \binom{N/2+l}{l-k} p^{l-k}(1-p)^{N/2+k}, & \text{for } l > 0, \\ \binom{N/2-l}{k-l} p^{k-l}(1-p)^{N/2-k}, & \text{for } l < 0. \end{cases} \tag{20}$$

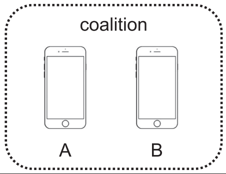
This stochastic process may be considered as a random walk in one dimension. Given the initial state $\boldsymbol{\pi}(0)$, the state $\boldsymbol{\pi}(t)$ is updated at each training round by multiplying it by the transition matrix $\mathbf{P}$:

$$\boldsymbol{\pi}(t+1) = \mathbf{P}\pi(t). \tag{21}$$

As $t \to \infty$, we have the steady state probability as:

$$\pi^s = \mathbf{P}\pi^s. \tag{22}$$

We consider $p = c/(N/2)$, where c is constant and much smaller than N. As N increases, the number of clients that change sides every training round stays constant c, i.e., c clients will change their decision. Given the definition of c, we can approximate the steady state probability as below.



decision (A)	decision (B)	decison (C)	**winner**
cooperated	defected	cooperated	**B**
cooperated	defected	defected	**A**
defected	cooperated	cooperated	**A**
defected	cooperated	defected	**B**

Fig. 6. An example on coalition formation with three clients (A, B and C). When the clients (A and B) in the coalition make opposite decisions, the client C can never be the winner in the minority.

Theorem 3. *In the stochastic MG-based algorithm, given the value of c, the steady state probability π_k^s can be expressed by the incomplete gamma function:*

$$\pi_k^s = \frac{\gamma(|k| + 1/2, c)}{2c(|k| - 1/2)}, \tag{23}$$

where $\gamma(s,x) = \int_0^x t^{s-1}e^{-t}\mathrm{d}t$.

Proof. The proof can be found in Appendix D, available in the online supplemental material. □

Fig. 5 illustrates the steady state probability versus the difference of the number of clients in two sides. The curve is roughly Gaussian distributed for small c, and forms two peak values for larger values of c. For a smaller c, the stochastic MG principle achieves a smaller volatility compared to the cases with a larger c.

7 Coalition MG-Based Client Participation Decision Algorithm

Recall that there is no information sharing in both standard MG-based decision and stochastic MG-based decision algorithms. We then propose a coalition MG-based client participation decision algorithm by taking advantage of cooperation among some clients.

As shown in Fig. 6, suppose that three clients (A, B and C) serve for the model update in the FL system. Two clients (A and B) make decision in a coalition way, and the other one (C) is free in decision making. When the clients in the coalition make the opposite decisions, e.g., A is cooperated, and B is defected or A is defected and B is cooperated. In this situation, the client C will surely be in the majority no matter what decision it made, either cooperated or defected.

Motivated by this example, we propose a coalition MG-based client participation decision algorithm. In general, the client participation decision should be modeled as an non-transferable utility (NTU) game in a partition form, as the value of a coalition $\mathcal{S}$ will have a strong dependence on how the players in $\mathcal{N}\backslash\mathcal{S}$ are structured [41]. The coalitional game in a partition form is inherently complex to solve.

Fortunately, there exists a simple and efficient coalition formation solution for our problem. In our solution, clients can form coalition pairs with two members in each coalition. Two members in the same coalition need to always make opposite decisions. With this design principle, the NTU game can be degenerated into a transferable utility (TU) game in a characteristic form.

To analyze the stability of the coalitional game, we introduce the concept of coalitional stability in our paper, captured by the notion of the core, i.e., the set of outcomes such that no subgroup of players has the incentive to deviate. We define the mathematical property of super-additivity as follows [41].

Definition 9. *A TU game $(\mathcal{N}, v)$ is said to be super-additive if and only if*

$$v(\mathcal{S}_1 \cup \mathcal{S}_2) \geq v(\mathcal{S}_1) + v(\mathcal{S}_2), \forall \mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{N}, \text{ s.t. } \mathcal{S}_1 \cap \mathcal{S}_2 = \varnothing.$$

A game is super-additive if cooperation, i.e., the formation of a larger coalition out of disjoint coalitions, guarantees at least the value that is obtained by disjoint coalitions together. If the payoff of a game is super-additive, cooperation is always beneficial to all players. Players have the incentive to form the grand coalition $\mathcal{N}$ (i.e., the coalition of all players) since the payoff received from $v(\mathcal{N})$ is no less than the payoff received by the players in any disjoint set of coalitions they could form.

