Blockchain-Enhanced Federated Learning Market With Social Internet of Things

Pengfei Wang, *Member, IEEE*, Yian Zhao, Mohammad S. Obaidat, *Life Fellow, IEEE*, Zongzheng Wei, Heng Qi, *Senior Member, IEEE*, Chi Lin, *Senior Member, IEEE*, Yunming Xiao, and Qiang Zhang, *Member, IEEE*

***Abstract*—The machine learning performance usually could be improved by training with massive data. However, requesters can only select a subset of devices with limited training data to execute federated learning (FL) tasks as a result of their limited budgets in today's IoT scenario. To resolve this pressing issue, we devise a blockchain-enhanced FL market (BFL) to (i) make data in computationally bounded devices available for training with social Internet of things, (ii) maximize the amount of training data with given budgets for an FL task, and (iii) decentralize the FL market with blockchain. To achieve these goals, we firstly propose a trust-enhanced collaborative learning strategy (TCL) and a quality-oriented task allocation algorithm (QTA), where TCL enables training data sharing among trusted devices with social Internet of things, and QTA allocates suitable devices to execute FL tasks while maximizing the training quality with fixed budgets. Then, we devise an encrypted model training scheme (EMT) based on a simple but countervailable differential privacy methodology to prevent attacks from malicious devices. In addition, we also propose a contribution-driven delegated proof of stake (DPoS) consensus mechanism to guarantee the fairness of reward distribution in the block generation process. Finally, extensive evaluations are conducted to verify the proposed BFL could improve the total utility of requesters and average accuracy of FL models significantly.**

***Index Terms*—Federated learning, blockchain, social Internet of Things, task allocation, data sharing.**

Manuscript received 16 March 2022; revised 14 June 2022; accepted 30 June 2022. Date of publication 13 October 2022; date of current version 22 November 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2021ZD0112400; in part by the National Natural Science Foundation of China under Grant U1908214, Grant 62072069, and Grant 62172069; in part by the CCF-Tencent Open Fund under Grant IAGR20210116; in part by the Liaoning Revitalization Talents Program under Grant XLYC2008017; and in part by the Fundamental Research Funds for the Central Universities under Grant DUT20RC(3)039 and Grant DUT21TD107. *(Corresponding authors: Pengfei Wang; Qiang Zhang.)*

Pengfei Wang, Yian Zhao, Zongzheng Wei, Heng Qi, and Qiang Zhang are with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: wangpf@dlut.edu.cn; zhaoyian.zh@gmail.com; zongzhengwei@gmail.com; hengqi@dlut.edu.cn; zhangq@dlut.edu.cn).

Mohammad S. Obaidat is with the Computer Science Department and the Cybersecurity Center, The University of Texas Permian Basin, Odessa, TX 79762 USA, also with the King Abdullah II School of Information Technology, University of Jordan, Amman 11942, Jordan, also with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with the Department of Computer Science and Engineering, School of Engineering & Technology, Amity University, Noida, Uttar Pradesh 201301, India (e-mail: m.s.obaidat@ieee.org).

Chi Lin is with the School of Software Technology, Dalian University of Technology, Dalian 116024, China, and also with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116621, China (e-mail: c.lin@dlut.edu.cn).

Yunming Xiao is with the Department of Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: yunming.xiao@u.northwestern.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2022.3213314.

Digital Object Identifier 10.1109/JSAC.2022.3213314

I. Introduction

WITH the proliferation of smart IoT devices, how to improve federated learning (FL) [1], [2], [3] performance has always been a great concern for all stakeholders in the IoT scenario. Unlike traditional machine learning algorithms that require centralized data to train a global machine learning model, FL leverages edge devices to train local machine learning models and aggregates them together to generate a global model. In general, FL training performance could be improved while the estimation variance could also be decreased [4], [5] by leveraging more data from various kinds of related devices.

However, it is not practical to leverage massive data to train an FL task mainly for two reasons in the IoT scenario. Firstly, FL requesters usually have fixed budgets to fulfil an FL task. The budgets are mainly for compensation for the resource consumption (*e.g.,* computing, communication, *etc.*) on IoT devices which participate in the FL training. Secondly, a number of IoT devices are computationally bounded and cannot support the training process in practice, although they hold valuable data that can contribute to the training. These data could not be utilized at all since the current FL paradigm only allows models to be trained locally, *i.e.,* on the device where data are generated.

To solve this pressing issue, our rationale is to share training data among trusted IoT devices. To be specific, the training data could be transferred from computationally bounded devices to trusted computationally capable devices by considering the device relationship with social Internet of things (SIoT) [6]. As a result, requesters could maximize the amount of training data with given budgets for an FL task. Meanwhile, we also guarantee the sustainable development of the market by addressing the possible malicious participants as well as market fairness. Without sustainable development, the market would fail as few devices will eventually take the most portion if not all of the market. Such a monopoly

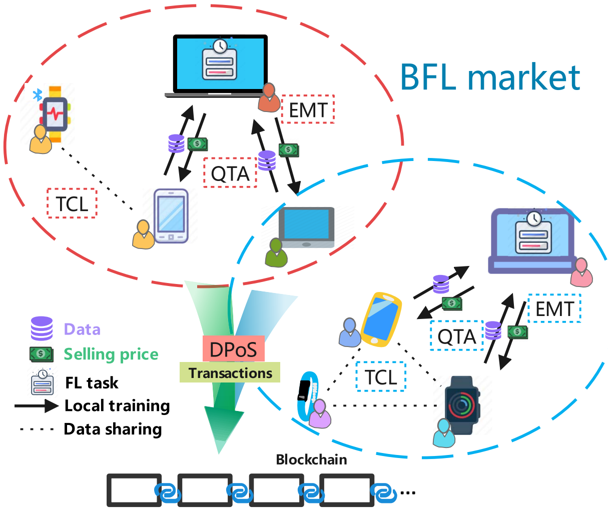


Fig. 1. An example of BFL market with SIoT.

will drive away the requesters as they cannot benefit from utilizing the data on various devices with a reasonable price.

Nevertheless, it is challenging to achieve the above goals as it calls for solutions to three major problems. The first problem is how to organize all IoT devices and their trust relationship, and share data among them. This is challenging because of the heterogeneity of devices [7], *i.e.,* devices are different in computing resources, data volume, and data quality, *etc*. The second problem is how to allocate FL tasks to all computationally capable devices while maximizing the training quality of requesters given limited budgets. The third problem is how to organize the FL tasks and IoT devices in blockchain and prevent attacks from malicious devices, thus ensure that all FL tasks in the blockchain system can conduct model training securely.

Our solution is to devise a blockchain-enhanced federated learning (BFL) market with SIoT, as depicted in Figure 1. In the market, devices could buy or sell their FL services, *e.g.,* requesting an FL task, executing a local model training process, *etc.* In addition, computationally bounded IoT devices can share data to their trusted computationally capable devices with SIoT to increase the training data volume. Blockchain is leveraged to standardize the market order and record the related transactions. More concretely, it can prevent attacks from malicious devices, ensure that all FL tasks can conduct model training securely, and guarantee the fairness of reward distribution.

Towards a blockchain-enhanced federated learning market with social Internet of things, we make the following major contributions:

- We devise the BFL market to improve the federated learning performance by making data on computationally bounded devices available for training with SIoT. To the best of our knowledge, this is the first work to optimize FL with both blockchain and SIoT. We formulate the problem as the collaborative federated learning (CFL) problem, which is proven to be NP-hard.
- To provide a theoretically feasible solution for CFL problem, we propose a trust-enhanced collaborative learning strategy (TCL) and a quality-oriented task allocation algorithm (QTA) separately. TCL organizes the trust relationship of heterogeneous IoT devices, and guides the data sharing among mutually trusted devices with SIoT. QTA aims to allocate FL tasks to computationally capable devices by maximizing the training quality of the FL requesters with fixed budgets.
- Then, we devise an encrypted model training scheme (EMT) and a contribution-driven delegated proof of stake (DPoS) for blockchain to ensure the long-term stable operation of TCL and QTA in the actual BFL market by preventing attacks from malicious devices and ensuring the fairness of reward distribution separately.
- We further conduct extensive evaluations with real-world dataset and show that BFL could improve the system utility of all requesters by 65.7% on average compared with the other benchmarks while improving the overall FL model training accuracy.

The rest of this paper is organized as follows. In Section II, we depict the system architecture of the BFL market and formalize the CFL problem, which is proven to be NP-hard. Then, we devise the blockchain-enhanced federated learning market including TCL, QTA, EMT and DPoS in Section III. Extensive evaluations are conducted in Section IV. Section V reviews the related work, and we conclude the paper in Section VI.

II. Preliminaries

In this section, we elaborate the proposed system architecture of the BFL market and formulate the collaborative federated learning (CFL) problem in the IoT scenario.

A. System Architecture

We aim to devise a blockchain-enhanced federated learning market (BFL) in this paper. The system architecture is decentralized as shown in Figure 1, and all transactions and related operations are recorded in the blockchain. The BFL market consists of a requester set $\boldsymbol{R} = \{r_1, r_2, \ldots, r_M\}$ and a device set $\boldsymbol{E} = \{e_1, e_2, \ldots, e_N\}$. In the market, each requester $r_i (1 \leq i \leq M) \in \boldsymbol{R}$ can post an FL task k_i with deadline T_i. Correspondingly, each device $e_j (1 \leq j \leq N) \in \boldsymbol{E}$ has the amount of data $g_{i,j}$ for task k_i, and device e_j decides whether to participate in task k_i according to its capability and benefit or not. Device e_j could send the generated data to nearby trusted devices before the requester purchases the data if it is computationally bounded in our system architecture.

In our devised system architecture, an FL task allocation in the BFL market includes two stages. The first stage is that computationally bounded devices send their data to computationally capable devices they trust, and the second stage is that computationally capable devices are selected by the requester to participate in the corresponding FL task. Each FL task will be trained according to the allocation results. Specifically, computationally capable devices complete the local model training and submit their local update to the requester, which aggregates results and updates the global model. To ensure the robustness of the proposed blockchain based architecture, encrypted model training scheme and contribution-driven DPoS consensus mechanism are also should be devised to enhance the security of data transmission and fairness of the market.

B. Problem Formulation

The problem to be solved is to maximize the amount of training data purchased by requesters with given fixed budgets. When requester r_i posts an FL task k_i with the deadline T_i, all devices in device set $\boldsymbol{E}$ can decide whether to participate in this FL task or not. We consider that the maximum amount of data each device e_j has is $G_{i,j}$ for task k_i and the amount of data that device e_j can train in one CPU cycle is β_j. We define $\gamma_{i,j} = G_{i,j} - g_{i,j}$, where $g_{i,j}$ is the amount of data that device e_j can be trained before the deadline T_i, and $\gamma_{i,j}$ is the amount of data that cannot be trained before the deadline T_i by device e_j. In particular, $\gamma_{i,j} = 0$ for the device with sufficient computing resources, and $g_{i,j} = 0$ for the computationally bounded device. In fact, the maximum amount of data that can be trained in T_i is $\beta_j \times |T_i|$, where $|T_i|$ indicates the number of CPU cycles included in T_i. After completing the data sharing, the data volume of computationally capable device e_j is $\hat{g}_{i,j} = g_{i,j} + \sum_k x_{k,j}\gamma_{i,k}$, where $x_{k,j} \in \{0,1\}$ indicates that whether computationally bounded device e_k transfers data to computationally capable device e_j. Then, the computationally capable device e_j will have a selling price $v_{i,j}$ for its unit data.

