

Mobile Devices Strategies in Blockchain-based Federated Learning: A Dynamic Game Perspective

Sizheng Fan, *Student Member, IEEE*, Hongbo Zhang, Zehua Wang, *Member, IEEE*, and Wei Cai, *Member, IEEE*

Abstract—Leveraging various mobile devices to train the shared model collaboratively, federated learning (FL) can improve the privacy and security of 6G communication. To economically encourage the participation of heterogeneous mobile devices, an incentive mechanism and a fair trading platform are needed. In this paper, we implement a blockchain-based FL system and propose an incentive mechanism to establish a decentralized and transparent trading platform. Moreover, to better understand the mobile devices' behaviors, we provide economic analysis for this market. Specifically, we propose two strategy models for mobile devices: the discrete strategy model (DSM) and the continuous strategy model (CSM). Also, we formulate the interactions among the non-cooperative mobile devices as a dynamic game, where they adjust their strategies iteratively to maximize the individual payoff based on others' previous strategies. We further prove the existence of Nash equilibrium (NE) in two different models and propose algorithms to achieve them. Simulation results demonstrate the convergence of the proposed algorithms and show that the CSM can effectively increase the mobile devices' payoffs to 128.1 percent at most compared with DSM.

Index Terms—Federated learning, Nash equilibrium, dynamic game, blockchain

I. INTRODUCTION

6G with transformative technologies will transfer wireless communication from “connected things” to “connected intelligence” [1], which will be empowered by ubiquitous artificial intelligence (AI) [2]. However, in traditional machine learning (ML), the data are gathered to the centralized database for the model training at some powerful computing platforms such as a cloud data center, which have efficiency issues and cause privacy and safety concerns [3].

Fortunately, the blockchain-based federated learning (FL) system provides a promising solution to such a problem [4]. FL can assign machine learning tasks to mobile devices to train the shared model collaboratively. For example, Google used the FL to power applications such as next-word prediction through jointly learning users' behaviors across a large pool of mobile devices [5]. Since the raw data used for model training are stored in the local mobile devices instead of a cloud,

This work was supported by Project 61902333 supported by National Natural Science Foundation of China, in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS). (Corresponding author: Wei Cai.)

S. Fan, H. Zhang, and W. Cai are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172 China and Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China (e-mail: {sizhengfan, hongbozhang}@link.cuhk.edu.cn, caiwei@cuhk.edu.cn).

Z. Wang is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 2G9, Canada (e-mail: zwang@ece.ubc.ca).

users' privacy is enhanced. In FL, two parties are involved: the *requester*, the users who need model training, and the *workers*, namely the mobile devices that use their data to train FL models. The whole procedure of FL consists of several global iterations, with each covering several local iterations. Each mobile device trains every training model in its local iteration. In a global iteration, the requester sends the training model to mobile devices and updates the model after receiving several trained models. Moreover, using blockchain, FL can be implemented via decentralized data ledgers without requiring any central server, which mitigates the risks of single-point failures [6]. All network entities transparently trace any update events and user behaviors. Furthermore, through transaction logs, one can easily trace the origin where a model parameter is modified or updated during the training process. Also, the smart contracts on the blockchain can provide automatic and decentralized services [7]. Once the smart contract is deployed on the blockchain, the code cannot be modified, and it can provide a self-organized transaction process [8]. Overall, the blockchain-based FL systems can support a secure, decentralized platform and offer services to fulfill the application requirement.

Despite all the benefits, blockchain-based FL systems still face challenges. The first issue is *efficiency and scalability problem*. Although it would be an effective way for blockchain to serve as decentralized storage and replace the central FL servers through the literature review [9], how to choose an efficient consensus mechanism for FL is also a key challenge. Specifically, based on diverse consensus mechanisms, blockchains can be categorized into three main types: public, private, and consortium blockchain [10]. Zhang *et al.* proposed a reliable public blockchain-based FL system and designed an incentive mechanism to engage self-interested participants [11]. However, even though public blockchains¹ have better information transparency and auditability due to no access limitation, the consensus process among all nodes incur the high computational cost and long delay. Thus, public blockchains are unsuitable for energy-limited and time-sensitive scenarios, such as FL based on mobile devices. The second issue is *unrealistic ecosystem*. Many existing studies [14] assume that the mobile devices voluntarily participate in FL without asking for any returns. However, these assumptions are not realistic, as mobile devices' training models incur various costs, electricity, and bandwidth, to name a few. Thus, it is more reasonable to compensate the mobile devices when the requester wants to

¹The public blockchain, e.g., BitCoin [12], and Ethereum [13], is also referred to as permissionless blockchain since public participants reach the consensus through the Proof-of-Work (PoW) puzzle competition.

use resources in them. Under this incentive mechanism, mobile devices tend to choose strategies to maximize their payoffs in the whole FL process, given the fixed incentive mechanism. However, this may not be easy to achieve if the mobile device does not know other mobile devices' information, such as the total data size and the training cost² because of mobile devices' privacy concerns. Therefore, it is also a challenge to solve the problem of incomplete information among mobile devices.

To tackle these two issues, we implement a consortium blockchain-based resource trading system. Based on the system, we deploy a smart contract based incentive mechanism and conduct a comprehensive economic analysis of the blockchain-based FL resource trading market from the perspective of mobile devices.

We first consider the implementation of the blockchain system. Due to the intrinsic differences in the consensus model, consortium blockchains outperform public blockchains in terms of efficiency, scalability, and privacy³. However, there are two limitations to consortium blockchains: 1) *Pre-authorized Problem*: the consortium blockchain is a type of permissioned blockchain where the consensus is achieved by a set of pre-authorized nodes only; 2) *In-scope Consensus*: the consensus in permissioned blockchains is only recognized by nodes within miners, which is different from public blockchains, whose consensus will be recognized by all nodes [10]. These limitations do not apply in the scenario. First, *Pre-authorized Problem* does not exist in this application. The problem can be solved by off-chain authentication such as mobile phone number authentication and identity authentication, which is also a solution for the Sybil Attack [15] plaguing consortium blockchains. Second, *In-scope Consensus* is sufficient in this application. Transparency and auditability are only required within the context of the stakeholders, who are the requester and the mobile devices. In this case, the mobile devices also serve as the consensus nodes in the consortium blockchain since they are the stakeholders of the reward smart contract deployed by the requesters. According to these features, considering the real-time requirements and synchronization of FL, we adopt the consortium blockchain to work out a decentralized and transparent platform for reward allocation among mobile devices.

To engage self-interested mobile devices, we propose and deploy a smart contract based incentive mechanism based on the mobile devices' contribution in the current global iteration to motivate mobile devices to participate in the FL tasks. In this paper, we consider two kinds of strategy models: *discrete strategy model (DSM)* and *continuous strategy model (CSM)* from the perspective of mobile devices. DSM means that the mobile devices can decide whether to participate in the FL task or not. If a mobile device decides to join, all her/his data will be used to train the model. And CSM refers to that mobile devices can determine the percentage of data used in the FL task. Motivated by the iterative interaction of FL, we model the

mobile devices' decision problem as a dynamic game to solve the problem of incomplete information in the training process. Specifically, empowered by the consortium blockchain system, even though the mobile devices do not know others' personal information, including total data size and training cost, she/he can observe the histories of the other mobile devices' strategies and adjust her/his strategy to maximize the payoff in the FL task.

We summarize our key results and contributions as follows:

- *System Modeling and Formulation*: We propose and implement a smart contract based incentive mechanism to encourage mobile devices to participate in the FL tasks. In this case, two kinds of strategy models of mobile devices in the FL tasks are proposed.
- *Dynamic Game Analysis*: To the best of our knowledge, it is the first work in modeling dynamic game for FL, which builds a foundation for further research. We prove the existence of the Nash equilibrium (NE) and mixed-strategy NE of the two models and propose the two corresponding algorithms to achieve. The simulation results demonstrate the convergence of the two algorithms, and the CSM can effectively increase the mobile devices' payoffs compared with DSM.

The remainder of this paper is organized as follows: Section II presents the related works and analyzes their limitations. The dynamic game model and incentive mechanism are presented in Section IV. We present DSM and discuss the existence of NE in Section V. CSM and the existence of NE are discussed in Section VI. We conduct simulations to evaluate the effectiveness of the proposed method in Section VII. Lastly, Section VIII concludes this paper.

II. RELATED WORK

Many studies towards FL put their primary focus on improving the performance (e.g., [16]), resource allocation [17]–[19] and solving privacy and safety problems (e.g., [20]).