Suppose that we have the grand coalition $\mathcal{N}$, we make the following definitions based on the grand coalition $\mathcal{N}$ before we introduce the concept of the core.

Definition 10. *A payoff vector $\mathbf{r} \in \mathbb{R}^{\mathcal{N}}$ for dividing $v(\mathcal{N})$ is group rational if $\sum_{i \in \mathcal{N}} r_i = v(\mathcal{N})$.*

Definition 11. *A payoff vector $\mathbf{r} \in \mathbb{R}^{\mathcal{N}}$ is individually rational if each player can obtain a benefit no less than that by acting alone, i.e., $r_i \geq v(\{i\}), \forall i \in \mathcal{N}$.*

Given that a payoff vector is both group rational and individually rational, we can define the core as follows.

Definition 12. *Given a TU coalitional game, the core is defined in which no coalition $\mathcal{S} \subset \mathcal{N}$ has an incentive to reject the proposed payoff allocation, deviate from the grand coalition, and form a new coalition $\mathcal{S}$ instead. Mathematically, the core is given by:*

$$\mathcal{C} = \left\{ \mathbf{r} : \sum_{i \in \mathcal{N}} r_i, \text{ and } \sum_{i \in \mathcal{S}} r_i \geq v(\mathcal{S}), \forall \mathcal{S} \subseteq \mathcal{N} \right\} \tag{24}$$

where r_i is the reward allocated to player i.

The core guarantees that no group of players has an incentive to leave the grand core in order to form another coalition.

Coalition formation entails finding a coalitional structure that maximizes the total utility. A centralized coalition formation mechanism is usually much simple for execution. However, the number of partitions of a set $\mathcal{N}$ grows exponentially with the number of players in N and is given by a value known as the Bell number [42], [43]. Hence, finding an optimal partition by using a centralized approach is, in general, computationally complex and impractical.

For our problem, we judiciously explore the properties of the game to find a simple and efficient coalition formation solution. The principle is that clients can form coalition pairs with two members in each coalition. Two clients in the same coalition always make opposite decisions. With this decision principle, we have

$$v(\mathcal{S}_1 \cup \mathcal{S}_2) = v(\mathcal{S}_1) + v(\mathcal{S}_2), |\mathcal{S}_1| = |\mathcal{S}_2| = 2, \mathcal{S}_1 \cap \mathcal{S}_2 = \varnothing.$$

This implies that our coalitional game is stable since the formation of a large coalition out of disjoint coalitions equals to the value that is obtained by the disjoint coalition pairs together. More specifically, the expected reward of each member in the core equals to that of an independent client in any coalition pair.

According to the above principle, our algorithm composes of two key steps, i.e., the *coalition formation* step and the *decision-making* step, which will be described as follows.

Step 1: in the *coalition formation* step, each client pairs with another one in the FL system. The motivation for a client to join a coalition is that the potential winning probability will increase. Let $\mathcal{S}$ denote the set containing the clients that have been paired with each other, e.g., $\mathcal{S} = \{1, 2, \ldots, i\}$. When formulating coalitional pairs, we also take the communication efficiency into account. Let d_{ij} denote the distance measure between clients i and j. As the communication efficiency deteriorates with the distance between clients, the objective of client pairing is set as:

$$\min_{\mathcal{S}} \sum_{\{i,j\} \in \mathcal{S}} d_{i,j}. \tag{25}$$

Specifically, given N clients in the system, the number of pairs is $(N-1)/2$ when N is an odd number.

To minimize the objective in Eq. (25), we use the dynamic programming method to iteratively fill the pairing set $\mathcal{S}$ from the complete client set $\mathcal{N}$. Let $d(\mathcal{S})$ denote the sum of distances between clients as pairs in the set $\mathcal{S}$, and we have the recursive equation as:

$$d(\mathcal{S}) = \min\{|d_{i,j}| + d(\mathcal{S}/\{i\}/\{j\} | i, j \in \mathcal{S}\}. \tag{26}$$

The detailed coalition formation algorithm is illustrated in Algorithm 2. To guarantee the performance of the coalition formation, Algorithm 2 is performed on the parameter server. This is efficient for its convenience and flexibility for practical deployment.