The amount of data that can be purchased by the requester from the computationally capable device e_j is $y_{i,j}\hat{g}_{i,j}$, where $y_{i,j}$ represents the purchased percentage. The problem can be formulated to be a $collaborative\ federated\ learning$ (CFL) problem as presented as follows.

$$\max \sum_i \sum_j y_{i,j}\hat{g}_{i,j} \tag{1}$$

$$s.t. \sum_j x_{k,j} \in \{0,1\} \tag{1a}$$

$$\sum_k x_{k,j}\gamma_{i,k} + g_{i,j} \leq \beta_j \times |T_i| \tag{1b}$$

$$\sum_j y_{i,j}\hat{g}_{i,j}v_{i,j} = B_i \tag{1c}$$

$$c_{i,j} \geq b_{i,k} \tag{1d}$$

$$\mathcal{L}_{j,k} \geq \xi \tag{1e}$$

$$\parallel e_j, e_k \parallel \leq R_j \tag{1f}$$

$$x_{k,j} \in \{0,1\}, \quad y_{i,j} \in [0,1] \tag{1g}$$

where Equation 1 indicates that our objective is to maximize the amount of data for FL tasks while the following conditions are required to be satisfied simultaneously: (i) Equation 1a indicates that the computationally bounded device can only send its own data to at most one device. (ii) Equation 1b indicates that the device cannot train more data than its computing resource. (iii) Equation 1c indicates that the requester r_i can just select devices within the given budget. (iv) Equation 1d, Equation 1e and Equation 1f indicate the restrictions on the collaborative learning strategy: the payment needs to exceed the device reserve price, the trustworthiness of the collaborative learning needs to exceed the threshold, and the collaborative learning needs to be within the communication range, respectively. (v) Equation 1g illustrates whether the collaborative learning occurs and the proportion of data purchased by the requester respectively. We summarize the main notations in Table I.

Theorem 1: The CFL problem is NP-hard.

Proof: The BFL market contains a device set $\boldsymbol{E}$ and a requester set $\boldsymbol{R}$. The device set $\boldsymbol{E}$ consists of a computationally bounded device set $\boldsymbol{E}_l = \{e_1, e_2, \ldots, e_{N_l}\}$ and a computationally capable device set $E_c = \{e_1, e_2, \ldots, e_{N_c}\}$. We consider a special case of the CFL problem as follows. In this case, we assume that the values of trustworthiness $\mathcal{L}$ between any two devices exceed the threshold ξ, and the communication range of all devices is large enough, *i.e.,* all computationally bounded devices can send data to any computationally capable devices in the market. Each computationally bounded device $e_k (1 \leq k \leq N_l) \in \boldsymbol{E}_l$ needs to send its data to computationally capable device $e_j (1 \leq j \leq N_c) \in \boldsymbol{E}_c$ before deadline T_i and gets payment $c_{k,j}$. Besides, we assume that the requester r_i has sufficient budget to select all devices that have the relevant data to train. We define the amount of data participated in the FL task is proportional to the profit. As a result, the problem is simplified to the following: For M tasks with deadlines and profits posted by M requesters, the BFL market allocates N_c computationally capable devices to complete them. Our target is to complete all tasks with the maximum profit, *i.e.,* maximum the amount of data for each task. To sum up, the CFL problem in a special case is a job sequencing with deadline problem, which is a NP-hard problem obviously [8]. Therefore, the CFL problem is NP-hard at least. □

III. Blockchain-Enhanced Federated Learning Market With SIoT

In this section, we decompose the CFL problem into two stages, *i.e.,* data transmission and task allocation. The overall process of TCL, QTA EMT and DPoS in the BFL market is shown in Figure 2. We devise (i) trust-enhanced collaborative learning strategy (TCL) based on double data auction mechanism to ensure the trusted sharing of private data among devices and (ii) quality-oriented task allocation algorithm (QTA) based on greedy strategy to support data transactions after TCL completes data sharing. Besides, we also devise (iii) encrypted model training scheme (EMT) and (iv) contribution-driven DPoS consensus mechanism to enhance the security and fairness of the market respectively.

A. Trust-Enhanced Collaborative Learning Strategy

In this subsection, we firstly give an overview of TCL and propose an auction-based data sharing strategy, which aims to ensure that the computationally bounded devices send their data to trusted nearby computationally capable devices for FL training as much as possible. Then, we establish the trustworthiness model and mobility model of devices. We show

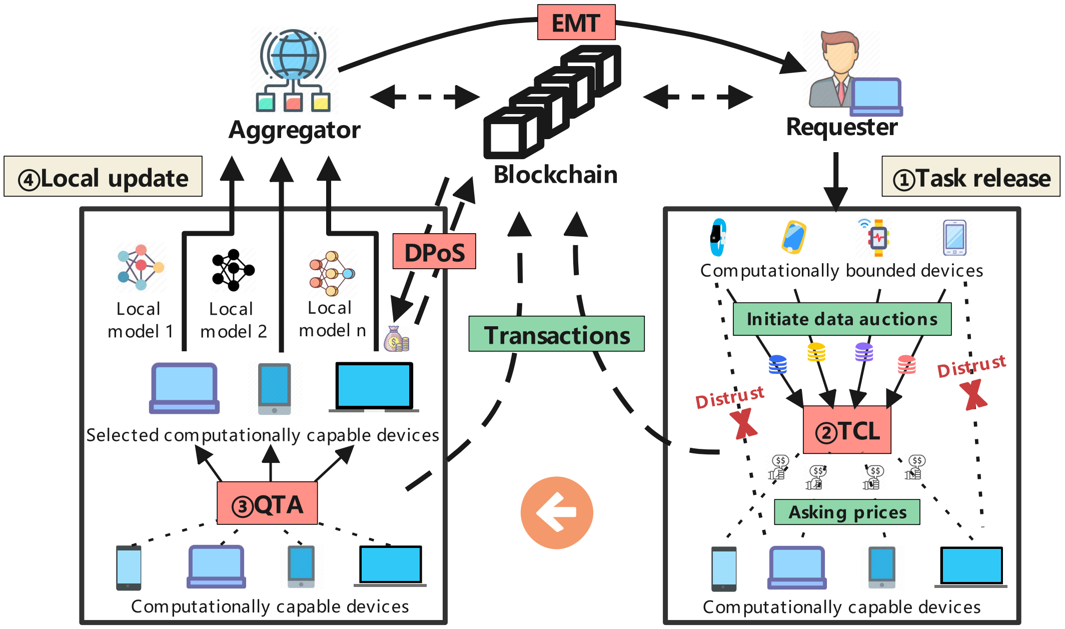


Fig. 2. The overall process of blockchain-enhanced federated learning market.

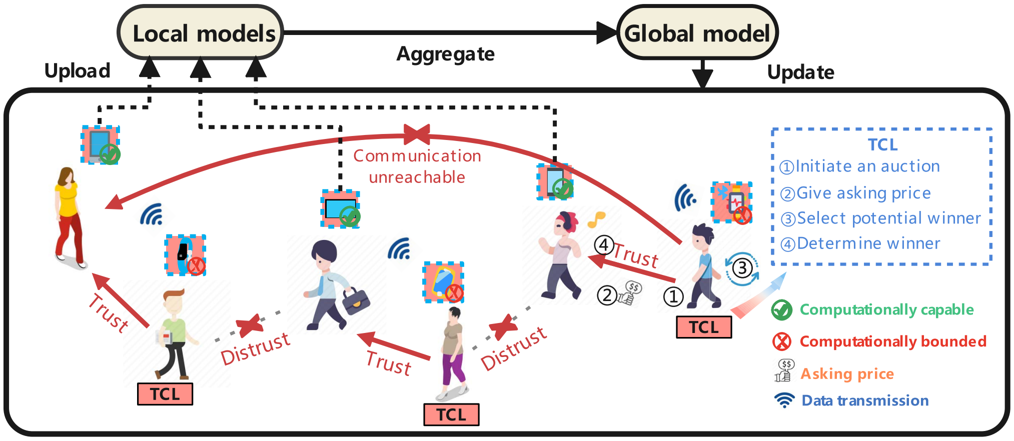


Fig. 3. The detailed process of TCL.

the details of TCL and implement it on the smart contract. Finally, we theoretically demonstrate that our proposed data auction mechanism is robust.

1) Overview of TCL : When a requester $r_i(1 \leq i \leq M) \in \boldsymbol{R}$ posts an FL task k_i in the BFL market, computationally capable devices which possess relevant data can participate in k_i whereas computationally bounded devices need to send their data to computationally capable devices through TCL. As shown in Figure 3, we comprehensively consider the sociality and mobility of devices in the SIoT scenario, and firstly propose the trustworthiness model and mobility model of devices. Then, we devise a double data auction algorithm based on the second price sealed auction to ensure that the data is auctioned at a truthful price. Computationally bounded devices can send data to their trusted devices within the communication range through data auction, which ensures the computationally bounded devices can also participate in model training and requester r_i can obtain more data under the same budget. Note that this process should be completed before requester r_i selects the devices in the BFL market.

2) Trustworthiness Model: Considering the sociality of IoT devices, data sharing among devices is limited. It can only take place in mutually trusted devices due to privacy problem. To achieve the goal, we establish a trustworthiness model by defining trustworthiness as the trust metric, which refers to the degree of trust that one device believes that another device will not disclose its data to others. According to the information theory, entropy is a natural measure of uncertainty, so we description the trustworthiness model based on entropy [9]. Firstly, we divide the trust probability interval 0 to 1 into eleven levels in the step of 0.1 for each device to choose, so as to obtain trust probability p_{ab} of device e_a to any device e_b. We map trustful probability p_{ab} to $[-1, 1]$ to calculate the trustworthiness between two devices. Specifically, we define

TABLE I

MAIN NOTATIONS EMPLOYED IN THIS PAPER

Notation	Explanation
$\boldsymbol{R}$	Requester set
r_i	The i-th requester in requester set $\boldsymbol{R}$
k_i	FL task posted by requester r_i
T_i	Task deadline of task k_i
$\boldsymbol{E}$	Device set
e_j	The j-th device in device set $\boldsymbol{E}$
$G_{i,j}$	The maximum data volume owned by e_j for task k_i
β_j	The upper limit that can be trained in one CPU cycle
$g_{i,j}$	The data volume that can be trained by e_j for task k_i
$\gamma_{i,j}$	The data volume that cannot be trained by e_j for task k_i
$\mathcal{L}_{i,j}$	Trustworthiness of device e_i to device e_j
ξ_j	Trustworthiness threshold selected by device e_j
R_j	Communication range radius of device e_j
$a_{j,k}$	Asking price of device e_j to device e_k
b_j	Budget of device e_j
$c_{i,j}$	Payment of requester r_i to device e_j
B_i	Budget of requester r_i
$v_{i,j}$	Selling price of device e_j to requester r_i
u	Utility of devices or requesters
$\tilde{\boldsymbol{E}}_i^c$	Device set selected by requester r_i.
W_i	Parameters of global model for task k_i
$\boldsymbol{\pi}$	Training result set
π_j	Training result of device e_j in result set $\boldsymbol{\pi}$
ϵ_j	Model accuracy of device e_j
θ	Model accuracy threshold for FL task
$\ell_{i,j}$	Contribution of device e_j for task k_i
S_j	Stake of device e_j
χ_j	Device e_j's proportion of votes

an entropy function $\varphi(\cdot)$ to compute the trustworthiness L as illustrated in Equation 2., where L is $\varphi(p) - 1$ when p is greater than or equal to 0 and less than 0.5. Otherwise, L is $1 - \varphi(p)$.

$$\varphi(p) = -p\log_2(p) - (1-p)\log_2(1-p) \tag{2}$$

In a complete social network, the trustworthiness can be calculated via concatenation and multipath propagations. The concatenation propagation is a trust propagation single chain containing multiple intermediate recommenders, and trustworthiness L_{ac} of device e_a to device e_c is L_{ab} times L_{bc}. The multipath propagation is a trust propagation multiple chains containing multiple direct recommenders, and trustworthiness L_{ad} can be obtained by multi-path weighted average, as illustrated in Equation 3.

$$L_{ad} = \frac{L_{ab}}{L_{ab} + L_{ac}} L_{ab} L_{bd} + \frac{L_{ac}}{L_{ab} + L_{ac}} L_{ac} L_{cd} \tag{3}$$

Notice that the social network graph with complete trustworthiness will be obtained through these two basic propagations. Last, each computationally bounded device can independently select trustworthiness threshold ξ and only when the trustworthiness between two devices exceeds this threshold can they carry out the collaborative learning.