A. Blockchain-based FL System

Some existing references focus on the combination of blockchain and FL to solve the concerns on the security and privacy. For example, Kang *et al.* [21] propose a blockchain-based reputation system for FL to avoid unreliable edge devices acting as workers for federated edge learning. Simulation results show that the proposed methods can obtain performance improvement. Feng *et al.* [22] proposed a blockchain-based FL framework to authenticate cross-domain unmanned aerial vehicles to avoid poisoning attacks. Shayan *et al.* [23] proposed a blockchain-based system named Biscotti that co-designs a privacy-preserving FL process. Based on the system, the requesters can detect poisoning attacks via comparing the effect with and without the model update on a database. Different from these previous work, we use the consortium blockchains to establish a transparent and decentralized trading platform among distinct mobile devices.

²The training cost is determined by the training model, task types, and various mobile devices' processors.

³Unlike public blockchains, the information on the consortium blockchain is not available to the public, which protects the privacy of parameters in federated learning to a certain extent.

B. Incentive Mechanism Design for FL

Game theory is often used in the design of incentive mechanisms in several scenarios, such as cloud computing [24] and crowdsourcing [25]. Similarly, proper incentives are also needed to compensate the mobile devices in FL. To motivate mobile devices to participate in the FL, a few studies work on designing the incentive mechanism. However, there are several limitations in these works. First, these existing works modeled the whole procedure of the FL as one global iteration. They neglect the fact that global iterations are dynamic in the entire process of the FL, and mobile devices can decide whether they are going to join or not in every global iteration. For example, Jiao *et al.* [26] did not consider the multiple global iterations in the FL task and proposed an auction-based incentive mechanism to select high-quality mobile devices to maximize social welfare at the beginning of the FL task. Second, the literature is usually modeled from the requester's perspective and ignores the view of the mobile device, e.g., maximizing the requester's payoff given a fixed budget. For example, Ding *et al.* [27] presented an analytical study on the optima incentive mechanism design using a contract from the server's perspective and further analyzed three information scenarios to real the impact of information asymmetry levels on the server's optimal strategy and minimum cost. Third, these existing works always assume that if the mobile devices are selected to participate FL task, they will use all their data in the FL task. Mobile devices can decide the proportion of data size in the FL task. For example, Fan *et al.* [28] proposed an auction-based incentive mechanism, and all mobile devices will report their data size to the requester. If the mobile devices are selected, they will use all their data to train the FL model. Apart from these earlier works, we consider the FL procedure from mobile devices' point of view. We build up mathematical models of mobile devices, analyze the FL using dynamic games, and maximize the payoff of mobile devices in the whole FL procedure. Moreover, we consider two different kinds of decision models, from "whether participate or not" to "how much to participate" of mobile devices and prove the existence of NE. Finally, we propose a learning-based heuristic algorithm to achieve the NE.

III. BLOCKCHAIN-BASED FEDERATED LEARNING SYSTEM

In this section, we first introduce the motivation to integrate the blockchain and FL. Then, we present the system users and system architecture.

A. The Integration of Blockchain and FL

Combining the blockchain and FL is natural because of the same decentralized character [29]. On the one hand, FL tends to involve multi-parties and leverage various devices to train the shared model collaboratively. On the other hand, blockchain can provide a transparent and secure platform for many parties involved in FL. With the blockchain technique, the participation and the allocation of rewards in the FL are encrypted and stored on blocks, and smart contracts provide a self-organized transaction process. Hence, The integration

of blockchain and FL aims to support a secure, decentralized platform for both parties involved in FL.

Blockchains are currently classified as public blockchain [30], [31] and consortium blockchain [7], which adopt different consensus scopes among participants. The public blockchain achieves consensus among overall participants. In other words, every miner needs to verify and sign the newly generated block and broadcast it to other miners, which might cause large network resource consumption and high delay. Besides, several consensus algorithms, such as the PoW, require participants to work out the sizeable computational puzzle and cause significant computational resource consumption and low performance. To compensate the miners for large resource consumption, the requesters also need to pay the high gas fee.

Compared with the public blockchain, the consortium blockchain uses Practical Byzantine Fault Tolerance (PBFT)⁴ to achieve a higher Transaction Per Second (TPS)⁵ and eliminates the gas fee's cost due to the narrowing consensus scope [32]. Since the recorded data only need to be verified within all mobile devices' scope rather than all participants, we can adopt the consortium blockchain in our system with all mobile devices participating in the consensus. With the consensus, all other devices can verify the FL task's participation and the allocation of rewards and generate new blocks. Moreover, the smart contract, which can provide an automatic and decentralized service, does not require gas.

Therefore, because of the real-time requirements of FL and the frequent interactions among workers and requesters, we decide to implement consortium blockchain as an infrastructure for FL. The detailed information of the system will be shown in III-C.

B. System Users

Therefore, because of the real-time requirements of FL and frequent interactions among workers and requesters, we decide to implement consortium blockchain as an infrastructure for FL. The detailed information of the system will be shown in III-C.

- **Requesters:** Requesters are individuals and companies who need FL tasks training. They will send task requirements, including task types and budgets, to the blockchain system. After every local iteration, they will obtain and evaluate the model parameters. Meanwhile, they will pay to mobile devices. The training progress will continue until the accuracy meets their requirements.
- **Mobile Devices:** Mobile devices include laptops, smartphones, etc. They perform a local model update using their dataset for serving FL and are incentivized via payment from the requester.

⁴Practical Byzantine Fault Tolerance (PBFT) is an algorithm that optimizes aspects of Byzantine Fault Tolerance (BFT) and has been implemented in several modern distributed computer systems, especially in some blockchain platforms.

⁵TPS refers to the number of atomic actions performed by a certain entity per second.

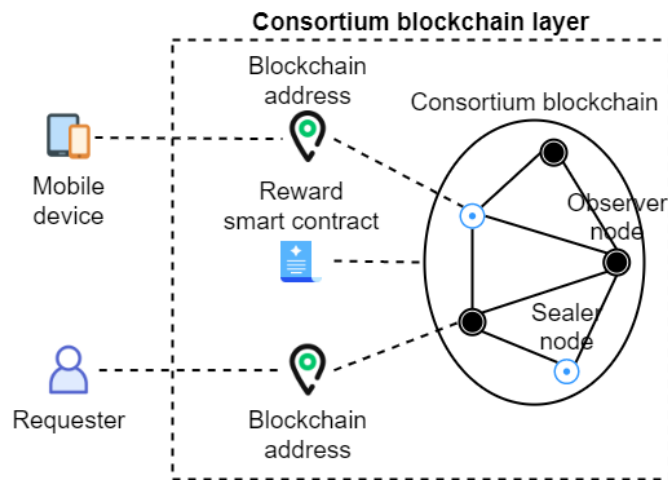


Fig. 1. System Architecture

C. Blockchain architecture

As shown in Fig. 1, the requester and mobile devices register on the consortium blockchains. Each has its address, a pair of asymmetric keys (public/private), and a certificate. Two kinds of nodes are on the consortium blockchain: *sealer nodes* and *observer nodes*. Both can send transactions, while only sealer nodes can join the consensus process. Hence, the selection of sealer nodes should be carefully considered. Otherwise, the recorded data and smart contracts on the consortium blockchain are at risk of being tampered with. Since the consensus can ensure that smart contracts can satisfy the profits of the majority, we can randomly choose initial mobile devices as sealer nodes without loss of generality. As we mentioned before, we can use the reward smart contract to allocate the corresponding rewards for mobile devices. The smart contract requires the requester and the mobile devices to reach a consensus since it involves both parties' interests. As the transaction initiator, the requester deploys the reward smart contract. The mobile devices that work as sealer nodes participate in the consensus verification to authenticate the deployed smart contract. There is a competitive relationship among the mobile devices, and they are all seeking a fair reward algorithm to ensure their profits. Hence, most mobile devices tend to maintain the fair reward algorithm rather than maliciously tamper with it to prevent the requester from quitting the transaction and losing what they should have gained. Moreover, the smart contract is open and transparent. If the smart contract is tampered with, the requester can immediately detect tampering and terminate the smart contract. With all these considerations, we let the mobile devices work as sealer nodes and the requester work as observer nodes, which is also shown in Fig. 1. When a mobile device wants to join the system, she/he sends the IP address to all the sealer nodes by off-chain means. Through the PBFT consensus process, she/he can join the network if more than two-thirds of the sealer nodes authenticate her/his IP address.

In addition to providing a platform for the reward smart contract, the consortium blockchain can also store transaction records. The history of the strategies in each global iteration

can be recorded on the blockchain. In other words, each device can observe all histories in previous global iterations.

IV. SYSTEM MODEL

A. FL Process

FL is a distributed privacy-preserving machine learning technique. It can make multiple mobile devices capable of collaboratively training a global shared model without the necessity of uploading private data that are stored locally to a central server [33].