Algorithm 2. Client Coalition Formation Algorithm

1: **Given**: $N, d_{i,j}, i, j \in \{1, \ldots, N\}, \mathcal{S} = \varnothing, \mathcal{R} = \mathcal{N} = \{1, \ldots, N\}$
2: **repeat**
3: Select the pairs i, j that minimize $d_{i,j}$;
4: $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\} \cup \{j\}$;
5: $\mathcal{R} \leftarrow \mathcal{R}/\{i\}/\{j\}$;
6: Find the clients to be added into $\mathcal{S}$ following Eq. (26);
7: **until** $\mathcal{R} = \varnothing$ or $|\mathcal{R}| = 1$.

Step 2: when we have two clients in each coalition, one client randomly selects the decision and the other one takes the opposite decision. When the total number of clients in the federation is even, then there must be one and only one winner in each coalition at each training round.

TABLE 4
Experimental Parameter Settings

Symbol	Definition	Default Value
N	the number of players/clients	1001
x	# of mini-batches per client	10
C	the recruitment budget per round	500
c	the parameter for stochastic MG	1,10
ψ	the cutoff value	50
M	the historical length	5
S	the size of strategy space	2
β	the weight for standard MG	1
η	the coefficient for cost function	1
T	the total number of training rounds	100

We conduct theoretical analysis on the advantages for joining a coalition in terms of winning probability (i.e., staying in the minority side). For completeness, we discuss two cases: (1) suppose that there are N_c clients joining coalitions, where $N_c/2$ clients choose cooperated strategies and others choose defected strategies; (2) the residual $N - N_c$ clients who are not in any coalition will make the random decision. Then we have the following proposition.

Proposition 4. *Let p_c and p_{nc} denote the winning possibility for clients in the coalitions and not in any coalition, respectively. Then we have the improvement on winning probability for joining a coalition as:*

$$\frac{1}{2(N - N_c)} \leq p_c - p_{nc} \leq 12. \tag{27}$$

Proof. The proof can be found in Appendix E, available in the online supplemental material. □

For the example in Fig. 6, when the three clients make random choice, the client not in a coalition can never be a minority (e.g., C). When they join a coalition (e.g., A or B), the winning probability increases to $1/2$.

It is worth discussing other related approaches for the formulation of coalitions. The recursive core method is applicable for the analysis of the dynamics of coalition formation while taking externalities into account [44], [45], [46]. As externalities are not considered in our work, the recursive core is not applicable for our analysis. For forming coalitions, some works generated client preference list over all possible coalitions and then ordered them based on preferences such as [47]. However, decentralization introduces significant communication overheads, which may increase the complexity and fail to achieve the optimality. Considering this, some works proposed to design centralized coalition formation mechanisms [42], [48], which are simpler and more efficient for execution.

8 PERFORMANCE EVALUATION

In this section, we conduct extensive experiments to evaluate the performance of our proposed MG-based client participation decision algorithms for federated learning.

8.1 Experimental Settings

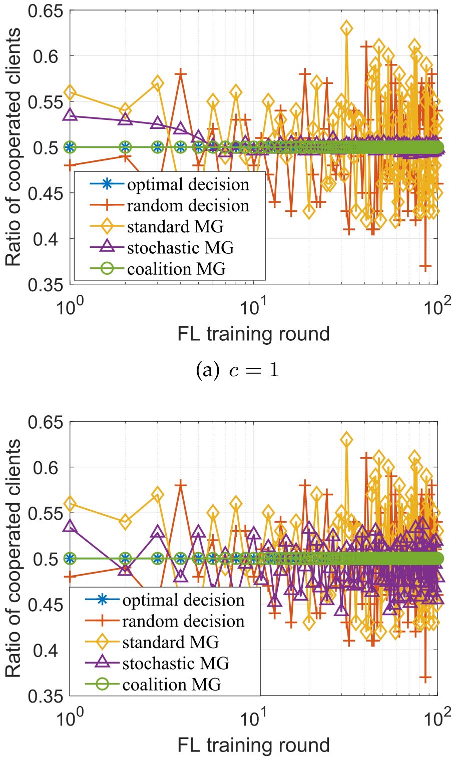
In our experiments, we use two datasets for evaluation, including the MNIST digit recognition dataset [49] and the CIFAR-10 image classification dataset [50]. We simulate a federated learning system that contains one parameter server and N clients. The parameter server is responsible for aggregating updated parameters from cooperated clients and broadcast the winning information to all clients. In default, there are 1001 clients in the system and each client conducts the model training independently. Based on the updated model parameters and winning information, each client iteratively decides whether to participate in the current training round or not. That is, each client needs to choose one strategy from "cooperated" or "defected" in each model training round. Table 4 lists the key parameters with corresponding values used in our experiments.