3) Mobility Model: The mobility of IoT devices is equally important as well. Successful data sharing can only be carried out when the communication conditions are met, and the result of device mobility is the change in the communication range of each device. The communication range of a mobile device may vary from device to device and depends on the relevant preferences or limitations of the device owner. We assume that it is realistic that the communication range of device e_i depends on its current location l_i, and each device in device set $\boldsymbol{E}$ broadcasts its location, and other devices can determine the set of devices that can communicate with. Each device e_i has a communication range radius R_i, which divides the communication range into different layers along the radius direction. We assume that devices located in the same layer have the same communication rate, and the larger the radius, the lower the communication rate. For the data sender, that is, the computationally bounded devices, the transmission delay is equal to the amount of data divided by the communication rate. The data can be successfully transmitted only when the device energy exceeds the transmission power multiplied by the transmission delay. The privacy protection of device location information is beyond the scope of this paper and will not be considered here.

4) Algorithm Design: In the BFL market, we consider a practical scenario, in which each computationally bounded device independently decides where its data goes. Devices with data required for task k_i form a device set $\boldsymbol{E}_i$, $\boldsymbol{E}_i(1 \leq i \leq M) \subseteq \boldsymbol{E}$. And all devices in $\boldsymbol{E}_i$ will determine whether they are capable of local training based on the size of the model. If device $e_j(1 \leq j \leq N) \in \boldsymbol{E}_i$ fails to fulfill the local training, it will transfer its data to other trusted devices. We assume that each computationally capable device can receive data from multiple computationally bounded devices, and each device is capable of receiving all asking prices from computationally capable devices within its communication range. The TCL is designed as follows.

We consider a double data auction initiated by computationally bounded device $e_k(1 \leq k \leq N) \in \boldsymbol{E}_i$, which needs to perform FL task k_i posted by requester r_i. Then, device $e_j \in \boldsymbol{E}_i$ within device e_k's communication range participates in the auction. The double data auction consists of the following four steps:

- **Initiate an auction:** The computationally bounded device e_k broadcasts its reserved data volume $\gamma_{i,k}$ and task k_i's deadline T_i. Besides, e_k has a budget of unit data $b_{i,k}$, which represents the final cost of data.
- **Buyers give asking price:** Each device calculates the maximum amount of data it can receive. If device e_j meets $g_{i,j} + \gamma_{i,k} \leq \beta_k \times |T_i|$, e_j will submit its asking price $a_{j,k}$ to e_k. In addition, asking price $a_{j,k}$ beyond the auction deadline will be discarded.
- **Select potential winner:** Device e_k receives an asking price set $\boldsymbol{A}_k = \{a_{1,k}, a_{2,k}, \ldots, a_{\lambda,k}\}$, where λ is the number of asking prices received within the deadline. Then device e_k removes the asking prices that are higher than its budget $b_{i,k}$ and obtains a new set $\tilde{\boldsymbol{A}}_k$. Next, device e_k sorts set $\tilde{\boldsymbol{A}}_k$ in ascending order, *i.e.,* $a_{j,k} \geq a_{j,k'} \geq \cdots \geq b_{i,k}$. Finally, device e_k selects the trusted device with the highest asking price as the potential winner and sends payment $c_{k,j}$ to e_j, which can be calculated as illustrated in Equation 4.

$$c_{k,j} = max\{a_{j,k'}, b_{i,k}\} \tag{4}$$

- **Determine the winner:** Device e_k determines the winner $\tilde{e}_j$ of the auction, and records this transaction in the ledger, which includes data volume $\gamma_{i,k}$ and payment $c_{k,j}$. Once device $\tilde{e}_j$ is selected by requester r_i, device e_k sends data to $\tilde{e}_j$ and gets payment $c_{k,j}$.

Note that the following situations are possible: If potential winner $\tilde{e}_j$ of auctioneer e_k is selected by other auctioneers at the same time, potential winner $\tilde{e}_j$ will choose one of them according to the utility in this auction u_j, which is computed as illustrated in Equation 5.

$$u_j = \gamma_{i,k}(a_{j,k} - c_{k,j}) \tag{5}$$

The final winner will exit the current auction and participate in the other auctions if it has surplus computing resources. If the potential winner of auctioneer e_k doesn't choose e_k, it will start the next round of auctions until no device gives the asking price or the time limit is reached. At the same time, the losers of auctions will continue to compete with other devices until all auctioneers stop the auction. Thus, it is possible that computationally bounded devices may fail to transfer their data within limited time.

In addition, devices that cannot complete the FL task before task k_i's deadline T_i also need to send a part of its data to the trusted device. For example, if device e_j has more data than $\beta_j \times |T_i|$, the extra part needs to be auctioned. Otherwise, if device e_j has the ability to complete the local training, it will not send data to trusted devices because the redundant data transmission will increase the total consumption of the BFL market.

5) Smart Contract of TCL : As shown in Algorithm 1, we devise a smart contract for TCL to automatically and efficiently execute the algorithm. The input of TCL includes device set $\boldsymbol{E}_i$ and requester set R, and the output is new device set $\tilde{\boldsymbol{E}}_i$, in which all devices are computationally capable for the requester. Firstly, we divide the device set $\boldsymbol{E}_i$ according to whether the device is computationally bounded. The computationally bounded devices are listed in one set $\boldsymbol{Q}_1$, while the devices with surplus computing resources are listed in another set $\boldsymbol{Q}_2$(lines 1 to 9). Note that $\boldsymbol{Q}_1 \cap \boldsymbol{Q}_2$ may not be equal to device set $\boldsymbol{E}_i$, the reason is that there may be some devices in $\boldsymbol{E}_i$ that can just complete tasks on time without surplus computing resources.

Secondly, each device e_k in computationally bounded device set $\boldsymbol{Q}_1$ initiates an auction and devices within e_k's communication range in device set $\boldsymbol{Q}_2$ give an asking price. The auction initiated by device e_k will expire after a period of time and it will receive a asking price set $\boldsymbol{A}_k$(lines 12 to 17). Then, device e_k removes asking prices which come from the untrusted devices or are lower than the budget $b_{i,k}$(lines 20 to 24). Next, device e_k sorts the remaining asking prices in ascending order, and selects the device with the highest asking price as the potential winner $\tilde{e}_j$(lines 25 to 26).

Thirdly, device e_k gives a payment $c_{k,j}$ to potential winner $\tilde{e}_j$. If potential winner $\tilde{e}_j$ selects e_k at the same time, a transaction record will be generated. At the end of this section, we elaborate on the data structure of transactions generated in TCL, QTA and EMT. For TCL, the transaction record includes the transaction ID, transaction type, address of device e_k, address of device $\tilde{e}_j$, data volume $\gamma_{i,k}$ and payment $c_{k,j}$. After the transaction is endorsed, it will be recorded in the ledger and written into the blockchain in the next block generate cycle.

Algorithm 1 Trust-Enhanced Collaborative Learning Strategy

```
Input: device set $\boldsymbol{E}_i$, requester set $\boldsymbol{R}$, trustworthiness matrix
    $\mathcal{L}$, trustworthiness threshold $\xi$
Output: computationally capable device set $\tilde{\boldsymbol{E}}_i$
1: $\boldsymbol{Q}_1 \leftarrow \emptyset$, $\boldsymbol{Q}_2 \leftarrow \emptyset$
2: for $e_i$ in $\boldsymbol{E}_i$ do
3:   if $e_i$ is computationally bounded or unable to complete
     the task on time then
4:     $\boldsymbol{Q}_1 \leftarrow \boldsymbol{Q}_1 \cup e_i$
5:   end if
6:   if $e_i$ has surplus computing resources then
7:     $\boldsymbol{Q}_2 \leftarrow \boldsymbol{Q}_2 \cup e_i$
8:   end if
9: end for
10: $\tilde{\boldsymbol{E}}_i = \boldsymbol{E}_i - \boldsymbol{Q}_1$
11: for $e_k$ in $\boldsymbol{Q}_1$ do
12:   The asking price set $\boldsymbol{A}_k \leftarrow \emptyset$
13:   for $e_j$ in $\boldsymbol{E}_i$ do
14:     if $e_j$ in the communication range of $e_k$ then
15:       $e_j$ gives a asking price $a_{j,k}$ to $e_k$
16:       $\boldsymbol{A}_k \leftarrow \boldsymbol{A}_k \cup \{a_{j,k}\}$
17:     end if
18:   end for
19:   for $a_{j,k}$ in $\boldsymbol{A}_k$ do
20:     if $a_{j,k} \leq b_{i,k}$ or $\mathcal{L}_{j,k} \leq \xi_k$ then
21:       $\boldsymbol{A}_k$ = $\boldsymbol{A}_k \backslash \{a_{j,k}\}$
22:     end if
23:   end for
24:   $e_k$ sorts asking price set $\boldsymbol{A}_k$ in ascending order, i.e.,
      $\boldsymbol{A}_k = \{a_{1,k}, a_{2,k}, \cdots, a_{\lambda',k}\}, a_{1,k} \leq a_{2,k} \leq \cdots \leq a_{\lambda',k}$
      where $\lambda'$ is the number of asking price
25:   $e_k$ selects the device $\tilde{e}_j$ with the highest asking price as
      the potential winner and gives a payment $c_{k,j}$ to $\tilde{e}_j$.
26: end for
27: return $\tilde{\boldsymbol{E}}_i$
```

6) Property Analysis: An economic-robust auction refers to an auction which can simultaneously achieve individual rationality, budget balance, and truthfulness [10]. As a result, our proposed TCL is an economic-robust auction as we demonstrate by theoretical analysis of the above three properties. Moreover, we also prove that TCL is computationally efficient.

Theorem 2: TCL ensures individual rationality for all devices.

Proof: If device e_j is the loser of an auction, its utility $u_j = 0$. If device e_j is the winner of an auction, e_j will be selected as the potential winner of at least one computationally bounded device e_k. If device e_k receives asking price other than e_j, the payment of e_k to potential winner $\tilde{e}_j$ will be the next highest asking price $a_{j,k'}$, $a_{j,k'} \leq a_{j,k}$. If device e_k only receives one asking price form device e_j, the payment is e_k's budget $b_{i,k}$, $b_{i,k} \leq a_{j,k}$. Thus, according to Equation 5, the utility of device e_j is greater than or equal to zero, which proves that the TCL strategy ensures individual rationality for all devices. □

Theorem 3: TCL is budget-balanced for all devices.

Proof: Device e_k removes all asking prices higher than budget $b_{i,k}$ after receiving the asking price set $\boldsymbol{A}_k$. Thus, budget $b_{i,k}$ is greater than all asking prices, and payment $c_{k,j} = max\{a_{j,k'}, b_{i,k}\}$, $a_{j,k} \geq a_{j,k'} \geq b_{i,k}$. Hence, we have $b_{i,k} \leq c_{k,j}$, which ensures budget balance for all devices. □

Theorem 4: TCL ensures computational efficiency.

Proof: We assume that there are m computationally bounded devices, where each device e_k can communicate with an average of n devices, and the time complexity of the sorting algorithm is $O(nlogn)$. Considering the worst case of reverse auction, if the potential winner $\tilde{e}_j$ selected by computationally bounded device e_k in each round of auction does not choose itself, device e_k will need to go through n rounds of the auction and the time complexity is $O(n^2 logn)$. Thus, the time complexity of TCL algorithm is $O(mn^2 logn)$, which can be completed in polynomial time and is computationally efficient. □

Theorem 5: TCL guarantees truthfulness for all devices.

Proof: Proof of this theorem is equivalent to proving that in each auction of computationally bounded device e_k, other devices e_j cannot enhance its utility by submitting an asking price $\tilde{a}_{j,k} \neq a_{j,k}$. This can be proved through the following cases.

Case 1: $\tilde{a}_{j,k} \neq a_{j,k}$ and device e_j loses the auction both $\tilde{a}_{j,k}$ and $a_{j,k}$. In this case, the utility u_j of device e_j from e_k's auction is zero.

Case 2: $\tilde{a}_{j,k} > a_{j,k}$ and device e_j wins with $\tilde{a}_{j,k}$ and loses with $a_{j,k}$. we assume that the data size of device e_k for requester r_i is $\gamma_{i,k}$. Because device e_j loses with $a_{j,k}$, we can get $\tilde{a}_{j,k} \geq a_{j,k'} \geq a_{j,k} \geq b_{i,k}$. According to equation 4, the payment $c_{k,j} = a_{j,k'}$, so the payment $c_{k,j} \geq a_{j,k}$. Thus, the device e_j's utility $u_j = \gamma_{i,k} \times (a_{j,k} - c_{k,j}) \leq 0$, and the device e_j cannot get a higher utility.

