TBCIM: Two-Level Blockchain-Aided Edge Resource Allocation Mechanism for Federated Learning Service Market

Lianbo Ma, *Senior Member, IEEE*, Ying Qian, Guo Yu, *Member, IEEE*, Zhetao Li, *Member, IEEE*, Liang Wang, *Member, IEEE*, Qing Li, *Senior Member, IEEE*, Xingwei Wang, and Guangjie Han, *Fellow, IEEE*

***Abstract*—With advances in the edge computing (EC) and federated learning (FL) technologies in jointcloud, the edge FL service market has emerged recently and it requires trading edge resources between model requesters and data owners to complete FL tasks, which needs to incentivize sufficient data owners to participate in model training tasks. However, the limitations of resource trading and incentive design for edge FL service market have not been well addressed. In this paper, we propose a two-level blockchain-aided resource trading mechanism for encouraging appropriate edge servers to compete for dynamic FL tasks from the market while incentivizing data owners to participate in the FL tasks. At the upper level, we apply the deep learning-based reverse auction to model the dynamics of the task server selection process, with the aim of maximizing the total social welfare of the edge FL service market, where the edge server, as a seller, considers not only the data contribution of edge devices but also the cost of using blockchain when bidding. At the lower level, the edge servers offer rewards in exchange for the data owners' participation, while the parameter aggregation is completed through the blockchain in a decentralized manner, which improves the FL's robustness. Then, we utilize the Stackelberg game to model the dynamic process that the data owners compete for the servers' revenue. We conduct extensive simulation experiments and the experimental results show that the proposed mechanism is able to get maximized social welfare and provide effective insights and strategies for the resource trading in the edge FL market to complete the federated training.**

***Index Terms*—Federated learning, two-level cooperation, incentive mechanism, jointcloud, mobile edge computing.**

Received 10 November 2023; revised 18 July 2024 and 25 March 2025; accepted 5 July 2025; approved by IEEE TRANSACTIONS ON NETWORKING Editor C. Brinton. Date of publication 29 July 2025; date of current version 18 December 2025. This work was supported by the National Key Research and Development Program of China under Grant 2022YFB4500800. *(Corresponding authors: Xingwei Wang; Liang Wang.)*

Lianbo Ma and Ying Qian are with the College of Software, Northeastern University, Shenyang 110819, China (e-mail: malb@swc.neu.edu.cn; 2110494@stu.mail.neu.edu.cn).

Guo Yu is with the Institute of Intelligent Manufacturing, Nanjing Tech University, Nanjing 211816, China (e-mail: gysearch@163.com).

Zhetao Li is with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China (e-mail: liztchina@hotmail.com).

Liang Wang is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: liangwang@nwpu.edu.cn).

Qing Li is with the Peng Cheng Laboratory, Shenzhen, Guangdong 518055, China (e-mail: liq@pcl.ac.cn).

Xingwei Wang is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: wangxw@mail.neu.edu.cn).

Guangjie Han is with the Department of Information Science and Engineering, Hohai University, Changzhou 213022, China (e-mail: hanguangjie@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TON.2025.3589017, provided by the authors.

Digital Object Identifier 10.1109/TON.2025.3589017

I. INTRODUCTION

THE past decade has witnessed an explosive growth of cutting-edge mobile and edge devices of Internet of Things (IoT) [1], which generate massive raw data at the edge network [2], [3]. Currently, with the development of jointcloud computing, the worldwide network has 7 billion IoT devices and 3 billion smartphones, and it is expected to reach 80 billion devices, which will result in 180 trillion gigabyte global data in 2025 [4]. The data from these edge devices can exhibit very valuable insights. However, processing such massive data through cloud computing would be ineffective due to the significant transmission delay and unreliability [5]. A natural solution is to deploy cloud computing services at the edge of the network where data is generated. This can be achieved by the edge computing (EC) [6], where the data at the edge network is exploited via artificial intelligence (AI) to enable various intelligent services, such as edge caching, model training, and model inference [7].

In EC, a large number of edge devices are required to process some important artificial intelligence (AI) tasks (e.g., object detection, image classification, and event prediction [8]) to enable data-driven intelligent applications, and one key to enable AI models is the ability of learning/training models using massive training data [9]. With the fast growth of data generated by various IoT devices, the learning/training of AI models becomes a dominant workload in distributed EC systems with limited resources [10], [11], [12]. Federated learning (FL) [13], [14], [15] is a promising solution to tackle the above problem, since it allows distributed users to collaboratively train a shared model while preserving all training data on their devices, which also protects data privacy [16], [17], [18].

Despite its prominent benefits, when applied in EC, FL comes with new challenges to handle [19], [20]. In the context of FL in EC (here called edge FL), the distributed participants, including edge servers (ESs) and edge devices (EDs), are profit-driven and aware of the value of their local data [21]. But the values of these local resources are not fully utilized in the edge FL system. One of the main reasons is the lack of an efficient service market mechanism for edge FL

2998-4157 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

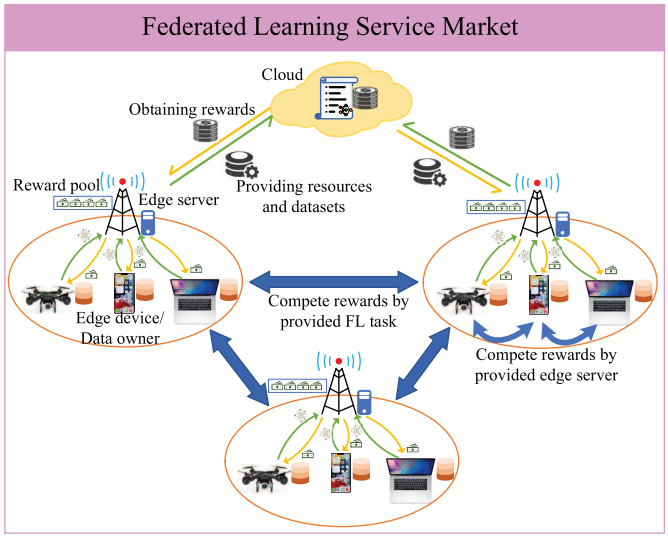


Fig. 1. Competitive relationships between different members of the edge FL service market. Edge server and its connected edge devices form an edge subsystem aiming to compete for rewards when performing the FL tasks published by the task allocation system; Different edge devices within the subsystem compete for rewards from the edge server.

applications, where users can sell their computing resources and local data for model training, and in return receive payment from buyers who require trained models [22]. In this way, users can trade their computing and data resources with rewards paid by model requester. However, the key challenge in establishing a successful FL service market lies in how to efficiently incentivize data owners to contribute sufficient training data [23], [24].

The above issue is more serious in the newest edge FL service market, as shown Fig. 1, which operates with a two-level topology, involving the cooperation at both cloud-edge level and edge-end level. Specifically, the cloud-edge cooperation refers to the resource trading between model requesters (from task allocation system) and ESs (which are connected by a set of resource/communication-constrained EDs). That is, the qualified ESs compete for those FS tasks published by model requester in the FL service market. The edge-end cooperation means that a set of EDs within the communication range submit trained gradients to their corresponding ESs for parameter aggregation. Hence, designing an incentivization model that captures this two-level cooperation in edge FL service market poses a significant challenge.

Moreover, it can be found that the above edge FL service market also faces significant challenges related to the centralized ESs that are responsible for model parameter aggregation, which can be fragile and result in loss for all associated EDs [17]. In addition, the presence of potentially malicious EDs would mislead the model training process via uploading incorrect masked gradients and unmasked shares.

To address the above issues, we in this paper introduce a two-level blockchain-aided resource trading model to facilitate the trading of edge resources for FL tasks. Different from existing studies [33], [36], [37], [38], which ignore some essential costs (e.g., storage and communication overheads) consumed by blockchain, we consider the cost of using blockchain in the blockchain-aided FL (BCFL) model, where an efficient edge FL service market is created to allow users to securely buy and sell computing resources. We then propose a deep learning-based reverse auction to select ESs and determine resource pricing, aiming to ensure maximum social welfare while meeting rationality and incentive compatibility requirements. Moreover, the Stackelberg game is utilized to model the competition behaviors among EDs for ESs revenue, where the blockchain is employed for parameter aggregation in smart contracts deployed on ESs, mitigating single-point failures and malicious participation. The effectiveness of the above mechanisms has been verified through extensive experiments.

To sum up, we highlight three key advantages compared to previous studies:

1) We propose a **T**wo-level **B**lock**C**hain-aided **I**ncentive **M**echanism termed as **TBCIM** that models the complex trading cooperation of different edge FL participants. Intuitively, this mechanism utilizes thecloud-edge-end cooperation to enable an efficient edge FL service market, where users can buy and sell computing resources in a credible way. Especially, the coordination by the ESs in proximity allows for more efficient computing resource allocation among communication-constrained EDs.
2) We design a deep learning-based reverse auction to model the cloud-edge cooperation process between model requesters and ESs. Different from conventional auctions, this deep learning-based auction can maximize total social welfare while satisfying the individual rationality and incentive compatibility constraints. Besides, a dynamic threshold scheme is devised to check whether the data quality (e.g., data distribution) of data owners can meet the requirements of consecutive training tasks.
3) We utilize the Stackelberg game to model the edge-end cooperation process that the EDs compete for the ESs' revenue, where the blockchain is introduced to implement parameter aggregation mechanism in smart contract (deployed in ESs), which avoids single-point failure and malicious participation. To ensure reasonable rewards in exchange for the data owners' participation, the gas fee of using blockchain for parameter aggregation is measured and considered as training cost in the design.

To the best of our knowledge, our proposed TBCIM is the first cloud-edge-end cooperation mechanism for incentivization of blockchain-aided FL service market. Note that, different from existing works [9], [22], [24], we deploy the blockchain in ESs rather than all the edge devices to perform parameter aggregation, and only allow ESs to participate in computationally-expensive mining competition for operating the blockchain, which thus reduces unnecessary mining overhead and recourse waste of resource-constrained EDs.

The rest of this paper is organized as follows. Section II presents the system model. Section III and Section IV elaborate the upper auction process and the lower Stackelberg process in detail, respectively. Section V gives the experiment

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

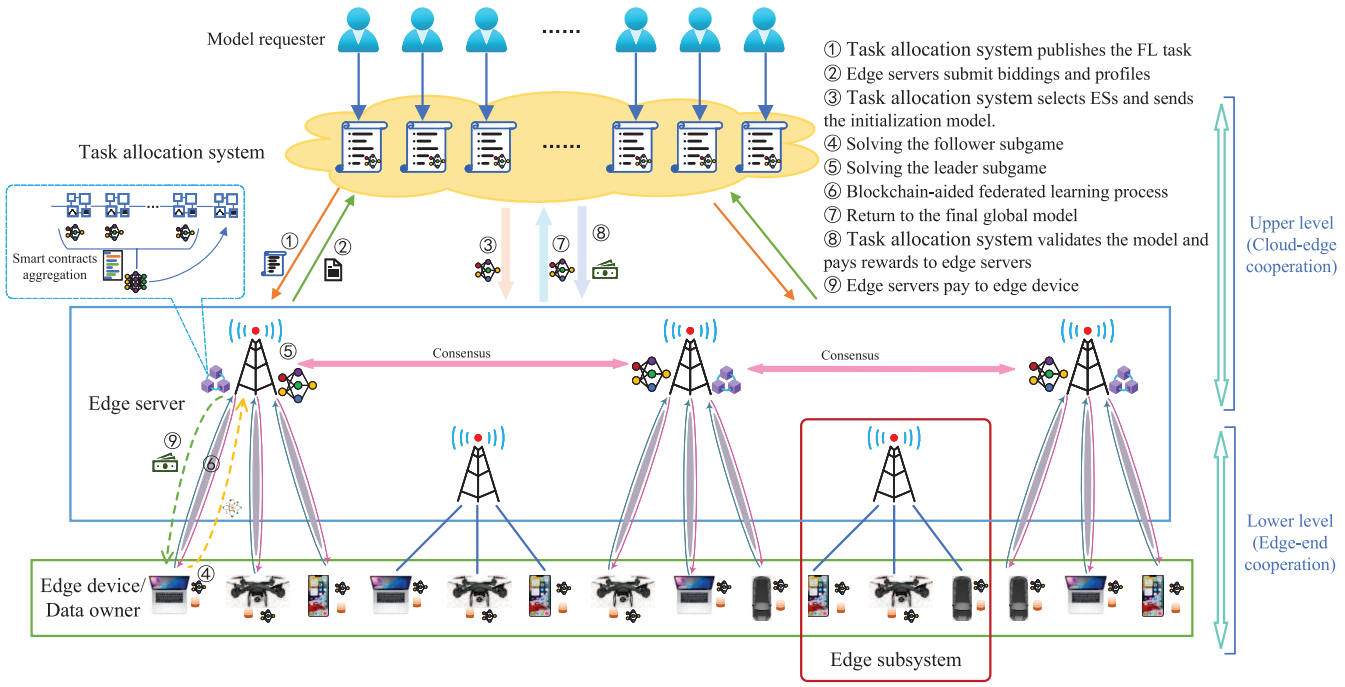


Fig. 2. An illustration of the proposed TBCIM incentive framework for edge FL service market, which is composed of upper-level reverse auction and lower-lever Starkelberg game.

results of the proposed method. Section VI reviews the related work. Section VII concludes the work.

