Hybrid Blockchain-Based Resource Trading System for Federated Learning in Edge Computing

Sizheng Fan, *Graduate Student Member, IEEE*, Hongbo Zhang, Yuchen Zeng, and Wei Cai, *Member, IEEE*

***Abstract*—By training a machine learning algorithm across multiple decentralized edge nodes, federated learning (FL) ensures the privacy of the data generated by the massive Internet-of-Things (IoT) devices. To economically encourage the participation of heterogeneous edge nodes, a transparent and decentralized trading platform is needed to establish a fair market among distinct edge companies. In this article, we propose a hybrid blockchain-based resource trading system that combines the advantages of both public and consortium blockchains. We design and implement a smart contract to facilitate an automatic, autonomous, and auditable rational reverse auction mechanism among edge nodes. Moreover, we leverage the payment channel technique to enable credible, fast, low-cost, and high-frequency payment transactions between requesters and edge nodes. Simulation results show that the proposed reverse auction mechanism can achieve the properties, including budget feasibility, truthfulness, and computational efficiency.**

***Index Terms*—Auction, blockchain, edge computing, Internet of Things (IoT), trade market.**

I. Introduction

ALONG with the rise of the Internet of Things (IoT) and social networking applications, the data used for machine learning are generated at the exponential growth rate at multiple decentralized edge nodes, including end devices, such as smartphones, laptops, different kinds of sensors, etc., [1]. To improve the intelligence of IoT applications, it is inevitable to use machine learning to train the models using the data stored in various kinds of IoT devices. The traditional machine learning is to gather the data at these edge nodes and make them the centralized massive data sets to be used for the model training at some powerful computing platforms such as a cloud data center [2]. The process of moving the distributed data to a central server will not only have efficiency issues but also cause privacy and safety concerns.

Fortunately, federated learning (FL) in edge computing provides a promising solution to such a problem [3], [4]. FL can distribute machine learning tasks to the edge nodes and leverage different edge nodes to train the shared model collaboratively. Since the raw data used for model training are stored at the local edge nodes themselves instead of the cloud, the protection for users' privacy is enhanced. In this article, we call the users who need model training as requesters. Edge nodes can provide model training services for the requesters by collecting the data from IoT devices such as laptops, smartphones, smartwatches.

Many current studies toward FL put their main focus on improving the performance [5]–[8], and solving privacy and safety problems [9], [10]. Among these studies, a pivotal and fundamental assumption is that the edge nodes are voluntarily participating in computing with no intention to ask for any returns, which is one type of volunteer computing [11]. However, these assumptions are impractical in reality, because training models at edge nodes will consume various resources, such as electricity, bandwidth, and computing resources. To this end, it is more reasonable to compensate the edge nodes when the requesters want to use the edge resources.

However, to determine the trading price of the resource is a crucial challenge. It is unfair for the requesters if the edge providers solely determine the price. The requesters need a fair and transparent algorithm for pricing in transactions. Meanwhile, there is a conflict of interest among different edge providers, who also seek for a fair selection algorithm to determine the edge nodes to provide the service for the requesters. The auction, as a mechanism, can not only help the requester to recruit workers but also facilitate edge nodes to participate in the FL tasks, which can solve the mentioned problems. Compared with other game theory such as the Stackelberg game, the auction can motivate the edge nodes to report their bids honestly, which can help the requester understand the information of the sellers and achieve a desirable result. The state-of-the-art work proposes to use the auction as a method of resource allocation [12], [13], which employs a centralized third party to charge fees and assign tasks, respectively. However, an untrustworthy third party may collude with certain edge nodes to make extra profits, which brings the unfairness to other edge nodes. To economically encourage these heterogeneous edge nodes that are deployed by distinct companies and individuals, we choose blockchain to offer them a decentralized and transparent resource trading system. Blockchain, as a decentralized trading system, provides a credible, faster, and transparent environment for requesters and edge nodes. By implementing the auction mechanism in the smart contract, we can incent requesters and edge nodes for FL in edge computing.

Manuscript received April 16, 2020; revised July 23, 2020 and August 10, 2020; accepted September 19, 2020. Date of publication October 14, 2020; date of current version February 4, 2021. This work was supported in part by the National Natural Science Foundation of China under Project 61902333; and in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society. *(Corresponding author: Wei Cai.)*

The authors are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Shenzhen 518172, China, and Shenzhen Institute of Artificial of Intelligence and Robotics for Society, Shenzhen, China (e-mail: sizhengfan@link.cuhk.edu.cn; hongbozhang@link.cuhk.edu.cn; yuchenzeng@link.cuhk.edu.cn; caiwei@cuhk.edu.cn).

Digital Object Identifier 10.1109/JIOT.2020.3028101

In this article, we fill the blank by proposing a hybrid blockchain-based resource trading system that combines the advantages of both public and consortium blockchains. The public blockchain, e.g., BitCoin [14], is also referred to as permissionless blockchain since the consensus is reached by public participants through the Proof-of-Work (PoW) puzzle competition. Besides, deploying and calling smart contracts on the public blockchain, e.g., Ethereum [15], requires paying for a high gas fee. Gas fees are payments made by users to compensate miners for the computing energy required to validate and pack transactions into blocks on the Ethereum. In the Ethereum, the miners need to carry out the PoW consensus algorithm to pack the transactions into blocks. The PoW consensus algorithm requires miners to compute for a computationally intensive puzzle, which would cost miners a lot of computing energy. Hence, users need to pay for a high gas fee to compensate miners when deploying or calling smart contracts on the public blockchain. In contrast, the consortium blockchain is a type of permissioned blockchain where the consensus is achieved by a set of preauthorized nodes only [16]. Due to the intrinsic differences in the consensus model, the consortium blockchain outperforms the public blockchain in terms of efficiency, scalability, and privacy, while the public blockchain achieves a higher degree of decentralization and higher credibility among a larger group of users. According to these features, we employ the consortium blockchain as the decentralized platform for auctions among requesters and edge nodes, because the bidding requires low-cost, high-frequency, and low-latency operations on smart contracts [17], while the transparency and audibility are only required within the group of stakeholders, who are the requesters and edge providers in this context. Besides, the consortium blockchain will record the useful information, such as the winners and reward allocation in the auction. Under this circumstance, the edge nodes will also serve as the consensus nodes in the consortium blockchain, since they are the stakeholders on the consensus over the auction smart contract deployed by the requesters. On the other hand, we adopt the public blockchain as our payment solution, since the digital currency issued by the public blockchain is better accepted by the general public. To overcome the performance issues of the public blockchain, we also integrate the payment channel technique to provide users with credible, faster, lower cost, and higher frequency transactions using multiple off-chain micropayment transactions. Another important component of the proposed trading platform is an incentive mechanism that can motivate the participation of edge nodes. In this work, we design and implement a smart contract-based rational reverse auction mechanism to validate the capability of maximizing the valuation of the requester in the platform. The major contributions of this article are shown as follows.

1) *System Design and Implementation:* We present the very first opensource implementation of a resource trading system for FL in edge computing. In the hybrid-blockchain framework, the cryptocurrency in the consortium blockchain is not accepted. Hence, we use the public blockchain to solve it. We leverage the payment channel technique to solve the performance problem in the public blockchain, enabling credible, fast, low-cost, high-frequency payment transactions among data requesters and edge companies. Besides, we let requesters and edge nodes interact with the public and consortium blockchains separately instead of cross-blockchain interaction, which is still an unmatured technology in state-of-the-art works.
2) *Smart Contract-Based Incentive Mechanism:* We design a data quality-driven reverse auction (DQDRA) and implement it in the smart contract to facilitate automatic, autonomous, and auditable auctions among edge nodes. We prove that DQDRA satisfies budget feasibility, individual rationality (IR), truthfulness, and computational efficiency.

The remainder of this article is organized as follows. We review the related work in Section II and illustrate the system overview in Section III. Then, in Section IV, we model and formulate the target problem. The mechanism design and the analysis on the auction model are presented in Sections V and VI, respectively. Afterward, the testbed implementation and simulation results are shown in Section VII. Section VIII concludes this article.

II. Related Work

A. Edge and Blockchain for FL

As the studies showing that more than 90% of data will be stored and processed locally in the near future, people are paying increasing attention recently to FL in edge computing and put more studies focus on the usage of the resources and privacy issue [1], [18]. Since edge computing has a limited computation and communication resources for the optimal learning performance, Wang *et al.* [1] proposed an algorithm to determine the frequency of global aggregation to ensure that the available resource is efficiently used. Qian *et al.* [18] emphasized the necessity to design an optimal service placement scheme to be placed on the edge nodes and raised the problem that the existing placement strategies do not care about the user's privacy. Therefore, they proposed a privacy-aware service placement (PSP) to solve the problems.