Case 3: $\tilde{a}_{j,k} > a_{j,k}$ and device e_j wins the auction with both $\tilde{a}_{j,k}$ and $a_{j,k}$. Because the device e_j wins the auction with both $\tilde{a}_{j,k}$ and $a_{j,k}$, we can get $\tilde{a}_{j,k} > a_{j,k} \geq a_{j,k'} \geq b_{i,k}$. According to equation 4, the payment $c_{k,j}$ is the same in both asking price $\tilde{a}_{j,k}$ and $a_{j,k}$. According to equation 5, the utility of device e_j is $\tilde{u}_j = \gamma_{i,k} \times (a_{j,k} - c_{k,j}) = u_j$, so device e_j cannot get a higher utility.

Case 4: $\tilde{a}_{j,k} < a_{j,k}$ and device e_j wins with $a_{j,k}$ and loses with $\tilde{a}_{j,k}$. In this case, the utility of device e_j for asking price $\tilde{a}_{j,k}$ is zero, so device e_j cannot get a higher utility corresponding to $a_{j,k}$.

Case 5: $\tilde{a}_{j,k} < a_{j,k}$ and device e_j wins the auction with both $\tilde{a}_{j,k}$ and $a_{j,k}$. Similar to **Case 3**, we can get $a_{j,k} > \tilde{a}_{j,k} \geq a_{j,k'} \geq b_{i,k}$. According to Equation 4, the payment $c_{k,j}$ is the same in both asking price $\tilde{a}_{j,k}$ and $a_{j,k}$. According to Equation 5, the utility of device e_j is $\tilde{u}_j = \gamma_{i,k} \times (a_{j,k} - c_{k,j}) = u_j$, so the device e_j cannot get a higher utility. □

B. Quality-Oriented Task Allocation Algorithm

In this subsection, we still start with the overview of QTA, which is leveraged to allocate the FL tasks posted by the requesters in $\boldsymbol{R}$ to suitable computationally capable devices in device set $\boldsymbol{E}_i$, and the goal of QTA is to maximize the amount of data under fixed budgets. Then we detail the algorithm process and implement it on the smart contract.

1) Overview of QTA : We get a device set $\tilde{\boldsymbol{E}}_i$ after TCL, and each device in $\tilde{\boldsymbol{E}}_i$ can complete the data training within the deadline T_i. Thus, requester r_i can select devices in device set $\tilde{\boldsymbol{E}}_i$ to participate in FL task k_i. We devise a task allocation strategy based on the greedy strategy named QTA. At any time, requesters select devices that are idle in the market, and give priority to devices with low selling price to participate in the FL tasks.

2) Algorithm Design: We consider that requester r_i has a budget B_i for its task k_i. Each device $e_j \in \tilde{\boldsymbol{E}}_i$ has data volume $g_{i,j}$ and a selling price $v_{i,j}$ for task k_i. In fact, each device e_j will generate a selling price vector $\mathcal{V}_j = \{v_{1,j}, v_{2,j}, \ldots, v_{M,j}\}$ for the smart contract. The smart contract gets a selling price vector $\mathcal{V}_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,\eta}\}$ for requester r_i, where η is total number of prices. The QTA consists of the following three steps:

- **Selling price update:** According to payment $c_{k,j}$ and data volume $\gamma_{i,k}$ of computationally bounded devices, each computationally capable device e_j updates selling price $v_{i,j}$ and data volume $\hat{g}_{i,j}$ for FL task k_i.
- **Computationally capable devices quotation:** Each device $e_j \in \tilde{\boldsymbol{E}}_i$ sends its new selling price $v_{i,j}$ to the smart contract. Smart contract gets a selling price vector $\mathcal{V}_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,\eta}\}$ for requester r_i.
- **Requester greedy selection:** According to selling price vector $\mathcal{V}_i$, the smart contract selects the device by using the greedy strategy until the budget B_i runs out.

The new selling price $v_{i,j}$ in step 1 will be calculated as follows. Due to the limitation of computing resources, computationally bounded devices hope to send data to computationally capable devices to get rewards. The data price of computationally capable devices is the sum of their data and the cost of calculation, while the computationally bounded devices only include their data. Therefore, payment $c_{j,k}$ of computationally bounded device e_k is less than the data price of computationally capable devices in TCL. We assume that computationally capable device e_j will receive data volume set $\boldsymbol{\gamma} = \{\gamma_{i,1}, \gamma_{i,2}, \ldots, \gamma_{i,\mu}\}$ from multiple computationally bounded devices with payment set $\boldsymbol{c} = \{c_{1,j}, c_{2,j}, \ldots, c_{\mu,j}\}$, where μ indicates the number of computationally bounded devices received. The original selling price of computationally capable device e_j is $\tilde{v}_{i,j}$ and the data volume is $g_{i,j}$. We get the total data volume as illustrated in Equation 6 and the new selling price $v_{i,j}$ as illustrated in Equation 7.

$$\hat{g}_{i,j} = g_{i,j} + \sum_{k=1}^{\mu} \gamma_{i,k} \tag{6}$$

$$v_{i,j} = \frac{g_{i,j}}{\hat{g}_{i,j}} \tilde{v}_{i,j} + \sum_{k=1}^{\mu} \frac{\gamma_{i,k}}{\hat{g}_{i,j}} c_{k,j}. \tag{7}$$

With the increase of purchase data from computationally bounded devices, selling price $v_{i,j}$ of computationally capable devices decrease gradually. Besides, if the remaining budget of requester r_i is not enough to purchase all data of device e_j, r_i will purchase a part of the entire data. The purchase percentage $y_{i,j}$ is illustrated in Equation 8.

$$y_{i,j} = \begin{cases} 0, & B_i \geq 0 \\ \dfrac{B_i}{\hat{g}_{i,j} v_{i,j}}, & 0 < B_i < \hat{g}_{i,j} v_{i,j} \\ 1, & B_i \leq \hat{g}_{i,j} v_{i,j} \end{cases} \tag{8}$$

Next, we define the utility of computationally bounded device e_k, computationally capable device e_j and requester r_i. The utility of computationally bounded devices is the difference between payment and unit data price multiplied by the total amount of transaction data, which can be defined as

$$u_k^{limited} = \gamma_{i,k} \left(c_{k,j} - b_{i,k})\right) \tag{9}$$

The actual energy consumed by device e_j to calculate the unit data as

$$\Gamma(e_j) = \kappa n_j f_j^2 \tag{10}$$

where κ is the effective capacitance parameter of computing chipset for device e_j, n_j represents the number of CPU cycles that device e_j processes unit data and f_j indicates the CPU-cycle frequency of device e_j. Thus, the utility of computationally capable device e_j can be defined as

$$u_j^{capable} = \hat{g}_{i,j} y_{i,j} \left(v_{i,j} - \Gamma(e_j)\right) \tag{11}$$

where $y_{i,j}$ is the percentage of the data selected by requester r_i. The utility of requester r_i can be defined as

$$u_i^{requester} = \sum_j y_{i,j} \hat{g}_{i,j} \tilde{v}_{i,j} \tag{12}$$

When device e_j completes task k_i, the utility will comply with Equation 11, where they are greater than or equal to zero. If device e_j fails to participate in task k_i or fails to submit the local updates before the deadline, the utility of computationally capable device e_j and computationally bounded devices sending data to e_j is equal to zero.

A globally optimal solution can be achieved by QTA when requesters post the FL task in a particular order. However, the assumption that there exists a global coordinator to decide the order of requesters is not practical. Even when we consider this assumption to be true, finding the order is an NP-hard problem by itself according to Theorem 1. Instead, we consider an asynchronous market where all requesters post the FL task independently from each other. Let us consider when any requester joins the BFL market. Our proposed greedy strategy ensures that the requester will maximize the training data from all available IoT devices in the current market, *i.e.,* excluding the devices that have been selected by earlier requesters. Therefore, our proposed greedy-selection-based QTA is optimal in an asynchronous market.

3) Smart Contract of QTA : As shown in Algorithm 2, we devise a smart contract for QTA. The input of QTA includes the requester set $\boldsymbol{R}$ and the computationally capable device set $\tilde{\boldsymbol{E}}_i$ generated in TCL, and the output is the selected computationally capable devices set $\tilde{\boldsymbol{E}}_i^c$ by requester r_i. Firstly, each requester r_i sends its budget B_i to the smart contract, and each device $e_j \in \tilde{\boldsymbol{E}}_i$ updates selling price $v_{i,j}$ and maximum data volume $\hat{g}_{i,j}$ (lines 2 to 3). Secondly, each device e_j sends its selling price $v_{i,j}$ and maximum data volume $\hat{g}_{i,j}$ to smart contract (line 4). The smart contract gets all selling prices and sorts them in ascending order to get selling prices vector $\boldsymbol{V}_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,\eta}\}$, where η is total number of price (line 5). Thirdly, each requester r_i purchases training data according to the selling price vector $\boldsymbol{V}_i$ (lines 6 to 18). Once the task allocation is completed, a transaction will be generated, which includes the transaction ID, transaction type, address of device e_j, address of requester r_i, data volume $\hat{g}_{i,j} y_{i,j}$ and selling price $v_{i,j}$. After the transaction is endorsed, it will be recorded on the blockchain in the next block generate cycle.

Algorithm 2 Quality-Oriented Task Allocation Algorithm

```
Input: requester set $\boldsymbol{R}$, computationally capable device set $\tilde{\boldsymbol{E}}_i$
Output: selected device set $\tilde{\boldsymbol{E}}_i^c$
1: $\boldsymbol{E}_i^c \leftarrow \emptyset$
2: Each requester $r_i \in \boldsymbol{R}$ sends the budget $B_i$ to smart contract
3: Each device $e_j \in \tilde{\boldsymbol{E}}_i$ updates the selling price $v_{i,j}$ and data volume $\hat{g}_{i,j}$ for FL task $k_i$
4: Each device $e_j$ sends selling price $v_{i,j}$ and maximum data volume $\hat{g}_{i,j}$ to smart contract
5: For each requester $r_i$, smart contract gets all selling prices and sorts it in ascending order, i.e., $\boldsymbol{V}_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,\eta}\}$, $v_{i,1} \leq v_{i,2} \leq \cdots \leq v_{i,\eta}$, where $\eta$ is the number of selling price
6: Smart contract executes the allocation function:
7: for $a_{i,j}$ in $\boldsymbol{V}_i$ do
8:   if $B_i > 0$ then
9:     if $B_i \geq v_{i,j} \times \hat{g}_{i,j}$ then
10:      $y_{i,j} \leftarrow 1$, $B_i \leftarrow B_i$ - $v_{i,j}\hat{g}_{i,j}y_{i,j}$
11:    else
12:      $y_{i,j} \leftarrow \frac{B_i}{a_{i,j}\hat{g}_{i,j}}$, $B_i \leftarrow 0$
13:    end if
14:    $\tilde{\boldsymbol{E}}_i^c \leftarrow \tilde{\boldsymbol{E}}_i^c \cup e_j$
15:  else
16:    break
17:  end if
18: end for
19: return $\tilde{\boldsymbol{E}}_i^c$
```

After the execution of QTA, we complete the task allocation and establish the task relationship between requester r_i and device set $\tilde{\boldsymbol{E}}_i$. As a result, we generate a device set $\tilde{\boldsymbol{E}}_i^c$, each of which will train the local data for task k_i. Notice that if there are no devices in device set $\tilde{\boldsymbol{E}}_i^c$, the task allocation will fail and restart.

C. Encrypted Model Training Scheme

This subsection proposes EMT to ensure that FL tasks can be securely trained in the BFL market. In order to ensure the consistency, we introduce EMT with the same structure as QTA as follows.

1) Overview of EMT : After requester r_i completes the QTA, it will get its device set $\tilde{\boldsymbol{E}}_i^c$. Each device $e_j \in \tilde{\boldsymbol{E}}_i^c$ will start training after the requester sends them the model and initial parameters with EMT. In EMT, we first propose a simple, countervailable differential privacy noise to encrypt the local update information submitted by computationally capable devices, then we add the validation of local update to ensure that the global model is trained efficiently and stably. After receiving the local update submitted by devices, requester r_i aggregates verified local updates. In this process, the differential privacy noise is eliminated. Finally, requesters carry out the back propagation for global update.