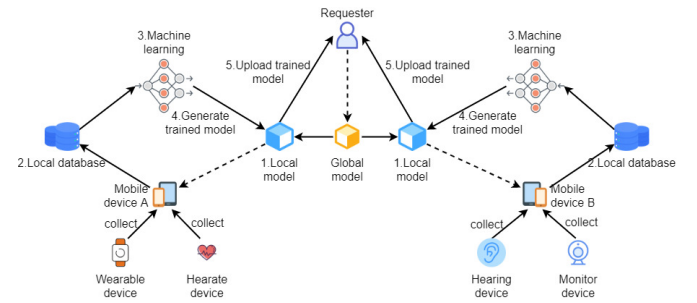


Fig. 2. Federated Learning Process

As depicted in Fig. 2, we consider a wireless FL scenario with a requester and several mobile devices. M mobile devices, where $\mathcal{M} = \{m_1, \dots, m_j, \dots, m_M\}$, that have the computing capacity and local datasets $\mathcal{D} = \{D_1, \dots, D_j, \dots, D_N\}$ to provide FL services for the requester. We denote d_j to represent the data size of m_j . To protect privacy, the requester with the FL task can train the machine learning model on these mobile devices' without data collection. The requester first sends the model to all participating mobile devices (step 1). In several iterations, every mobile device will minimize the training loss over its local data (steps 2-4). The loss function on the local data of m_j is written as follow:

$$L_j = \frac{1}{d_j} \sum_{n \in D_j} f_n(\mathbf{w}) \quad (1)$$

where L_j refers to the training loss over m_j 's local data and $f_n(\mathbf{w})$ is the loss function of sample n .

Then, each participating mobile device will send the new weight and the gradient to the requester to update the global model (step 5). Then, the goal of FL is to minimize the global loss function $L(\mathbf{w})$, which is defined as follow [34]

$$\min_{\mathbf{w}} L(\mathbf{w}) = \frac{\sum_{j=1}^N d_j L_j}{D} \quad (2)$$

After receiving the weights and gradients, the requester evaluates the model updates from all participating mobile devices using several mechanisms. Two schemes are used frequently in this procedure to identify unreliable participants in the FL task for potential attack risk. The first one is the rejection of negative influence (RONI) for independent and identically distributed (IID) data scenarios and another is FoolsGold mechanism, aiming for non-IID data scenarios [35] [36]. In this case, the requester can take advantage of

these useful schemes to effectively remove the unreliable local model updates from the unreliable mobile devices and refuse to pay them. The requester integrates all the reliable local model updates into an average value in Eq.(3), (4) and sets the average value as the new global model for the next iteration. Without interference from the unreliable participants, the newly generated value is more accurate, boosting the efficiency of the model training. Then, the requester returns the new global model to the reliable participants and begins the next global iteration.

$$\mathbf{w}_g^{t+1} = \frac{1}{N} \sum_{j=1}^N \mathbf{w}_j^t \quad (3)$$

$$\nabla J^{t+1} = \frac{1}{N} \sum_{n=1}^N \nabla J^n \quad (4)$$

where \mathbf{w}_g^{t+1} and ∇J^{t+1} refers to average weights and average gradients in t th iteration. They will be set as the initial model parameters in $t + 1$ th iteration.

B. Dynamic Game for Blockchain-based Federated Learning

The FL task procedure consists of several global iterations, as shown in Fig. 3. The FL task consists of several global iterations. In every global iteration, the requester will announce a fixed reward B for every global iteration, and the workers will decide whether to participate in every global iteration or not. We model the interactions between the requester and the mobile devices as a dynamic game. The interactions can be illustrated in the following order.

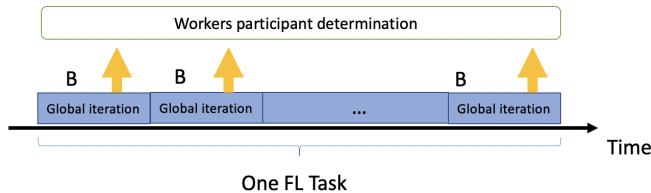


Fig. 3. The component of FL

As shown in Fig. 4, combined with the consortium blockchain-based FL system mentioned in Section III, the requester will publish her/his task requirement, including data type, models types (i.e., convolutional neural network (CNN), recurrent neural network (RNN)), the maximal computational time T_{max} and the budget B at the beginning of FL task (step 1). The interaction between the requester and the mobile devices in one global iteration will be shown as follows. Given the fixed requester's fixed budget B , the maximal computational and communication time T_{max} (step 2), every mobile device $m_j \in \mathcal{M}$ can decide her/his strategy in the current global iteration according to other mobile devices' strategies in the past and record it on the blockchain (step 3). Upon receiving the mobile devices' information, the smart contract will calculate the reward (step 4). The mobile devices will receive the requester's model and begin training with their local datasets (step 5). After finishing the training process, the

mobile devices send the model parameters to the requester (step 6). Steps 3, 4, 5 and 6 in Fig. 4 are repeated in each global iteration. We assume that mobile devices keep connecting in every global iteration. Each mobile device's goal is to choose the best strategy in one global iteration that maximizes her/his expected payoff during the whole FL task.

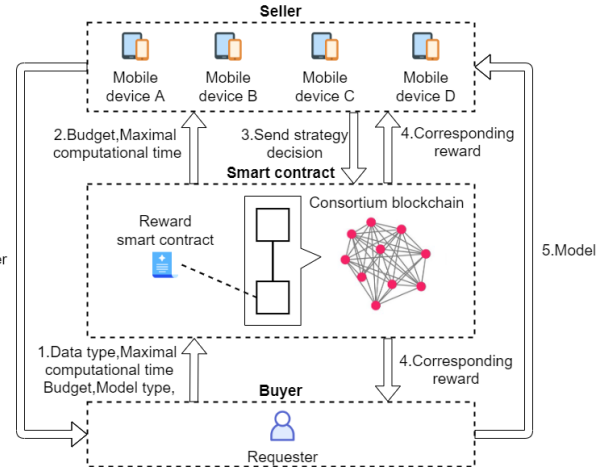


Fig. 4. Transaction Process

C. Incentive Mechanism Design

In our FL model, the requester will provide a budget B in every global iteration, and M mobile devices compete for the budget. The mobile devices will send their strategy decisions (i.e., the used data sizes) to the smart contract in every global iteration. We use $s_{j,n}$ to denote the used data size that the mobile device m_j chooses in the n th global iteration,

Most existing incentive mechanisms in FL are with high complexity [26], [28]. However, an effective incentive mechanism in FL should be low complex and easy to implement. The real-time requirement in every global iteration is that mobile devices receive compensation from the requester before the next global iteration, which can help them make decisions in the following global iterations. Besides, mobile devices often face problems such as lack of power and low signal. Hence, an incentive mechanism should be with lower computing complexity, saving mobile devices' waiting time. The incentive mechanism proposed in this paper is shown as follows.

$$p_{j,n} = B \frac{s_{j,n}}{s_{j,n} + \sum_{i \neq j} s_{i,n}} \quad (5)$$

From Eq.(5), we can understand that the requester proportionally allocates its fixed budget to the set of mobile devices based on their data size in the current global iteration.

V. DSM

In this section, we first study the DSM. In this model, each mobile device decides whether to participate in the current global iteration or not to maximize her/his payoff in the whole FL task. Next, we discuss pure strategy NE and mixed strategy NE in one global iteration and propose the algorithm to achieve it.

A. Problem Formulation

The players of the DSM are the set \mathcal{M} of mobile devices.

The strategy of $m_j \in \mathcal{M}$ is to decide to participate in with $\sigma_{j,n} = 1$ and give up with $\sigma_{j,n} = 0$ the n th global iteration, respectively. We denote the strategies of players in \mathcal{M} in the n th global iteration expect m_j as $\sigma_{-j,n}$.

Next, we define the mobile devices' payoffs in one global iteration. Denote m_j 's training cost in n th global iteration for FL task as $c_{j,n}$. $c_{j,n} = c_{j,n}^d + c_{j,n}^p + c_{j,n}^c$ is composed of three parts, m_j 's cost for using these data in one global iteration, namely data cost denoted as $c_{j,n}^d$, the cost for computing data in one global iteration, namely computing cost denoted as $c_{j,n}^p$, and the cost for uploading the weights and gradients to the requester in one global iteration, namely communication cost denoted as $c_{j,n}^c$. For notational convenience, we assume that the training cost of m_j is related to the used data size in the current global iteration [37], which can be represented as follows.

$$c_{j,n} = \theta_j d_{j,n} \sigma_{j,n} \quad (6)$$

where $d_{j,n}$ refers to the used data size of m_j in the n th global iteration. θ_j refers to the cost to train the unit data and is related to both the model types and the m_j .