Besides, many works have put blockchain into consideration to solve the concerns on the security and privacy of FL using edge computing. Shayan *et al.* [19] proposed a decentralized public peer-to-peer (P2P) system named Biscotti that co-designs a privacy-preserving FL process using blockchain. The task publisher can detect the poisoning attack by comparing the influence with and without the model update on a database. Once the performance of the model update on the database reduces beyond a predefined threshold, the task published will reject the local model update. Fung *et al.* proposed the FoolsGold scheme to identify malicious edge nodes based on the principle that honest edge nodes can be separated from malicious edge nodes by the diversity of their gradient updates. The malicious edge nodes can be detected with the cosine similarity method as their gradient updates are more similar to each other than honest edge nodes [20].

B. Incentive Mechanism for FL

For the FL coupling with edge computing, it is not practical in the real world to make all the edge nodes voluntarily

participate in the FL because model training will incur costs [21]. Therefore, many research efforts are devoted to the development of various incentive mechanisms to help motivate edge nodes to participate in FL.

Kang *et al.* introduced reputation as the metric to measure the reliability of the mobile devices and proposed a contract theory to incent mobile devices to participate in the FL task. The task publisher needs to design different contracts for different types of workers, and mobile devices can only choose the contract matching their own types. The numerical result shows that this scheme is efficient in selecting reliable mobile devices and can improve the efficiency of FL [22]. Zhan *et al.* proposed a two-stage Stackelberg game model as the incentive mechanism for FL. The server announces a total reward while each edge node determines its training strategy to maximize its own utility. Then, the authors propose the deep reinforcement learning-based incentive mechanism [23]. However, both papers adopt the assumption that all the training data in each edge node have the same quality and are independently and identically distributed (IID), which is actually impossible.

C. Payment Channel

There are severe scalability problems of cryptocurrencies on blockchains because it takes a long time to write all backlogged transactions into a blockchain if there is a burst of transactions. The payment channel can solve such a problem, which is studied and utilized by many researchers [24], [25].

There are two kinds of payment channels that are unidirectional payment channel and bidirectional payment channel [16]. In the unidirectional payment channel, which only allows the single directional transaction, it is the payer's responsibility to create the payment channel and lock some deposit in it. The payer's signature will be sent to the payee. The payee can get paid whenever he/she wants to by sending the signature to the blockchain, and the payment channel is closed. In the bidirectional payment channel, both payer and payee can pay the other by signing a signature, and sending the signature to the other party with the prerequisite that both of them need to contribute to the deposit.

However, there is a chance that someone can cheat for some money if he/she is a liar. This issue can also be solved by using the timestamp and the challenging period. When one party finds that the other is going to close the payment channel dishonestly, he/she can continuously broadcast the other's signature and update the splitting plan until the end of the challenging period. When allowing both parties to close the payment channel, the splitting plan is locked after the challenging period, and either of them can finally close the payment channel. The cryptocurrency flows to each one's wallet.

III. System Overview

In this section, we first introduce the system users in the hybrid blockchain-based resource trading system. Then, we introduce the design of the system, including edge-based FL and hybrid blockchain system. Finally, we discuss the transaction process in the system.

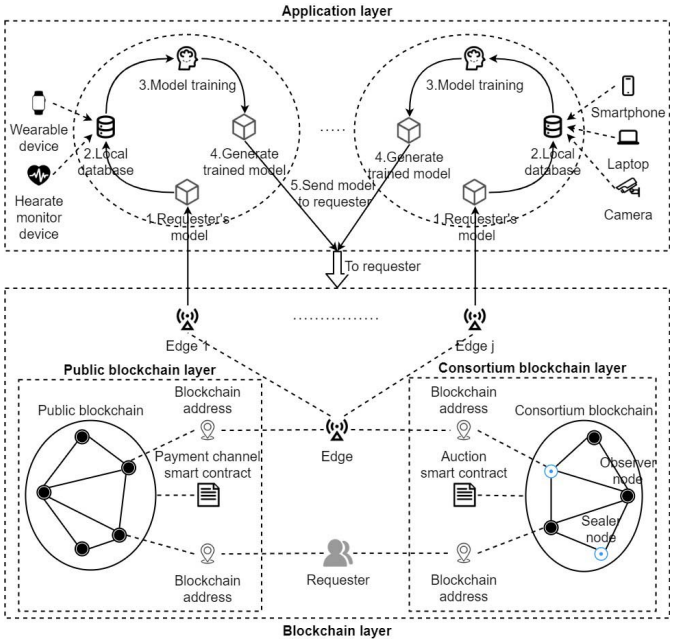

Fig. 1. Hybrid blockchain system.

A. System Users

In this section, we introduce the system users in the hybrid blockchain-based resource trading system, including requesters, edge nodes, and IoT devices. The detailed introductions are as follows.

1) *Requesters:* Requesters are individuals or companies who need model training. They send task requirements, including task types and budgets, to the blockchain system. Then, after paying to edge nodes via the blockchain system, they can get the results from edge nodes.
2) *Edge Nodes:* Edge nodes equipped with storage and computing capacity act as service providers in our system. They perform a local model update using their data set for serving FL and are incentivized via payment from requesters. Additionally, they collect data from IoT devices and pay them according to data usage.
3) *IoT Devices:* IoT devices, including laptops, smartphones, smartwatches, etc., upload a set of data into edge nodes to earn money from requesters that request their data.

B. Edge-Based FL

FL is a favorable and superior distributed privacy-preserving machine learning technique. It can make multiple edge nodes capable of collaboratively training a global shared model without the necessity of uploading private data that are stored locally to a central server. As shown in Fig. 1, we consider a wireless communication infrastructure with a set of edge nodes, which are deployed by different companies in the application layer. These edge nodes equipped with enough computational capacity and storage can receive and store different kinds of data, such as application data and sensing data from a variety of devices. In order to protect privacy, the

requester with the FL task can train the deep learning model on the data of these edge nodes without data collection. The requester first sends the model to all participating edge nodes (step 1). Every edge node will minimize the training loss over its local data in several iterations, and the number of iterations is decided by the requester (steps 2–4). After that, each participating edge node will send the new weight and the gradient to the requester to update the global model (step 5).

C. Hybrid Blockchain System

In this section, we introduce the design of our hybrid blockchain system, which is composed of the public and consortium blockchains.

1) Consortium Blockchain: Compared with the public blockchain, the consortium blockchain achieves a higher TPS and eliminates the cost of the gas fee due to the narrowing of consensus scope. In the consortium blockchain, there are two kinds of nodes that are sealer nodes and observer nodes. Both of them can send transactions on the consortium blockchain. However, only sealer nodes can join the consensus process.

As a method to carry out transactions, the auction smart contract is repeatedly deployed and called by the requesters and the edge nodes. Requesters deploy the auction smart contract on the blockchain as the auction sponsor, and edge nodes participate in the auction by calling the smart contract. However, deploying and calling the smart contract on the public blockchain requires paying a high gas fee to the consensus node. The problems that the public blockchain has a low performance may also lead to the inefficiency of the auction smart contract. Xiong *et al.* [26], [27] try to use edge computing to solve the limited computing resources problem in lightweight devices. However, they still cannot solve the high gas fee problem in public blockchain. Hence, we provide requesters with the consortium blockchain to deploy the auction smart contract.

However, the selection of sealer nodes needs to be carefully considered. If the sealer nodes are not selected correctly, the data blocks and smart contracts on the consortium blockchain are at risk of being tampered with. Therefore, the selection of sealer nodes is essential to ensure the success of the deployment of the auction smart contracts on the consortium blockchain. Since the auction smart contract only involves the interests of both parties in the transaction, it is acceptable as long as both parties in the transaction reach a consensus. As the transaction initiator, the requesters deploy the auction smart contract. The deployed smart contract needs to be co-authenticated by the participants, edge nodes. The edge nodes participate in the consensus verification as sealer nodes. Because the algorithm of the auction smart contract is fixed and there is a competitive relationship among the edge nodes in the auction, the tampering of a few malicious nodes will not be agreed by other edge nodes. Moreover, the smart contract is open and transparent. If the smart contract is tampered with, the requester can immediately detect the tampering and terminate the smart contract. Most edge nodes tend to maintain the fair selection algorithm rather than maliciously tamper with it to prevent the requester from quitting the transaction and lose what they should have gained. In addition, the newly added sealer nodes also need to achieve consensus verification from the existing sealer nodes. Through the practical Byzantine fault tolerance (PBFT) consensus process, the new sealer node is allowed to join if more than two-thirds of the sealer nodes agree.

In addition to providing a platform for the auction smart contract, the consortium blockchain can also store transaction records. At the end of the transaction, the transaction record is signed by both parties and verified by the sealer nodes. Because the transaction record has the requester's signature, the sealer nodes cannot tamper with it.

2) Public Blockchain: Compared with the consortium blockchain, the public blockchain has a higher degree of decentralization and higher credibility. Hence, the digital currency on the public blockchain is accepted by most people.

If the payment procedures are also deployed in the consortium blockchain, the liquidity of the digital currency may have issues. Although the digital currency on the consortium blockchain is recognized by the members of the consortium, it is not directly equivalent to the common currency for the others. Therefore, we deploy the payment procedures in the public blockchain where the digital currency is accepted by most people.