2) Algorithm Design: We get device set $\tilde{\boldsymbol{E}}_i^c = \{e_1, e_2, \ldots, e_\sigma\}$ after QTA, where σ is the total number of device set $\tilde{\boldsymbol{E}}_i^c$. At this stage, we conduct FL on these devices. The EMT consists of the following four steps:

- **FL initialization:** The smart contract sends parameters of the global model W_i, noise δ_j to each device $e_j \in \tilde{\boldsymbol{E}}_i^c$ for FL task k_i. In order to make noise δ_j play a role of the privacy protection, we use $(\varepsilon, \zeta) - differential\ privacy$ mechanism [11]. Noise $\delta_j \in Y \sim N(0, \sigma^2)$ is generated randomly, where $\zeta \in (0,1), \sigma > \sqrt{2ln(2.5/\zeta)\Delta f}/\varepsilon$ and Δf represents the maximum $L2$ distance to query the output of the adjacent dataset. We generate σ noise for task k_i, which are recorded as $\boldsymbol{\delta} = \{\delta_1, \delta_2, \ldots, \delta_\sigma\}$. In order to ensure the elimination of noise after aggregation, we define the noisy key $\tilde{\delta} = -\sum_j \delta_j$.
- **Local training:** Each device e_j conducts the local training and obtain final parameters of the local model $\widetilde{W}_{i,j}$. Then, device e_j calculates parameters difference and completes encryption with differential privacy as illustrated in Equation 13.

  $$\Delta W_{i,j} = \widetilde{W}_{i,j} - W_i + \delta_j. \tag{13}$$

  Lastly, device e_j generates a signature $\varrho_{i,j}$ and sends signature $\varrho_{i,j}$ and parameters difference $\Delta W_{i,j}$ to the smart contract.
- **Model validation:** Requester r_i sends the test data and accuracy threshold θ to the smart contract. Then, the smart contract uses the parameters of the local model $\widetilde{W}_{i,j} = \Delta W_{i,j} + W_i$ to get the accuracy $\vartheta_{i,j}$ of the validation data for each device e_j. The smart contract will discard parameters difference $\Delta W_{i,j}$ and update noisy key $\tilde{\delta} = \tilde{\delta} + \delta_j$ if the accuracy $\vartheta_{i,j}$ is less than threshold θ.
- **Model aggregation:** The smart contract aggregates all parameters of the local model to complete the validation and decrypt it as illustrated in Equation 14.

  $$W_i = W_i + \sum_j \Delta W_{i,j} + \tilde{\delta}. \tag{14}$$

3) Smart Contract of EMT : As described in Algorithm 3, we devise a smart contract for EMT. The input of EMT includes selected device set $\tilde{\boldsymbol{E}}_i^c$, parameters of the global model W_i and accuracy threshold θ, and the output is the updated parameters of the global model W_i'. Firstly, the smart contract generates a differential noise set $\boldsymbol{\delta} = \{\delta_1, \delta_2, \ldots, \delta_\sigma\}$ and a noisy key $\tilde{\delta}$, then it sends differential noise set $\boldsymbol{\delta}$ and initial parameters W_i to device set $\tilde{\boldsymbol{E}}_i^c$ (lines 1 to 2). Secondly, device e_j trains the local model and obtains updated parameters $\widetilde{W}_{i,j}$. Then, device e_j generates a signature $\varrho_{i,j}$ and encrypts the difference of parameters like $\Delta W_{i,j} = \widetilde{W}_{i,j} - W_i + \delta_j$, which is encapsulated with signature $\varrho_{i,j}$ and the final result of the FL task is $\pi_{i,j} = (\Delta W_{i,j} || \varrho_{i,j})$(lines 3 to 7). Thirdly, requester r_i sends validation dataset and accuracy threshold θ to the smart contract. For each received result $\pi_{i,j}$, the smart contract utilizes $\widetilde{W_i} = W_i + \Delta W_{i,j}$ to test the validation dataset and obtain the corresponding accuracy $\epsilon_{i,j}$. If accuracy $\epsilon_{i,j} < \theta$, the smart contract will discard result $\pi_{i,j}$ and update the noisy key $\tilde{\delta} = \tilde{\delta} + \delta_j$(lines 9 to 14). Finally, the smart contract aggregates all local updates and noisy key $\tilde{\delta}$. Because the sum of noisy key $\tilde{\delta}$ and other differential privacy noise are zero, the noise is eliminated in the aggregation. In terms of security, noisy key $\tilde{\delta}$ is generated by the smart contract and saved in the smart contract, so the security can be guaranteed. Then, the smart contract encapsulates its signature ϑ_i and the difference of parameters ΔW_i as $(\Delta W_i || \vartheta_i)$, which is sent to requester r_i to update the global model (lines 15 to 17). At the same time, a transaction can be generated, which includes the transaction ID, transaction type, address of device e_j, address of requester r_i and accuracy $\epsilon_{i,j}$. After the transaction is endorsed, it will be recorded in the ledger and written into the blockchain in the next block generate cycle.

D. Consensus Mechanism

In order to ensure the consistency of transaction records of all devices, we propose a consensus mechanism for the BFL market in this section. We first introduce the conventional PoS consensus mechanism. Then, we propose the contribution model and the contribution-driven DPoS consensus mechanism.

1) Conventional PoS Consensus Mechanism: Although PoW is currently the most widely used consensus algorithm in blockchain platforms, and its reliability has been extensively verified, PoW is not without its flaws. On the contrary, its large consumption of energy has been criticized, and the centralization caused by mining pools has been criticized has also been controversial. To solve these problems, the PoS consensus mechanism comes into being. In the PoS mechanism, each entity use coinage as a measure of its equity. The coinage is defined as $coin \times t$, where t represents time. The more stake an entity has, the more likely it is to become the next block producer. As a result, the PoS no longer requires entities to perform a large number of hash operations, which greatly reduces the energy consumption [12]. However, the characteristics of the PoS mechanism also bring some new problems. Due to the low cost of malicious attacks, the blockchain system is vulnerable to uninterested attacks by entities, and is prone to the Matthew effect [13], thereby increasing the gap between the rich and the poor, and ultimately a few rich devices have the right to generate blocks and mint coins, which aggravates the degree of centralization.

2) Contribution Model for Devices: In order to establish a stable and reliable BFL market, we establish a contribution model to evaluate the service quality of devices in the market. The contribution of each device is associated with FL tasks that the device has participated in the past. Therefore,

Algorithm 3 Encrypted Model Training Scheme

```
Input: selected device set $\tilde{\boldsymbol{E}}_i^c$, parameters of the global model $W_i$, accuracy threshold $\theta$
Output: updated parameters of the global model $W_i'$
The smart contract generates the differential privacy noise set $\boldsymbol{\delta} = \{\delta_1, \delta_2, \ldots, \delta_\sigma\}$ and noisy key $\tilde{\delta}$
The smart contract sends parameters of the global model $W_i$ and noise $\delta_j$ to device $e_j \in \tilde{\boldsymbol{E}}_i^c$ for FL task $k_i$
for $e_j$ in $\tilde{\boldsymbol{E}}_i^c$ do
    device $e_j$ gets parameters of the global model $W_i$ and obtains parameters of the local model $\widetilde{W}_{i,j}$ after the local training
    $e_j$ generates a signature $\varrho_{i,j}$ and encrypts the parameters difference $\Delta W_{i,j} = \widetilde{W}_{i,j} - W_i + \delta_j$
    $e_j$ sends result $\pi_{i,j} \leftarrow (\Delta W_{i,j} || \varrho_{i,j})$ to the smart contract
end for
The smart contract receives result set $\boldsymbol{\pi}_i \leftarrow \{\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,\sigma}\}$, where $\sigma$ is the total number of devices
for $\pi_{i,j}$ in $\boldsymbol{\pi}_i$ do
    The smart contract verifies signature $\varrho_{i,j}$ and tests the validation dataset with parameters of the local model $\widetilde{W}_{i,j} \leftarrow \Delta W_{i,j} + W_i$ to get accuracy $\epsilon_{i,j}$
    if $\epsilon_{i,j} < \theta$ then
        The smart contract discards $\Delta W_{i,j}$ and updates noisy key $\tilde{\delta} = \tilde{\delta} + \delta_j$
    end if
end for
The smart contract aggregates parameters difference $\Delta W_i = \sum_{j=1}^{\sigma} \Delta W_{i,j} + \tilde{\delta}$ and generates a signature $\vartheta_i$.
The smart contract sends $(\Delta W_i || \vartheta_i)$ to requester $r_i$
Requester $r_i$ updates parameters of the global model $W_i' \leftarrow W_i + \Delta W_i$
return $W_i'$
```

the contribution can intuitively reflect the service quality of devices. For FL task k_i, the contribution $\ell_{i,j}$ of devices e_j can be measured by the accuracy $\epsilon_{i,j}$ of the local update submitted by devices on the testset, which is recorded in transactions on the blockchain. Considering that accuracy $\epsilon_{i,j} \in [0, 1]$, in order to reward devices that provide high-quality data and punish devices that provide low-quality data, we define the contribution $\ell_{i,j} \in [-1, 1]$ of each device e_j. Referring to the calculation method of the trustworthiness, we also propose a contribution calculation equation based on entropy as illustrated in Equation 15, and $\ell_{i,j}$ is $\psi(\epsilon_{i,j}) - 1$ when $\epsilon_{i,j}$ is greater than or equal to 0 and less than 0.5, and $\ell_{i,j}$ is $1 - \psi(\epsilon_{i,j})$ otherwise.

$$\psi(\epsilon) = -\epsilon \log_2(\epsilon) - (1 - \epsilon) \log_2(1 - \epsilon) \tag{15}$$

Since device e_j may have different contributions in each task, the average contribution is used to measure the contribution of device.

$$\tilde{\ell_{i,j}} = \frac{1}{k} \sum_{j=1}^{k} \ell_{i,j} \tag{16}$$

3) Contribution-Driven DPoS Consensus Mechanism: The conventional PoW/PoS mechanism has certain limitations, whether it is the computing power that dominates in PoW, or the entity that owns a large number of coins in PoS, can obtain the right to verify the transaction and get rewards from it [14]. Therefore, the conventional PoW/PoS mechanism motivates devices in the market to pursue either computing power or coins, both of which ignore the issue of quality of service, which is crucial in the market. To solve this problem, we define the stake as the weighted sum of the coins accumulated in the transaction and the average contribution in the FL task, and propose a contribution-driven DPoS consensus mechanism.

We define the stake S_j of device e_j as Equation 17 and the vote distribution strategy as Equation 18 in our proposed contribution-driven DPoS consensus mechanism. On the one hand, the introduction of the contribution model makes the devices no longer take the pursuit of the data volume as the only goal, and the data quality also affects their stakes. As a result, the devices not only tend to provide more data, but also to provide high-quality data, which reduces the rounds of the global model training, so as to reduce the energy consumption of the BFL market. On the other hand, considering that computationally bounded devices are incapable of storing and calculating the distributed ledgers, the stake of computationally bounded devices are delegated to winners of the latest auction in TCL, and those computationally capable devices as their delegators.

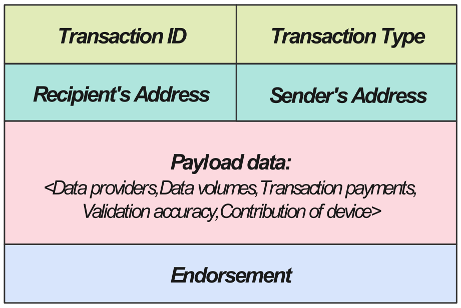
$$S_j = w \times coinage + (1 - w) \times \sum_i \tilde{\ell_{i,j}} \tag{17}$$

$$\chi_j = \frac{S_j}{\sum_{i=1}^{N} S_i} \tag{18}$$

In addition, if device e_j fails to complete task k_i within the deadline T_i, it will be punished, such as being unable to get any payment. If stake S_j of device e_j are less than zero, it will not be allowed to participate in the data sharing and model training. In this way, the market access threshold can be raised to prevent the emergence of a large number of computationally bounded devices in the market and disrupt the reasonable competition order.