To evaluate the performance of our client participation decision algorithms, we use random decision algorithm and optimal decision algorithm as the baselines. All the compared algorithms are listed as below:

- *Standard MG* (Standard MG-based client participation decision algorithm) proposed in Section 5.
- *Stochastic MG* (Stochastic MG-based client participation decision algorithm) proposed in Section 6.
- *Coalition MG* (Coalition MG-based client participation decision algorithm) proposed in Section 7.
- *Random Decision*, in which each client randomly makes the cooperated or defected participation decision with equal probability [20].
- *Optimal Decision (FedAvg)*, in which clients' participation decisions are made by the coordination server optimally such as FedAvg [3].

The evaluation metrics include the attendance, volatility, utility, and model accuracy. In detail, we have

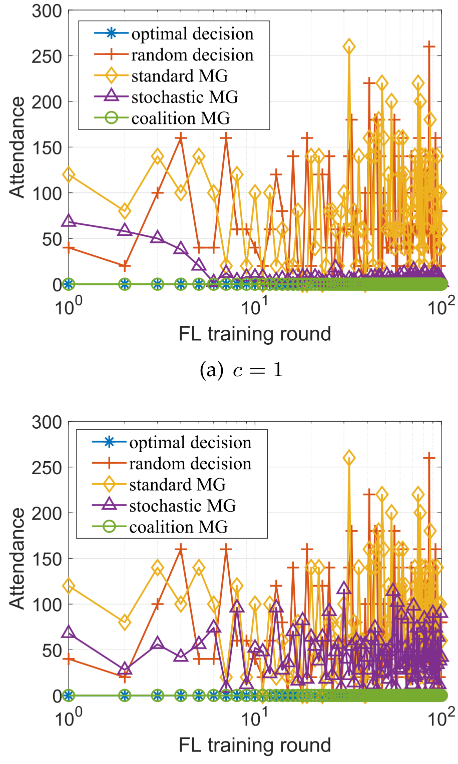
- *Attendance*, which is the collective sum of the difference in the crowd of the two sides, the cooperated side and the defected side. A larger attendance means a bigger difference between the crowd of the two sides, and vice versa. The formal definition can be referred in Eq. (11).
- *Volatility*, which describes the fluctuation around the mean attendance (i.e., cut-off value). The volatility metric can be regarded as a measure of the quality of cooperation in the game. A smaller value of the volatility means a high quality of cooperation in the game, i.e., a better performance can be achieved. On the contrary, a larger value of the volatility indicates that players tend to be disordered in the game. The volatility can be calculated according to Eq. (16).
- *Utility*, which can be calculated as the received reward from the PS minus the training cost for each client. In the formulated game, each client aims to maximize its own utility. The calculation formula for the client utility can be obtained with Eq. (5).
- *Model accuracy*, which is verified on two datasets. The MNIST dataset contains 60,000 grayscale images of handwritten digits from 1 to 10 [49]. The CIFAR-



(a) $c = 1$

(b) $c = 10$

Fig. 7. The ratio of cooperated clients versus the FL training round.



(a) $c = 1$

(b) $c = 10$

Fig. 8. The attendance versus the FL training round.

10 dataset consists of 60,000 colour images in 10 classes, with 6,000 images per class [50]. Both datasets are divided into two sets: training set and test set. The training set includes 50,000 randomly selected images and the rest 10,000 images are used for testing.

We compare how these metrics are affected by the system parameters under different algorithms.

8.2 Performance Comparison

8.2.1 Attendance

Fig. 7 presents the ratio of cooperated clients over the total number of clients versus the FL training round. Given that the budget from the PS can at most incentivize $\psi = 50$ clients (50% cooperators in the FL system). From the results, the *coalition MG* scheme can achieve a similar performance as that of the *optimal decision*. While for other schemes, more than 50% cooperators might exist in the FL system. That is, some cooperators might receive a negative utility, which is unprofitable for these clients. The *stochastic MG* scheme can achieve better performance compared to that of the *standard MG* scheme or the *random decision* scheme. Specifically, with the FL model training process, we have the ratio of cooperated clients approaching the optimal value 0.5. Fig. 7b illustrates the ratio of cooperated clients versus the FL training round when $c = 10$. We find that the *stochastic MG* scheme cannot easily approach the optimal value. This is due to a higher exploration probability for the case with $c = 10$ compared to the case with $c = 1$.