II. System Model

The following is the detailed description of the proposed two-level blockchain-aided model.

A. System Overview

Taking Fig. 2 as an example, we can see that the proposed TBCIM is composed of a set of EDs and a set of ESs in the FL service market, e.g., $ES = \{ES_1, ES_2, \ldots, ES_S\}$. Each ES_s $(s = 1, \ldots, S)$ is connected to a set $ED_s = \{ED_{s,1}, ED_{s,2}, \ldots, ED_{s,N}\}$ with N EDs. The key components of proposed framework are shown as follows:

1) *Task Allocation System*: The system is deployed at cloud (or jointcloud), and it is responsible to perform the publication of the FL tasks to ESs while selecting data owners to act as FL workers. The main goal of this system is to achieve optimal model performance and earn the corresponding bonus from the customer. Consequently, the system from cloud (or jointcloud) delivers the gains to ESs, indirectly encouraging EDs to actively participate in the model training process.
2) *Edge Servers*: There are a set of ESs as nodes of the blockchain, in which each ESs is connected to a set of EDs (clients). We make the assumption that this connection is maintained throughout training. Each ES sells its resources to execute smart contracts and consensus (e.g., mining) of the blockchain. As an intermediary component that connects the cloud and EDs, the ESs collect the profile of each ED and ensure a balanced distribution of gains between the overall system and the connected EDs.
3) *Edge Devices/Data Owners*: Each ED as client in FL trains the model using its local dataset and subsequently generates transactions to store the parameters of the local model in the blockchain. Then, the smart contract in blockchain aggregates all local model parameters and stores the aggregated parameters in blockchain as global model parameters for next updates. EDs start next local updates by downloading blocks containing latest global parameters to update the local model.

In the edge FL system, the centralized ES, which is responsible for model parameter aggregation, is fragile due to single-point failure and may bring loss to all its corresponding EDs [27], [28]. Moreover, some EDs may be malicious and their dishonest behavior would mislead model training. If no auditability, dishonest ED may upload incorrect masked gradients and unmasked shares to central ES [29]. To tackle the above issue, we introduce blockchain into the edge FL service market that leverages smart contracts to facilitate efficient interactions in resource trading, which protects against unreliable behaviors of untrustworthy ESs and preserve privacy of ESs and EDs [30].

Blockchain-aided federated learning: Given EDs' limited resources, we deploy the blockchain on multiple edge servers to avoid single-point failures. Our architecture addresses the single-point failure limitation of traditional FL frameworks by enabling multiple ESs to cooperatively train a single FL model through decentralized orchestration. When an ES fails, only its connected EDs are affected, while others continue FL operations seamlessly via blockchain smart contracts. ED act as blockchain users, storing local updates via transactions.

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

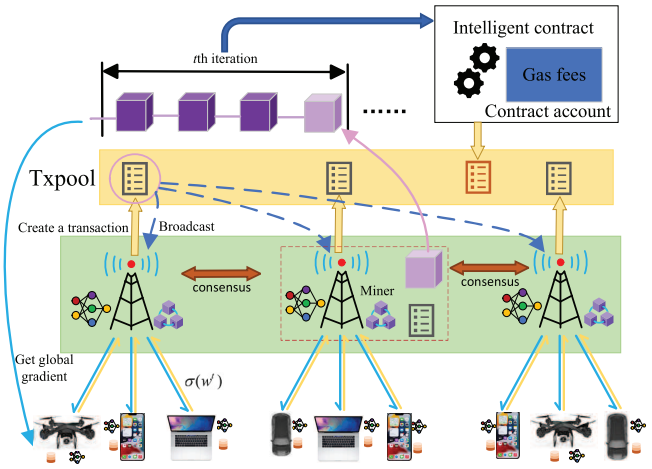


Fig. 3. The BCFL training process. Edge devices download the global model from blockchain for local training and upload the local update results via creating transactions and submitting them to nearby blockchain nodes. The blockchain verifies the current update for malicious behavior by comparing it with historical update records. Verified transactions are then broadcast to other blockchain nodes and then stored in the transaction pool. Smart contracts aggregate the current round of updates and then store the global update results on blockchain to initiate the next iteration.

As shown in Fig. 3, the blockchain verifies and broadcasts transactions from end devices, storing them in the transaction pool for recording. We use Proof of Stake (PoS) to determine the ES with bookkeeping rights. In each consensus round, a pseudorandom number within $[0, 1]$ is generated, and the probability of each ES being selected is its revenue share relative to the total. From ESs with probabilities exceeding the random number, one is stochastically selected as the block-producing node, ensuring higher-revenue ESs have proportionally greater selection chances. The blockchain identifies malicious nodes by calculating the cosine distance between an end device's submitted gradient and its historical records. Let ω_i^o denote updates on the blockchain and ω_i^t denote the local update from $ED_{s,i}$ in round t. Using the cosine distance formula, we obtain $\frac{1}{t-1}\sum_{\omega_i^j \in \omega_i^o} \frac{\omega_i^j \omega_i^t}{|\omega_i^j||\omega_i^t|} \leq \epsilon$, where ϵ is a hyperparameter.

Then, based on the framework as shown in Fig. 2, we can model the interactions between task allocation system, ESs and EDs through a two-level cooperation model, where at the upper (cloud-edge) level, a deep learning-based reverse auction is employed to model the resource trading process between cloud and ESs, aiming to maximize the total social welfare of the market, where the ESs act as sellers and the task allocation system acts as a buyer; at the lower (edge-end) level, the Stackelberg game is utilized to model such dynamic process that the EDs with available data compete for the servers' rewards, where the EDs act as followers to build a follower subgame and the ES acts as a leader to build a leader subgame.

The detailed steps are presented as following:

1) Task allocation system publishes FL task to candidate ESs at the upper level.
2) All candidate ESs submit biddings and data resource profiles to task allocation system according to the quality of their data resources, which are provided by its corresponding EDs at the lower level.
3) The task allocation system in cloud determines optimal ESs according to the reverse auction, and then sends the initialized model to the selected ESs.
4) In the edge subsystem, each ES first determines its strategy for how much resources it will provide. Then, the EDs solve the follower subgame according to the ES's strategy.
5) The solution of the follower subgame obtained in the previous step is substituted into the leader subgame to compute the reward of ES.
6) At the lower level, the BCFL training process begins after solving the Starkelberg game.
7) The BCFL training process is completed when the model accuracy meets the FL task requirement. Then the ES sends the global model to the task allocation system.
8) The task allocation system verifies the accuracy of the received model and pays rewards to ESs.
9) ESs pay the rewards to EDs.

Among the above steps, the blockchain-aided federated training process, where the aggregation of one training model is conducted in blockchain deployed on ESs, can be described as follows: The global model parameters shared by EDs in a learning round involve creating transactions and reading blocks. The global model ω is optimized via minimizing the loss $F(\omega)$ on the union of all local datasets. After every ω^t local update on each ED, each ED creates a transaction and send to the transaction pool. For each transaction, ESs only need to store it in the blockchain through mining, and then the smart contract aggregates all the local gradients. It is assumed that the smart contract automatically executes the aggregation function at a fixed time τ_t and stores the aggregated result ω_{t+1} in the blockchain for reading by EDs.

B. Upper-Level Reverse Auction Through Deep Learning

At the upper level, the task allocation system is responsible to determine optimal ESs (or edge subsystems, each of which consists of an cluster head (i.e., ES) and the corresponding members (i.e., EDs)) to complete the FL tasks from task allocation system, according to the data quality and reported value of data resource provided by ESs. In our target edge FL service market, there are multiple ESs competing for each FL task, it is needed to determine the optimal subsystem specified by the ES, which can have best data quality and obtain the maximization of its social welfare. To achieve this goal, we propose a deep learning-based reverse auction mechanism to model the dynamic process between the task allocation system and ESs, where S ESs bid for the task T in the market. Specifically, each ES can calculate the combination of EDs with the best contribution to the model and the lowest cost according to the data quantity and data quality of the ED it owns, and the system selects ESs that provide services for the FL task according to the bids of all ESs. In fact, our proposed deep learning-based reverse auction mechanism has advantages of guaranteeing bidder honesty and maximizing revenue for the seller (i.e., ES).

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

Moreover, we also devise a metric (as shown in (27)) to quantify potential contribution of each ES's datasets to the task completion, where two significant attributes of local data, i.e., the data size and data distribution, need to be considered. As shown in [31] and [32], the dataset size plays a significant role in the improvement of data quality, where more training data easily generates better learning performance. For the data distribution, the training data is typically assumed to follow independent and identically distributed (IID) pattern. However, in the edge FL scenario, the local data in each ED is with unique feature and usually exhibits non-IID mode, which severely degrades the prediction performance of the FL models [41]. Therefore, our proposed metric that integrates the above attributes can well evaluate the data quality of data owners.

C. Lower-Level Stackelberg Game

At the lower level, each ES offers rewards in exchange for the participation of EDs (i.e., the data owners), aiming to attract more Eds to participate in the edge subsystem. In each subsystem, the ES (cluster head) aims to attract more EDs to contribute their datasets, so that the subsystem can provide high-quality datasets with good data coverage for each FL task. Note that the improvement of data coverage and data quality will benefit the prediction performance of trained model, but also incurs the increase of the cost consumed by ES.

To incentivize the participation of the EDs, each ES prepares a shared reward pool for all EDs in the edge subsystem. The reward for each distributed ED is dependent upon the proportion of the ED's contribution in the subsystem, i.e., its data quality relative to that of the total data in the subsystem. Note that, a subsystem that can provide a high reward pool is more preferred to EDs. Accordingly, when more EDs participate in that subsystem, the reward pool has to be shared by a large number of EDs. We employ a Stackelberg game model to model the above dynamic process that the EDs compete for the ES' reward in the subsystem.

III. Upper-Level Reverse Auction Through Deep Learning

A. Reverse Auction Framework

In the edge FL service market, ESs try to compete for opportunities to provide various resources with the aim of getting maximum revenue for their distributed training tasks. We utilize a reverse auction to model the interactions between the task allocation system and ESs, where the task allocation system serves as both auctioneer and buyers (buying datasets and resources), and the ESs play as both bidders and sellers (selling datasets and resources).