The payoff of m_j in the n th global iteration of FL task can be defined as follows.

$$V_{j,n}^D((\sigma_{j,n}, \sigma_{-j,n})) = \mathbb{1}_{t_{j,n} \leq t_{max}} (-\theta_j d_{j,n} + p_{j,n}) \cdot \sigma_{j,n} \quad (7)$$

where $p_{j,n}$ is defined in Eq.(5) and means the compensation from the requester to m_j in the n th global iteration, and

$$\mathbb{1}_{t_{j,n} \leq T_{max}} = \begin{cases} 1, & t_{j,n} \leq T_{max} \\ 0, & \text{otherwise.} \end{cases}$$

which means that only mobile devices satisfies the condition $t_{j,n} \leq T_{max}$ can participate in the n th global iteration to train the model. The payoff of mobile device m_j is related to not only her/his strategy $\sigma_{j,n}$ but also the strategies $\sigma_{-j,n}$ of other players in \mathcal{M} except m_j in the n th global iteration.

Further, we define the overall payoff of m_j over the whole federated learning task as

$$V_j^D = \sum_{n=1}^{\infty} \delta_j^{n-1} V_{j,n} \quad (8)$$

where $\delta_j \in (0, 1)$ is the discount factor, which represents the m_j 's patience about the future [38].

In conclusion, the Discrete Computation Decision Game (DCDG) is defined as follows.

Definition 1.(DCDG in the n th global iteration)

- Players: the set of mobile devices \mathcal{M} .
- Strategies: $\sigma_{j,n} \in \{0, 1\}, \forall j \in \mathcal{M}$.
- payoff: $V_{j,n}(\sigma_{j,n}, \sigma_{-j,n}), \forall j \in \mathcal{M}$.

where $V_{j,n}$ refers to the payoff of mobile device m_j in n th global iteration, and the corresponding NE are shown as follows.

Definition 2.(NE) A NE of DCDG in n th global iteration is a profile $\sigma^* = \{\sigma_{j,n}, j \in \mathcal{N}\}$ such that for each mobile device $m_j \in \mathcal{M}$,

$$V_{j,n}(\sigma_{j,n}^*, \sigma_{-j,n}^*) \geq V_{j,n}(\sigma_{j,n}, \sigma_{-j,n}^*)$$

B. NE Analysis

m_j make decisions in DCDG to maximize her/his overall payoff in the whole FL task. Obviously, to get the maximum value of V_j , we just need to get the maximum value of each item $V_{j,n}$ in Eq.(8). In DSM, if m_j does not take part in n th global iteration, the payoff is 0. So, m_j will choose to participate in the n th global iteration if $V_{j,n}(1, \sigma_{-j,n}) > 0$, and choose to give up the n th global iteration if $V_{j,n}(1, \sigma_{-j,n}) < 0$.

Lemma 1. A Decision Game profile is an NE, if and only if

$$(2\sigma_{j,n} - 1) \cdot V_{j,n}^D(1, \sigma_{-j,n}^*) \geq 0, \forall m_j \in \mathcal{M}$$

Proof: According to the definition of NE, if $(2\sigma_{i,n} - 1)V_{i,n}^D(1, \sigma_{-i,n}^*) < 0$, the best strategy for m_j is to give up the n th global iteration and the strategy is $\sigma_{j,n} = 0$, hence $(2\sigma_{j,n} - 1)(V_{j,n}^D(1, \sigma_{-j,n}^*)) \geq 0$. If $(2\sigma_{j,n} - 1)V_{j,n}^D(1, \sigma_{-j,n}^*) > 0$, the best strategy for m_j is $\sigma_{j,n} = 1$, hence $(2\sigma_{j,n} - 1)(V_{j,n}^D(1, \sigma_{-j,n}^*)) \geq 0$. The necessity has been proved. Then, we proof the sufficiency. Suppose $V_{j,n}^D(1, \sigma_{-j,n}^*) < 0$. Then the desired $\sigma_{j,n}$ that satisfies the condition in Lemma 1 is $\sigma_{j,n} = 0$, which is the best strategy of m_j . Suppose $V_{j,n}^D(1, \sigma_{-j,n}^*) > 0$. Then the desired $\sigma_{j,n}$ that satisfies the condition in Lemma 1 is $\sigma_{j,n} = 1$, which is also the best strategy of m_j . The proof is now completed.

According to the Lemma 1, the DCDG may not always have a pure strategy NE. Next, we will discuss the existence of mixed-strategy NE in DCDG and give an algorithm to achieve it.

C. Mixed-Strategy NE

The mixed strategy of m_j in the n th global iteration is simply a probability distribution over her/his pure strategy, i.e., m_j has the probability $\alpha_{j,n} \in [0, 1]$ to participate in the n th global iteration and the probability $1 - \alpha_{j,n}$ to give up the current global iteration. For notation convenience, we denote the mixed strategy profile of all mobile devices in the n th global iteration as

$$\alpha_n = \{\alpha_{j,n}, j \in \mathcal{M}\}$$

According to the von Neumann Morgenstern criterion [39], the expected payoff of m_j can be defined as

$$\omega_{j,n}(\alpha_{j,n}, \alpha_{-j,n}) = \alpha_{j,n} \cdot \bar{V}_{j,n}(1, \alpha_{-j,n}) + (1 - \alpha_{j,n}) \cdot \bar{V}_{j,n}(0, \alpha_{-j,n}) \quad (9)$$

where $\bar{V}_{j,n}^D(1, \alpha_{-j,n})$ and $\bar{V}_{j,n}^D(0, \alpha_{-j,n})$ are m_j 's expected payoffs when choosing to participate the n th global iteration and give up the n th global iteration, respectively. In our model, if m_j choose 0 in the n th global iteration, her/his payoff will be 0, i.e., $\bar{V}_{j,n}^D(0, \alpha_{-j,n}) = 0$. Note that $\bar{V}_{j,n}^D(1, \alpha_{-j,n})$ refers to the expected values over all possible strategies over all mobile devices expect m_j when m_j choose to participate in the current global iteration. Specifically, $M - 1$ other mobile devices form a subset \mathcal{M}_{-j} , and 2^{M-1} possible strategies combination of those mobile devices form a set

$\Sigma_{-j,n}$. Mobile device $m_i \in \mathcal{M}_{-j}$ has the probability $\alpha_{i,n}$ to participate in and $1 - \alpha_{i,n}$ to give up the current global iteration respectively. Then, the probability of $\sigma_{-j,n} \in \Sigma_{-j,n}$ can be represented as follows.

$$\phi(\sigma_{-j,n}) = \prod_{m_i \in \mathcal{M}_{-j}} \alpha_{i,n} \cdot \sigma_{i,n} \quad (10)$$

Then, the expected payoff of mobile device m_j can be calculated by

$$\begin{aligned} \overline{V_{j,n}^D}(\sigma_{j,n}, \alpha_{-j,n}) &= \sum_{\sigma_{-j,n} \in \Sigma_{-j,n}} \phi(\sigma_{-j,n}) V_{j,n}^D(\sigma_{j,n}, \sigma_{-j,n}), \\ \sigma_{j,n} &\in \{0, 1\} \end{aligned} \quad (11)$$

where $V_{j,n}^D(\sigma_{j,n}, \sigma_{-j,n})$ is the overall payoff of m_j under the pure strategy profile.

Definition 3. (Mixed-Strategy NE) A *mixed-strategy NE* of DCDG is a probability profile α^* such that for each worker $m_j \in \mathcal{M}$:

$$\omega_{j,n}(\alpha_{j,n}^*, \alpha_{-j,n}) \geq \omega_{j,n}(\alpha_{j,n}, \alpha_{-j,n}), \forall \alpha_j \in [0, 1]$$

Lemma 2 There exists at least one mixed-strategy NE in the DCDG.

Proof. The DCDG is a finite game including M players, and each player has two strategies. Hence, there exists at least one mixed-strategy equilibrium [40].

Combined with the consortium blockchain system mentioned in Section III, we propose to use a smoothed best response updated algorithm [41] to compute the mixed-strategy NE. All the players in DCDG (i.e., all the mobile devices) can record their strategies on the blockchain and make decisions based on past strategies. The detailed information will be shown in Alg.1.

Algorithm 1 Smoothed best response algorithm

Require: α^0, γ, Δ

Ensure: α^*

```

1: Initialize  $\leftarrow 0, f \leftarrow 0$ 
2: while  $f = 0$  do
3:    $n \leftarrow n + 1$ 
4:   for  $j = 1 : N$  do
5:     Calculate  $\overline{V_{j,n}^D}(1, \alpha_{-j,n-1})$ 
6:      $\alpha_{j,n+1} = \frac{e^{(1/\lambda)\overline{V_{j,n}^D}(1, \alpha_{-j,n})}}{e^{(1/\lambda)\overline{V_{j,n}^D}(1, \alpha_{-j,n})} + 1}$ 
7:   end for
8:   if  $|\alpha^n - \alpha^{n-1}| < \Delta$  then
9:      $f \leftarrow 1$ 
10:  end if
11: end while
12: return  $\alpha^n$ 

```

At the beginning of the next global iteration, m_j will compute the expected payoff when choosing to participate in this global iteration according to the other mobile devices' strategies recorded on the consortium blockchain in the past (line 4-line 5). Then, each mobile device will update her/his mixed strategy at the next global iteration according to Eq.(12)

(line 6). The mixed-strategy NE will be achieved when the condition is satisfied (line 8-line 10).

$$\alpha_{j,n+1} = \frac{e^{(1/\lambda)\overline{V_{j,n}^D}(1, \alpha_{-j,n})}}{e^{(1/\lambda)\overline{V_{j,n}^D}(1, \alpha_{-j,n})} + 1} \quad (12)$$

where λ refers to the freedom degree⁶ of m_j . According to the result in [41], we can obtain that such a smoothed best response with some learning rules converges to the mixed-strategy NE.