The voting process of our proposed contribution-driven DPoS consensus mechanism is divided into four steps:

- **Proportion of votes:** Requester r_i independently chooses a contribution weight w according to its preferences for competence and integrity. Specially, if the weight $w = 1$, the stake of devices is only determined by the coinage. If the weight $w = 0$, the stake of devices is only determined by the average contribution. If the weight $w \in (0, 1)$, the stake of the device is determined by the weighted sum of coinage and the average contribution, and as the w increases, the influence of coinage is becoming more and more prominent. After the smart contract determines weight w, the stake of all devices can be calculated as illustrated in Equation 17 and the proportion of votes can be calculated as illustrated in Equation 18.



Transaction ID	Transaction Type
Recipient's Address	Sender's Address
Payload data: <Data providers,Data volumes,Transaction payments, Validation accuracy,Contribution of device>	
Endorsement	

Fig. 4. The data structure of transactions on the blockchain.

- **Election delegators:** Computationally bounded devices select the winners in the latest data auction as the delegators according to the transaction records and vote for them. As a result, the stake of delegator e_j is $S_j = \sum_k S_k$, where S_k is the stake of the computationally bounded devices.
- **Voting:** All delegators compete for the right to generate the next block according to the consensus mechanism, and the winner gets the reward.
- **Reward distribution:** Once the block is confirmed, the block publisher will receive the corresponding reward, who will distribute the reward to its supporters according to the Shapley value [15].

Meanwhile, the system stores the transactions generated by above algorithms on the blockchain. Here, we give the data structure of these transactions. As shown in Figure 4, the data structure of the transaction includes transaction ID, transaction type, recipient's address, sender's address, payload data, and endorsement. The transaction ID is the unique identifier of the current transaction record. The transaction type includes TCL, QTA, and EMT. The payload data of TCL and QTA includes their respective data volume and transaction fee, and EMT records the validation accuracy. The endorsement indicates the validity of the transaction. In the BFL market, the information transfer between devices will be written on blockchain according to the data structure.

IV. Performance Evaluation

This section evaluates the performance of our proposed algorithms. Firstly, we introduce the default settings, dataset, benchmarks and metrics in detail. Secondly, to demonstrate that our proposed algorithms can ensure an FL task to maximize the amount of training data with given budgets, we evaluate the performance of the BFL market on data utilization, average model accuracy and total utility of all requesters. We also evaluate the performance of the proposed contribution-driven DPoS consensus mechanism on average reward percentile of poor devices to demonstrate that our proposed consensus mechanism can guarantee the fairness of reward distribution in the block generation and reduce the wealth inequality among devices. Lastly, we evaluate the performance of blockchain in the execution time and throughput of the auction and the FL task allocation.

A. Evaluation Default Settings

This subsection describes the default settings for our evaluations. We consider the BFL market consisting of requesters and devices. To reduce the parameter search space, we set the number of requesters is 50 and that of devices is 500 in our evaluation. Considering that the storage resources of computationally bounded devices may also be limited, we set that the average amount of data owned by computationally bounded devices is half of computationally capable devices, and the unit price of data is also half of computationally capable devices. With reference to [16], the trustworthiness threshold is randomly selected over [0,1] and the number of devices within the communication range of each device is randomly selected over [4,10]. We randomly set the duration of the FL tasks posted by requesters, and we ensure that each requester posts the FL task at least once. We set the number of global model iterations to 200 via mini-batch SGD optimizer, and the size of mini-batch is set to 50 with reference to [17].

B. Dataset

In this subsection we introduce the dataset used in our evaluations. We utilize a well-known image classification dataset named CIFAR-10 [18], which consists of 60000 32×32 colour images in 10 classes with 50000 training images and 10000 test images, and per class includes 6000 images. We separate the training set of CIFAR-10 into devices during the initialization phase. Specifically, each device samples without replacement from the scrambled training set according to the initialized data volume. In order to match the training set and devices, the mean of the devices at initialization is set as the total data volume of the training set divided by the total number of devices. We train the model according to part of the data purchased by requesters, and test the average accuracy of the global model on test set. Besides, in order to reasonably simulate the direct trust probability between devices, we utilize the social connectedness index (SCI), a measure of the social connectedness between different geographies [19], between Facebook users as the initial trust probability of different devices. Precisely, SCI measures the intensity of connectedness between locations and reflects the relative probability that two individuals across two locations become friends with each other. Thus, SCI can properly simulate the initial trust probability between two devices.

C. Benchmarks

This subsection introduces the benchmarks of our evaluations. To the best of our knowledge, our paper is the first to optimize FL with both blockchain and SIoT. Hence, there are no similar benchmark algorithms. To evaluate the effectiveness of the proposed algorithms, on the one hand, we compare the effectiveness of the TCL and QTA. For TCL, due to TCL is an improved algorithm devised based on the first-price sealed auction (FPSB) [20], it is suitable to select FPSB as the benchmark. In FPSB, the requester pays the winner the highest asking price, whereas in TCL the requester pays the second highest asking price. Compared with the FPSB, TCL can better reflect the real asking price of the buyer for the data. For QTA, we choose not to use QTA as the benchmark, that is, the requester randomly selects computationally capable devices to participate in its FL task. On the other hand, we compare the impact of whether data sharing is allowed in the market. Specifically, random selection and QTA are the situations where data sharing is not allowed, and FPSB+TCL and QTA+TCL are the situations where data sharing is allowed. As a result, we choose these four cases for comparative evaluation. In addition, DPoS is an improvement algorithm to PoS, and we introduce the contribution degree for DPoS in this paper. In order to compare the respective effectiveness of DPoS and contribution, we choose four combinations of PoS, Contribution-based PoS, DPoS, and Contribution-based DPoS for comparison.

D. Metrics

In this subsection, we describe the metrics in our evaluations. Firstly, We evaluate the effectiveness of algorithms we proposed in the BFL market with three basic metrics, including the data utilization, the average model accuracy, and the total utility of requesters. The data utilization represents the amount of data purchased by the requesters divided by the total amount of data in the BFL market. The model accuracy represents the average global model accuracy obtained by the requesters in the market and measures the training quality of FL tasks. The total utility of requesters can be calculated by Equation 12, which reflects the total value of the data obtained by the requesters. Secondly, we evaluate the average reward percentile of poor devices, which is equal to the average reward received by poor devices in the process of generating blocks divided by the average reward received by all devices. Here, the poor devices are those whose coinage is significantly lower than the median. The average reward percentile of poor devices can measure the effectiveness of our proposed consensus mechanism in reducing the gap between the rich and the poor devices in the market. Thirdly, we evaluate the performance of the execution time and the throughput of the data auction and FL task in the BFL market. The execution time of the data auction represents the average time for each round of auction, and the execution time of FL task allocation is the average time spent by requesters in selecting computationally capable devices. The throughput of data auction represents the number of data auctions that can be completed per second, and throughput of FL task allocation represents the number of FL tasks that are successfully allocated per second.

E. Data Utilization

To demonstrate that our proposed algorithms make full use of the data of all devices under fixed budgets for requesters, we adjust the percentage of computationally bounded devices in the market from 10% to 80% to observe the data utilization under different scenarios. The evaluation conditions follow the default settings. As shown in Figure 5, we consider four scenarios where the requesters randomly select devices to participate in FL tasks (dashed line and circle marks), select devices by using QTA (dashed line and star marks), complete data sharing by using FPSB and then select devices by using QTA (solid line and triangle marks), and complete data sharing by using TCL and then select devices by using QTA (solid line and cross marks), respectively. Notice that there is no unit for data utilization, or its unit can be considered as 1, and the data utilization may exceed one because the data of each device in the market can be reused by multiple requesters.

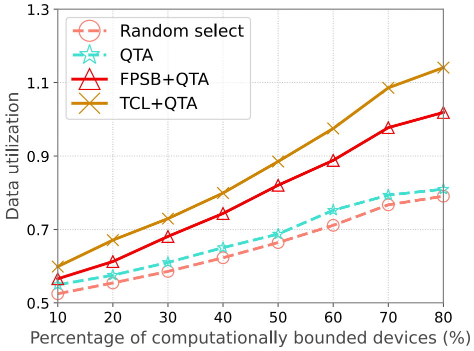


Fig. 5. Performance on data utilization.

We use numerical simulation to simulate the data utilization of the market in a fixed period of time. When the percentage of computationally bounded devices is relatively low, the data utilization is not much different in the four scenarios because the amount of data owned by computationally bounded devices only account for a small proportion of the total amount of data in the market, and hence whether to allow data sharing has little impact on the data utilization of the market. Nevertheless, we observe that QTA brings higher data utilization than random selection. When the percentage of computationally bounded devices increases gradually, the data utilization under scenarios of FPSB+QTA and TCL+QTA are significant higher than other two scenarios because allowing data sharing enables computationally bounded devices to send their data to computationally capable devices through auction, and requesters can choose computationally capable devices to participate in FL tasks at a lower price, so as to purchase more data under fixed budgets.

We further explain the performance difference in a more intuitive way. We first look at FPSB+QTA and TCL+QTA. Because TCL takes the second highest price as the payment of the winner, it helps to reduce the unit data price of the computationally capable devices. As a result, the requesters can purchase more data under the same budget compared to FPSB. It thus follows that the TCL obtains higher data utilization than FPSB. We now look at QTA and random selection. Given that there is no data sharing, the unit price of data of the computationally capable device remains the same, regardless of the changes of the portion of computationally bounded devices. Thus, the amount of data available to the requesters remains barely changed. Nevertheless, because the requesters choose a lower data price under QTA scenario, the requesters get more data under the same budget. Hence, the data utilization of QTA is higher than the random selection.

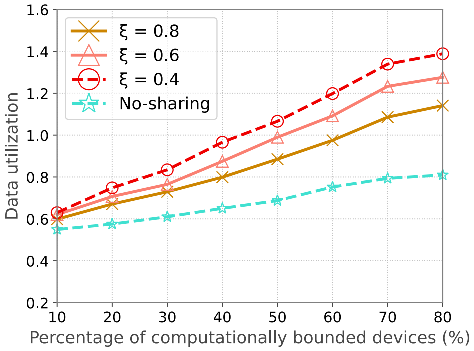


Fig. 6. Performance on data utilization by different trustworthiness thresholds ξ.

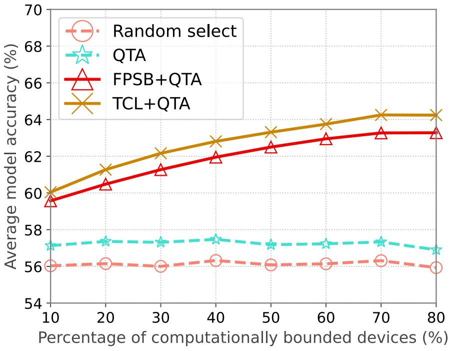


Fig. 7. Performance on average model accuracy.

The above results demonstrate that our proposed algorithms improve the data utilization in the BFL market. Notice that when the percentage of computationally bounded devices reaches 70% or more, the growth of data utilization slows down in all scenarios. The reason is that almost all the computing resources in the market are occupied. Hence, although the number of computationally bounded devices is still increasing, there are no computationally capable devices that have extra computing resources for the additional data.

Next, we evaluate the impact of different trustworthiness thresholds of computationally bounded devices on data utilization. As shown in Figure 6, we set the trustworthiness threshold to 0.4, 0.6 and 0.8 respectively. In fact, the difference in data utilization caused by the change of trustworthiness threshold is the result of the joint influence of the trustworthiness model and mobility model. After the trustworthiness threshold is changed, the newly added trusted devices may not be able to share data due to the limitation of the communication range. When the percentage of computationally bounded devices is relatively low, the data utilization where there is no data sharing is obviously lower than that in other cases, and the impact of different trustworthiness thresholds is not obvious. But as the percentage of computationally bounded devices increases, the impact of different trustworthiness thresholds increases as well. The lower the trustworthiness threshold, the more sufficient the data circulation in the market. As more computationally bounded devices send their data to surrounding trusted devices, the data utilization in the market becomes higher.