Based on the reverse auction framework, the interactions between the task allocation system and ESs can be formulated as follows:

- The auctioneer starts the reverse auction and broadcasts the FL task requirement $\Gamma(\mathbb{P}_r, B_r)$ to all ESs, where $\mathbb{P}_r$ is the actual distribution, served as a reference distribution, and B_r is the bandwidth requirement.
- Each ES submits the type profile $\mathbf{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_S\}$. The ES's type $\mathbf{t}_s$ involves the bid b_s revealing its private service cost/valuation c_s, the size d_s and the Earth Mover's Distance (EMD) value θ_s that evaluates the distance between the data distributions of local data and task-required data.
- The auctioneer employs multi-layer neural networks to encode the reverse auction to determine whether each bid wins or not, where bidder valuations are the input and allocation/payment decisions are the output, i.e., it selects winning bids $\mathbb{W}$ as workers, computes the allocation result G and payment P, and then informs the auction outcomes (i.e., the set of winning bids and their payments) to ESs.
- The auctioneer sends the initialized model $\mathcal{M}$ and some hyper-parameters, such as learning rate, step size, etc., to the winners, i.e., the ESs that provide computing and data services for current task.
- Each winning ES performs the BCFL training process according to the corresponding FL training task using datasets from EDs.
- The auctioneer pays the payment p_s to each winning ES.

B. Reverse Auction Formulation as Learning Problem

The ES can estimate the cost to provide the bidding price in the auction according to the amount of data owned by the connected EDs and the gas fees of the blockchain. Formally, each ES has a bidding space denoted as V_s, and the bidding space for all ESs is given by $V = \prod_{s=1}^{S} V_s$. A reverse auction (g, p) can be denoted as a pair of winner determination rules $g_s : V \to 2^S$ and payment rules $p_s : V \to \mathbb{R}_{\geq 0}$ (which can be randomized). Given bids $b = (b_1, \ldots, b_S) \in V$, the reverse auction computes a winner determination $g(b) \in 2^S$, and payment $p(b) \in \mathbb{R}^s_{\geq 0}$.

The task allocation system possesses information about the distributions $\mathcal{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_n)$ but lacks knowledge of the bidders' actual costs, denoted as c. Bidders submit their valuations, which may not always be truthful. A reverse auction is employed to determine the task allocation to the bidders and to payments for them. Each ES gives a bid b_s according to the cost. We ignore the cost of running the task allocation system. Then the social welfare is

$$U(b) = U_d(g(b)) - \sum p_s(b_s), \tag{1}$$

where $g(b)$ represents the winner determination results based on the bid b, and $U_d(g(b)) = q(D, \theta)$ represents the data utility proxy for the total utility derived from the winner determination results $g(b)$, where the total utility is determined by the workers' EMD values $\theta = \sum_{i \in \mathbb{W}} \theta_i / |\mathbb{W}|$ and data size $D = \sum_{i \in \mathbb{W}} D_i$ as detailed in Section IV-B. Let b_{-s} denote the valuation profile $b = (b_1, \ldots, b_S)$ without element b_s, and $V_{-s} = \prod_{j \neq s} V_j$ represents the possible bid profiles of all ESs other than ES_s.

Then, we formulate the optimization of the reverse auction design as a machine learning problem, where the conventional loss function, aiming to estimate error against a target label, is replaced with the negated social welfare on bidding derived from $\mathcal{F}$. Given a parametric class of reverse auctions, i.e., $(g^w, p^w) \in \mathcal{M}$, where w represents the parameters in $\mathbb{R}^d$ (for $d \in \mathbb{N}$), and a sample of ES valuation profiles,

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

i.e., $L = \left\{b^{(1)}, \ldots, b^{(L)}\right\}$, which are independently derived from $\mathcal{F}$. Our goal is to achieve an optimal reverse auction from all auctions in $\mathcal{M}$ satisfying incentive compatibility by minimizing the negated social welfare $-U(b)$.

To guarantee the incentive-compatibility, we introduce several constraints in the machine learning problem for the reverse auction mechanism. Specifically, for each ES, the *ex post regret* is used to estimate the extent to which a reverse auction deviates incentive compatibility. By fixing the bids of others, the ex-post regret of an ES is defined as the maximum potential utility improvement brought by its strategically deceptive bid from the set of all possible dishonest of bids. In this work, we focus on *expected ex-post regret* of an ES, formulated as

$$rgt_s(w) = \mathbf{E}\left[\max_{b'_s \in V_s} U^w\left(b'_s, b_{-s}\right) - U^w\left(b_s, b_{-s}\right)\right], \quad (2)$$

where for given the model parameters w, the expectation is calculated with respect to $b \sim \mathcal{F}$ and $U^w(b) = U_d(g^w(b)) - \sum p_s^w(b_s)$. It is assumed that $\mathcal{F}$ has full support on the bid profile space V, with recognizing that the regret is non-negative. The reverse auction ensures incentive compatibility if and only if $rgt_s(w) = 0, \forall s \in ES$.

Considering the above, we reconstruct the objective function of the deep neural network model to minimize the expected loss:

$$\min_{w \in \mathbb{R}^d} \mathbf{E}_{b \sim \mathcal{F}}\left[\sum p_s(b_s) - U_d(g(b))\right]$$
$$\text{s.t.} \quad rgt_s(w) = 0, \quad \forall s \in ES. \quad (3)$$

The loss function objective is to minimize the expected negative social welfare provided that the *expected ex post regret* of each ES is zero.

$$\widehat{rgt}_s(w) = \frac{1}{\mathcal{L}} \sum_{\ell=1}^{\mathcal{L}} \max_{b'_s \in V_i} U^w\left(b'_s, b_{-s}^{(\ell)}\right) - U^w\left(b^{(\ell)}\right). \quad (4)$$

and aim to minimize the empirical loss, with the empirical regret (equal to zero for all ESs) constraints:

$$\min_{w \in \mathbb{R}^d} \frac{1}{\mathcal{L}} \sum_{\ell=1}^{\mathcal{L}} \left[\sum_{s=1}^{S} p_s(b_s) - U_d(g(b))\right]$$
$$\text{s.t.} \ \widehat{rgt}_s(w) = 0, \quad \forall s \in ES. \quad (5)$$

C. Neural Network Architecture and Training

We then give the neural network architecture, termed as RANet, to model reverse auctions. The architecture consists of two logically distinct components, i.e., the winner determination and payment networks.

The architecture comprises two key components: a randomized allocation network denoted as $g^w : \mathbb{R}^s \to [0,1]^s$ and a payment network represented as $p^w : \mathbb{R}^s \to \mathbb{R}> 0^s$. Both neural networks are structured as feed-forward, fully-connected models incorporating tanh activation functions. The input layer of these neural networks consists of bids, denoted as b_s, signifying the cost associated with c_s.

The allocation neural network gives a result of allocation probabilities $z_1 = g_1(b), \ldots, z_s = g_s(b)$ for each ES. These results of allocation probabilities are calculated using a softmax activation function. On the other hand, the payment neural network generates payment values for each ES, representing the expected payment that the ES should receive for a given bid profile.

Algorithm 1 RANet Training

Input: Batches $\mathcal{L}_1, \ldots, \mathcal{L}_T$ of size B
  Initialize: $w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^n$
  for $t = 0$ to T **do**
    Receive batch $\mathcal{L}_t = \left\{b^{(1)}, \ldots, b^{(B)}\right\}$
    Initialize misreports $b_s'^{(\ell)} \in V_s, \forall \ell \in [B], i \in N$
    for $r = 0$ to R **do**

$$\forall \ell \in [B], i \in N:$$
$$b_s'^{(\ell)} \leftarrow b_s'^{(\ell)} + \gamma \nabla_{b'_s} U^w\left(b'_s, b_{-s}^{(\ell)}\right)$$

    end for
    Compute regret gradient: $\forall \ell \in [B], i \in N$:

$$g_{\ell,s}^t = \nabla_w \left[U^w\left(b'_s, b_{-s}^{(\ell)}\right) - U^w\left(b^{(\ell)}\right)\right]\Big|_{w=w^t}$$

    Compute Lagrangian gradient using (7) and update w^t: $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{C}_{\rho_t}(w^t, \lambda^t)$
    if t is a multiple of Q **then**
      $\lambda_s^{t+1} \leftarrow \lambda_s^t + \rho_t \widehat{rgt}_i\left(w^{t+1}\right), \quad \forall s \in ED_s$
    else
      $\lambda^{t+1} \leftarrow \lambda^t$
    end if
  end for

In order to guarantee that the bids of the EDs in the designed reverse auction mechanism are truthful, i.e., the ED's cost should not be less than its expected revenue for the allocation. After the computation of the sigmoid layer, each network will provide each ED with a price score $\tilde{p}_s \in [0,1]$ regarding the price paid by the ES to the ED. It then outputs a payment $p_s = \tilde{p}_s z_s b_s$, where the vectors of z_s are obtained from the allocation neural network. An architectural overview is depicted in Fig. 4, demonstrating how the social welfare and regret are calculated based on the parameters of the winner allocation network and payment network.

We utilize the Lagrangian augmentation algorithm to solve the learning problem with constraints in (5) in terms of parameters w of neural network. We give the Lagrangian function and add a quadratic penalty term to ensure the incentive compatibility of the reverse auction:

$$\mathcal{C}_\rho(w;\lambda) = \frac{1}{\mathcal{L}} \sum_{\ell=1}^{\mathcal{L}} \left[\sum_{s=1}^{S} p_s(b_s) - U_d(g(b))\right] + \sum_{s \in ES} \lambda_s \widehat{rgt}_s(w) + \frac{\rho}{2}\left(\sum_{s \in ES} \widehat{rgt}_s(w)\right)^2, \quad (6)$$

where $\lambda_s \in \mathbb{R}^n$ is a vector of Lagrange multipliers, and $\rho > 0$ is the coefficient adjusting the quadratic penalty term. The model parameters and Lagrange multipliers are updated as follows: (a) $w^{\text{new}} \in \text{argmin}_w \mathcal{C}_\rho\left(w^{\text{old}}; \lambda^{\text{old}}\right)$ and (b) $\lambda_s^{\text{new}} = \lambda_s^{\text{old}} + \rho \widehat{rgt}_s\left(w^{\text{new}}\right), \forall s \in ES$.

We employ Algorithm 1 to solve the constrained training problem. Our approach involves dividing the training

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

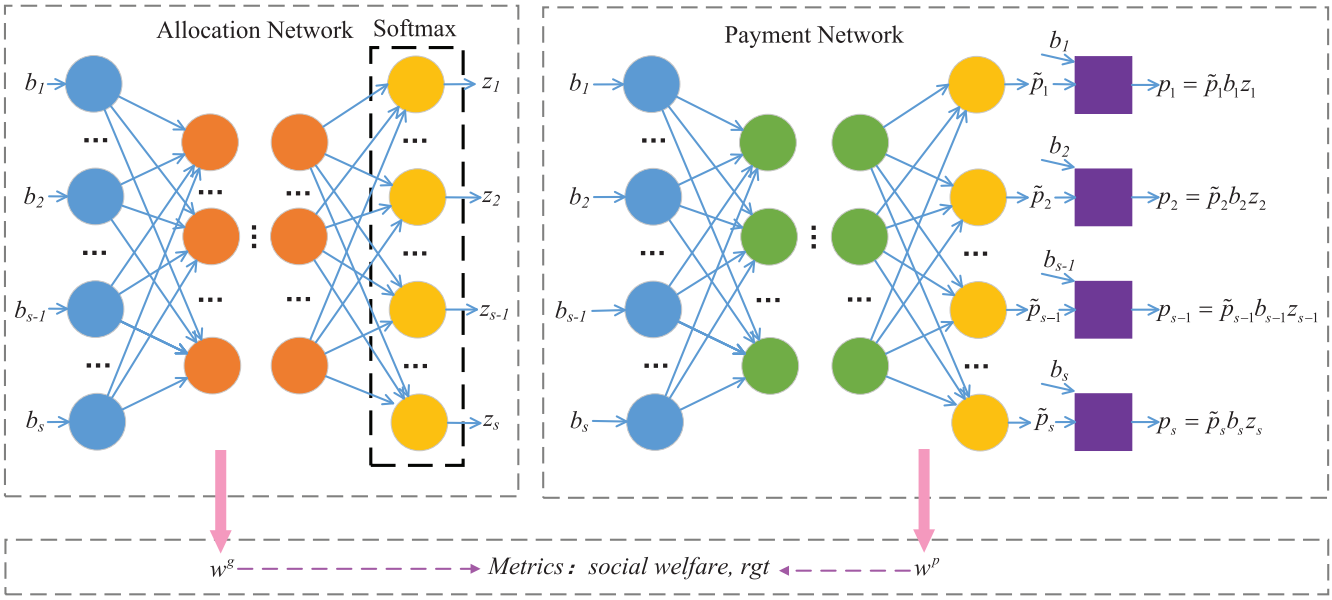


Fig. 4. The edge server allocation network and payment network. On the left is the allocation network, where the input is the bids from s edge servers $(b_1, b_2, \ldots, b_s)$ and the output is the winning probabilities of the edge servers $(z_1, z_2, \ldots, z_s)$. The network weights are denoted as w^g. On the right is the payment network, where the input is the bids from s edge servers $(b_1, b_2, \ldots, b_s)$ and the output is the payment ratios for each edge server $(p_1, p_2, \ldots, p_s)$. The payment ratio is multiplied by the allocation probability and the bid gives the final payment price. The network weights are denoted as w^p. By calculating social welfare and the rgt value from the payment and allocation results, the overall loss function is obtained, which is then used to update the weights of both networks.