VI. CSM

In this section, we talked about CSM. Specifically, the mobile devices in CSM would adopt not only "participate in" or "not participate in" but also "how much to participate in". We investigate NE's existence and propose a learning-based heuristic algorithm to achieve NE based on CSM.

A. Problem Formulation

As the same as DSM, the players of CSM are also the set of mobile devices \mathcal{M} participating in the FL task.

Unlike DSM, the strategy of $m_j \in \mathcal{M}$ is to decide how much to participate in the FL task in the n th global iteration.

The training cost in CSM of m_j is the same as the training cost in DSM. So, the payoff of m_j in the n th global iteration of the FL task in CSM can be defined as follows:

$$V_{j,n}^C((\sigma_{j,n}, \sigma_{-j,n})) = \mathbb{1}_{t_j \leq t_{max}} (-\theta_j \sigma_{j,n} s_j + p_{j,n}) \quad (13)$$

where $\sigma_{j,n} \in [0, 1]$ denotes the percentage of data used for federated learning in the n th global iteration. Hence, $-\theta_j \sigma_{j,n} s_j$ refers to the training cost when m_j ' strategy is $\sigma_{j,n}$ in the current local iteration. Besides, we use $\sigma_{-j,n}$ to represent the strategies of other mobile devices in \mathcal{M} .

More formally, the Continues Computation Decision Game (CCDG) and the corresponding NE are defined as follows:

Definition 4.(CCDG in one global iteration)

- Players: the set of mobile devices \mathcal{M} .
- Stategies: $\sigma_{j,n} \in [0, 1], \forall m_j \in \mathcal{M}$.
- Payoffs: $V_{j,n}^C(\sigma_{j,n}, \sigma_{-j,n}), \forall m_j \in \mathcal{M}$.

Definition 5.(NE) A NE of the CCDG in n th global iteration is a profile $\sigma^* = \{\sigma_{j,n}, m_j \in \mathcal{M}\}$ such that for $m_j \in \mathcal{M}$,

$$V_{j,n}^C(\sigma_{j,n}^*, \sigma_{-j,n}^*) \geq V_{j,n}^C(\sigma_{j,n}, \sigma_{-j,n}^*)$$

B. NE Analysis

Next, we talk about the existence of NE in CSM. m_j 's strategy $\sigma_{j,n}$ belongs to a closed and bounder convex set $[0, 1]$. And $V_{j,n}^C$ is a continuous and quasi-concave function of $\sigma_{j,n}$. According to the Existence Theorem II of NE [43], the NE of CCDG exists.

⁶A larger freedom degree refers to that the mobile device is more likely to act randomly [42]

C. Learning-based Heuristic Algorithm

We have proved the existence of NE in VI-B. In this subsection, we propose a learning-based heuristic algorithm for the perspective of mobile devices to achieve NE in CSM.

In CSM, every mobile device can change its payoff by adjusting the percentage of data used in the current global iteration of the FL task. If the FL system can achieve NE, each mobile device will achieve NE. In this case, every mobile device can obtain her/his maximum utility and is unwilling to change the strategy. Thus, the objective function of m_j can be represented as follows.

$$\begin{aligned} \max V_j^C(B, (\sigma_{j,n}, \sigma_{-j,n})) \\ \text{s.t. } \sigma_{j,n} \in [0, 1] \end{aligned}$$

where V_j^C represents the overall payoff of m_j under CSM in the FL task.

The value of n th item in the objective function is related to the used data size that the mobile devices choose in the n th global iteration and is unrelated to the other global iterations. Therefore, to maximize the V_j^C , we need to maximize each item's value in this objective function. Thus, the objective function can be transformed as follows.

$$\begin{aligned} \max \delta^{n-1} (-\theta_j \sigma_{j,n} s_j + B \frac{\sigma_{j,n} s_j}{\sigma_{j,n} s_j + \sum_{i \neq j} \sigma_{i,n} s_i}) \\ \text{s.t. } \sigma_j \in [0, 1] \end{aligned}$$

Obviously, in order to make sure that the payoff of the mobile device is not negative, the following condition should be satisfied

$$-\theta_j + B \frac{1}{\sigma_{j,n} s_j + \sum_{i \neq j} \sigma_{i,n} s_i} \geq 0 \quad (14)$$

Next, we let

$$\begin{aligned} g(\sigma_{j,n}) &= \frac{\partial [\delta^{n-1} (-\theta_j \sigma_{j,n} s_j + B \frac{\sigma_{j,n} s_j}{\sigma_{j,n} s_j + \sum_{i \neq j} \sigma_{i,n} s_i})]}{\partial \sigma_{j,n}} \\ &= \delta^{n-1} (-\theta_j s_j + B \frac{s_j \sum_{i \neq j} \sigma_{i,n} s_i}{(\sigma_{j,n} s_j + \sum_{i \neq j} \sigma_{i,n} s_i)^2}) \end{aligned} \quad (15)$$

and analyze its characteristics: We can observe that $\delta^{n-1} > 0$, so we can omit the effect of this part. Otherwise $g(\sigma_{j,n})$ decreases with $\sigma_{j,n}$ and converges to $-\theta_j s_j < 0$ when $\sigma_{j,n} \rightarrow \infty$. If $\sigma_{j,n} = 0$, $g(0) = \delta^{n-1} (-\theta_j s_j + B \frac{s_j}{\sum_{i \neq j} \sigma_{i,n} s_i})$. According to Eq.(14), we can get $g(0) > 0$. So, the objective function have maximum value in $[0, +\infty]$. However, $\sigma_{j,n} \in [0, 1]$ and we need to decide the sign of $g(1)$. We discuss two scenarios here.

First, if $g(1) > 0$ and $g(\sigma_{j,n})$ is continuous with $\sigma_{j,n}$. So, the objective function can achieve maximum value if $\sigma_{j,n} = 1$.

Second, if $g(1) < 0$ and $g(\sigma_{j,n})$ is continuous with $\sigma_{j,n}$. So, the objective function has only one maximum value in $[0, 1]$.

Thus, we let $-\theta_j s_j + B \frac{s_j \sum_{i \neq j} \sigma_{i,n} s_i}{(\sigma_{j,n} s_j + \sum_{i \neq j} \sigma_{i,n} s_i)^2} = 0$. By solving this equation, we can obtain that the optimal choice for $\sigma_{j,n}$ in the n th global iteration, denoted by $\sigma_{j,n}^*$.

Given the used data size of other mobile devices, we can get

$$\sigma_{j,n}^* = \sqrt{\frac{B \sum_{i \neq j} \sigma_{i,n} s_i}{\theta_j s_j^2}} - \frac{\sum_{i \neq j} \sigma_{i,n} s_i}{s_j} \quad (16)$$

Summarize the two cases mentioned above and conclude the optimal choice for mobile devices.

$$\sigma_{j,n}^* = \min(1, \sqrt{\frac{B \sum_{i \neq j} \sigma_{i,n} s_i}{\theta_j s_j^2}} - \frac{\sum_{i \neq j} \sigma_{i,n} s_i}{s_j}) \quad (17)$$

Next, we consider the homogeneous mobile devices and the heterogeneous mobile devices, respectively.

Homogeneous Mobile Devices: To better comprehend the meaning of the mobile device's optimal choice, we simplify the conditions and consider that all mobile devices are homogeneous to obtain the analytical solution of the optimal strategy in each global iteration for m_j . When all mobile devices are homogeneous (i.e., they have the same θ_j to compute the unit data and s_j is the same), they have the same optimal strategy, which can be represented as follows:

$$\sum_{i \neq j} \sigma_{i,n}^* s_i = (N-1) \sigma_{j,n}^* s_j \quad (18)$$

And we can get the expression of $\sigma_{j,n}^*$

$$\sigma_{j,n}^* = \frac{(N-1)B}{N^2 \theta_j s_j} \quad (19)$$

Under this situation, every mobile device can calculate the optimal data size and achieve NE easily.

Heterogeneous Mobile devices: Moreover, we consider a practical condition that all mobile devices are heterogeneous. When all mobile devices are heterogeneous (i.e., they have different θ_i and d_j), they cannot calculate the explicit expression of $\sigma_{j,n}$ in every global iteration. Fortunately, mobile devices can obtain others' information from the consortium blockchain system. Hence, we propose a learning-based heuristic algorithm to get the optimal strategy from the perspective of mobile devices.