However, on a decentralized public blockchain, due to the lack of a centralized third party that endorses the credit, there is no guarantee that the edge nodes will still provide services after receiving the requesters' payments. To solve this problem, we ask the requesters to make micropayments several times according to the iteration number with a fixed interval during the model training. By breaking large transactions into micropayment transactions, the risk of each transaction can be reduced. However, due to the low performance of the public blockchain, it takes a longer time to confirm a large number of transactions with micropayments on the public blockchain, which leads to the reduction of transaction efficiency. Therefore, we implement the payment channel technique using the smart contract to provide requesters and edge nodes with faster transactions using multiple off-chain micropayment transactions and provide them with the public blockchain to deploy the payment channel smart contract.

The payment channel technique reasonably designs the off-chain transactions to ensure the security of the off-chain transactions for both parties [16]. We briefly introduce the design of the payment channel smart contract to keep the transaction running properly.

In the payment channel smart contract, the requirement to withdraw the money in a transaction is signed by the requester and the edge node. Before the requester deposits the digital currency into the payment channel smart contract, it will request a refund transaction (RT) from the edge nodes to make his money available. To avoid the requester's double-spending, the locktime of RT is greater than that of the following commitment transaction (CT) and settlement transaction (ST). With RT, the requester can safely deposit the digital currency into the smart contract. After that, if the requester needs to transfer money to the edge nodes, it can simply send a CT to the edge nodes. CT indicates the amount that the requester transfers to the edge nodes and is signed by the requester. The

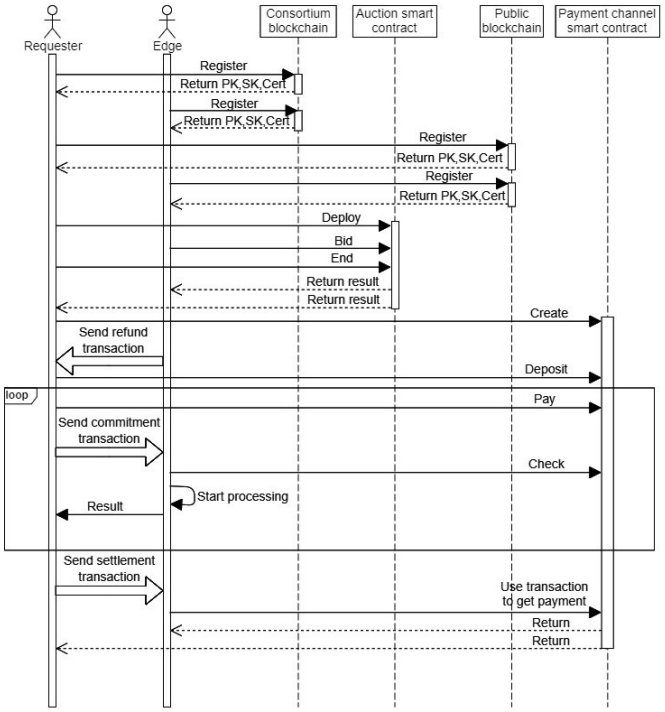

Fig. 2. Sequence diagram of the transaction process.

edge nodes can use the transaction to withdraw money from the smart contract simply by signing it. We call the transaction in which the money is last withdrawn as ST. To avoid the requester's double-spending, the locktime of CTs gradually decreases and is greater than that of ST.

D. Transaction Process

Fig. 2 illustrates the sequence diagram of the transaction process in our hybrid blockchain-based resource trading system. In this sequence diagram, we use single line arrows to represent on-chain transactions and use bold line arrows to represent off-chain transactions.

System Initialization: First, the requester and edge nodes register on the public and consortium blockchains, respectively. Each of them has its address, a pair of asymmetric keys (public/private), and a certificate.

Requesting and Biding: If a requester wants to request a model training service, he first will create an auction smart contract that contains his task requirements, including task types, budget for one global iteration in the FL task, etc. Then, the requester deploys the auction smart contract on the consortium blockchain, and all edge nodes are informed by the deployed auction smart contract. After receiving the requirement of the FL task, the edge nodes need to decide whether to attend this task or not. Once an edge node decides to take part in this auction, it needs to call the auction smart contract and send his bid, including bid price, and the information of its data set. After receiving the bid from all the edge nodes, the auction smart contract would select a set of winners according to the bid of edge nodes, and determine the payment to every winner.

Evaluating and Paying for FL Tasks: After receiving the weights and gradients, the requester carries out the quality evaluation of the model updates from all participating edge nodes using some mechanisms. Two schemes are used frequently in this procedure to identify unreliable edge nodes in FL. One is the reject on negative influence (RONI) for IID data scenarios, and the other is the FoolsGold mechanism, aiming for non-IID data scenarios [28], [29]. The requester can take the advantages of these useful schemes to effectively remove the unreliable local model updates from the unreliable edge nodes, and refuse to pay them. After evaluating the performance, the requester sends the payment to the winners. First, the requester deploys a payment channel smart contract to every reliable winner. Before depositing the digital currencies into those smart contracts, the requester asks the edge nodes to send an RT. Then, the requester can pay for the computing task to the reliable winners via the payment channel. Besides, the requester also sends the off-chain CTs to the reliable winners directly. After checking the payment channel, winners start to process data according to the requirements of the requester. The payment and processing procedures keep looping until the training process completes. When the training processes complete, the requester generates STs for the winners and sign with its signatures. The winner can verify if the information in the transaction record is correct. If so, the winner signs the transaction with his signature and uploads it to the consortium blockchain. If edge nodes want to withdraw digital currencies in the payment channel smart contract, they only need to end the payment channel smart contract by using the ST, and the smart contract will return the digital currencies to both parties based on the balance.

Consensus Process: In the consortium blockchain, we use the PBFT consensus algorithm for the consensus process [30]. Compared to other consensus algorithms, the PBFT algorithm has benefits, such as low latency, high efficiency, and high stability. With low latency and high efficiency, the consortium blockchain can satisfy the demands of high-frequency transactions. Without the loss of generality, the block producer of a new block rotates among consensus nodes (edge nodes) to guarantee fairness. When the blocks that include their transactions are verified and uploaded into the blockchain, the transactions are finished.

IV. Auction Model and Problem Formulation

Based on our system design, we have an auction implemented by a smart contract on the consortium blockchain to encourage both the requester and edge nodes to participate in our system. In this section, we discuss the auction model and formulate the FL training problem as an auction. Moreover, Table I summarizes some significant notations used in the model.

A. Data Quality Metrics

The evaluation of data quality is significant for requesters when they are selecting edge nodes. In this section, we consider two reasonable data quality metrics.

TABLE I
KEY NOTATIONS

Symbol	Definition	Symbol	Definition	Symbol	Definition
$\mathcal{N}$	the set of edge nodes	$\mathcal{D}$	the set of dataset of $\mathcal{N}$	n_j	the jth edge node
D_j	the dataset of n_j	σ_j	the EMD of n_j	c_j	the total cost of n_j
$\mathcal{P}_j(y=i)$	the distribution of class i in n_j	$\mathcal{P}_a(y=i)$	the reference distribution of class i	k	the number of local iteration in one global iteration
c_j^p	the computing cost of n_j	ζ	the size of the local model	c_j^d	the data cost of n_j
c_j^c	the communication cost of n_j	r_j	the number of CPU cycles of n_j	h_j	the channel gain between n_j and the requester
β_j	the effective capacitance parameter of computing chipset for edge node n_j	ρ_j	the transmission power between n_j and requester	f_j	the number of CPU frequency of n_j
x_j	the flag showing whether n_j is selected	u_j	the utility of n_j in one global iteration	$\Phi(\cdot)$	the total valuation of requester
N_0	the background noise	B	the budget of requester	b_j	the bid price of n_j
$\mathcal{N}_{k-1}^*$	the winner set in the first $k-1$ iterations	$\widetilde{\mathcal{N}}^*$	the winner set over $\widetilde{\mathcal{N}}_{-j}$	$\widetilde{\mathcal{N}}$, $\widetilde{\mathcal{N}}_{-j}$	the edge nodes set except n_j
$Q_{j\|\mathcal{N}_{k-1}^*}$	the gain contribution of the kth iteration if n_j is selected by the platform	$\alpha_{k,l}$	a bid such that n_k can substitute n_l to win in the lth loop over the new set $\widetilde{\mathcal{N}}$	$\beta_{k,l}$	the highest bid such that n_k can substitute n_l to win

Data Size Metric: For data quality metric, the requesters want more data in the FL model within his/her budget as more data means a better performance [31]. The experiment conducted in [23] was to measure the model accuracy under different amounts of the training data. The test accuracy of the training model can be regarded as a concave function.