F. Average Model Accuracy

In this evaluation, we conduct the federated learning by sampling from the dataset according to the amount of data purchased by the requester and calculate the average model accuracy in the above four scenarios. The evaluation conditions follow the default settings as well. As shown in Figure 7, no matter how the percentage of computationally bounded devices changes, the average model accuracy under scenarios of random selection and QTA is barely changed and the accuracy for random selection is lower than that of the QTA, overall. Next, we look at the FPSB+QTA and TCL+QTA. The average model accuracies under scenarios of FPSB+QTA and TCL+QTA increase as the percentage of computationally bounded devices increases because of higher data utilization. Precisely, as a result of that TCL takes the second highest price as the payment of the winner, the requesters purchase more data with the same budget compared to the FPSB through analysis in Section IV-E. It thus follows that the TCL obtains higher average model accuracy than FPSB. Besides, because of the excessive number of computationally bounded devices, some of the computationally bounded devices cannot complete data auction successfully when the percentage of computationally bounded devices exceeds 70%. Therefore, the growth of data obtained by requesters slows down, and the growth of model accuracy obtained by requesters slows down as well. The evaluation shows that the average model accuracy of TCL+QTA is about 1.1% higher than FPSB+QTA, about 7% higher than the QTA and about 8% higher than the random selection. Hence, our proposed algorithms can make full use of the data of computationally bounded devices and improve the average model accuracy of requesters under fixed budgets.

G. Total Utility of Requesters

To demonstrate that our proposed algorithms can increase the utility of the requester, we adjust the percentage of computationally bounded devices in the market from 10% to 80% to calculate the total utility of requesters in different scenarios. As shown in Figure 8, similar to the trend of model accuracy, the total utility of requesters under scenarios of random selection and QTA is basically unchanged as the percentage of computationally bounded devices increases. The reason behind the similar trends is that both model accuracy and total utility of requesters are positively correlated with the total amount of data obtained by the requesters. Under the scenarios of TCL+QTA and FPSB+QTA, as the percentage of computationally bounded devices increases, the selling price of computationally capable devices decreases, and requesters are able to buy more data under fixed budgets and obtain higher utility according to Equation 7. Therefore, the total utility of requesters increases in the scenarios of TCL+QTA and FPSB+QTA as the percentage of computationally bounded devices increase, and that under TCL is higher than that under FPSB as the previous evaluation. This evaluation shows that the total utility of requesters in the case of TCL+QTA is about

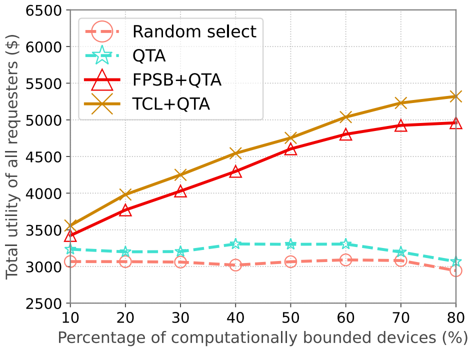


Fig. 8. Performance on total utility of requesters.

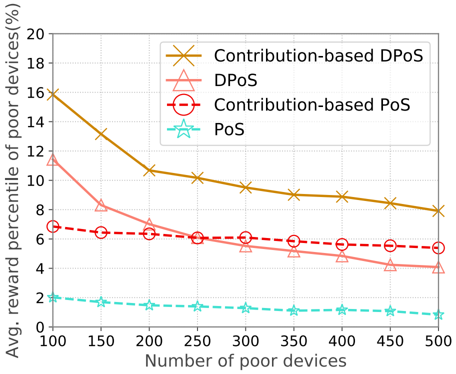


Fig. 9. Performance on average reward percentile of poor devices.

7% higher than that of FPSB+QTA, 74% higher than that of QTA and 116% higher than that of random selection.

H. Average Reward Percentile of Poor Devices

This subsection evaluates the average reward percentile of poor devices. If a market is monopolized by a small number of rich devices, fair competition rules of the market will be broken. More and more devices will eventually exit the market, and the training of FL tasks cannot be completed. Therefore, we consider that such a market is unstable. To demonstrate that our proposed consensus mechanism reduces wealth inequality and promotes the market stability, we compare the average reward of poor devices with four consensus mechanisms, as shown in Figure 9. We consider $1,000$ devices including poor devices, rich devices and ordinary devices in our evaluation, where the ten rich devices own a fifth of total coinage, and we adjust the number of poor devices in the market from 100 to 500 to evaluate the average reward percentile of poor devices [21]. We first simulate the BFL market running for a period of time and generate a fixed number of blocks. We calculate the average reward of each poor, rich and ordinary device during this period, then compute the percentile of reward of each poor device as the y-axis.

We first compare all four consensus mechanisms. Compared with the other three consensus mechanisms, we find that the contribution-driven DPoS mechanism can allocate more reward to poor devices. For example, when the number of poor devices in the market is 100, our proposed contribution-driven DPoS is 4.3% higher than DPoS, 9.1% higher than contribution-driven PoS, and 13.9% higher than PoS in the average reward of poor devices. This is most likely due to that unlike the conventional PoS consensus mechanism that only considers coinage, the stake of each device equals to the weighted sum of contribution and coinage in contribution-driven PoS consensus mechanism. Moreover, our proposed contribution-driven DPoS mechanism provides an incentive for both computationally bounded devices and computationally capable devices, and encourages them to provide data of a higher quality for FL tasks. Thus, our proposed consensus mechanism can reduce the wealth gap between the rich and poor devices and the average reward percentile of poor devices decreases as their number increases.

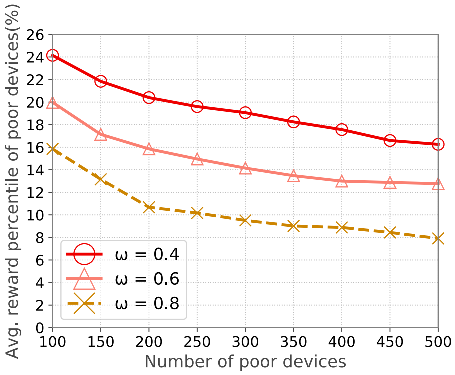
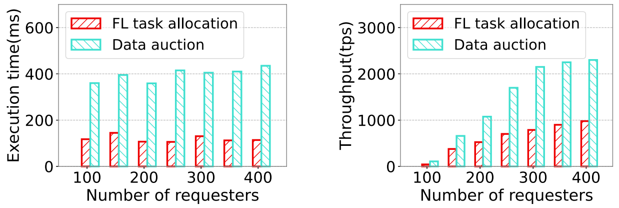


Fig. 10. Performance on average reward percentile of poor devices by different contribution weights w.

In addition, we also evaluate the impact of different contribution weights w in our proposed contribution-driven DPoS mechanism on the average reward of poor devices as shown in Figure 10. We have the same settings as in Figure 9, and assign 0.4, 0.6 and 0.8 to the weight w in Equation 17. The evaluation shows that with the increase of contribution weight, the proportion of contribution increases, and the average reward of poor devices increases. For example, when the number of poor devices in the market is 100, the average reward when $w = 0.4$ is 3.7% higher than that when $w = 0.6$, and 8% higher than that when $w = 0.8$ in the average reward of poor devices.

I. Performance of Blockchain

To demonstrate our BFL market is stable and efficient, we adjust the number of requesters from 100 to 400 to evaluate the overall system performance by simulating the execution time and throughput of data auctions and task allocations. For TCL, if a computationally bounded device is not chosen by its potential winner, it will conduct the next auction round. In the evaluation, we set the maximum number of rounds to a fixed number, which is decided by computationally bounded devices. As shown in Figure 11, with the increase of the number of requesters, the average execution time of the data auction and FL task allocation remain at a stable millisecond level, which is negligible for the training duration of FL tasks. In order to better reflect the performance of blockchain,



(a) Execution time

(b) Throughput

Fig. 11. Execution time and throughput of data auction and FL task allocation in blockchain.

we increase the number of requesters and devices in the same proportion in the evaluation. When the number of requesters exceeds 300, some FL tasks posted by requesters cannot be allocated immediately, as a result of the limitation of computing and communication resources of the blockchain. Therefore, the throughput tends to be stable and can be maintained at a high level. Based on the above analysis, our proposed BFL market is stable and efficient.

V. Related Work

Google is the first to propose the FL framework which goal is to develop a virtual keyboard for smartphones named Gboard [22] aiming to provide auto-correction, word completion, and next-word prediction features. In recent years, FL has been studied in various scenarios extensively. Wang et al. [23] propose an FL method between different retailers to train a high-precision model, which identifies the demographic characteristics of electric power consumers via extracting information from smart meter data, and then provides services for them. Li et al. [24] promote healthcare services and solve the problem of data islands between medical institutions. FL creates a collaborative medical environment between different hospitals to accelerate patient diagnosis and treatment without sacrificing user privacy.

FL can provide many advantages in the IoT scenario, so there are many works devoted to applying FL to various scenarios. In this scenario, FL mainly trains the global model on a large number of IoT devices and protects the data privacy of all IoT devices through localized training [25], [26]. With the support of FL, Otoum et al. [27] utilize the computing, communication and intelligent functions of IoT devices to realize energy trading and remote monitoring. Qu et al. [28] find a balance between privacy and low efficiency of fog computing, and propose a solution based on FL to improve the performance of fog computing. Lin et al. [29] propose an FL approach in intelligent healthcare with high data integrity and low privacy leakage. However, these works do not consider the limited computing resources widely existing in IoT devices, and they assume that all IoT devices can participate in FL tasks. In practice, many IoT devices are resource limited [30], which are not guaranteed to complete the model training within the deadline. Moreover, Nguyen et al. [31] also find many problems of federated learning in the IoT scenario, including the communication cost, security and privacy protection, *etc.*

Besides, decentralized FL has also attracted extensive attention due to the problems of single-point failure, communication bottleneck and trust in server centered FL paradigm. As a distributed computing architecture, blockchain has been widely studied in the Internet of things [32], [33], [34], [35] recently. Therefore, researchers have leveraged blockchain into FL to achieve a practical decentralized and secure solution in recent years. Li et al. [36] propose a blockchain based FL framework, which avoids centralized server and reduces attacks from malicious nodes. Khan et al. [37] utilize the blockchain to achieve the high-quality FL, in which the game based incentive mechanism can maximize the user utility in the set number of iterations, and the base station can maximize the FL performance using the user's best response strategy. Cui et al. [38] devise a decentralized asynchronous FL framework of blockchain authorization for anomaly detection in the IoT scenario, which ensures data integrity and prevents a single point of failure. Cross-Device FL [39] realizes the decentralized system by using the blockchain to protect the reputation of participating devices. Cao et al. [40] propose an asynchronous FL based on directed acyclic graph (DAG) named DAG-FL, which improves the efficiency of FL, and its special consensus algorithm avoids extra computational consumption. Zhang et al. [41] propose an FL method based on blockchain for device fault detection in industrial IoT, which solves the problem of heterogeneous data and achieves satisfactory results in accuracy and performance. However, the devices lack motivations to participate in FL because of their rationality and they need benefits from the requester as motivation, such as money.

On the safety of FL, Ma et al. [42] put forward challenges in FL application, including information disclosures and malicious attacks. On the one hand, the attackers pretend to be a participant to attack the FL model, resulting in a significant reduction in the accuracy of the model. On the other hand, the attackers can deduce the original data information through a small part of the original gradient information. Existing researches develop privacy protection solutions through differential privacy [43]. Unfortunately, the introduction of differential privacy noise may affect the performance of the global model. Sun et al. [44] utilize a homomorphic encryption to encrypt the parameters, and the malicious devices cannot infer the original data information according to the ciphertext, which ensures the security of the data level. However, the efficiency of homomorphic encryption is brutal to improve and the homomorphic multiplication of ciphertext through tensor product operation will lead to a sharp expansion of the ciphertext dimension.