In the n th global iteration, m_j needs to decide the used data size in the current global iteration. And every mobile device can observe the total budget in every global iteration and other mobile devices' behavior in the past global iterations. Thus, mobile devices can calculate $\sum_{i \neq j} \sigma_{i,n}$ based on the observation and assume that other mobile devices' strategies of the current global iteration are related to the decisions made in the past global iterations. The detailed information will be shown in Alg.2.

At the beginning of every global iteration, m_j firstly takes others' average used data size in the $[n-T, n-1]$ th global iterations as the other mobile devices' behaviors in the current global iteration (line 6). Then, they use the average result to calculate their optimal strategy according to Eq.(17) (line 7). After this global iteration, mobile devices can upload their strategies to the consortium blockchain (line 8).

Algorithm 2 Learning-based heuristic algorithm

Require: $s = \{s_1, \dots, s_M\}, T, B, N$
Ensure: σ^*

- 1: **for** $j = 1 : M$ **do**
- 2: $H_j \leftarrow \phi$
- 3: **end for**
- 4: **for** $n = 1 : N$ **do**
- 5: **for** $j = 1 : M$ **do**
- 6: Calculate $\sum_{i \neq j} \sigma_{i,n}$ based on the history
- 7: $\sigma_{j,n} = \min(s_j, \sqrt{\frac{B \sum_{i \neq j} \sigma_{i,n}}{\theta_j s_j^2} - \frac{\sum_{i \neq j} \sigma_{i,n} s_i}{s_j}})$
- 8: $H_j \leftarrow H_j \cup \sigma_{j,n}$
- 9: **end for**
- 10: **end for**
- 11: **return** σ_n^*

VII. SYSTEM IMPLEMENTATION AND EVALUATION

A. Test-bed Implementation

In order to better demonstrate our system, we implement a prototype. In this section, we introduce the enabling technologies and the prototype’s system deployment and demonstrate it with several shortcuts.

1) *Test-bed Specification:* In our prototype, we choose a workstation and let a laptop and a nuc act as two mobile devices. Next, we will introduce the configuration of these computers. The workstation we choose is an HP OMEN equipped with Intel i9-12900K CPU, NVIDIA RTX3090 GPU with 24 GB VRAM, 64GB RAM, 1T SSD, and 2T HDD. Meanwhile, we choose a Dell XPS 13 as a mobile device equipped with Intel i7-6500 CPU and Intel Graphics 520 GPU, 8 GB RAM, and 512GB. As for the other mobile device, we choose an Intel NUC equipped with i7-8809G, including both Intel CPU and AMD GPU, 16 GB RAM, and 512GB.



Fig. 5. The test-bed implementation

Fig. 5 illustrates the test-bed implementation of our system. We choose a workstation to act as a requester and let the other two laptops and NUC work as mobile devices in our test-bed implementation. To ensure they can ping with each

other, we configure their networks under the same gateway. The three devices can form a peer-to-peer network and build up a consortium blockchain, where the workstation works as an observer node and mobile devices work as sealer nodes. We can simulate the interactions between the requester and mobile devices with the consortium blockchain. The requester firstly deploys a reward smart contract on the consortium blockchain. Then, mobile devices can send their strategies by calling the functions on the smart contract. Finally, the requester can call the functions to end the game and let the smart contract to compute the rewards. As shown in Fig. 5, we simulate the whole procedure on the consortium blockchain. We consider two mobile devices in our test-bed implementation, and the initial mobile devices are randomly chosen. If a new mobile device wants to join the platform, it can join the consortium blockchain and participate in the consensus as a sealer node through the peer-to-peer network.

2) *Enabling Technologies:* The test-bed implementation includes several existing technologies. For the consortium blockchain platform, we choose FISCO-BCOS⁷, an open-source consortium blockchain platform. FISCO-BCOS also provides a web page which can directly demonstrate the information of the consortium blockchain, which is shown in Fig. 6. The upper layer shows the current status of the consortium blockchain, such as block number, transaction amount, and PBFT view that records the number of times leader switch in the PBFT consensus. Meanwhile, a line chart shows the transaction amount in the last 15 days. The middle layer demonstrates the information of nodes in the consortium blockchain. Moreover, the detailed information of block and transaction is demonstrated in the lower layer.

Besides, we can also use the provided python-SDK⁸ to develop and test the situation of the consortium blockchain. As for the programming language of the smart contract, we use Solidity⁹ to write the reward smart contract.

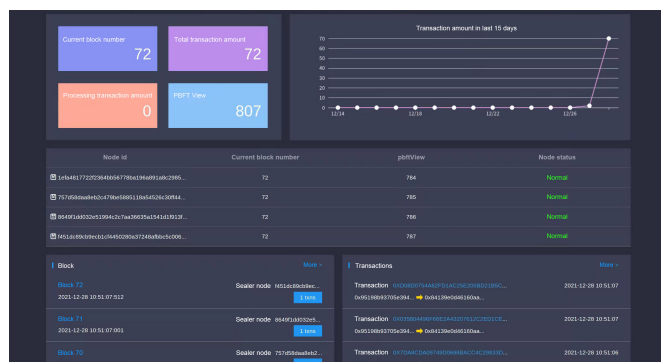


Fig. 6. The web page

3) *Smart Contract Deployment:* Initially, the requester deploys the reward smart contract in the consortium blockchain. Once it is deployed, the smart contract has a unique address, and the requester and edge nodes can call functions of the

⁷<http://fisco-bcos.org/>

⁸https://github.com/FISCO-BCOS/FISCO-BCOS-DOC/tree/release-2/docs/sdk/python_sdk

⁹<https://github.com/ethereum/solidity>

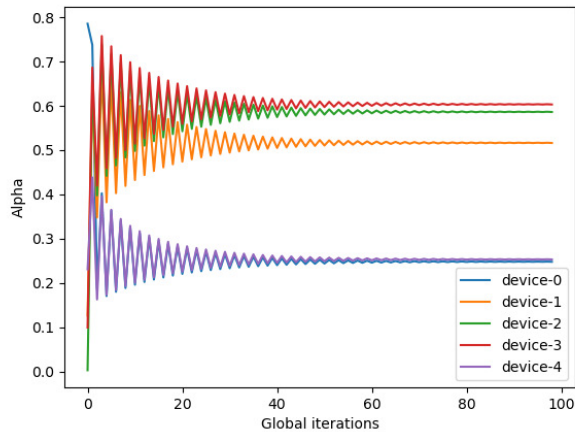


Fig. 7. The probability of mobile devices participating the current global iteration

smart contract through its address. Our proposed algorithm is implemented in the smart contract. Mobile devices can send their strategies to the smart contract by calling the corresponding functions. After all the mobile devices send their strategies, the smart contract can calculate their rewards and the results will be sent back to the requester and mobile devices.

B. Simulation Results

In this section, we show the numerical results from our evaluations to study the impacts of the incentive mechanism and the behaviors of the mobile devices in the FL task.

The simulation configurations are as follows: We consider FL scenarios at different scales, containing a requester and several mobile devices. Specifically, the number of participating mobile devices varies from 10 to 100. The budget of the requester varies from 1000 to 4000. Based on the parameter setting in the work of Fan *et al.* [28], we uniformly generate mobile devices' data size from 500 to 1500. According to the experiment conducted in [44], the value of θ_j obeys a normal distribution with a mean of 0.1 and a variance of 0.02 (i.e., $\theta_j \sim N(0.1, 0.0004)$).

1) *Simulation Results on mixed-strategy NE of DSM:* In this subsection, we use simulation results to show the convergence of DSM in mixed-strategy NE. At the beginning of the first global iteration, every mobile device that participates in the FL task chooses a probability $\alpha \in [0, 1]$. They will adjust their probability according to others' strategies until the system reaches equilibrium.

As shown in Fig. 7, we can observe that the system will achieve equilibrium before the 60th global iteration. Because of the heterogeneous characteristics, different devices have different probabilities in equilibrium states. Specifically, the probability of device-2 and device-3 to participate in every global iteration in mixed-equilibrium NE is around 0.6. And the probability of device-0 and device-4 is between 0.2 and 0.3 in equilibrium states. The simulation results are carried out 100

times, and we get the same results, which shows the dynamic game among non-cooperative mobile devices can achieve NE.

2) *Simulation Results on NE of CSM:* At the very beginning, every mobile device chooses a small data size to train the FL model. They will adjust their strategies according to others to maximize their payoff in the following global iterations. For every mobile device, if the difference between the data size used in the adjacent two global iterations is less than 10, it is considered to reach the equilibrium state.