Fig. 8 illustrates the attendance versus the FL training round. Again, we find that the *coalition MG* scheme can achieve performance close to the optimal value. For the *stochastic MG* scheme, we have the attendance changing from a large number to a number close to zero. This implies that the *stochastic MG* scheme can help clients keep achieving a non-negative reward.

8.2.2 Volatility

We also show the system volatility in different experimental settings as shown in Fig. 9. Again, we find that the *coalition MG* scheme can achieve the minimum system volatility among all five schemes. While the *stochastic MG* scheme can achieve a lower volatility compared to the *standard MG* scheme or the *random decision*. The *stochastic MG* scheme and the *coalition MG* scheme can reduce the volatility by 51%-100% compared to other baselines.

8.2.3 Utility

Fig. 10 shows the CDF of client utility over the FL training process. Recall that a negative utility will reduce motivation to participate in FL model training, therefore, a non-negative utility is preferred. For the *coalition MG* scheme and the *optimal decision*, we find that the clients will receive a non-negative utility from the start to the end of training. For the *stochastic MG* scheme, we have the CDF curve close to the optimal one, especially for the case with $c = 1$. While for the *standard MG* scheme and the *random decision*, the client utility is negative with a high probability. The *stochastic MG* scheme and the *coalition MG* scheme can improve the utility by 39%-48% compared to other baselines.

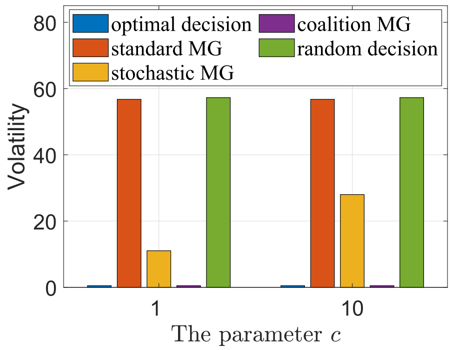
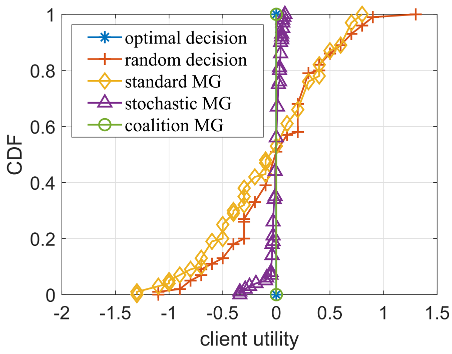
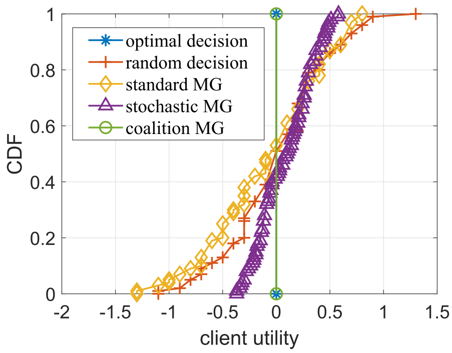


Fig. 9. The system volatility versus different parameter c.



(a) $c = 1$

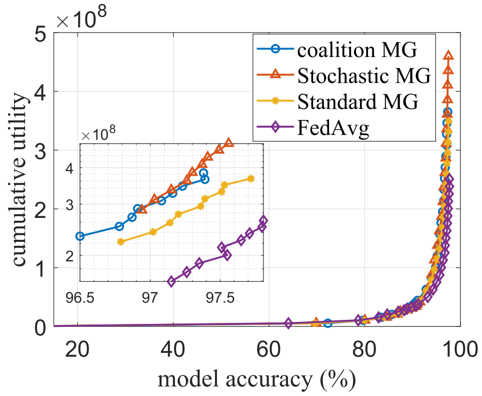


(b) $c = 10$

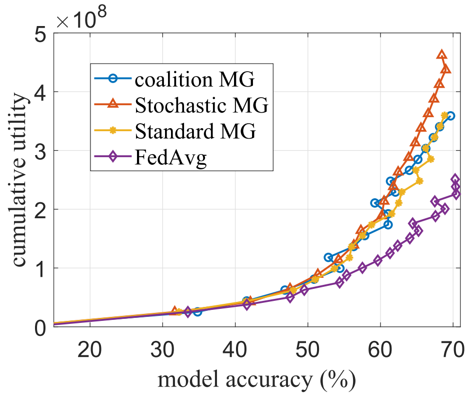
Fig. 10. The CDF of client utility over the FL training process.