sample $\mathcal{L}$ into mini-batches of size B. We iterate over the training samples for multiple passes, ensuring data shuffling after each pass. At iteration t, we denote the batch as $\mathcal{L}_t = \{b^{(1)}, \ldots, b^{(B)}\}$. The update w^{new} for model parameters involves unconstrained optimization of $\mathcal{C}\rho$ with respect to w, which is tackled by a gradient-based optimizer. We use $\widetilde{rgt}_s(w)$ to represent the empirical regret in (4) calculated on batch $\mathcal{S}_t$. The gradient $\nabla_w \mathcal{C}_\rho(w; \lambda^t)$ for fixed λ^t is expressed as follows:

$$\nabla_w \mathcal{C}_\rho(w; \lambda^t) = -\frac{1}{B}\sum_{\ell=1}^{B} \nabla \left[\sum_{s=1}^{S} p_s(b_s) - U_d(g(b))\right] + \sum_{s\in ES}\sum_{\ell=1}^{B} \lambda_s^t g_{\ell,s} + \rho \sum_{s\in ES}\sum_{\ell=1}^{B} \widetilde{rgt}_s(w) g_{\ell,s}, \tag{7}$$

where

$$g_{\ell,s} = \nabla_w \left[\max_{b'_s \in V_s} U^w\left(b'_s, b_{-s}^{(\ell)}\right) - U^w\left(b^{(\ell)}\right)\right]. \tag{8}$$

The computation of $\widetilde{rgt}_s$ and $g_{\ell,s}$ involves the presence of dishonest bids from the ESs. To generate these dishonest bid reports, we employ a gradient-based iterator, facilitating gradient propagation by evaluating the utility difference resulting from dishonest bids. In principle, we search for the dishonest bid that maximizes the utility difference among the ESs. To calculate the optimal deceptive bids, we conduct Q gradient updates: $b_s'^{(\ell)} = b_s'^{(\ell)} + \gamma \nabla_{b'_s} U^w\left(b_s^{(\ell)}, b_{-s}^{(\ell)}\right)$, with $\gamma > 0$.

D. Theoretical Analysis

Typically, the optimization problem of (5) in the reverse auction model is nonconvex, and then the solver is not guaranteed to achieve a globally optimal solution. In fact, the proposed deep learning-based method exhibits an excellent problem-solving performance, as shown in the following experiments, since the reverse auction is learned well, with the aim of closely matching the optimal reverse auction structure in settings where this is known, and thus it incurs very low regret.

In fact, the deep learning-based approach aims to constrain the difference between empirical and expected regrets in terms of the quantity of sampled valuations, which can be extended to the revenue scenario. Our constraints are used for the reverse auction coming from a finite capacity class, and indicate that solving (5) by using substantial sample results can obtain a reverse auction that can achieve near-optimal expected revenue with minimal expected regret.

Then, we can theoretically validate the target reverse auctions according to a set of general allocation rules, including: the allocation rule $g: V \to [0,1]^{2^M}$ transforms the valuation profiles into a vector of allocation probabilities for ES, where $g(b) \in [0,1]$ denotes the probability that the ES wins the auction, and the payment function $p: V \to R^n$ transforms the valuation profiles into a payment for each ES $p_s(b) \in \mathbb{R}$. For simplicity, the superscripts "w" can be omitted. As before, $\mathcal{M}$ denotes a class of auctions (g, p), and it is assumed that the allocation and payment rules in $\mathcal{M}$ are continuous, while the set of valuation profiles V is a compact set.

The inner product of any two vectors, denoted as a and b is represented as $\langle a, b\rangle = \sum_{i=1}^{d} a_i b_i$. Therefore, the L_1-norm of matrix A is defined as $\|A\|_1 = \max_{1\le j\le \ell} \sum_{i=1}^{k} A_{ij}$.

Let $\mathcal{U}$ be the class of social welfare functions for ES defined on auctions in $\mathcal{M}$, i.e.,

$$\mathcal{U} = \Big\{U: V \times V \to \mathbb{R} \mid U(g(b)) - \sum p_s(b_s) \quad (g, p) \in \mathcal{M}\Big\} \tag{9}$$

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

We treat b_s as a vector of length 2^M, and use the inner product to represent the social welfare function $U(b_s) = \langle b_s, g_s(b)\rangle - p_s(b)$.

Consider $rgt \circ \mathcal{U}$ as the set encompassing all regret functions for ES. Then it can be defined with respect to social welfare functions within $\mathcal{U}$:

$$\begin{aligned} rgt \circ \mathcal{U} = \{f_s : V \to \mathbb{R} \mid f_s(b) \\ = \max_{b'_s} U(b'_s, b_{-s}) - U(b_s, b_{-s})\}. \end{aligned} \tag{10}$$

The $\ell_{\infty,1}$ distance between two social welfare functions, denoted as U and U', is formally defined as $\max_{b,b'_s} \sum_i |U(b'_s, b_{-s}) - U(b_s, b_{-s})|$. For this distance metric, $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$ represents the minimum number of balls, each having a radius of ϵ, required to encompass the set $\mathcal{U}$. Let $\mathcal{N}_\infty(\mathcal{U}, \epsilon)$ represent the minimum number of balls, each with a radius of ϵ, needed to cover the set $\mathcal{U}$ under this distance metric.

Furthermore, we represent the class of allocation functions as $\mathcal{G}$ for each ES, $\mathcal{G}_s = \{g_s : V \to 2^M \mid g \in \mathcal{G}\}$. Likewise, we represent the class of payment functions using $\mathcal{P}$ and $\mathcal{P}_s = \{p_s : V \to \mathbb{R} \mid p \in \mathcal{P}\}$. We represent the covering number of $\mathcal{P}$ as $\mathcal{N}_\infty(\mathcal{P}, \epsilon)$ the context of the $\ell_{\infty,1}$ distance metric. Additionally, we utilize $\mathcal{N}_\infty(\mathcal{P}_s, \epsilon)$ to denote the covering number for $\mathcal{P}_s$ in the context of the ℓ_∞ distance metric.

Consider a class of functions denoted as $\mathcal{F}$, where each function $f : Z \to [-c, c]$. Given a sample denoted as $\mathcal{S} = \{z_1, \ldots, z_L\}$, comprising a set of data points from Z, the empirical Rademacher complexity of $\mathcal{F}$ is defined as:

$$\hat{\mathcal{R}}_L(\mathcal{F}) := \frac{1}{L}\mathbf{E}_\xi \left[\sup_{f\in\mathcal{F}} \sum_{z_i \in S} \xi_s f(z_s)\right], \tag{11}$$

where ξ belongs to the set $\{-1, 1\}^L$, and each σ_i is independently sampled from a uniform distribution over $\{-1, 1\}$.

Lemma 1: Suppose $L = \{z_1, \ldots, z_\mathcal{L}\}$ is an independent and identically distributed sample drawn from some distribution D and covering Z. Then for all $f \in \mathcal{F}$, the probability of drawing $\mathcal{S}$ from D is at least $1 - \delta$,

$$\mathbf{E}_{z\in D}[f(z)] \le \frac{1}{\mathcal{L}} \sum_{i=1}^{L} f(z_i) + 2\hat{\mathcal{R}}_\mathcal{L}(\mathcal{F}) + 4c\sqrt{\frac{2\log(4/\delta)}{\mathcal{L}}}. \tag{12}$$

We evaluate the performance of the reverse auction class by employing a concept often used in the ranking literature known as covering numbers. Specifically, we introduce the $\ell_{\infty,1}$ distance between reverse auctions (g, p) and (g', p') belonging to the set $\mathcal{M}$. This distance is defined as:

$$\max_{b\in V} \sum_s |g_s(b) - g'_s(b)| + \sum_s |p_s(b) - p'_s(b)|. \tag{13}$$

For any given $\epsilon > 0$, we denote $\mathcal{N}_\infty(\mathcal{M}, \epsilon)$ as the minimum number of balls, each having a radius of ϵ, that are required to cover the entirety of $\mathcal{M}$ when measured using the $\ell_{\infty,1}$ distance.

The proof entails directly applying Lemma 1 to the set of revenue functions established on $\mathcal{M}$, denoted as:

$$\begin{aligned} rev \circ \mathcal{M} = \Bigg\{ f : V \to \mathbb{R} \mid f(v) = \sum_{i=1}^{n} p_i(v), \\ \text{for some } (g, p) \in \mathcal{M}\}. \end{aligned} \tag{14}$$

Theorem 1 Assume the bid b_s is less than or equal to 1 for each bidder ES_s. Define $\mathcal{M}$ as a reverse auction class that satisfies individual rationality. The value range of δ is $(0, 1)$. For any (g^w, p^w) belonging to $\mathcal{M}$, the probability of sampling $\mathcal{L}$ samples in $\mathcal{F}$ is at least $1 - \delta$, then the bound of revenue is:

$$\begin{aligned} &\mathbf{E}_{b\sim\mathcal{F}}\left[\sum p_s(b_s) - U_d(g(b))\right] \\ &\le \frac{1}{\mathcal{L}} \sum_{\ell=1}^{\mathcal{L}} \left[\sum_{s=1}^{S} p_s(b_s) - U_d(g(b))\right] + 2s\Delta_\mathcal{L} + Cs\sqrt{\frac{\log(1/\delta)}{\mathcal{L}}}, \end{aligned} \tag{15}$$

and the regret of revenue is:

$$\frac{1}{S}\sum_{s=1}^{S} rgt_s(w) \le \frac{1}{S}\sum_{s=1}^{S} \widehat{rgt}_s(w) + 2\Delta_\mathcal{L} + C'\sqrt{\frac{\log(1/\delta)}{\mathcal{L}}}, \tag{16}$$

where $\Delta_\mathcal{L} = \inf_{\epsilon>0}\left\{\frac{\epsilon}{S} + 2\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{M},\epsilon/2))}{\mathcal{L}}}\right\}$, C, and C' are distribution-independent constants.

Subsequently, it limits the Rademacher complexity component within this class by establishing a connection with the covering number associated with the payment class $\mathcal{P}$, and this limitation is influenced by the covering number related to the reverse auction class of $\mathcal{M}$.