IID: The data distribution is another significant index to evaluate the quality of the data. We usually assume that the training data are IID [32]. But actually, the local data are user specific and usually non-IID in FL. The Earth mover's distance (EMD) metric [33] provides a method to qualify the influence of nonindependence and nonidentity on the data. We consider a K classification problem defined over a compact $\mathcal{X}$ and a label space $\mathcal{Y} = [K]$, where $[K] = 1, \ldots, K$. The edge node n_j's training data set $D_j = \{\boldsymbol{x_j}, y_j\}$ distributes over $\mathcal{X} \times \mathcal{Y}$ following the distribution $\mathcal{P}_j$. Thus, the EMD σ_j of D_j can be calculated by [34]

$$\sigma_j = \sum_{i=1}^{K} \left\| \mathcal{P}_j(y=i) - \mathcal{P}_a(y=i) \right\| \tag{1}$$

where $\mathcal{P}_a$ refers to the reference distribution, which is the public knowledge of all users, including requesters and edge nodes, and $\mathcal{P}_j(y=i)$ denotes the probability for the ith class in D_j.

B. Auction Model

We consider a wireless edge-based FL scenario. N edge nodes, where $\mathcal{N} = \{n_1, \ldots, n_j, \ldots, n_N\}$, that have the computing capacity and local data sets $\mathcal{D} = \{D_1, \ldots, D_j, \ldots, D_N\}$ are deployed to provide FL services for the requester. We denote d_j to represent the data size of edge node j, and the total data size of all participating edge nodes is $\sum_{j=1}^{N} d_j = D$. Given the FL task that deals with the information announced by the platform, each edge node decides whether it takes part in this task or not and submits multidimensional information, revealing their intended data size, the EMD metrics, and bid prices.

Denote n_j's total cost for the FL task as c_j. $c_j = c_j^d + c_j^p + c_j^c$ is composed of three parts, n_j's cost for using these data, namely, data cost denoted as c_j^d, the cost for computing data, namely, computing cost denoted as c_j^p, and the cost for uploading the weights and gradients to requesters, namely, communication cost denoted as c_j^c. For each edge node, the data cost c_j^d is mainly from data usage in every local iteration, and is closely related to the data size d_j. Therefore, c_j^d in one global iteration of the FL task can be calculated as follows:

$$c_j^d = k\alpha_j d_j \tag{2}$$

where $\alpha_j > 0$ means the cost of unit data usage in one local iteration of the FL task and k refers to the number of local iterations in one global iteration. The computation cost of edge node n_j for one global iteration is mainly from the CPU energy consumption, which can be written as follows [5]:

$$c_j^p = k\gamma_j r_j d_j f_j^2 \tag{3}$$

where γ_j refers to the effective capacitance parameter of computing chipset for edge node n_j, r_j means the number of CPU cycles for an edge node n_j to perform one sample of data in local model training, and f_j is the CPU frequency of edge node n_j. After a global iteration, all participating edge nodes send their model parameters through the wireless network to the requester. The energy consumption of communication can be written as follows [5]:

$$c_j^c = \frac{\zeta \cdot \rho_j}{W \ln\left(1 + \frac{\rho_j h_j}{N_0}\right)} \tag{4}$$

where ζ is a constant which is the size of the local model update with the same value for all edge nodes. W refers to the transmission bandwidth and ρ_j means the transmission power of the edge node n_j. h_j is the channel gain of the P2P link

between edge node n_j and the requester. N_0 is the background noise. The utility of edge nodes in one global iteration of the FL task can be defined as

$$u_j = \begin{cases} p_j - c_j, & \text{if edge node } n_j \text{ is a winner} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where p_j means the reward from the requester to edge node n_j. If an edge node is not selected by the platform, its payment and utility are both 0. We assume that each edge node is strategic and aims to maximize its own utility. However, with truthfulness, the dominant strategy of edge nodes is to submit their true cost.

For requesters, we adopt the assumption that the performance of the FL model is related to the data size and the EMD metric. According to the experiment in [33] and [35], if the EMD metric $\sigma_j < 1$, it does not have a significant influence on FL performance. Thus, we limit σ_i in $[0, \sigma_{\max}]$ to guarantee the performance of the model, and assume that the valuation of the requester is solely related to the data size when $\sigma_j < \sigma_{\max}$.

The experiment conducted in [23] illustrates the relationship between the model accuracy and the training data size. Thus, the formula can be written as follows:

$$R(n_j) = \lambda g(d) \tag{6}$$

where $\lambda > 0$ is a system parameter and $g(d)$ is a concave function with respect to the amount of total training data, which means that the marginal value of data decreases as the increasing number of data in the FL task. In this article, let $g(d) = \sqrt{d}$ and $g(0) = 0$. Thus, the total valuation of the requester can be defined as

$$\Phi(\mathcal{N}) = \lambda \sqrt{\sum_{n_j \in \mathcal{N}} \left(\min\{1, x_j\} \cdot d_j\right)} \tag{7}$$

where x_j refers to the flag showing whether edge node n_j is selected by the system or not. The details are as follows:

$$x_j = \begin{cases} 1, & \text{if edge node } n_j \text{ is selected} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

C. Problem Formulation

In this section, we adopt the assumption that the platform aims to maximize the requester's valuation with a limited budget B

$$\begin{aligned} \max \ & \lambda \sqrt{\sum_{n_j \in \mathcal{N}} \left(\min\{1, x_j\} \cdot d_j\right)} \\ \text{s.t.} \ & \sum_{n_j \in \mathcal{N}^*} p_j \le B \\ & x_j \in \{0, 1\}. \end{aligned}$$

First, we define submodular functions, and all the results of this article are based on it. The submodular function is defined as follows.

Definition 1 (Submodular Function): For a finite set γ, function $f : 2^{\gamma} \to \mathcal{R}$ is submodular if

$$f(\mathcal{C} \cup i) - f(\mathcal{C}) \ge f(\mathcal{Z} \cup i) - f(\mathcal{Z})$$

for any $\mathcal{C} \subseteq \mathcal{Z} \subseteq \gamma$ and $y \in \gamma \backslash \mathcal{Z}$. Moreover, a submodular function f is nondecreasing if $f(\mathcal{C}) \le f(\mathcal{Z})$ for any $\mathcal{C} \subseteq \mathcal{D}$.

Lemma 1: The revenue $\Phi(\mathcal{N})$ in (7) is a nonnegative, nondecreasing submodular function.

Proof: In order to proof Lemma 1, we need to show that

$$\Phi(\mathcal{C} \cup i) - \Phi(\mathcal{C}) \ge f(\mathcal{Z} \cup i) - f(\mathcal{Z})$$

for any $\mathcal{C} \subseteq \mathcal{Z} \subseteq \mathcal{N}$ and $i \in \mathcal{N} \backslash \mathcal{Z}$. According to (5), we can get the following.

Given sets $\mathcal{C}$, $\mathcal{Z}$, and i, we can have

$$\begin{aligned} \Phi(\mathcal{C} \cup i) - \Phi(\mathcal{C}) &= \sqrt{s + d_i} - \sqrt{s} \\ \Phi(\mathcal{Z} \cup i) - \Phi(\mathcal{Z}) &= \sqrt{z + d_i} - \sqrt{z} \end{aligned}$$

where s and z refer to the data contributions of sets $\mathcal{C}$ and $\mathcal{Z}$, respectively. Since $\mathcal{C} \subseteq \mathcal{Z} \subseteq \gamma$ and $d_j \ge 0$ for every data set in edge nodes, we can obtain $s \le z$. Thus

$$\left(\sqrt{s + d_i} - \sqrt{s}\right) - \left(\sqrt{z + d_i} - \sqrt{z}\right) \ge 0.$$

So $\Phi(N)$ is a nondecreasing function. Thus, $\Phi(\mathcal{N})$ is a submodular function. ■

Our objective is to design an incentive mechanism satisfying the four properties defined as follows.

Definition 2 (Budget Feasibility): According to the auction mechanism, the total cost of the requester should be less than B, which can be written as $\sum_{n_j \in \mathcal{N}} p_j \le B$.

Definition 3 (IR): The utility obtained by edge nodes participating in the auction is nonnegative, which can be written as $u_j = p_j - c_j \ge 0$.

Definition 4 (Truthfulness): The FL platform is truthful if $u_j(c_j, b_{-j}) \ge u_j(b_j, b_{-j})$ where b_j and c_j stand for its true cost and bids with $b_j \ne c_j$, respectively. b_j refers to the bid submitted by edge nodes except n_j. Truthfulness can also be called individual compatibility (IC).

Definition 5 (Computational Efficiency): The result of an auction algorithm can be obtained in polynomial time.

V. Mechanism Design

In this section, we first discuss our problem formulation. Then, we talk about the mechanism design based on our system design and auction model. In our system, the requester is a buyer and several edge nodes are potential sellers, which satisfies the reverse auction model. Thus, we propose a basic reverse auction within the fixed budget to maximize the valuation of the requester, namely, DQDRA. DQDRA is composed of two phases that are winner selection and payment determination. The first phase determines the winner set according to the bids submitted by all edge nodes, and the reward of the winner set is determined in the payment determination procedure.

As shown in Fig. 3, at the beginning of the auction, the buyer will publish the task, including the data type, the total budget, and the maximum of the EMD value through the smart contract (step 1). After receiving the data type of the task, all of the edge nodes will decide whether to participate in this auction or not (step 2). For those edge nodes who want to join in this task, they will send their bid prices, the number of data, and the EMD of its data set to the smart contract (step 3).