8.2.4 Model Inference Accuracy

Fig. 11 demonstrates the cumulative client utility versus the FL model accuracy on MNIST and CIFAR-10 datasets. As the progress of FL training, the cumulative utility increases with the model accuracy. However, the increasing rate is different. Our MG-based distributed algorithms outperform the state-of-the-art active FedAvg algorithm. For time-varying client participation, the *stochastic MG* scheme can achieve the highest expectation on utility due to its stochastic decision-making design, which can be referred in Section 6. While the *coalition MG* and *standard MG* schemes can also achieve superior performance compared to the *FedAvg* solution. The system scalability for the *standard MG* scheme is slightly lower than that of the *stochastic MG* scheme, due to the unsettled attendance volatility problem. While for the *coalition MG* scheme, as the coalition is unchanged with training, it cannot get the best performance as well. Without regard to utility, the accuracy of all schemes approach the optimal model accuracy with the training progress.



(a) MNIST



(b) CIFAR-10

Fig. 11. The cumulative client utility versus the FL model accuracy.

9 Conclusion

This paper formulated a minority game (MG) model of autonomous client participation decision for federated learning. We first proposed an MG-based framework for autonomous client participation problem. We found that the performance cannot be continuously improved with the evolutionary training process in federated learning. Facing this challenge, we then proposed an extended MG-based algorithm, namely, stochastic MG-based decision, where clients gradually change their model contribution level in a stochastic manner. Besides, we also observed that clients can cooperate with each other to improve the possibility of taking the winner action. By taking competition and cooperation among clients into consideration, we further proposed a coalition MG-based participation decision algorithm. The results indicate that the stochastic MG-based algorithm and the coalition MG-based algorithm can improve the utility by 39%-48% and reduce the volatility by 51%-100% compared to other baselines. Moreover, the FL model accuracy achieved by our scheme is comparable to that achieved by centralized algorithms. As the future work, we plan to investigate the extension of our MG-based algorithms to scenarios with multiple federated learning tasks.

REFERENCES

[1] ACUMEN, "Machine learning market (by component: Hardware, software, services; by enterprise size: Smes, large enterprises, by end-use) – global industry analysis, market size, opportunities and forecast, 2019–2026," 2019, accessed: 26 May 2020. [Online]. Available: https://www.acumenresearchandconsulting.com/machine-learning-market

[2] M. Hu, L. Zhuang, D. Wu, Y. Zhou, X. Chen, and L. Xiao, "Learning driven computation offloading for asymmetrically informed edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1802–1815, Aug. 2019.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[4] L. Lyu *et al.*, "Towards fair and privacy-preserving federated deep models," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 11, pp. 2524–2541, Nov. 2020.

[5] M. Shayan, C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Biscotti: A blockchain system for private and secure federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1513–1525, Jul. 2021.

[6] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," in *Proc. Int. Conf. Learn. Representations Workshop*, 2016, pp. 1–10.

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

[8] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," in *Proc. IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2020.

[9] M. A. Heikkila, A. Koskela, K. Shimizu, S. Kaski, and A. Honkela, "Differentially private cross-silo federated learning," 2020, *arXiv:2007.05553*.

[10] X. Liang, Y. Liu, J. Luo, Y. He, T. Chen, and Q. Yang, "Self-supervised cross-silo federated neural architecture search," 2021, *arXiv:2101.11896*.

[11] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A stackelberg game perspective," *IEEE Netw. Lett.*, vol. 2, no. 1, pp. 23–27, Mar. 2020.

[12] 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.

[13] R. Zeng, C. Zeng, X. Wang, B. Li, and X. Chu, "A comprehensive survey of incentive mechanism for federated learning," 2021, *arXiv:2106.15406*.

[14] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/TETC.2021.3063517.

[15] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications* Cambridge, U.K.: Cambridge Univ. Press, 2012.

[16] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*.

[17] W. Yang *et al.*, "Gain without pain: Offsetting DP-injected noises stealthily in cross-device federated learning," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2021.3102030.

[18] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. VincentPoor, "Age-based scheduling policy for federated learning in mobile edge networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 8743–8747.

[19] M. M. Wadu, S. Samarakoon, and M. Bennis, "Federated learning under channel uncertainty: Joint client scheduling and resource allocation," 2020, *arXiv:2002.00802*.

[20] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," 2020, *arXiv:2001.10402*.

[21] H. Yu *et al.*, "A sustainable incentive scheme for federated learning," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 58–69, Jul./Aug. 2020.