Given our assumption that auctions within $\mathcal{M}$ adhere to individual rationality and that valuation functions are confined to the interval $[0, 1]$, it follows that, for any v, $p_i(v) \le 1$. According to the definition of the covering number $\mathcal{N}_\infty(\mathcal{P}, \epsilon)$ for the payment class, we can establish that, for any $p \in \mathcal{P}$, there exists an $f_p \in \hat{\mathcal{P}}$ with a cardinality $|\hat{\mathcal{P}}| \le \mathcal{N}_\infty(\mathcal{P}, \epsilon)$. This guarantees that $\max_v \sum_i |p_i(v) - f_{p_i}(v)| \le \epsilon$. Initially, we focus on bounding the Rademacher complexity for a given $\epsilon \in (0, 1)$.

$$\begin{aligned} &\hat{\mathcal{R}}_\mathcal{L}(rev \circ \mathcal{M}) \\ &= \frac{1}{\mathcal{L}}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{\mathcal{L}} \sigma_\ell \cdot \sum_s p_s\left(v^{(\ell)}\right)\right] \\ &= \frac{1}{\mathcal{L}}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{\mathcal{L}} \sigma_\ell \cdot \sum_s f_{p_s}\left(v^{(\ell)}\right)\right] \\ &\quad + \frac{1}{\mathcal{L}}\mathbf{E}_\sigma\left[\sup_p \sum_{\ell=1}^{\mathcal{L}} \sigma_\ell \cdot \sum_s p_i\left(v^{(\ell)}\right) - f_{p_s}\left(v^{(\ell)}\right)\right] \\ &\le \frac{1}{\mathcal{L}}\mathbf{E}_\sigma\left[\sup_{\hat{p}\in\hat{\mathcal{P}}} \sum_{\ell=1}^{\mathcal{L}} \sigma_\ell \cdot \sum_s \hat{p}_i\left(v^{(\ell)}\right)\right] + \frac{1}{\mathcal{L}}\mathbf{E}_\sigma\|\sigma\|_1\epsilon \\ &\le 2\ n\sqrt{\frac{2\log(\mathcal{N}_\infty(\mathcal{P}, \epsilon))}{\mathcal{L}}} + \epsilon. \end{aligned} \tag{17}$$

The final inequality holds because

$$\sqrt{\sum_\ell \left(\sum_i \hat{p}_i\left(v^\ell\right)\right)^2} \le \sqrt{\sum_\ell \left(\sum_i p_i\left(v^\ell\right) + n\epsilon\right)^2} \le 2\ n\sqrt{\mathcal{L}}. \tag{18}$$

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

Next, we prove that $\mathcal{N}_\infty(\mathcal{P},\epsilon) \le \mathcal{N}_\infty(\mathcal{M},\epsilon)$ holds true. For any $(g,p) \in \mathcal{M}$, consider $(\hat{g},\hat{p})$ such that, for all v,

$$\sum_s |g_s(b) - \hat{g}_s(b)| + \sum_s |p_s(b) - \hat{p}_s(b)| \le \epsilon. \tag{19}$$

Therefore, for any $p \in \mathcal{P}$, for all $b, \sum_s |p_s(b) - \hat{p}_s(b)| \le \epsilon$, which implies $\mathcal{N}_\infty(\mathcal{P},\epsilon) \le \mathcal{N}_\infty(\mathcal{M},\epsilon)$. Applying Lemma 1 and $\sum_s p_s(b) \le n$ for any b, with probability of at least $1-\delta$,

$$\begin{aligned}
&\mathbf{E}_{b\sim\mathcal{F}}\left[\sum p_s(b_s) - U_d(g(b))\right] \\
&\le \frac{1}{\mathcal{L}}\sum_{\ell=1}^{\mathcal{L}}\left[\sum_{s=1}^{S} p_s(b_s) - U_d(g(b))\right] \\
&\quad + 2\cdot\inf_{\epsilon>0}\left\{\epsilon + 2n\sqrt{\frac{2\log\left(\mathcal{N}_\infty(\mathcal{M},\epsilon)\right)}{\mathcal{L}}}\right\} + Cn\sqrt{\frac{\log(1/\delta)}{\mathcal{L}}}.
\end{aligned} \tag{20}$$

Therefore, the revenue bound is proven.

Proof of regret bound: Let's define the class of summation regret functions:

$$\begin{aligned}
\overline{rgt}\circ\mathcal{U} = \Bigg\{ & f : V \to \mathbb{R} \mid f(v) = \sum_{i=1}^{n} r_i(v) \\
& \text{for some } (r_1,\ldots,r_n) \in rgt\circ\mathcal{U}\Bigg\}.
\end{aligned} \tag{21}$$

As defined by the covering number $\mathcal{N}_\infty(\mathcal{U}_s,\epsilon)$, there exists a set $\hat{\mathcal{U}}_s$ containing no more than $\mathcal{N}_\infty(\mathcal{U}_s,\epsilon/2)$ elements. This set $\hat{\mathcal{U}}_s$ is constructed to ensure that for any $u_s \in \mathcal{U}_s$, there exists a corresponding $\hat{u}_s \in \hat{\mathcal{U}}_s$ satisfying

$$\sup_{b,b'_i} |U(b'_s, b_{-s}) - U(b_s, b_{-s})| \le \epsilon/2. \tag{22}$$

For any $u_s \in \mathcal{U}_s$, taking $\hat{u}_s \in \hat{\mathcal{U}}_s$ satisfying the above condition, then for any b,

$$\begin{aligned}
&\left|\max_{b'_s\in V}\left(u_s\left(b'_{-s}\right) - u_s\left(b_{-s}\right)\right) - \max_{\bar{b}_s\in V}\left(\hat{u}_s\left(\bar{b}_{-s}\right) - \hat{u}_s\left(b_{-s}\right)\right)\right| \\
&\le \left|\max_{b'_s} u_s\left(b'_{-s}\right) - \max_{\bar{b}_i}\hat{u}_s\left(\bar{b}_{-s}\right) + \hat{u}_s\left(b_{-s}\right) - u_s\left(b_{-s}\right)\right| \\
&\le \left|\max_{b'_s} u_s\left(b'_{-s}\right) - \max_{\bar{b}_i}\hat{u}_s\left(\bar{b}_{-s}\right)\right| + \left|\hat{u}_s\left(b_{-s}\right) - u_s\left(b_{-s}\right)\right| \\
&\le \left|\max_{b'_s} u_s\left(b'_{-s}\right) - \max_{\bar{b}_i}\hat{u}_s\left(\bar{b}_{-s}\right)\right| + \epsilon/2.
\end{aligned} \tag{23}$$

Let $b^*_s \in \arg\max_{b'_s} u_i\left(b'_{-s}\right)$ and $\hat{b}^*_s \in \arg\max_{\bar{b}_s}\hat{u}_s\left(\bar{b}_{-s}\right)$, then

$$\begin{aligned}
&\max_{b'_s} u_s\left(v'_{-i}\right) = u_i\left(v^*_i, v_{-i}\right) \le \hat{u}_s\left(b^*_s\right) + \frac{\epsilon}{2} \le \hat{u}_s\left(\hat{v}^*_{-i}\right) + \frac{\epsilon}{2} \\
&= \max_{\bar{b}_s}\hat{u}_s\left(\bar{b}_{-s}\right) + \frac{\epsilon}{2}, \text{ and} \\
&\max_{\bar{b}_s}\hat{u}_s\left(\bar{b}_{-s}\right) = \hat{u}_s\left(\hat{b}^*_{-s}\right) \le u_s\left(\hat{b}^*_{-s}\right) + \frac{\epsilon}{2} \le u_s\left(b^*_{-s}\right) + \frac{\epsilon}{2} \\
&= \max_{b'_s} u_s\left(b'_{-s}\right) + \frac{\epsilon}{2}.
\end{aligned} \tag{24}$$

Thus, for all $u_s \in \mathcal{U}_s$, there exists $\hat{u}_s \in \hat{\mathcal{U}}_s$ such that for any bid profile b,

$$\begin{aligned}
&\left|\max_{b'_s}\left(u_s\left(b'_{-i}\right) - u_s\left(b_{-s}\right)\right) - \max_{\bar{b}_s}\left(\hat{u}_s\left(\bar{b}_{-s}\right) - \hat{u}_s\left(b_{-s}\right)\right)\right| \\
&\le \epsilon,
\end{aligned} \tag{25}$$

which implies $\mathcal{N}_\infty(rgt\circ\mathcal{U}_i,\epsilon) \le \mathcal{N}_\infty(\mathcal{U}_i,\epsilon/2)$. Therefore, the regret bound is proven.

Algorithm 2 Client Selection

for each ED set, $s = 1$ to S **do**
  The server sends $\mathbb{P}_a$ to each $ED_{s,n}$.
  for each ED in ED_s, $n = 1$ to N **do**
    $ED_{s,n}$ calculates the EMD value based on Eq.(25) and sends it to the ES_s.
  end for
  ES_s calculates the average EMD value and selects the workers.
  if $\theta_{s,n} \ge \theta_s$ **then**
    Add $ED_{s,n}$ to $\mathbb{W}_s$.
  end if
end for

IV. Lower-Level Stackelberg Game

The selection of EDs that are more relevant to the data distribution required for the target task can lead to higher model performance. This section outlines the process of selecting such relevant clients, i.e., first selecting clients with a certain degree of correlation and then determining the data size sold by each client through a game, followed by allocating the resulting profits to each client accordingly.

A. Relevant Clients Selection

In the process of selecting an ED (client), each ES takes into consideration the data quality provided by the ED. Specifically, it assesses whether the distribution of local datasets available to the ED aligns with the distribution required for the given task. Notably, the reduction in accuracy primarily results from divergence in weights, a metric quantifiable by the EMD. A larger EMD value indicates a greater divergence in terms of weights, which in turn makes a detrimental impact on the overall quality of the global model. Consequently, we employ a dynamic client selection approach in each round. This approach can ensure the provisioning of high-quality data, at the same time bolster the training efficiency with cost saving. The process of client selection is shown in Algorithm 2.

We are addressing an L class classification problem defined within the confines of a compact space $\mathbb{X}$ and a label space $\mathbb{Y}$. The data samples from the ED, denoted as $D_{s,n} = \{\mathbb{X}_{s,n}, \mathbb{Y}_{s,n}\}$, are distributed across $\mathbb{X}\times\mathbb{Y}$ according to the distribution $\mathbb{P}_{s,n}$. Let $\theta_{s,n}$ represent the EMD for $D_{s,n}$. To be precise, given the actual distribution $\mathbb{P}_a$ for the entire population, the EMD is computed as follows:

$$\theta_{s,n} = \sum_{j\in\mathbb{Y}} \left\|\mathbb{P}_{s,n}(y=j) - \mathbb{P}_a(y=j)\right\|, \tag{26}$$

where $\mathbb{P}_a$ represents the distribution of FL task demands. It is acquired through historical data from the task allocation system.

Let $\boldsymbol{\theta_s} = \{\theta_{s,1},\ldots,\theta_{s,n}\}$ denote the set of EMD values of all ESs. Based on the data size and the EMD value of the workers set (EDs) $\mathbb{W}_s$, the quality of the model can be formulated as

$$\begin{aligned}
q(\mathbb{W}_s) &= q(D(\mathbb{W}_s), \overline{\theta}(\mathbb{W}_s)) \\
&= \alpha(\overline{\theta}) - a_1 \mathrm{e}^{-(a_3 D(\mathbb{W}_s))^{\alpha(\overline{\theta})}},
\end{aligned} \tag{27}$$

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

where $D(\cdot)$ is the function of getting the data size (i.e., $D(\mathbb{W}_s) = \sum_{i\in\mathbb{W}_s} d_i$, and d_i is the data size of each worker), $\overline{\theta}(\cdot)$ is the function of obtaining average EMD (i.e., $\overline{\theta}_s(\mathbb{W}_s) = \frac{\sum_{i\in\mathbb{W}_s}\theta_i}{|\mathbb{W}_s|}$ and $\overline{\theta}(\emptyset) = 0$), $\alpha(\overline{\theta}) = \exp\left(-\left(\frac{\overline{\theta}}{a_4}+a_3\right)^2\right) < 1$, and $a_1,\ldots,a_4$ are positive curve fitting parameters. The first term $\alpha(\overline{\theta})$ shows that the increasing average EMD leads to the degradation of model performance. The exponential component, $-a_1 e^{-(a_2 D(\mathbb{W}_s))^{\alpha(\overline{\theta})}}$, accounts for diminishing returns as the overall data size expands. Consequently, we can express the data utility, denoted as U_d, as a linear function of $q(\cdot)$:

$$\begin{aligned} U_d(\mathbb{W}_s) &= U_d(D(\mathbb{W}_s), \overline{\theta}(\mathbb{W}_s)) \\ &= a_5 q(\mathbb{W}_s) \\ &= a_5\left(\alpha(\overline{\theta}) - a_1 e^{-(a_2 D(\mathbb{W}_s))^{\alpha(\overline{\theta})}}\right), \end{aligned} \tag{28}$$

where a_5 is the profit per unit performance.