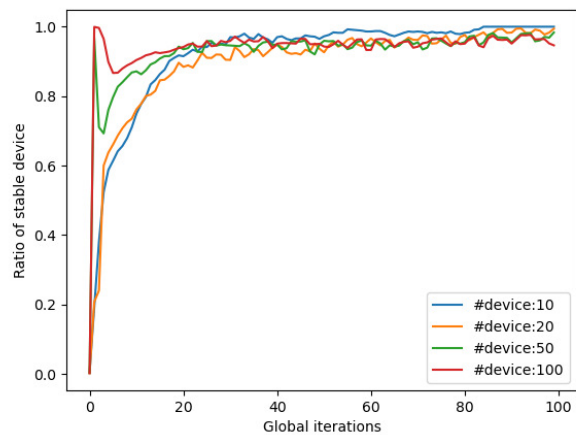


Fig. 8. Ratio of stable devices under different number of mobile devices

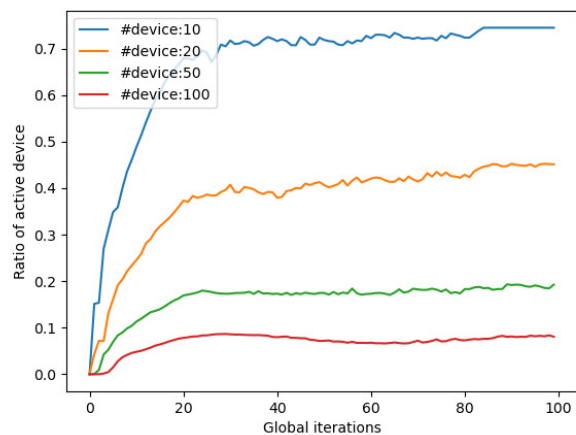


Fig. 9. Ratio of active device under different devices

Fig. 8 shows that the ratio of stable devices when the budget is fixed at 3000, and the number of devices varies from 10 to 100. We believe that the strategy of a mobile device is the same as the strategy of the last epoch, which can be considered stable. At the beginning of the FL, we notice that the ratio of stable devices increases dramatically when the number of mobile devices is 50 and 100, respectively. The reason is that when the budget is fixed, the competition becomes more intense as the numbers of mobile devices grow. Hence, most mobile devices choose to give up this task, and the ratio of stable devices is high. In this case, some mobile

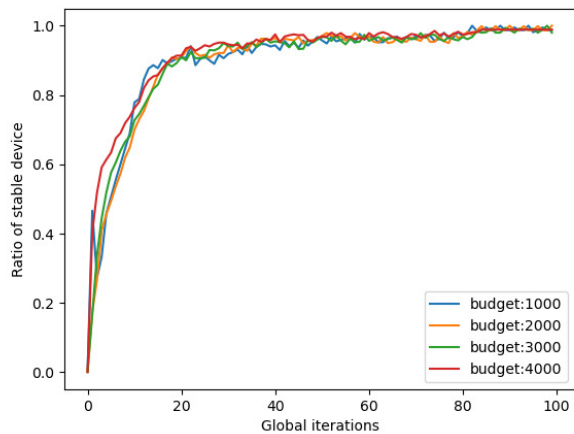


Fig. 10. Ratio of stable devices under different budget

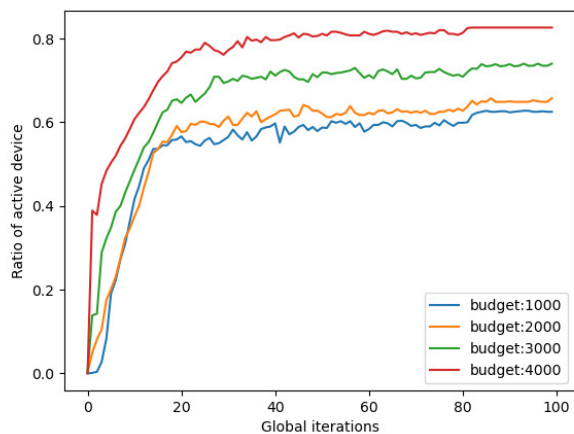


Fig. 11. Ratio of active devices under different budget

devices with small θ decide to participate in the FL task and keep changing their strategies, resulting in the ratio of stable devices declining. However, when the market is small (i.e., the number of mobile devices is 10 and 20), the ratio of stable devices increases stable. Moreover, we can observe that all mobile devices can reach equilibrium states under four different mobile devices. Specifically, more than 90 percent of mobile devices can reach equilibrium states before 20 global iterations. Before the 20th global iteration, the ratio of stable devices increased with the rising number of mobile devices. As the number of global iterations increased, the stable devices' ratio almost reached parity.

From Fig. 9, we investigate the ratio of active workers when the number of mobile devices varies from 10 to 100. Because of the intense competition and low compensation, some mobile devices may choose to give up this task during the FL training. Hence, we use active devices to refer to the mobile devices that participate in the FL tasks. (i.e., their strategies are not 0 in the current global iteration). We can observe that with the increasing number of mobile devices,

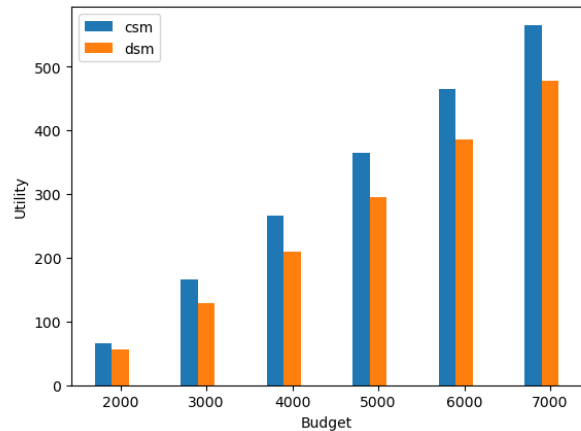


Fig. 12. The comparison of the mobile devices' average payoffs between CSM and DSM

the ratio of active devices decreases, and the total number of active mobile devices keep stable under the fixed budget. Even though the number of mobile devices in the FL becomes large, the requester cannot recruit more mobile devices under the fixed budget. Because of the intense competition, many mobile devices will give up this FL task initially.

Fig. 10 shows that the ratio of stable devices when the number of mobile devices is fixed at 10, and the budget in one global iteration varies from 1000 to 4000. From this figure, we can obtain that the ratio of stable devices increases sharply. The reason is that the small budget (i.e., $B = 1000$) is not attractive to mobile devices. Hence, most mobile devices choose to give up this task. However, some mobile devices observe others' behaviors and participate in this task. They begin to keep changing their strategies, resulting in a decrease in the ratio of stable devices. For a larger budget (i.e., $B = 3000$ and $B = 4000$), the ratio of stable devices keeps increasing until they become stable. Moreover, we can observe that the budget has little impact on the convergence speed.

From Fig. 11, we investigate the ratio of active workers when the budget varies from 1000 to 4000 with an increment of 1000. At the beginning of the FL, we notice that the ratio of active devices increases rapidly when the budget is 4000. When the system reaches an equilibrium state, we can find that the ratio of active devices rises with the increasing budget. This is because that mobile devices are more willing to participate in the FL task with more budget.

C. Simulation Results on the Comparison of DSM and CSM

Fig. 12 compares the average expected payoffs of mobile devices in DSM with the average payoffs of mobile devices in CSM when the number of devices is fixed at 10, and the budget in one global iteration varies from 2000 to 7000. This figure shows that the mobile devices' average payoff increases with the rising budget under CSM and DSM, respectively. Compared with DSM, the CSM can effectively improve the mobile devices' payoffs to 128.1 percent when the budget is 3000. That is because mobile devices' strategies in CSM

are more flexible. They can change the percentage of data used in every global iteration according to the other mobile devices' behaviors in the past global iterations. However, mobile devices in DSM can only choose to participate in or not participate in the current global iteration. If many mobile devices decide to simultaneously participate in one global iteration, their compensation will reduce. Hence, mobile devices can achieve higher average payoffs in CSM.

VIII. CONCLUSION

This paper presents and implements a consortium blockchain-based system for FL, which provides a transparent and traceable trading market. A requester coordinates mobile devices to train a shared global model in the design, with raw data stored locally. To motivate mobile devices to participate in the FL tasks, we first propose a simple and easy-to-implement incentive mechanism based on the mobile devices' used data size in the current global iteration. Moreover, we offer a dynamic game-based theoretical framework to investigate the mobile devices' behaviors based on the incentive mechanism. Specifically, we propose two kinds of strategy models, namely DSM and CSM, and analyze the mobile devices' behaviors using the dynamic game. With the consortium blockchain, the systems can record the mobile devices' behaviors and reward allocation in every global iteration of FL. Hence, every mobile device can observe other participants' past behaviors and adjust their strategies in the current global iteration. Then, we prove the existence of NE and propose two algorithms to achieve NE in two strategy models. Lastly, we use simulation results to demonstrate the two algorithms' convergence. Moreover, the CSM can effectively increase the mobile devices' payoffs to 128.1 percent at most compared with DSM.