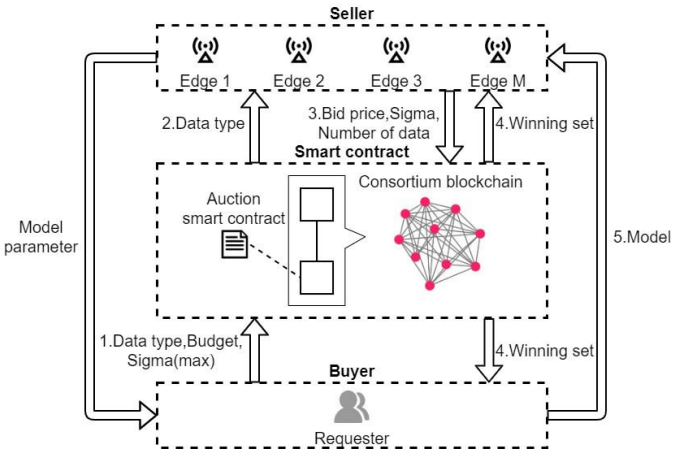

Fig. 3. Auction model.

Algorithm 1 Winner Selection

Input: $\mathcal{N}$, $\mathcal{S} = \{\boldsymbol{s_1}, \cdots, \boldsymbol{s_j}, \cdots, \boldsymbol{s_N}\}$, $\boldsymbol{s_j} = \{b_j, d_j, \sigma_j\}$, B, σ_{max}

Output: $\mathcal{N}^*$

1: Initialize $\mathcal{N}_t \leftarrow \phi, \mathcal{N}^* \leftarrow \phi, k \leftarrow 1, i \leftarrow 0$
2: $\mathcal{N}_t \leftarrow \{n_j \in \mathcal{N} : \sigma_j \leq \sigma_{max}\}$
3: $i \leftarrow \arg\max_{j:n_j \in \mathcal{N}_t} \frac{Q_{j|\mathcal{N}^*_{k-1}}}{b_j}$
4: **while** $\mathcal{N}_t \neq \phi$ and $b_i \leq \frac{B}{2} \times \frac{Q_{i|\mathcal{N}^*_{k-1}}}{\Phi(N^* \cup n_i)}$ **do**
5: $\quad \mathcal{N}^* \leftarrow \mathcal{N}^* \cup n_i$, $\mathcal{N}_t \leftarrow \mathcal{N}_t \backslash n_i$
6: $\quad i \leftarrow \arg\max_{j:n_j \in \mathcal{N}_t} \frac{Q_{j|\mathcal{N}^*_k}}{b_j}$
7: $\quad k \leftarrow k + 1$
8: **end while**
9: **return** $\mathcal{N}^*$

The smart contract will decide the winning set once receiving the information from the edge nodes and return the winning set to edge nodes and the requester, respectively (step 4). The requester will send the initial model to the winners via wireless communication, and the winners will begin to train the model with their local data sets (step 5).

A. Winner Selection

We first describe a monotone greedy allocation algorithm that aims to maximize the valuation of the requester with a fixed budget in one global iteration.

Once receiving the multidimensional information, including bids, the data size, and the EMD metrics from all edge nodes that are willing to take part in this auction, the smart contract creates two sets that are the winner set and the candidate set, respectively, (line 1). According to the EMD metrics of all edge nodes, the smart contract selects the edge nodes that satisfy the maximal EMD metrics set by the platform (line 2). In order to achieve our objective function to maximize the valuation of requesters, we define a new parameter called marginal contribution as follows:

$$Q_{j|\mathcal{N}^*_{k-1}} = \Phi\left(\mathcal{N}^*_{k-1} \cup n_j\right) - \Phi\left(\mathcal{N}^*_{k-1}\right) \tag{9}$$

where $\mathcal{N}^*_{k-1}$ refers to the winner set in the first $k-1$ iterations, and $\Phi(\mathcal{N}^*_{k-1})$ means the contribution of the winners in the first $k-1$ iterations. $Q_{j|\mathcal{N}^*_{k-1}}$ refers to the gain contribution of the kth iteration if edge node n_j is selected by the platform. In the phase of the greedy winner selection, winners are selected by marginal density $[(Q_{j|\mathcal{N}^*_{k-1}})/b_j]$. Thus, in each iteration, the platform will select the largest marginal density of edge node n_j (line 6) in $\mathcal{N}_t$ and add n_j into the winner set $\mathcal{N}^*$ (line 5). If several candidates have the same marginal density, we will randomly select one of them. According to the greedy mechanism and Lemma 1, we can get

$$\frac{Q_{1|\mathcal{N}^*_0}}{b_1} \geq \frac{Q_{2|\mathcal{N}^*_1}}{b_2} \geq \cdots \geq \frac{Q_{s|\mathcal{N}^*_{s-1}}}{b_s} \tag{10}$$

where $\mathcal{N}_s$ refers to the feasible candidate set that contains s winners. Following the order, we greedily add edge nodes into the winner set until currently considered edge node n_k violates the budget feasible mechanism, which is defined as

$$b_k \leq \frac{B}{2} \times \frac{Q_{k|\mathcal{N}^*_{s-1}}}{\Phi\left(\mathcal{N}^*_s\right)}. \tag{11}$$

B. Payment Determination

After finishing the winner selection procedure, the remaining job is to decide the payments for winners. For each winner n_k, similar to the winner selection, we will recompute the maximal marginal density over $\widetilde{\mathcal{N}} = \mathcal{N} \backslash n_k$ (lines 3 and 4). We use $\widetilde{\mathcal{N}}^*$ to represent the winner set over $\widetilde{\mathcal{N}}$. Let l be the index of $\widetilde{\mathcal{N}}^*$ and n_l refers to the edge node selected by platform in the lth loop over $\widetilde{\mathcal{N}}$. According to the marginal density, the platform can calculate a bid $\alpha_{k,l}$ such that n_k can substitute n_l to win in the lth loop over the new set $\widetilde{\mathcal{N}}$ (line 8). $\alpha_{k,l}$ can be written as

$$\alpha_{k,l} = \frac{b_l \times Q_{k|\widetilde{\mathcal{N}}^*}}{Q_{l|\widetilde{\mathcal{N}}^*}}$$

and according to the budget feasibility, the bid should satisfy (line 4)

$$\alpha_{k,l} \leq \frac{B}{2} \times \frac{Q_{k|\widetilde{\mathcal{N}}^*}}{\Phi\left(\widetilde{\mathcal{N}}^*\right)}$$

where $\Phi(\mathcal{N}^*)$ means the total revenue if n_k is selected in the lth loop and $Q_{k|\widetilde{\mathcal{N}}^*}$ refers to the marginal contribution of edge node n_k in the lth iteration over the set $\widetilde{\mathcal{N}}^*$.

VI. Analysis of Auction Model

In this section, we give the theoretical analyses of DQDRA. Next, we prove the IR, incentive compatibility, computational efficiency, and trustfulness.

Lemma 2: DQDRA achieves trustfulness.

Proof: Assume edge node n_j bids b_j other than its truthful cost c_j. We first consider the scenario where $b_j > c_j$.

Case 1: n_j wins with both c_j and b_j. From the critical payment, the payment of the wining edge node is independent to its bid. Therefore, in either case, the winning edge node

Algorithm 2 Payment Determination

Input: $\mathcal{N}$, $\mathcal{N}^*$, B
Output: Payment set for winners $\mathbf{P}$
1: **for** $n_j \in \mathcal{N}^*$ **do**
2: $\mathcal{N}_{-j} \leftarrow \mathcal{N} \backslash n_j$
3: $\widetilde{\mathcal{N}}^* \leftarrow \phi$, $l \leftarrow 1$, $p_j = 0$, $s \leftarrow 0$
4: $i \leftarrow \arg\max_{s:n_s \in \mathcal{N}_{-j}} \frac{\mathcal{Q}_{s|\widetilde{\mathcal{N}}^*_{l-1}}}{b_s}$, $\widetilde{\mathcal{N}}^* \leftarrow n_i$
5: **while** $\mathcal{N}_{-j} \neq \phi$ and $b_i \leq \frac{B}{2} \times \frac{Q_{i|\widetilde{\mathcal{N}}^*_{l-1}}}{\Phi(\widetilde{\mathcal{N}}^*)}$ **do**
6: $\widetilde{\mathcal{N}}^* \leftarrow \widetilde{\mathcal{N}}^* \cup n_i$, $\mathcal{N}_{-j} \leftarrow \mathcal{N}_{-j} \backslash n_i$
7: $\alpha_{j,l} = \frac{b_l \times Q_{j|\widetilde{\mathcal{N}}^*_{l-1}}}{Q_{l|\widetilde{\mathcal{N}}^*_{l-1}}}$, $\beta_{j,l} = \frac{B}{2} \times \frac{Q_{j|\widetilde{\mathcal{N}}^*_{l-1}}}{\Phi(\widetilde{\mathcal{N}}^*)}$
8: $p_j = \max_l(p_j, \min(\alpha_{j,l}, \beta_{j,l}))$
9: $l \leftarrow l + 1$
10: **end while**
11: $\mathbf{P} \leftarrow \mathbf{P} \cup p_j$
12: **end for**
13: **return P**

receives the same payment p_j. So $u_j(c_j, b_{-j}) = u_j(b_j, b_{-j}) = p_j - c_j$.