B. Profit Distribution Based on Stackelberg Game

With the expansion of data volume, EDs will inevitably suffer from the increase of data and computing related expenses. Therefore, each $ED_{s,n}$ is associated with a unit cost of local data denoted as $\gamma_{s,n}$, and the cost of local data, $c^d_{s,n}$, can be expressed as

$$c^d_{s,n} = d_{s,n}\gamma_{s,n}. \tag{29}$$

In addition, the ED has to evaluate the computation and communication expenses to estimate its service cost in the role of a worker. Based on experimental findings concerning the energy consumption during FL training [24], the computation cost, denoted as $c^c_{s,n}$, for the data owner $ED_{s,n}$ can be defined as a linear function of the data size $d_{s,n}$, as expressed below:

$$c^{\mathrm{c}}_{s,n} = d_{s,n}\delta_1\delta_{\mathrm{g}} M\beta_{s,n}C^2_{s,n}, \tag{30}$$

where $\beta_{s,n}$ represents the unit computational cost of the data owner $ED_{s,n}$. Given that the structures of the global model and the local model remain identical when applying FedAvg, we utilize M to denote the model size and $C_{s,n}$ to represent the amount of computing resources allocated by $ED_{s,n}$ to the training process.

For the wireless transmission in our work, the wireless channel is assumed to remain slow-fading and stable, and it adopts the frequency-division multiple-access communication scheme due to its implementation simplicity and minimal communication interference. However, more advanced wireless communication configurations may also be employed. Subsequently, the communication cost is defined as

$$c^{\mathrm{m}}_{s,n} = P^{\mathrm{m}}_{s,n}\frac{M}{R}\delta_{\mathrm{g}}\varphi_{s,n}, \tag{31}$$

where $\frac{M}{R}\delta_{\mathrm{g}}$ is the total time for model transmission, and $\varphi_{s,n}$ is the data owner $ED_{s,n}$'s unit energy cost for communication.

In BCFL, EDs are trained by creating transactions and invoking smart contracts to complete for the FL parameter aggregation, while the blockchain is deployed on ESs to occupy the computational and storage resources of ESs. To estimate the cost of using blockchain, we employ the calculation of the gas fees used by the blockchain:

$$c^b_s = a_6 g_e p_g |\mathbb{W}_s|, \tag{32}$$

where g_e is the amount of gas required by each ED to invoke a smart contract, p_g is the cost of one unit of gas, a_6 is the factor to convert gas into cost, and $|\mathbb{W}_s|$ denotes the size of workers set.

Each ED invests different computational resources in training to complete the task. Then, the more computational resources an ED invests, the faster its computation, and thus the shorter its running time. Accordingly, the amount of computational resources invested by an ED affects its own revenue. Deriving from (28), we can obtain the revenue of ES_s:

$$\pi_s = U(\mathbb{W}_s) - \sum_{i=1}^{|\mathbb{W}_s|} V_i - c^b_s, \tag{33}$$

where $V_i = p_t\frac{C_i d_i}{\kappa\theta_i}$ is the fee paid by ES to each worker i in $\mathbb{W}_s$. Here, $\frac{C_i d_i}{\kappa\theta_i}$ denotes the total ED utility (C_i is the amount of computing resources that worker i devotes to training, and θ_i is the EMD value of worker i) and p_t is the price of one unit of utility.

Then, the profit function of worker i in $\mathbb{W}_s$ is

$$\pi_i = V_i - c^{\mathrm{m}}_i - c^d_i - c^{\mathrm{c}}_i, i \in \mathbb{W}_s. \tag{34}$$

We model the interactions between ESs and EDs via the Stackelberg game, where ESs play as leaders, EDs as followers, and both of them have the goal of maximizing their own utility.

C. Equilibrium Solution

We utilize the backward induction method to determine the Stackelberg equilibrium, obtaining solutions for both ES and its workers. In detail, we begin by setting the first derivative to zero to solve for the equilibrium solution of the follower's game. Subsequently, we substitute the solution from the leader's subgame with that from the follower's subgame and then calculate the leader's subgame solution.

To achieve the equilibrium solution for the followers subgame, we take the first derivative of the profit function of each ED in workers set with respect to C_i:

$$\begin{aligned} \frac{\partial\pi_i}{\partial C_i} &= \frac{\partial V_i - c^{\mathrm{m}}_i - c^d_i - c^{\mathrm{c}}_i}{\partial C_i} \\ &= \frac{p_t d_i}{\kappa\theta_i} - 2d_i\delta_1\delta_{\mathrm{g}} M\beta_i C_i. \end{aligned} \tag{35}$$

By taking (35) equal to zero, we get the equilibrium computing resources:

$$C^*_i = \frac{p_t d_i}{2d_i\delta_1\delta_{\mathrm{g}} M\beta_i\kappa\theta_i}. \tag{36}$$

Furthermore, to ensure the existence and uniqueness of the Nash equilibrium, the revenue function of each ED should be strictly concave.

After establishing the functional correlation between the computational resources of individual workers and the unit

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

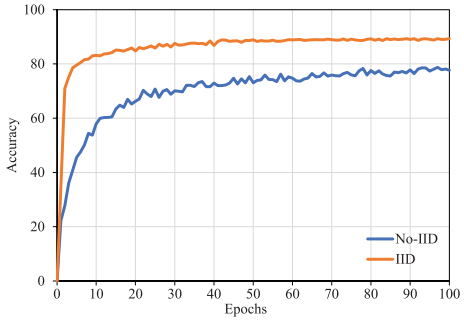


Fig. 5. The training accuracy comparison between the IID and No-IID settings.

price of computational resources, we can proceed to derive the leader's subgame solution for ES. Bringing (36) into (33), we have

$$\pi_s = U(\mathbb{W}_s) - \sum_{i=1}^{|\mathbb{W}_s|} \frac{p_t^2 d_i^2}{2 d_i \delta_1 \delta_g M \beta_i \kappa \theta_i} - c_s^b. \tag{37}$$

Clearly, π_s is a strctly concave function of p_t, and thereby there is an unique p_t^* to maximize the profit of ES. Thus, the uniqueness of this Stackelberg game is proved.

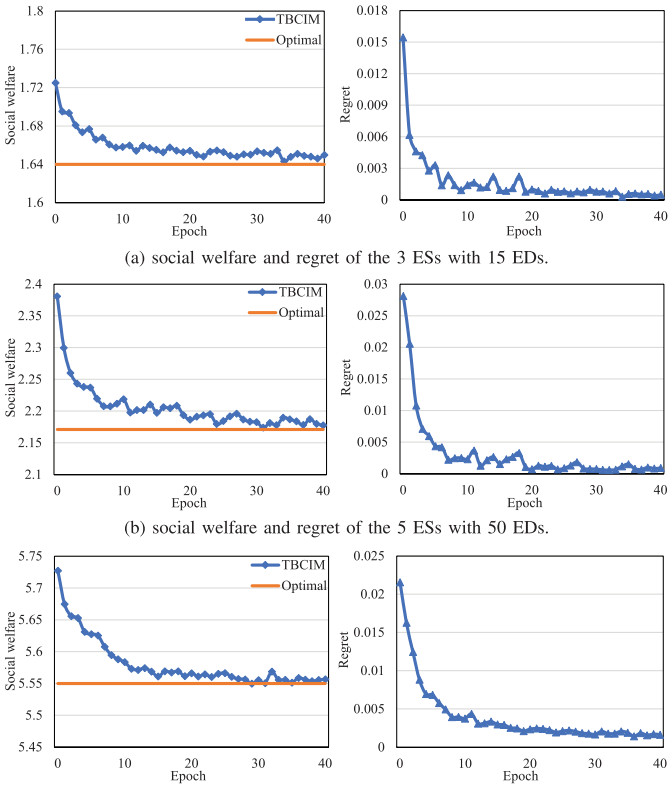
V. Experimental Evaluation

In this section, we experimentally demonstrate that our methodology is capable of retrieving near-optimal auction outcomes for nearly all scenarios with known optimal solutions. Moreover, it excels at discovering novel auction solutions in cases where analytical solutions are absent. Subsequently, we introduce the experimental setting, present various experimental results, and engage in discussions regarding our proposed TBCIM.

A. Experimental Setting

BCFL Setup: For simplicity, we consider 3, 5, and 10 ESs, respectively, and each ES randomly generates 3 to 10 EDs in our experiment. Smart contracts use FedAvg algorithm to aggregate models. We set the federated learning rate to be $\eta = 0.01$, while fixing the number of global epochs and local epochs at $\delta_g = 100$ and $\delta_1 = 5$, respectively. We divide CIFAR-10 according to the mixture distribution principle, and then assign it to the EDs as the No-IID setting between the EDs. We allocate non-independently and identically distributed data to each ES's EDs based on the Dirichlet distribution. In addition, we conduct a comparison of accuracy changes using various data owner selection strategies. Fig. 5 illustrates the results in the No-IID setting, where EDs are randomly chosen at a rate of 0.5. These observations show that the No-IID setting indeed plays a negative impact on the training accuracy. Furthermore, the data distribution of the task requirements is randomly generated by the mixture distribution. We empirically set $a_1 = 0.87$, $a_2 = 10^{-5}$, $a_3 = 0.13$, $a_4 = 10^5$, $a_5 = 0.64$, $a_6 = 3.48 \times 10^{-2}$, $g_e = 7.68 \times 10^4$ and $M = 0.5$ in the experiments.

Reverse Auction Setup. We implement an end-to-end reverse auction using the TensorFlow deep learning library.



(a) social welfare and regret of the 3 ESs with 15 EDs.

(b) social welfare and regret of the 5 ESs with 50 EDs.

(c) social welfare and regret of the 10 ESs with 100 EDs.

Fig. 6. The social welfare and regret of different FL scales.

The training involves a dataset containing 500,000 bidding samples, while the performance evaluation is conducted based on a distinct dataset consisting of 10,000 bidding samples. The parameter ρ in the augmented Lagrangian framework is deliberately set to 1. We utilize the Adam optimizer with a fixed learning rate of 0.001 to iteratively update the weights w^t for each batch. Following each w^t update, we execute a sequence of $R = 10$ misreport update steps. Subsequently, we preserve the optimized misreports for the current batch and employ them to initialize misreports for the same batch in subsequent updates.

B. Simulation Results and Discussion

In this section, we show the numerical results from our evaluations based on the simulations.

1) Discusion of Impacts on Social Welfare: We conduct several simulations to evaluate the performance of our proposed TBCIM framework. Besides, we also investigate the impact of the proposed framework on the bids and revenues of each ES, as well as the truthfulness of the bids.

Fig. 6 shows the convergence curves of social welfare (blue curve) and regret (green curve) for our proposed TBCIM mechanism under various settings of edge servers and edge devices. As illustrated, the high initial social welfare values indicate that the mechanism has not learned incentive-compatible features, allowing for higher revenue through dishonest bidding, which results in relatively high regret values. As the training epochs progress, social welfare gradually approaches the optimal solution. The machine

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

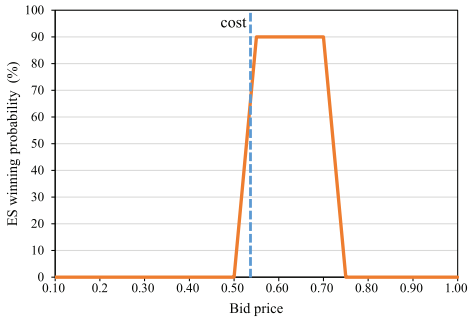


Fig. 7. The relationship between the probability of ES winning and the bidding.