In our future work, we aim to investigate the requester's behaviors based on mobile devices' analysis results and hope to consider the impact of the requester's strategies on mobile devices. Moreover, we will extend the scenario from one requester to multiple requesters.

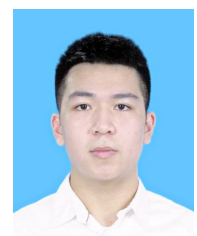
REFERENCES

- [1] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.
- [2] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6g communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, 2020.
- [3] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," *arXiv:1905.06641v2*, 2019.
- [4] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.
- [5] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.
- [6] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, 2021.
- [7] H. Zhang, S. Fan, and W. Cai, "Decentralized resource sharing platform for mobile edge computing," in *International Conference on 5G for Future Wireless Networks*, pp. 101–113, Springer, 2020.
- [8] Y. Du, Z. Wang, and V. C. M. Leung, "Blockchain-enabled edge intelligence for iot: Background, emerging trends and open issues," *Future Internet*, vol. 13, no. 2, 2021.
- [9] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus," *IEEE Network*, vol. 35, no. 1, pp. 234–241, 2021.
- [10] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. Leung, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53019–53033, 2018.
- [11] Z. Zhang, D. Dong, Y. Ma, Y. Ying, D. Jiang, K. Chen, L. Shou, and G. Chen, "Refiner: A reliable incentive-driven federated learning system powered by blockchain," *Proc. VLDB Endow.*, vol. 14, p. 2659–2662, jul 2021.
- [12] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," tech. rep., 2008. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>.
- [13] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
- [14] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching," *IEEE Internet of Things Journal*, pp. 1–1, 2020.
- [15] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*, pp. 251–260, Springer, 2002.
- [16] L. Li, J. Wang, X. Chen, and C. Z. Xu, "Multi-layer coordination for high-performance energy-efficient federated learning," in *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2020.
- [17] W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 536–550, 2021.
- [18] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, C. Miao, and D. I. Kim, "Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3640–3653, 2021.
- [19] Q. Hu, S. Wang, Z. Xiong, and X. Cheng, "Nothing wasted: Full contribution enforcement in federated edge learning," *IEEE Transactions on Mobile Computing*, 2021.
- [20] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," pp. 1175–1191, 10 2017.
- [21] J. Kang, Z. Xiong, X. Li, Y. Zhang, D. Niyato, C. Leung, and C. Miao, "Optimizing task assignment for reliable blockchain-empowered federated edge learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1910–1923, 2021.
- [22] C. Feng, B. Liu, K. Yu, S. K. Goudos, and S. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5g-enabled uavs," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3582–3592, 2022.
- [23] M. Shayan, C. Fung, C. J. Yoon, and I. Beschastnikh, "Biscotti: A blockchain system for private and secure federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 7, pp. 1513–1525, 2020.
- [24] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat, and Z. Zhao, "A blockchain based witness model for trustworthy cloud service level agreement enforcement," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1567–1575, IEEE, 2019.
- [25] H. Wang, H. Zhou, G. Yang, and T. Xiao, "Dcrowd: A decentralized service model for credible crowdsourcing based on game theory and smart contracts," in *2021 IEEE International Conference on Services Computing (SCC)*, pp. 23–33, 2021.
- [26] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Transactions on Mobile Computing*, 2020.
- [27] N. Ding, Z. Fang, and J. Huang, "Optimal contract design for efficient federated learning with multi-dimensional private information," *IEEE Journal on Selected Areas in Communications*, 2020.
- [28] S. Fan, H. Zhang, Y. Zeng, and W. Cai, "Hybrid blockchain-based resource trading system for federated learning in edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2252–2264, 2021.
- [29] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 33–39, 2018.
- [30] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. M. Leung, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53019–53033, 2018.

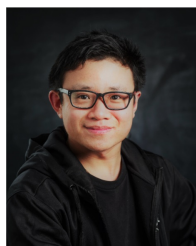
- [31] Z. Hong, Z. Wang, W. Cai, and V. Leung, "Blockchain-empowered fair computational resource sharing system in the d2d network," *Future Internet*, vol. 9, no. 4, p. 85, 2017.
- [32] M. Castro, B. Liskov, *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, pp. 173–186, 1999.
- [33] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6g communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, 2020.
- [34] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1387–1395, 2019.
- [35] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.
- [36] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10700–10714, 2019.
- [37] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1387–1395, IEEE, 2019.
- [38] D. T. Hoang, X. Lu, D. Niyato, P. Wang, D. I. Kim, and Z. Han, "Applications of repeated games in wireless networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2102–2135, 2015.
- [39] J. Von Neumann and O. Morgenstern, "Theory of games and economic behavior, 2nd rev," 1947.
- [40] S. Li and J. Huang, "Price differentiation for communication networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 703–716, 2014.
- [41] D. Fudenberg, F. Drew, D. K. Levine, and D. K. Levine, *The theory of learning in games*, vol. 2. MIT press, 1998.
- [42] Q. Ma, L. Gao, Y. Liu, and J. Huang, "Economic analysis of crowd-sourced wireless community networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1856–1869, 2017.
- [43] G. Debreu, "A social equilibrium existence theorem," *Proceedings of the National Academy of Sciences*, vol. 38, no. 10, pp. 886–893, 1952.
- [44] Y. Zou, S. Feng, D. Niyato, Y. Jiao, S. Gong, and W. Cheng, "Mobile device training strategies in federated learning: An evolutionary game approach," in *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 874–879, IEEE, 2019.



Sizheng Fan [S'20] received the B.Eng. degree in Automation from Beijing Institute of Technology, China in 2018. He is currently working towards Ph.D. in Computer and Information Engineering at The Chinese University of Hong Kong, Shenzhen, China. He is working as an Research Assistant in Human-Cloud Systems Laboratory. His current research interests include blockchain, game theory and crowdsourcing. He is a student member of the IEEE and the CCF.



Hongbo Zhang received the B.Eng. degree in Computer Science and Engineering from The Chinese University of Hong Kong, Shenzhen, China in 2020. He is currently working towards M.Phil. with the School of Science and Engineering in The Chinese University of Hong Kong, Shenzhen, China. He is working as a Research Assistant in Human-Cloud Systems Laboratory. His current research interests include blockchain, game theory, machine learning and edge computing.



Zehua Wang [S'11-M'17] received his Ph.D. degree from The University of British Columbia (UBC), Vancouver in 2016 and was a Postdoctoral Research Fellow in the Wireless Networks and Mobile Systems (WiNMoS) Laboratory directed by Prof. Victor C.M. Leung from Feb. 2017 to Aug. 2018. Before that, he received his bachelor's and master's degrees from Wuhan University and Memorial University, respectively. He is now an Adjunct Professor in the Department of Electrical and Computer Engineering at UBC, Vancouver and the CTO at Intellium Technology Inc., BC, Canada. He was a recipient of the Four Year Doctoral Fellowship at UBC from 2012 to 2016 and the Graduate Support Initiative Awards at UBC in 2014 and 2015. He received the Chinese Government Award for Outstanding Self-Financed Students Abroad in 2015. He is an Editor of *Wireless Networks* and served as the guest editors for a few special issues in top journals including *IEEE Access* and *ACMSpringer Mobile Networks & Applications*.



Wei Cai [S'12-M'16] received the B.Eng. degree in Software Engineering from Xiamen University, China in 2008, the M.S. degree in Electrical Engineering and Computer Science from Seoul National University, Korea, in 2011, and the Ph.D. degree in Electrical and Computer Engineering from The University of British Columbia (UBC), Vancouver, Canada, in 2016. From 2016 to 2018, he was a Postdoctoral Research Fellow with UBC. He is currently an assistant professor of Computer Engineering, School of Science and Engineering at The Chinese University of Hong Kong, Shenzhen. He is serving as the director of the CUHK(SZ)-White Matrix Joint Metaverse Laboratory. Dr. Cai has co-authored more than 70 journal and conference papers in the area of interactive multimedia and distributed/decentralized systems. His recent research interests are mainly in the topic of human-centered computing for metaverse, including blockchain, digital game, Web 3.0, and computational art. He serves as a Technical Program Committee member in ACM MM, MMSys, and NOSSDAV. He is an associate editor of *IEEE Transactions on Cloud Computing*. He was a recipient of the 2015 Chinese Government Award for the Outstanding Self-Financed Students Abroad, the UBC Doctoral Four-Year-Fellowship from 2011 to 2015, and the Brain Korea 21 Scholarship. He also received the best student paper award from ACM BSC12019 and the best paper awards from CCF CBC2018, IEEE CloudCom2014, SmartComp2014, and CloudComp2013.