[22] H. Yu *et al.*, "A fairness-aware incentive scheme for federated learning," in *Proc. AAAI/ACM Conf. AI Ethics Soc.*, 2020, pp. 393–399.

[23] 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.

[24] D. Challet, M. Marsili, and Y.-C. Zhang, "Modeling market mechanism with minority game," *Physica A-Stat. Mechanics Appl.*, vol. 276, no. 1, pp. 284–315, 2000.

[25] D. Challet, M. Marsili, and Y.-C. Zhang, *Minority Games: Interacting Agents in Financial Markets*. New York, NY, USA: Oxford Univ. Press, Inc., 2005.

[26] S. Ranadheera, S. Maghsudi, and E. Hossain, "Minority games with applications to distributed decision making and control in wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 184–192, Oct. 2017.

[27] S. Ranadheera, S. Maghsudi, and E. Hossain, "Computation offloading and activation of mobile edge computing servers: A minority game," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 688–691, Oct. 2018.

[28] M. Hu, Z. Xie, D. Wu, Y. Zhou, X. Chen, and L. Xiao, "Heterogeneous edge offloading with incomplete information: A minority game approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 9, pp. 2139–2154, Sep. 2020.

[29] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 843–852.

[30] R. L. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. H. Ngo, "Predicting sample size required for classification performance," *BMC Med. Inform. Decis. Mak.*, vol. 12, no. 1, pp. 1–10, 2012.

[31] C. Tang, X. Li, M. Cao, Z. Zhang, and X. Yu, "Incentive mechanism for macrotasking crowdsourcing: A zero-determinant strategy approach," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8589–8601, Oct. 2019.

[32] A. Sinha and A. Anastasopoulos, "Distributed mechanism design with learning guarantees for private and public goods problems," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4106–4121, Oct. 2020.

[33] D. Justus, J. Brennan, S. Bonner, and A. S. McGough, "Predicting the computational cost of deep learning models," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 3873–3882.

[34] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, "On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 4, pp. 832–847, Apr. 2017.

[35] L. Yang, B. Tan, V. W. Zheng, K. Chen, and Q. Yang, *Federated Recommendation Syst.* Berlin, Germany: Springer, 2020, pp. 225–239.

[36] W. B. Arthur, "Inductive reasoning and bounded rationality," *Amer. Econ. Rev.*, vol. 84, no. 2, pp. 406–411, 1994.

[37] Zhang and Yi-Cheng, "Evolving models of financial markets," *Europhys. News*, vol. 29, no. 2, pp. 51–54, 1998.

[38] C. H. Yeung and Y.-C. Zhang, *Minority Games*. New York, NY: Springer, 2009, pp. 5588–5604.

[39] A. Cavagna, J. P. Garrahan, I. Giardina, and D. Sherrington, "Thermal model for adaptive competition in a market," *Phys. Rev. Lett.*, vol. 83, no. 21, 1999, Art. no. 4429.

[40] G. Reents, R. Metzler, and W. Kinzel, "A stochastic strategy for the minority game," *Physica A: Statist. Mechanics Appl.*, vol. 299, no. 1, pp. 253–261, 2001.

[41] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications.* New York, NY, USA: Cambridge Univ. Press, 2011.

[42] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, "Coalition structure generation with worst case guarantees," *Artif. Intell.*, vol. 111, no. 1, pp. 209–238, 1999.

[43] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, "Coalitional game theory for communication networks," *IEEE Signal Process. Mag.*, vol. 26, no. 5, pp. 77–97, Sep. 2009.

[44] B. Ma, M. H. Cheung, and V. W. Wong, "Interference management for multimedia femtocell networks with coalition formation game," in *Proc. IEEE Int. Conf. Commun.*, 2013, pp. 6112–6117.

[45] F. Pantisano, M. Bennis, W. Saad, R. Verdone, and M. Latva-aho, "Coalition formation games for femtocell interference management: A recursive core approach," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2011, pp. 1161–1166.

[46] L. Á. Kóczy, "A recursive core for partition function form games," *Theory Decis.*, vol. 63, no. 1, pp. 41–51, 2007.

[47] J. S. Ng *et al.*, "Reputation-aware hedonic coalition formation for efficient serverless hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, to be published, doi: 10.1109/TPDS.2021.3139039.

[48] D. Ray, "A game-theoretic perspective on coalition formation," Oxford Univ. Press, pp. 1–336, 2007.

[49] 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.
[50] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Master's thesis, Dept. Comput. Sci., Univ. Toronto, Citeseer, Apr. 2009.
[51] W. Feller, *An Introduction to Probability Theory and its Applications*. New York, NY, USA: Wiley, 1970.