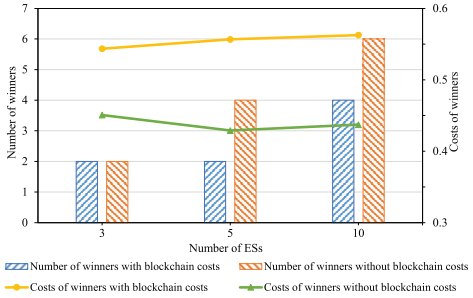


Fig. 8. The impact of blockchain costs on ES wins and profits.

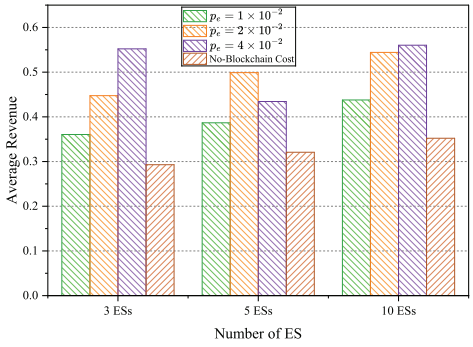


Fig. 9. The impact of blockchain costs on ES wins and profits.

learning solver adjusts the regret by iterating the Lagrange multiplier, initially prioritizing social welfare and then shifting focus to regret in later iterations. Consequently, as iterations increase, the machine learning model reduces regret, ultimately approaching zero. When $rgt = 0$, the optimal solution is achieved. This demonstrates that the machine learning model can learn incentive-compatible features during the auction process. Throughout the training process, the reverse auction mechanism shows remarkable performance, with social welfare nearing the approximate optimal solution and regret becoming negligible. However, in some instances, due to the presence of small non-zero regret, the social welfare obtained by the learned reverse auction mechanism may differ from that of the optimal incentive-compatible auction.

2) Truthfulness: In order to evaluate the truthfulness of ES bids, we randomly select an ES and observe the relationship between its bid and the probability of winning in relation to costs. The results are shown in Fig. 7. When the ES's bid is approximately equal to its cost, the probability of ES winning in the auction is maximized. Due to the influence of ex-post regret, there exists a minimal difference between the cost and the bid. From Fig. 7, we can find that both extremely low and high bids cannot lead to winning in the auction. The reason is that when an ES offers a lower bid, the difference between its revenue and cost reduces the overall revenue of the market, preventing the revenue from reaching an optimal solution. Clearly, when an ES provides an excessively high bid, it is highly improbable to secure a victory in the reverse auction. Based on the above results, we can conclude that RANet is able to learn the requirement for truthfulness in their reverse auction, thereby motivating ESs to refrain from making dishonest bids.

3) Discusion of Impacts on Number of Blockchain Worker Selection: Fig. 8, shows the comparison of the revenue with and without considering the blockchain cost, aiming to illustrate its impact on the performance of the reverse auction. As the number of ESs increases, the realized social welfare increases accordingly, while the number of selected workers decreases. This outcome can be attributed to the fact that with more ESs, they can offer higher bids, leading to more truthful disclosure of their actual value. Consequently, ESs perceive competing for the task as profitable, resulting in a more effective TBCIM framework that motivates both ESs and data owners to actively participate in federated learning training. The increase of data owners competing in the task allocation system is conductive to provide more task choices, and motivate participants to make genuine offers. This helps enable a competitive environment, which encourages participation in the training process.

In addition, Fig. 8 also illustrates the variation in the number of selected ESs under two scenarios: one considering blockchain costs and the other not considering blockchain costs. In the settings with 5 ESs and 10 ESs, compared to the test scenario without considering the blockchain costs, the number of selected ESs decreases by 2 in the test scenario with the blockchain costs. This is because that, with the inclusion of blockchain costs, ESs tend to submit higher bids, and thus lose become less attractive to the task allocation system. In the setting of 3 ESs, the number of workers remain unchanged due to the requirement for a certain amount of training data. In a word, reducing workers would lead to a decrease in BCFL training accuracy and an increase in training time.

4) Impacts on ES's Average Revenue With Blockchain: In the same FL structure, we compare the cost of joining blockchain with the cost of not joining blockchain. Fig. 9 shows the average revenue of participants under different blockchain joining costs, represented by varying the gas price p_e. We analyze three scenarios with gas costs set as $p_e = 1 \times 10^{-2}$, $p_e = 2 \times 10^{-2}$, and $p_e = 4 \times 10^{-2}$, respectively, and compare them with the scenario without blockchain involvement (No-Blockchain Cost). Results indicate that when the gas cost is relatively low ($p_e = 1 \times 10^{-2}$ or 2×10^{-2}), joining blockchain brings higher average revenue compared to the No-Blockchain scenario. However, when gas cost increases

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

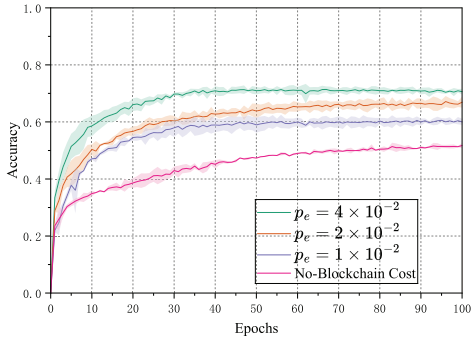


Fig. 10. Impact of blockchain cost on FL accuracy.

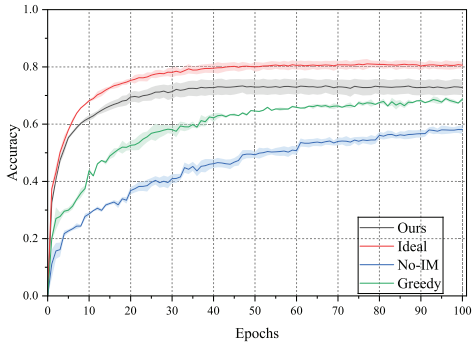


Fig. 11. Comparison of accuracy between different incentive mechanisms and the ideal scenario.

to a higher level ($p_e = 4 \times 10^{-2}$), the revenue advantage significantly diminishes. This implies that blockchain involvement can enhance participants' revenue only if the gas cost is controlled within an acceptable range.

5) Impacts on Accuracy With Blockchain: Fig. 10 presents the convergence behavior of model accuracy under varying blockchain participation costs (p_e), compared to the No-Blockchain scenario. When p_e is relatively high (e.g., 4×10^{-2}), the accuracy converges more rapidly and achieves a significantly higher final value, as the elevated cost incentivizes participation from contributors with higher-quality data. In contrast, lower blockchain costs (e.g., $p_e = 1 \times 10^{-2}$ or 2×10^{-2}) yield more moderate accuracy improvements, though they still surpass the No-Blockchain baseline. These results demonstrate that introducing blockchain participation costs effectively filters and encourages high-quality ESs, thereby enhancing both convergence speed and accuracy in FL.

6) Impact of the Incentive Mechanism on Accuracy: : Fig. 11 compares FL accuracy across four scenarios: Ideal, No-IM, Greedy, and our proposed mechanism (Ours). Results reveal that No-IM shows the poorest performance, with slowest convergence and lowest accuracy, highlighting the necessity of effective incentives for participant engagement and data quality. While Greedy outperforms No-IM, its sole focus on bid values limits accuracy gains and convergence speed. In contrast, our method maintains within 5% of Ideal accuracy, significantly surpassing both No-IM and Greedy. This demonstrates our mechanism's effectiveness in balancing bid value and data quality, improving both convergence and final accuracy.

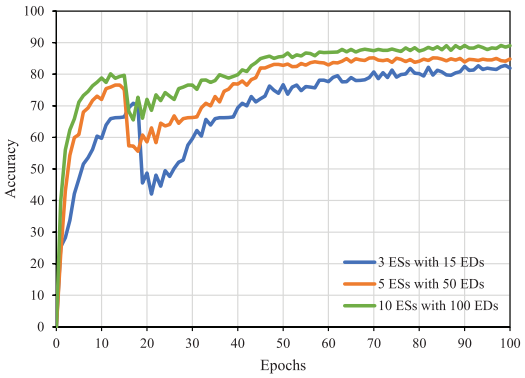


Fig. 12. The training accuracy with 20% of the ESs attacked.

7) Training Accuracy With Attacked ESs: In addition to the aforementioned metrics, we also pay attention to the change in accuracy when ESs are subjected to single-point attacks, as it impacts the stability and robustness of the TBCIM mechanism. As shown in Fig. 12, we have plotted the convergence curve of our proposed TBCIM mechanism when subjected to a single-point attack. We assume that 20% of the servers become unavailable during training between 15-30 epochs. From the figure, it can be observed that when some servers are unavailable, there is fluctuation in the overall model accuracy. However, after training for a certain number of epochs, convergence is still achieved. This is due to the fact that the failure of server results in the loss of its all workers, reducing available training data. Additionally, there is a probability that the failed servers are among the selected ESs, which generally possess relatively better data quality. As the subsequent training progresses, the TBCIM mechanism will transition into a new stable configuration. Consequently, further training will result in the convergence of the overall model.

VI. Related Work

Recently, with the development of jointcloud, the FL service market has emerged with the urgent need of efficient resource trading mechanisms [26]. Fan et al. [40] establish a decentralized and transparent FL resource trading market, where untrustworthy third parties can be prevented via recording transaction records on blockchain. Then, Deng et al. [22] develop a blockchain-based FL trading platform, aiming to ensure data privacy for client participants. However, existing incentive designs only focus on the incentivation of data owners in FL system rather than the resource trading of FL service market.

To promote fairness, as well as enhance participation of trustworthy entities, researches have explored various FL incentive designs, including centered around competition [34], reputation [35], and contract theory [36]. Toyoda and Zhang [37] maximize participants' profits through duplicated competition mechanism. Gao et al. [38] propose to quantify the contributions of participants for the fairness. Furthermore, Kang et al. [39] integrate the reputation evaluation and the contract theory to incentivize high-reputation nodes to possess high-quality data. Several recent studies focus on incentive design of hierarchical FL paradigms, where the

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

TABLE I
THE COMPARATION BETWEEN OUR WORK AND EXISTING MECHANISMS

Reference	EC	BC	CR	DD	NE	ME	T	ML
[40]	✓	✓	✓	×	×	×	×	×
[33]	✓	✓	×	×	×	×	×	×
[22]	✓	✓	✓	×	×	×	×	×
[9]	✓	✓	✓	×	×	×	✓	×
[37]	×	✓	×	×	×	×	✓	×
[38]	×	✓	✓	×	×	×	×	×
[39]	×	✓	×	×	×	×	×	×
[43]	×	✓	✓	×	×	✓	✓	×
[44]	×	✓	×	×	×	×	×	×
[45]	×	✓	×	×	×	×	×	×
TBCIM	✓	✓	✓	✓	✓	✓	✓	✓

EC: Applied in edge computing scenarios.
BC: Applying blockchain technology.
CR: Considering communication resources.
DD: Considering the data distribution of data owners.
NE: Nash equilibrium.
ME: Involving multiple ESs and multiple EDs.
T: Truthfulness.
ML: Using machine learning-aided methods to solve.

data owners upload their local parameters to edge servers (i.e., cluster heads) rather the model owner for intermediate aggregation [42], [43]. However, such designs are not applicable to the actual requirement of edge FL service market, where different edge servers compete for FL tasks published by the dynamic market.

Recently, the utilization of blockchain in FL systems (i.e., BCFL) emerges and tends to be popular. One line of research on BCFL is to improve the security of FL through distributed point-to-point network [25]. In this way, a trusted client selection process can be achieved through the immutability and auditability of blockchain [24]. Wang et al. [9] design a blockchain-based incentive mechanism to balance training overhead and model performance in hierarchical federated learning. Similarly, Xu et al. [33] leverage blockchain to authorize participants for detecting and mitigating malicious nodes. In fact, the blockchain can be regarded as a pivotal tool to enable transparent economic mechanism design [45]. For instance, Rehman et al. [44] propose a blockchain-aided decentralized reputation system to ensure reliable collaborative model training in EC environments. However, the introduction of blockchain leads to the increase of storage and computational overheads. Existing studies address mining-related costs using tokens, but cannot reduce essential expenses (e.g., storage and communication costs) in blockchain. Moreover, these studies do not consider the competitive relationship between ESs/EDs in FL service market.