Case 2: n_j wins with c_j but loses with b_j. Thus, $u_j(c_j, b_{-j}) > u_j(b_j, b_{-j}) = 0$.

Case 3: n_j wins with b_j but loses with c_j. It means that $[(Q_{j|\mathcal{N}^*_{k-1}})/b_j] > [(Q_{j|\mathcal{N}^*_{k-1}})/c_j]$ and thus $b_j < c_j$, which contradicts with the assumption. Thus, this case will not happen.

Case 4: n_j loses with both c_j and b_j. Thus, $u_j(b_j, b_{-j}) = u_j(c_j, -b_{-j}) = 0$.

Next, we will discuss another scenario where $b_j < c_j$.

Case 1: n_j wins with both c_j and b_j. From the critical payment, the payment of the wining edge node is independent to its bid. Therefore, in either case, the winning edge node receives the same payment p_j. So $u_j(c_j, b_{-j}) = u_j(b_j, b_{-j}) = p_j - c_j$.

Case 2: n_j wins with b_j but loses with c_j. Under this assumption, $b_j < c_j$. There are two conditions if n_j loses with true cost c_j. For the first condition, $b_j \leq (B/2) \times [(Q_{j|\mathcal{N}^*_{k-1}})/(\Phi(\mathcal{N}^*_{j|k-1} \cup n_j))] < c_j$ and $p_j \leq (B/2) \times [(Q_j^{k-1})/(\Phi(\mathcal{N}^*_{j|k-1} \cup n_j))]$ according to the pricing mechanism. Thus, $b_j < c_j$ and $u(b_j, b_{-j}) < u(c_j, b_{-j}) = 0$. For the second condition, we assume that there are in total l edge nodes winning in this auction. According to the winner selection procedure, we can get $[(Q_{1|\mathcal{N}^*_0})/b_1] \geq [(Q_{2|\mathcal{N}^*_1})/b_2] \geq \cdots \geq [(Q_{l|\mathcal{N}^*_{l-1}})/b_l] \geq [(Q_{j|\mathcal{N}^*_{l-1}})/c_j]$. With the pricing mechanism, we will use $\mathcal{N} \backslash n_j$ to derive another $\widetilde{\mathcal{N}}^*$. Let r = $\arg\max_{l:n_l \in \widetilde{\mathcal{N}}^*} b^v_{j,l}$, then $[(Q_{j|\mathcal{N}^*_{r-1}})/b^v_{j,r}] \geq [(Q_{l|\mathcal{N}^*_{r-1}})/b_l] \geq [(Q_{j|\mathcal{N}^*_{r-1}})/c_j]$. Therefore, $b^v_{j,r} < c_j$ and $u_j(b_j, b_{-j}) \leq u_j(c_j, b_{-j}) = 0$.

Case 3: n_j wins with c_j but loses with b_j. It means that $[(Q_j^{k-1})/c_j] > [(Q_j^{k-1})/b_j]$ and thus $c_j < b_j$, which contradicts with the assumption. Thus, this case will not happen.

Case 4: n_j loses with both c_j and b_j. Thus, $u_j(b_j, b_{-j}) = u_j(c_j, -b_{-j}) = 0$. ■

Lemma 3: DQDRA achieves IR.

Proof: In this part, we will illustrate the IR of all the edge nodes in the FL platform. According to Lemma 2, the platform is truthful. Thus, edge nodes will submit their cost as bids in order to get the maximal utility, i.e., $c_j = b_j$.

For loser n_j, $u_j(b_j, b_{-j}) = 0$, which satisfies IR mentioned before.

For winner n_j, we assume that it is selected in the lth iteration for payment determination and take the place of n_k in $\widetilde{\mathcal{N}}_*$. According to the winner selection procedure and payment determination, we can obtain $l \geq k$

$$b_k \leq \frac{B}{2} \times \frac{Q_{k|\mathcal{N}^*_{k-1}}}{\Phi\left(\mathcal{N}^*_{k-1} \cup n_k\right)} = \frac{B}{2} \frac{Q_{k|\widetilde{\mathcal{N}}^*_{l-1}}}{\Phi\left(\widetilde{\mathcal{N}}^*_{l-1}\right) \cup n_k} = \beta_{k,l}. \tag{12}$$

According to the winner selection algorithm, we can obtain

$$\frac{Q_{k|\mathcal{N}^*_{k-1}}}{b_k} \geq \frac{Q_{l|\mathcal{N}^*_{k-1}}}{b_l}. \tag{13}$$

So we can get $b_k \leq \min(\alpha_{k,l}, \beta_{k,l)} \leq p_k$. ■

Lemma 4: DQDRA satisfies budget feasibility.

Proof: Some of the budget feasible mechanisms [36]–[38] rely on the compensation determination mechanism to prove the property of budget feasibility. We omit its proof here. ■

Lemma 5: DQDRA satisfies computational efficiency.

Proof: The procedure to find the winner in edge nodes set $\mathcal{N}_t$ with the maximum marginal density in Algorithm 1 has the time complexity of $\mathcal{O}(|\mathcal{N}_t|)$ (line 7). Since the number of edge nodes is at most $|\mathcal{N}_t|$, the winner selection procedure has the time complexity of $\mathcal{O}(|\mathcal{N}_t|^2)$. In Algorithm 2, in order to decide the payment for every winner edge node, each while-loop executes similar steps in lines 6–11. The payment determination process generally has the time complexity of $\mathcal{O}(|\mathcal{N}_t|^3)$. Since $|\mathcal{N}_t| \leq |\mathcal{N}|$, the running time of payment determination is upper bounded by polynomial time $\mathcal{O}(|\mathcal{N}|^3)$. ■

VII. System Implementation and Evaluation

A. Testbed Implementation

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

1) Testbed Specification: The terminal in our testbed implementation is Dell XPS 13 equipped with 8-GB RAM, Intel i7-6500 CPU, and Intel Graphics 520. The edge computing server in our testbed implementation is iMac equipped with 8-GB RAM, Intel i5-4570 CPU, and NVIDIA GeForce GT 755M. The edge computing server is also equipped with three wireless access points. Two of them are TP-LINK TL-WR841N, which adopts the IEEE 802.11b/g/n standard with up to 300 Mb/s data rate and the other one is TP-LINK TL-WR842N, which adopts the IEEE 802.11b/g/n standard with up to 300 Mb/s data rate.

Fig. 4 illustrates the testbed implementation of our system. We use a workstation and two edge computing servers to work as edge nodes and use a laptop to work as the requester. We equip each edge server with a wireless access point to work as

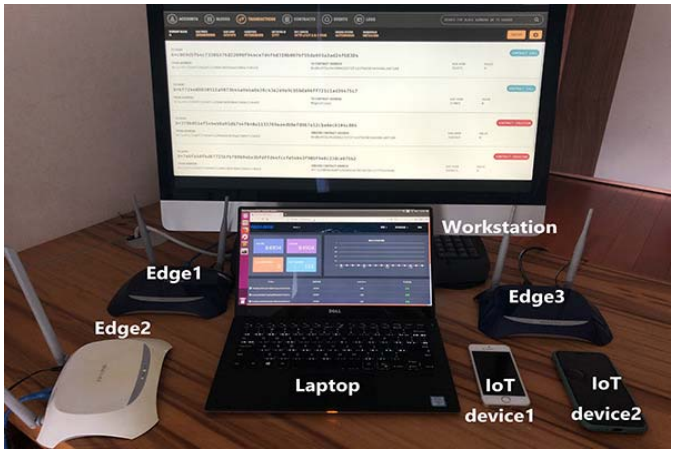

Fig. 4. Demonstration of the testbed implementation.

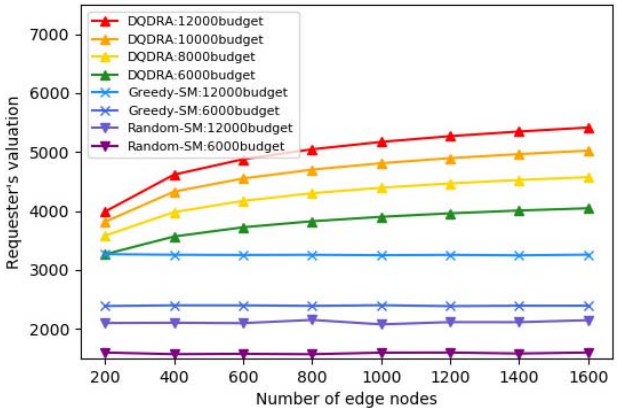

Fig. 5. Impact of the number of edge nodes on the requester's valuation.

an edge node. We use two smartphones to work as IoT devices. The requester can submit their computational tasks through the wireless network. Then, the edge computing server runs the computational tasks and returns the results to the requester through the wireless network. The workstation displays the transactions on the public blockchain. The laptop shows the information on the consortium blockchain.