VI. Conclusion

In this paper, we propose a set of algorithms in the BFL market aiming to make data in computationally bounded devices available for federated learning with social Internet of things, and ensure each FL task maximizes the amount of training data with fixed budgets. We propose a trust-enhanced collaborative learning strategy (TCL) and a quality-oriented task allocation algorithm (QTA), where TCL enables training data sharing among trusted devices with social Internet of things, and QTA guides the FL task allocation to devices and maximizes the training quality with fixed budgets. To ensure the long-term stable operation of TCL and QTA in BFL market, we devise an encrypted model training scheme (EMT) to prevent the attack from malicious devices, and a contribution-driven delegated proof of stake (DPoS) consensus mechanism to guarantee the fairness of reward distribution by reducing the wealth inequality. As a result, TCL and QTA theoretically achieve our goal. EMT and DPoS ensure the security and fairness of the BFL market respectively, so that TCL and QTA can bring tangible benefits. To sum up, the four algorithms complement each other and jointly achieve the ultimate goal of the BFL market. Finally, extensive evaluations are conducted to show that the proposed BFL could improve the total utility of all requesters by 65.7% on average compared with the benchmarks while improving the overall FL model training accuracy.

REFERENCES

[1] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, “Federated learning,” *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 13, no. 3, pp. 1–207, 2019.

[2] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, “Fast-convergent federated learning,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[3] Z. Qin, G. Y. Li, and H. Ye, “Federated learning and wireless communications,” *IEEE Wireless Commun.*, vol. 28, no. 5, pp. 134–140, Oct. 2021.

[4] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, “A review of applications in federated learning,” *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106854.

[5] (2021). *Is More Data Always Better for Building Analytics Models?*. [Online]. Available: https://analyticsindiamag.com/is-more-data-always-better-for-building-analytics-models/#:~:text=Having%20more%20data%20certainly%20increases,natural%20noise%20of%20the%20data

[6] L. Atzori, A. Iera, G. Morabito, and M. Nitti, “The social Internet of Things (SIoT)—When social networks meet the Internet of Things: Concept, architecture and network characterization,” *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128612002654

[7] Q. Li et al., “A survey on federated learning systems: Vision, hype and reality for data privacy and protection,” *IEEE Trans. Knowl. Data Eng.*, early access, Nov. 2, 2021, doi: 10.1109/TKDE.2021.3124599.

[8] N.-T. Nguyen and B.-H. Liu, “The mobile sensor deployment problem and the target coverage problem in mobile wireless sensor networks are NP-hard,” *IEEE Syst. J.*, vol. 13, no. 2, pp. 1312–1315, May 2018.

[9] Y. L. Sun, Z. Han, W. Yu, and K. J. R. Liu, “A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks,” in *Proc. 25TH IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2006, pp. 1–13.

[10] X. Zhou and H. Zheng, “TRUST: A general framework for truthful double spectrum auctions,” in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 999–1007.

[11] J. He, L. Cai, and X. Guan, “Differential private noise adding mechanism and its application on consensus algorithm,” *IEEE Trans. Signal Process.*, vol. 68, pp. 4069–4082, 2020.

[12] D. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. L. Njilla, “Data provenance in the cloud: A blockchain-based approach,” *IEEE Consum. Electron. Mag.*, vol. 8, no. 4, pp. 38–44, Jul. 2019.

[13] R. K. Merton, “The Matthew effect in science: The reward and communication systems of science are considered,” *Science*, vol. 159, no. 3810, pp. 56–63, Jan. 1968.

[14] Y. Huang, Y. Zeng, F. Ye, and Y. Yang, “Incentive assignment in hybrid consensus blockchain systems in pervasive edge environments,” *IEEE Trans. Comput.*, vol. 71, no. 9, pp. 2102–2115, Sep. 2021.

[15] C. Huang, X. Mi, and B. Kang, “Basic probability assignment to probability distribution function based on the Shapley value approach,” *Int. J. Intell. Syst.*, vol. 36, no. 8, pp. 4210–4236, Aug. 2021.

[16] M. Wang, G. Wang, Y. Zhang, and Z. Li, “A high-reliability multifaceted reputation evaluation mechanism for online services,” *IEEE Trans. Services Comput.*, vol. 12, no. 6, pp. 836–850, Nov. 2019.

[17] C. Xu, S. Liu, Z. Yang, Y. Huang, and K.-K. Wong, “Learning rate optimization for federated learning exploiting over-the-air computation,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3742–3756, Dec. 2021.

[18] G. H. A. Krizhevsky and V. Nair. *The Cifar-10 Dataset*. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[19] Facebook. (2021). *Facebook Social Connectedness Index*. [Online]. Available: https://data.humdata.org/dataset/social-connectedness-index#

[20] W. Zhang, X. Wang, G. Han, Y. Peng, and M. Guizani, “SFPAG-R: A reliable routing algorithm based on sealed first-price auction games for industrial Internet of Things networks,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 5016–5027, May 2021.

[21] Y. Du et al., “Blockchain-aided edge computing market: Smart contract and consensus mechanisms,” *IEEE Trans. Mobile Comput.*, early access, Jan. 4, 2022, doi: 10.1109/TMC.2021.3140080.

[22] A. Hard et al., “Federated learning for mobile keyboard prediction,” 2018, *arXiv:1811.03604*.

[23] Y. Wang, I. L. Bennani, X. Liu, M. Sun, and Y. Zhou, “Electricity consumer characteristics identification: A federated learning approach,” *IEEE Trans. Smart Grid*, vol. 12, no. 4, pp. 3637–3647, Jul. 2021.

[24] J. Li et al., “A federated learning based privacy-preserving smart healthcare system,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2021–2031, Mar. 2022.

[25] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, “Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, Jun. 2021.

[26] S. A. Rahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, “A survey on federated learning: The journey from centralized to distributed on-site learning and beyond,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2020.

[27] S. Otoum, I. A. Ridhawi, and H. Mouftah, “A federated learning and blockchain-enabled sustainable energy-trade at the edge: A framework for industry 4.0,” *IEEE Internet Things J.*, early access, Jan. 5, 2022, doi: 10.1109/JIOT.2022.3140430.

[28] Y. Qu et al., “Decentralized privacy using blockchain-enabled federated learning in fog computing,” *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5171–5183, Jun. 2020.

[29] H. Lin, K. Kaur, X. Wang, G. Kaddoum, J. Hu, and M. M. Hassan, “Privacy-aware access control in IoT-enabled healthcare: A federated deep learning approach,” *IEEE Internet Things J.*, early access, Sep. 15, 2021, doi: 10.1109/JIOT.2021.3112686.

[30] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, “Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 668–682, Mar. 2019.

[31] D. C. Nguyen et al., “Federated learning meets blockchain in edge computing: Opportunities and challenges,” *IEEE Internet Things J.*, vol. 8, no. 16, Aug. 2021, pp. 12806–12825.

[32] B. Yin, Y. Wu, T. Hu, J. Dong, and Z. Jiang, “An efficient collaboration and incentive mechanism for Internet of vehicles (IoV) with secured information exchange based on blockchains,” *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1582–1593, Mar. 2020.

[33] Y. Wu, H.-N. Dai, and H. Wang, “Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2300–2317, Feb. 2021.

[34] Y. Wu, Z. Wang, Y. Ma, and V. C. M. Leung, “Deep reinforcement learning for blockchain in industrial IoT: A survey,” *Comput. Netw.*, vol. 191, May 2021, Art. no. 108004.

[35] Y. Wu, H.-N. Dai, H. Wang, and K.-K.-R. Choo, “Blockchain-based privacy preservation for 5G-enabled drone communications,” *IEEE Netw.*, vol. 35, no. 1, pp. 50–56, Jan. 2021.

[36] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, “A blockchain-based decentralized federated learning framework with committee consensus,” *IEEE Netw.*, vol. 35, no. 1, pp. 234–241, Jan. 2021.

[37] L. U. Khanet al., “Federated learning for edge networks: Resource optimization and incentive mechanism,” *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, Oct. 2020.

[38] L. Cui et al., “Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3492–3500, May 2021.

[39] M. H. U. Rehman, A. M. Dirir, K. Salah, E. Damiani, and D. Svetinovic, "TrustFed: A framework for fair and trustworthy cross-device federated learning in IIoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8485–8494, Dec. 2021.

[40] M. Cao, B. Cao, W. Hong, Z. Zhao, X. Bai, and L. Zhang, "DAG-FL: Direct acyclic graph-based blockchain empowers on-device federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[41] W. Zhang et al., "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2021.

[42] C. Maet al., "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul./Aug. 2020.

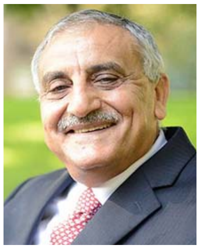
[43] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 968–979, May 2020.

[44] J. Cui, L. Wei, H. Zhong, J. Zhang, Y. Xu, and L. Liu, "Edge computing in VANETs—An efficient and privacy-preserving cooperative downloading scheme," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1191–1204, Jun. 2020.

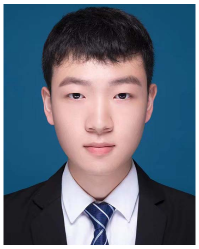
Pengfei Wang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in software engineering from Northeastern University (NEU), China, in 2013, 2015, and 2020, respectively.

From 2016 to 2018, he was a Visiting Ph.D. Student with the Department of Electrical Engineering and Computer Science, Northwestern University, IL, USA. He is currently an Associate Professor with the School of Computer Science and Technology, Dalian University of Technology (DUT), China. He has authored more than 30 papers on high-quality journals and conferences, such as IEEE/ACM Transactions on Networking, IEEE INFOCOM, IEEE Transactions on Intelligent Transportation Systems, DAC, IEEE ICNP, IEEE ICDCS, IEEE Internet of Things Journal, and JSA. He also holds a series of patents in U.S. and China. His research interests are ubiquitous computing, big data, and AIoT.

Yian Zhao was with the Dalian University of Technology (DUT), China, in 2019, where he is currently a Junior Undergraduate with the School of Computer Science and Technology. His current research interests include blockchain and federated learning.



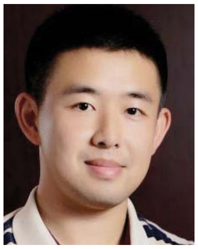
Mohammad S. Obaidat (Life Fellow, IEEE) received the M.S. and Ph.D. degrees in computer engineering from The Ohio State University, Columbus, OH, USA. He is now the Founding Dean of the College of Computing and Informatics, The University of Shrajah, United Arab Emirates. He has received extensive research funding and published to date (2019) about 1,000 refereed technical articles-about half of them are journal articles, over 70 books, and over 70 book chapters. He is a fellow of SCS. He is the editor-in-chief of three scholarly journals and an editor of many other international journals.



Zongzheng Wei received the B.S. degree from the Hebei University of Technology, China, in 2022. He is currently pursuing the master's degree with the School of Computer Science and Technology, Dalian University of Technology (DUT), China. His current research interests include federated learning.

Heng Qi (Senior Member, IEEE) received the B.S. degree from Hunan University in 2004 and the M.E. and Ph.D. degrees from the Dalian University of Technology, China, in 2006 and 2012, respectively.

He has been a JSPS Oversea Research Fellow with the Graduate School of Information Science, Nagoya University, Japan, from 2016 to 2017. He is currently an Associate Professor at the School of Computer Science and Technology, Dalian University of Technology. He has published over 100 technical papers, such as the IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Transactions on Multimedia, and INFOCOM. His research interests include computer networking and multimedia computing.



Chi Lin (Senior Member, IEEE) received the B.E. and Ph.D. degrees from the Dalian University of Technology (DUT), China, in 2008 and 2013, respectively.

He has been an Assistant Professor with the School of Software, DUT, since 2014, where he has been an Associate Professor since 2017. He has authored over 50 scientific papers including INFOCOM, SECON, ICDCS, IEEE Transactions on Mobile Computing and *ACM Transactions on Embedded Computing Systems*. His research interests include pervasive computing, cyber-physical systems (CPS), and wireless sensor networks. In 2015, he was awarded ACM Academic Rising Star.

Yunming Xiao received the B.S. degree in computer science from the Beijing University of Posts and Telecommunications, China, in 2019. He is currently pursuing the Ph.D. degree with the Computer Science Department, Northwestern University, USA. His research interests include network measurement and edge network design.

Qiang Zhang (Member, IEEE) received the B.S. degree in electronic engineering and the M.S. and Ph.D. degrees in circuits and systems from the School of Electronic Engineering, Xidian University, Xi'an, China, in 1994, 1999, and 2002, respectively.

He is currently the Dean and a Professor with the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests are artificial intelligence, neural networks, DNA computing, and big data.