As known, the implementation of blockchain in BCFL consumes storage and computational resources, thereby increasing the expenses of data owners. Existing researches solve this issue through tokens. Nevertheless, data owners primarily operate as users of the blockchain, and the tokens merely cover the expenses associated with mining. Without adequately considering other costs like storage and communication related to the blockchain, it is hard to obtain the real cost of completing the training task, which will further impact the bidding of servers and hinder the resources trading.

To address the above limitations, we propose a two-level blockchain-aided resource trading mechanism to perform a federated cloud-edge-end cooperation, where the resource trading between edge servers and FL tasks from the task allocation system, the incentivation of data owners for the given task, and the cost of using blockchain are considered comprehensively. For clarity, the comparative study between our work and existing incentive mechanisms is summarized in Table I.

VII. CONCLUSION

This paper introduces a novel blockchain-aided two-level incentive mechanism for federated learning in MEC-based jointcloud. Our approach combines a reverse auction mechanism to maximize social welfare with the Stackelberg game model to distribute benefits among data owners. Our TBCIM considers both the bidding price of ESs for providing training services and the costs incurred by servers for offering blockchain services, ensuring an incentive-compatible auction process. To enhance the authenticity of server bids, a deep neural network reverse auction is proposed to address the reverse auction process and yield social welfare approximation that closely approaches the optimal solution for single-item auctions. Furthermore, the Starkelberg game model provides a unique equilibrium solution while considering resource constraints. As a future direction, the asynchronous mode of this mechanism for blockchain-based edge federated learning service systems will be investigated. At the same time, we will incorporate the mobility and scalability attributes of edge computing to address the challenges pertaining to communication, security, and distance that arise during the dynamic changes in BCFL.

REFERENCES

[1] Z. Cai, X. Zheng, J. Wang, and Z. He, "Private data trading towards range counting queries in Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4881–4897, Aug. 2023, doi: 10.1109/TMC.2022.3164325.

[2] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and communications technologies for sustainable development goals: State-of-the-art, needs and perspectives," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2389–2406, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2812301.

[3] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016, doi: 10.1109/JIOT.2016.2584538.

[4] P. Zhang, C. Wang, C. Jiang, and Z. Han, "Deep reinforcement learning assisted federated learning algorithm for data management of IIoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8475–8484, Dec. 2021, doi: 10.1109/TII.2021.3064351.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[7] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, L.-N. Tran, S. Gong, and E. Dutkiewicz, "Dynamic federated learning-based economic framework for Internet-of-Vehicles," *IEEE Trans. Mobile Comput.*, vol. 22, no. 4, pp. 2100–2115, Apr. 2023, doi: 10.1109/TMC.2021.3122436.

[8] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9148862.

[9] X. Wang, Y. Zhao, C. Qiu, Z. Liu, J. Nie, and V. C. M. Leung, "InFEDge: A blockchain-based incentive mechanism in hierarchical federated learning for end-edge-cloud communications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3325–3342, Dec. 2022, doi: 10.1109/JSAC.2022.3213323.

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

[10] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1310–1321.
[11] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 308–318.
[12] H. Hua, Y. Li, T. Wang, N. Dong, W. Li, and J. Cao, "Edge computing with artificial intelligence: A machine learning perspective," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–35, Sep. 2023.
[13] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8229–8249, Jun. 2022, doi: 10.1109/JIOT.2022.3150363.
[14] 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, 3rd Quart., 2021, doi: 10.1109/COMST.2021.3090430.
[15] Q. Li et al., "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, Apr. 2023, doi: 10.1109/TKDE.2021.3124599.
[16] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, "Blockchain-empowered federated learning: Challenges, solutions, and future directions," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–31, Nov. 2023.
[17] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.I.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020, doi: 10.1109/TNNLS.2019.2944481.
[18] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Oct. 2020.
[19] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019, doi: 10.1109/JSAC.2019.2904348.
[20] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020, *arXiv:2003.02133*.
[21] J. Guo, J. Wu, A. Liu, and N. N. Xiong, "LightFed: An efficient and secure federated edge learning system on model splitting," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2701–2713, Nov. 2022, doi: 10.1109/TPDS.2021.3127712.
[22] Y. Deng, T. Han, and N. Zhang, "FLeX: Trading edge computing resources for federated learning via blockchain," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Vancouver, BC, Canada, May 2021, pp. 1–2, doi: 10.1109/INFOCOMWKSHPS51825.2021.9484628.
[23] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021, doi: 10.1109/TMC.2020.2994639.
[24] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "FLChain: A blockchain for auditable federated learning with trust and incentive," in *Proc. 5th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2019, pp. 151–159, doi: 10.1109/BIGCOM.2019.00030.
[25] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.
[26] Y. J. Kim and C. S. Hong, "Blockchain-based node-aware dynamic weighting methods for improving federated learning performance," in *Proc. 20th Asia–Pac. Netw. Oper. Manag. Symp. (APNOMS*, 2019, pp. 1–4, doi: 10.23919/APNOMS.2019.8893114.
[27] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019, doi: 10.1109/COMST.2018.2886932.
[28] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4585–4600, Jun. 2019, doi: 10.1109/JIOT.2018.2871706.
[29] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, Aug. 2018, doi: 10.1109/MCOM.2018.1701095.
[30] D. Zhang, F. R. Yu, and R. Yang, "Blockchain-based multi-access edge computing for future vehicular networks: A deep compressed neural network approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12161–12175, Aug. 2022, doi: 10.1109/TITS.2021.3110591.
[31] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012.
[32] Y. Jiao, P. Wang, S. Feng, and D. Niyato, "Profit maximization mechanism and data management for data analytics services," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2001–2014, Jun. 2018, doi: 10.1109/JIOT.2018.2819706.
[33] Y. Xu et al., "BESIFL: Blockchain-empowered secure and incentive federated learning paradigm in IoT," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6561–6573, Apr. 2023, doi: 10.1109/JIOT.2021.3138693.
[34] Y. Deng et al., "AUCTION: Automated and quality-aware client selection framework for efficient federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1996–2009, Aug. 2022, doi: 10.1109/TPDS.2021.3134647.
[35] J. S. Ng et al., "Reputation-aware hedonic coalition formation for efficient serverless hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2675–2686, Nov. 2022, doi: 10.1109/TPDS.2021.3139039.
[36] J. Pang, J. Yu, R. Zhou, and J. C. S. Lui, "An incentive auction for heterogeneous client selection in federated learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5733–5750, Oct. 2023, doi: 10.1109/TMC.2022.3182876.
[37] K. Toyoda and A. N. Zhang, "Mechanism design for an incentive-aware blockchain-enabled federated learning platform," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 395–403, doi: 10.1109/BigData47090.2019.9006344.
[38] L. Gao, L. Li, Y. Chen, C. Xu, and M. Xu, "FGFL: A blockchain-based fair incentive governor for federated learning," *J. Parallel Distrib. Comput.*, vol. 163, pp. 283–299, May 2022.
[39] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019, doi: 10.1109/JIOT.2019.2940820.
[40] S. Fan, H. Zhang, Y. Zeng, and W. Cai, "Hybrid blockchain-based resource trading system for federated learning in edge computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2252–2264, Feb. 2021, doi: 10.1109/JIOT.2020.3028101.
[41] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
[42] W. Y. B. Lim et al., "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 536–550, Mar. 2022, doi: 10.1109/TPDS.2021.3096076.
[43] P. Li and S. Guo, "Incentive mechanisms for device-to-device communications," *IEEE Netw.*, vol. 29, no. 4, pp. 75–79, Jul. 2015, doi: 10.1109/MNET.2015.7166194.
[44] M. H. ur Rehman, K. Salah, E. Damiani, and D. Svetinovic, "Towards blockchain-based reputation-aware federated learning," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 183–188, doi: 10.1109/INFOCOMWKSHPS50562.2020.9163027.
[45] N. B. Somy et al., "Ownership preserving AI market places using blockchain," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Atlanta, GA, USA, Jul. 2019, pp. 156–165, doi: 10.1109/BLOCKCHAIN.2019.00029.

Lianbo Ma (Senior Member, IEEE) received the B.Sc. degree in communication engineering and the M.Sc. degree in communication and information systems from Northeastern University, Shenyang, China, in 2004 and 2007, respectively, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2015. He is currently a Professor with Northeastern University. He has published over 90 journal articles, books, and refereed conference papers. His current research interests include computational intelligence and machine learning.

Ying Qian received the B.Sc. degree in information management and information systems from Tianjin University of Technology, Tianjin, China, in 2017. She is currently pursuing the Ph.D. degree with Northeastern University, Shenyang, China. Her current research interests include federated learning, deep learning, and edge computing.

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.

Guo Yu (Member, IEEE) received the B.Sc. degree in information and computing science and the M.Eng. degree in computer technology from Xiangtan University, Xiangtan, China, in 2012 and 2015, respectively, and the Ph.D. degree in computer science from the University of Surrey, Guildford, U.K., in 2020. He is currently an Associate Professor at Nanjing Tech University. His current research interests include evolutionary optimization and machine learning.

Zhetao Li (Member, IEEE) received the B.Eng. degree in electrical information engineering from Xiangtan University in 2002, the M.Eng. degree in pattern recognition and intelligent system from Beihang University in 2005, and the Ph.D. degree in computer application technology from Hunan University in 2010. From December 2013 to December 2014, he was a Post-Doctoral Researcher of wireless network with Stony Brook University. He is currently a Professor with the College of Information Science and Technology, Jinan University. He is a member of CCF.

Liang Wang (Member, IEEE) received the Ph.D. degree from Shenyang Institute of Automation (SIA), Chinese Academy of Sciences, Shenyang, China, in 2014. He is currently a Professor with the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. His research interests include ubiquitous computing, mobile crowd sensing, and crowd computing.

Qing Li (Senior Member, IEEE) received the B.Sc. degree in computer science and technology from Dalian University of Technology, Dalian, China, in 2008, and the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2013. He is currently an Associate Researcher with the Peng Cheng Laboratory, Shenzhen, China. His research interests include network function virtualization, network caching/computing, intelligent self-running networks, and edge computing.

Xingwei Wang received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 1989, 1992, and 1998, respectively. He is currently a Professor at Northeastern University. He has published over 100 journal articles, books, and refereed conference papers. His current research interests include cloud computing and future internet.

Guangjie Han (Fellow, IEEE) received the Ph.D. degree from Northeastern University, Shenyang, China, in 2004.

In February 2008, he finished his work as a Post-Doctoral Researcher with the Department of Computer Science, Chonnam National University, Gwangju, South Korea. From October 2010 to October 2011, he was a Visiting Research Scholar with Osaka University, Suita, Japan. From January 2017 to February 2017, he was a Visiting Professor with the City University of Hong Kong, China. From July 2017 to July 2020, he was a Distinguished Professor with Dalian University of Technology, Dalian, China. He is currently a Professor with the Department of Internet of Things Engineering, Hohai University, Changzhou, China. He has over 500 peer-reviewed journals and conference papers, in addition to 160 granted and pending patents. His H-index is 66 and i10-index is 287 in Google Citation (Google Scholar). The total citation count of his articles raises above 16 000 times. His current research interests include the Internet of Things, industrial internet, machine learning and artificial intelligence, mobile computing, security, and privacy. He is a Fellow of U.K. Institution of Engineering and Technology (FIET). He served as the chair for the organizing and technical committees in many international conferences. He has been awarded the 2020 IEEE Systems Journal Annual Best Paper Award and the 2017–2019 IEEE Access Outstanding Associate Editor Award. He has served on the editorial boards for up to ten international journals, including IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE SYSTEMS JOURNAL. He has guest-edited several special issues in IEEE journals and magazines, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *IEEE Communications Magazine*, IEEE WIRELESS COMMUNICATIONS, and *Computer Networks*.

Authorized licensed use limited to: Peng Cheng Laboratory. Downloaded on December 31,2025 at 02:51:35 UTC from IEEE Xplore. Restrictions apply.