2) Enabling Technologies: The testbed implementation includes several existing technologies. For the consortium blockchain platform, we choose FISCO-BCOS,[1] which is an opensource consortium blockchain platform. FISCO-BCOS also provides an information port that can directly demonstrate the information of the consortium blockchain. With the information port, requesters and edge nodes can check the information of transactions and the smart contract. Besides, we can also use the provided python-SDK[2] to develop and test the situation of the consortium blockchain. As for the programming language of the smart contract, we use Solidity[3] to write the auction smart contract. For the public blockchain platform, we choose Ganache,[4] which is an opensource Ethereum blockchain platform. It provides not only the testing Ethereum blockchain but also a user-friendly user interface (UI) for the requesters and edge nodes to check the transactions of the Ethereum blockchain. As for the compilation and the deployment of the smart contract on our testing Ethereum blockchain, we use the Truffle[5] framework.

3) Smart Contract Deployment: Initially, the requester deploys the auction smart contract in the consortium blockchain. The deployed smart contract has its unique address in the consortium blockchain. The requester and edge nodes can call functions on the smart contract through its consortium blockchain address. The requester can start an auction by calling the start() function along with the required parameters in the auction. Then, edge nodes can send bids by calling the bid() function along with their bids' parameters. When the requester wants to end the auction, he can call the handleBids() function with his budget. The handleBids() function returns the requester with a winner list. By calling the paymentDetermination() function with the winner list, the smart contract can proceed and generate a payment list automatically for the requester. The payment list provides the requester with the amount he needs to pay to the corresponding edge nodes. By deploying the payment channel smart contract on the public blockchain, the requester can pay the bills through the public blockchain address of edge nodes. Through the payment channel, the requester can pay the bills during model training iterations at regular intervals.

[1]http://fisco-bcos.org/

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

[3]https://github.com/ethereum/solidity

[4]https://www.trufflesuite.com/ganache

[5]https://www.trufflesuite.com/truffle

B. Simulation Results

In this section, we show the numerical results from our evaluations based on the simulations. We also compare the performance of DQDRA with GREEDY-SM and RANDOM-SM mechanisms proposed in the work of Chen *et al.* [39], which are budget feasible mechanisms and satisfy all the desirable properties. GREEDY-SM is based on the greedy algorithm to select winners in an auction until the total payment satisfies the budget feasibility mechanism. Compared with GREEDY-SM, RANDOM-SM has a fixed probability of returning the maximal marginal contribution or the result of GREEDY-SM with the same budget. Based on the parameter setting in the work of Zheng *et al.* [37], unless otherwise stated, the simulation parameters are as follows. There are $N = 500$ edge nodes taking part in the auction for FL tasks, and the budget of the requester is fixed at 12 000. We uniformly generate edge node n_j's data size from 500 to 1500, the EMD value σ_i from 1 to 1.4.

1) Impacts on Requester's Valuation: Fig. 5 shows the requester's valuation when the budget is fixed at certain amounts, and the number of edge nodes varies from 200 to 1600. From the figure, we can observe that the requester's valuation of DQDRA increases when the number of edge nodes increases, while that of Greedy-SM and Random-SM keep stable. The reason is that the requester can select edge nodes with lower costs and higher data amounts under a certain budget.

However, under 6000 and 12 000 budget, the requester's valuation of Greedy-SM and Random-SM is stable with the increasing number of edge nodes, which illustrates that the

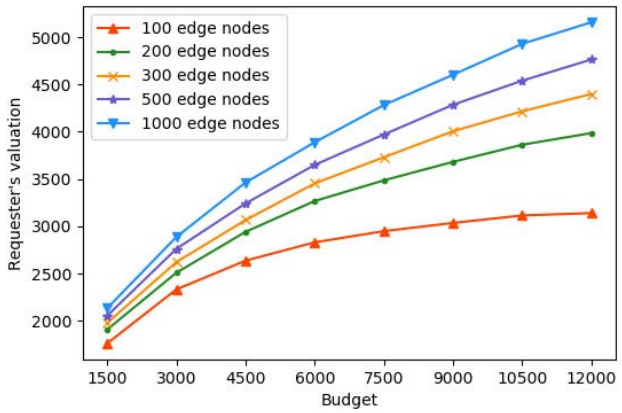

Fig. 6. Impact of the budget and number of edge nodes on the requester's valuation.

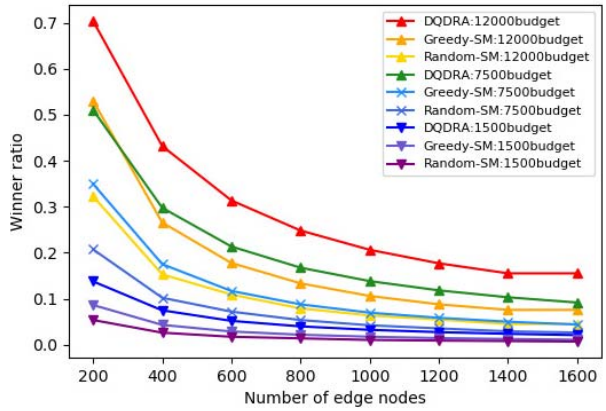

Fig. 7. Impact of number of edge nodes on the winner ratio.

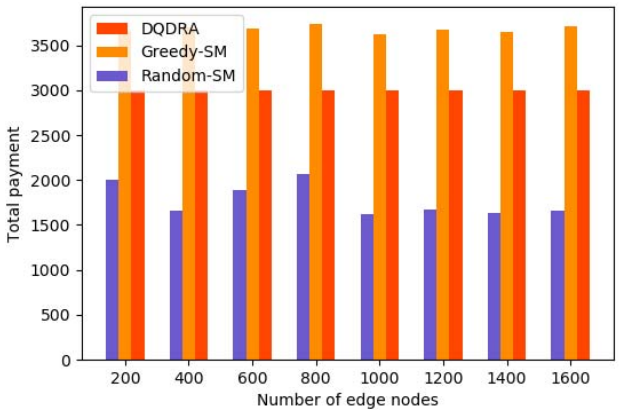

Fig. 8. Impact of the number of edge nodes on the total payment.

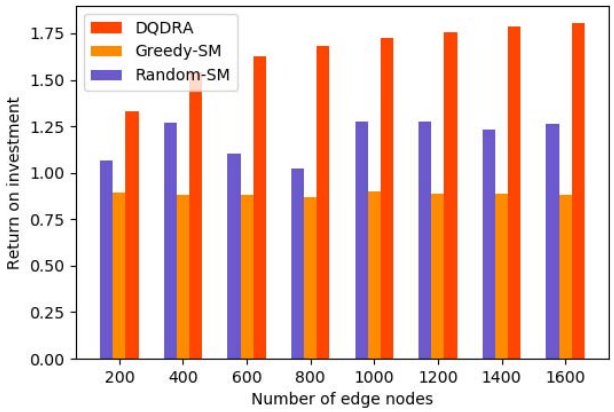

Fig. 9. Impact of the number of edge nodes on the ROI.

requester's valuation is saturated under these two budget in these two algorithms. Besides, with the same number of edge nodes, the requester's valuation increases when the budget increases. This is because the requester can have more budget to select edge nodes. The requester's valuation of DQDRA is larger than the valuation of the requester in the other two algorithms, which illustrates that the performance of DQDRA is the best among all the three algorithms.

We now show the requester's valuation of DQDRA in Fig. 6. The number of edge nodes varies from 100 to 1000, and the budget changes from 1500 to 12 000 with an increment of 1500. We can observe that the requester's valuation grows along with the increase in the budget. This is due to the reason that with more budget, the requester can recruit more edge nodes to perform the FL task. We can also see that when the number of edge nodes increases, the requester's valuation increases. Compared with the other four different kinds of edge nodes, the requester's valuation with 100 edge nodes increases slowly and will keep stable when the budget is higher. This is because the requester will finally recruit all of the edge nodes with the increasing budget.

2) Impacts on Winner Ratio: Fig. 7 shows the winner ratio of DQDRA, GREEDY-SM, and RANDOM-SM by fixing the budget at 1500, 7500, and 12 000, respectively, and varying the number of edge nodes from 200 to 1600.

Under a certain budget, we can observe that the three algorithms' winner ratios decrease as the number of edge nodes gets larger. It is because a larger number of edge nodes leads to more intense competition within the limited budget. Compared with Random-SM and Greedy-SM, the winner ratio of DQDRA is the highest. Under a certain number of edge nodes, we can find that the three algorithms' winner ratios increase with the increasing budget. It is because a larger budget can recruit more edge nodes to perform the task. Compared with Random-SM and Greedy-SM, when the number of edge nodes is 200, the winner ratios of DQDRA and Greedy-SM are about 0.7 and 0.5, respectively, and the winner ratio of Random-SM is only around 0.3. It is because to guarantee bounded approximation in the worst cases, Random-SM has a 40% probability of choosing the first maximal marginal contribution edge node. In conclusion, the winner ratio of DQDRA is bigger than Greedy-SM and Random-SM, which means that the performance of DQDRA is better than the other two.