Miao Hu (Member, IEEE) received the BS degree and the PhD degree in communication engineering from the Beijing Jiaotong University, Beijing, China, in 2011 and 2017, respectively. He is currently an associate professor with the School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China. From September 2014 to September 2015, he was a visiting scholar with the Pennsylvania State University, PA, USA. His research interests include edge/cloud computing and distributed networks.

Di Wu (Senior Member, IEEE) received the BS degree from the University of Science and Technology of China, Hefei, China, in 2000, the MS degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003, and the PhD degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2007. He was a postdoctoral researcher with the Department of Computer Science and Engineering, Polytechnic Institute of New York University, Brooklyn, NY, USA, from 2007 to 2009, advised by Prof. K. W. Ross. He is currently a professor and the associate dean with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. He was the recipient of the IEEE INFOCOM 2009 Best Paper Award, IEEE Jack Neubauer Memorial Award, and etc. His research interests include edge/cloud computing, multimedia communication, Internet measurement, and network security.

Yipeng Zhou (Member, IEEE) received the bachelor's degree in computer science from the University of Science and Technology of China and the PhD and Mphil degrees from Information Engineering (IE) Department, Chinese University of Hong Kong. He is currently a senior lecturer in computer science with School of Computing, Faculty of Science and Engineering, Macquarie University, and the recipient of ARC DECRA in 2018. From August 2016 to February 2018, he was a research fellow with Institute for Telecommunications Research (ITR) of University of South Australia. From 2013.9–2016.9, He was a lecturer with College of Computer Science and Software Engineering, Shenzhen University. He was a postdoctoral fellow with the Institute of Network Coding (INC) of The Chinese University of Hong Kong (CUHK) from August 2012 to August 2013. His research interests include federated learning, privacy preserved learning and video streaming.

Xu Chen received the PhD degree in information engineering from the Chinese University of Hong Kong, in 2012. He is currently a full professor with Sun Yat-sen University, Guangzhou, China, and the vice director of the National and Local Joint Engineering Laboratory of Digital Home Interactive Applications. He was a postdoctoral research associate with Arizona State University, Tempe, USA, from 2012 to 2014, and a humboldt scholar fellow with the Institute of Computer Science, University of Goettingen, Germany, from 2014 to 2016. He was a recipient of the Prestigious Humboldt Research Fellowship awarded by Alexander von Humboldt Foundation of Germany, 2014 Hong Kong Young Scientist Runner-Up Award, 2016 Thousand Talents Plan Award for Young Professionals of China, 2017 IEEE Communication Society Asia–Pacific Outstanding Young Researcher Award, 2017 IEEE ComSoc Young Professional Best Paper Award, the Honorable Mention Award of 2010 IEEE international conference on Intelligence and Security Informatics, the Best Paper Runner-Up Award of 2014 IEEE International Conference on Computer Communications (INFOCOM), and the Best Paper Award of 2017 *IEEE International Conference on Communications*. He is currently an area editor of *IEEE Open Journal of the Communications Society*, an associate editor of the *IEEE Transactions Wireless Communications*, *IEEE Internet of Things Journal* and *IEEE Journal on Selected Areas in Communications (JSAC)* Series on Network Softwarization and Enablers.

Min Chen (Fellow, IEEE) is currently a full professor with the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST) since February 2012. He is the director with the Embedded and Pervasive Computing (EPIC) Lab, and the director with the Data Engineering Institute, HUST. He is the founding chair of IEEE Computer Society Special Technical Communities on Big Data. He was an assistant professor with the School of Computer Science and Engineering at Seoul National University. He worked as a postdoctoral fellow with the Department of Electrical and Computer Engineering at University of British Columbia for three years. He has 300+ publications, including 200+ SCI papers, 100+ IEEE Transactions/Journals papers, 34 ESI highly cited papers and 12 ESI hot papers. He has published 12 books, including Big Data Analytics for Cloud/IoT and Cognitive Computing (2017) with Wiley. His Google Scholar Citations reached 32,500+ with an h-index of 86. His top paper was cited 3,650+ times. He was selected as Highly Cited Researcher since 2018. He got IEEE Communications Society Fred W. Ellersick Prize in 2017, and the IEEE Jack Neubauer Memorial Award in 2019. His research interests include cognitive computing, 5G Networks, wearable computing, Big Data analytics, robotics, machine learning, deep learning, emotion detection, and mobile edge computing, etc.