3) Impacts on Total Payment and Return on Investment: As shown in Fig. 8, our proposed auction-based mechanism is budget feasible, which means that the total payment of all the winner edge nodes is lower than the budget. From this figure, we can observe that the total payment of Greedy-SM is larger than DQDRA and Random-SM. This is because DQDRA always considers both the marginal contributions and the bid prices of edge nodes compared with Greedy-SM.

Through Fig. 9, we investigate the return on investment (ROI) in all the three algorithms when the number of edge nodes varies from 200 to 1600, and the budget is fixed at 12 000. From the figure, we can observe that when the number of edge nodes increases, ROI increases in DQDRA, whereas ROI keeps stable in both Greedy-SM and Random-SM. The reason is that when the number of edge nodes becomes larger, the competition among edge nodes becomes more intense, and the requester can recruit more edge nodes at a lower price. However, Greedy-SM and Random-SM just choose edge nodes according to the number order, so the number of edge nodes does not influence ROI. ROI of DQDRA is the highest among the ROI of the three algorithms, which illustrates that DQDRA is better than the other two.

VIII. Conclusion

In this article, we proposed a hybrid blockchain-based resource trading system for FL in edge computing. To avoid the cost overhead in cross-blockchain data synchronization, our system enables the requesters and edges to interact with the public and consortium blockchains separately, achieving higher credibility and better system performance. We also proposed the DQDRA using the smart contract in the consortium blockchain, facilitating automatic, autonomous, and auditable auctions among edge nodes. We proved that DQDRA satisfies budget feasibility, IR, truthfulness, and computational efficiency. Moreover, we integrated the payment channel into the public blockchain using the smart contract to enable credible, fast, low cost, and high-frequency payment transactions among requesters and edge nodes. Simulation results showed that the proposed DQDRA performs better than other existing budget-feasible reverse auction mechanisms. For future work, DQDRA will be further validated through real-world experiments based on the MNIST data set in FL. Another direction is to study the trading market with multiple requesters, where the requesters will compete with each other for recruiting edge nodes. In this case, we can try other auction models, such as the double auction mechanism.

References

[1] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[2] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," 2019. [Online]. Available: arXiv:1905.06641v2.

[3] 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 Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[4] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep./Oct. 2019.

[5] 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 *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 1387–1395.

[6] 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 Things J.*, vol. 7, no. 10, pp. 9441–9455, Oct. 2020.

[7] S. Shen, Y. Han, X. Wang, and Y. Wang, "Computation offloading with multiple agents in edge-computing–supported IoT," *ACM Trans. Sens. Netw.*, vol. 16, p. 8, Dec. 2019.

[8] X. Wang, X. Li, S. Pack, Z. Han, and V. C. M. Leung, "STCS: Spatial-temporal collaborative sampling in flow-aware software defined networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 999–1013, Jun. 2020.

[9] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security* Oct. 2017, pp. 1175–1191.

[10] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. 23rd Int. Conf. Artif. Intell. Stat.*, Jul. 2018, pp. 2938–2948.

[11] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732–1744, Jun. 2016.

[12] Z. Li, Z. Yang, S. Xie, W. Chen, and K. Liu, "Credit-based payments for fast computing resource trading in edge-assisted Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6606–6617, Aug. 2019.

[13] Z. Li, Z. Yang, and S. Xie, "Computing resource trading for edge-cloud-assisted Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3661–3669, Jun. 2019.

[14] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available:https://bitcoin.org/bitcoin.pdf

[15] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project, Zug, Switzerland, Yellow Paper, 2014.

[16] W. Cai, Z. Wang, J. Ernst, Z. Hong, and C. Feng, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53019–53033, 2018.

[17] N. Herbaut and N. Negru, "A model for collaborative blockchain-based video delivery relying on advanced network services chains," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 70–76, Sep. 2017.

[18] Y. Qian, L. Hu, J. Chen, X. Guan, M. Hassan, and A. Alelaiwi, "Privacy-aware service placement for mobile edge computing via federated learning," *Inf. Sci.*, vol. 505, pp. 562–570, Dec. 2019.

[19] M. Shayan, C. Fung, C. Yoon, and I. Beschastnikh, "Biscotti: A ledger for private and secure peer-to-peer machine learning," Nov. 2018. [Online]. Available: https://arxiv.org/abs/1811.09904.

[20] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," 2018. [Online]. Available: arXiv:1808.04866.

[21] Z. Zhou, L. Pengju, F. Junhao, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.

[22] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *Proc. IEEE VTS Asia–Pac. Wireless Commun. Symp. (APWCS)* Singapore, Aug. 2019, pp. 1–5.

[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] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016. [Online]. Available: https://lightning.network/

[25] C. Decker and R. Wattenhofer, *A Fast and Scalable Payment Network With Bitcoin Duplex Micropayment Channels*. Cham, Switzerland: Springer, Aug. 2015.

[26] Z. Xiong, J. Kang, D. Niyato, P. Wang, and H. V. Poor, "Cloud/edge computing service management in blockchain networks: Multi-leader multi-follower game-based ADMM for pricing," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 356–367, Mar./Apr. 2019.

[27] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4585–4600, Jun. 2019.

[28] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.

[29] 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 Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[30] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Oper. Syst. Design Implement. (OSDI)*, vol. 99, 1999, pp. 173–186.

[31] Y. Jiao, P. Wang, S. Feng, and D. Niyato, "Profit maximization mechanism and data management for data analytics services," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2001–2014, Jun. 2018.

[32] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016. [Online]. Available: https://arxiv.org/abs/1610.02527.
[33] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018. [Online]. Available: arXiv:1806.00582.
[34] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, early access, May 14, 2020, doi: 10.1109/TMC.2020.2994639.
[35] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," 2019. [Online]. Available: arXiv:1912.06370.
[36] W. Jin, M. Xiao, M. Li, and L. Guo, "If you do not care about it, sell it: Trading location privacy in mobile crowd sensing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 1045–1053.
[37] Z. Zheng, F. Wu, X. Gao, H. Zhu, S. Tang, and G. Chen, "A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 9, pp. 2392–2407, Sep. 2017.
[38] Y. Singer and M. Mittal, "Pricing mechanisms for crowdsourcing markets," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1157–1166.
[39] N. Chen, N. Gravin, and P. Lu, "On the approximability of budget feasible mechanisms," in *Proc. 22nd Annu. ACM-SIAM Symp. Discrete Algorithms*, 2011, pp. 685–699.

Sizheng Fan (Graduate Student Member, IEEE) received the B.Eng. degree in automation from Beijing Institute of Technology, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree in computer and information engineering with the Chinese University of Hong Kong, Shenzhen, China.

He is a Research Assistant with the Human-Cloud Systems Laboratory, Chinese University of Hong Kong. His current research interests include blockchain, federated learning, game theory, and crowdsourcing.

Mr. Fan is a student member of CCF.

Hongbo Zhang received the B.Eng. degree in computer science and engineering from the Chinese University of Hong Kong, Shenzhen, China, in 2020, where he is currently pursuing the M.Phil. degree with the School of Science and Engineering.

He is a Research Assistant with the Human-Cloud Systems Laboratory, Chinese University of Hong Kong. His current research interests include blockchain, game theory, machine learning, and edge computing.

Yuchen Zeng received the B.Eng. degree in computer science and engineering from the Chinese University of Hong Kong, Shenzhen, China, in 2020. He is currently pursuing the master's degree in software systems engineering with the Computer Science Department, University College London, London, U.K.

He was a Research Assistant with the Human-Cloud Systems Laboratory, Chinese University of Hong Kong. His current research interests include blockchain and edge computing.

Wei Cai (Member, IEEE) received the B.Eng. degree in software engineering from Xiamen University, Xiamen, China, in 2008, the M.S. degree in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2011, and the Ph.D. degree in electrical and computer engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2016.

From 2016 to 2018, he was a Postdoctoral Research Fellow with UBC. In August 2018, he joined the School of Science and Engineering, Chinese University of Hong Kong, Shenzhen, China, where he is currently an Assistant Professor. He has completed a Visiting Research with the National Institute of Informatics, Tokyo, Japan; Hong Kong Polytechnic University, Hong Kong; and Academia Sinica, Taipei, Taiwan. He has coauthored more than 50 journal and conference papers in the area of cloud-edge computing, interactive multimedia and blockchain systems. His recent research interests include human–computer interaction, multimedia, distributed computing, and computer network.

Dr. Cai was a recipient of the 2015 Chinese Government Award for the Outstanding Self-Financed Students Abroad, 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 BSCI2019 and the best paper awards from CCF CBC2018, IEEE CloudCom2014, SmartComp2014, and CloudComp2013. He is serving as an Associate Editor for IEEE TRANSACTIONS ON CLOUD COMPUTING.