2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) | 979-8-3315-6532-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/Trustcom66490.2025.00149

A Blockchain-Enabled AIoT Framework for Secure Metaverse in Wireless Communication Networks

Danhuai Zhao[†], Chen Tian[†], Zhenyu Ju[¶], Yi Rong[‡], Xiaoming He[§], Wanting Yang[‡‡]
[†]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[¶]China Mobile Zijin Innovation Institute, Nanjing, China
[‡]College of Computer Science and Software Engineering, Hohai University, Nanjing, China
[§]College of Internet of Things, Nanjing University of Posts and Telecommunication, Nanjing, China
[‡‡]Singapore University of Technology and Design, Singapore, Singapore
E-mail: zhaodanhuai@nju.edu.cn, tianchen@nju.edu.cn, juzhenyu@js.chinamobile.com,
rongyi1220@163.com, hexiaoming@njupt.edu.cn, wanting_yang@sutd.edu.sg

***Abstract*—The wireless communication network, comprising billions of cloud, edge, and end devices, enables emerging applications such as the metaverse—an immersive virtual environment where experiences and user interactions converge. However, user interactions in the metaverse generate substantial amounts of private data (*e.g.*, identity, location), raising significant privacy concerns despite their utility in training machine learning models. Artificial Intelligence of Things (AIoT) offers solutions to these challenges, with Federated Learning (FL) serving as a decentralized framework that enables collaborative model training without exposing local data. Nevertheless, deploying FL in large-scale environments like the metaverse increases vulnerability to malicious attacks. To address this issue, we propose a blockchain-based FL architecture that enhances trust and security. The architecture integrates a multi-task FL strategy with blockchain sharding to boost system throughput and reduce resource consumption. By partitioning the blockchain into smaller shards, we lower computational demands and enable concurrent training of multiple models, improving efficiency. We also design a shard creation algorithm based on bipartite matching and a bandwidth scheduling mechanism that prioritizes reliable devices with informative data. Experimental results show that our architecture outperforms existing baselines across multiple evaluation metrics.**

***Index Terms*—Wireless communication networks, AIoT, Federated learning, Blockchain, Metaverse**

I. Introduction

With the rapid advancement of information technology, global communication demands are growing exponentially, overwhelming traditional networks. To address this, wireless communication networks—large-scale systems comprising billions of cloud, edge, and end devices—have been proposed. These networks enable groundbreaking applications like the metaverse [1], [2], a virtual space for immersive, interactive experiences. Supported by technologies such as AR/VR, blockchain, edge computing, AI, and 6G, the metaverse is rapidly evolving [3], [4]. However, its reliance on sensitive user data raises significant privacy and security concerns, including risks of digital crimes involving personal information [5], [6].

Corresponding author: Chen Tian (tianchen@nju.edu.cn)

The integration of AI and IoT has led to Artificial Intelligence of Things (AIoT), which uses AI to process massive IoT-generated data. Since this data often contains sensitive information, security and privacy are critical in AIoT. Federated Learning (FL) has emerged as a key privacy-preserving solution [7], [8]. Instead of centralizing data, FL enables edge devices to collaboratively train models locally, sharing only model updates (gradients) with a central server [9]. This approach minimizes data exposure, significantly reducing leakage risks.

Applying FL in the metaverse faces a key challenge: *trust issues* [10]. Since FL relies on multiple participants for model training, ensuring all are trustworthy is difficult. In the metaverse, the large number of participants increases malicious actors who may submit false data, disrupting training. Thus, verifying the credibility of all devices in FL tasks remains unresolved.

Blockchain provides a decentralized, tamper-proof ledger for recording FL activities, enhancing trust among participants in the metaverse [11]. It enables secure transactions, identity verification, and decentralized governance without third-party intermediaries. By integrating FL with blockchain, model training becomes private yet transparent—users can update models locally and share only encrypted updates via blockchain, keeping personal data secure [12]. However, despite these advantages, three key challenges remain when combining FL and blockchain in the metaverse.

1) Resource Constraints: The metaverse faces computing and bandwidth limitations that challenge FL and blockchain adoption. Edge devices must handle both model training and demanding tasks like real-time rendering, but their restricted resources struggle with growing model sizes and complex FL workloads. Bandwidth also fails to scale with increasing devices [13]. While some FL studies optimize

2324-9013/25/$31.00 ©2025 IEEE
DOI 10.1109/Trustcom66490.2025.00149

resource efficiency, they often limit device participation to meet energy and latency requirements, neglecting critical factors like data quality and distribution.

2) Blockchain Scalability Issues: The expanding metaverse user base strains blockchain's limited scalability in throughput, networking, and storage. While blockchain sharding—splitting the network into parallel transaction-processing shards—offers a solution, it introduces new challenges: compromised security, inconsistent shard states, and interoperability gaps. For example, single-shard attacks by malicious actors disrupt operations, while unreliable edge devices in FL can delay model convergence or submit corrupted updates.

3) Emerging Security Threats: The FL-blockchain integration in the metaverse introduces new vulnerabilities. While blockchain secures model updates against tampering, it cannot prevent initial poisoning attacks from malicious edge devices that deliberately bias shared models. This risks flawed avatar-related decisions. Current reputation systems [14] mitigate attacks by scoring participant reliability, but they overlook two critical factors: update accuracy, and overall device performance.

In our work, a blockchain-based FL architecture in the metaverse is presented to provide trustworthiness and security. The architecture designs a multi-task FL strategy, which adopts the blockchain sharding technique to improve throughput and reduce the usage of computing and communication resources. In particular, multiple blockchain shards are implemented, aiming to realize parallel model learning. To select suitable devices during the process of model training, the data's attributes as well as the devices' reliability are simultaneously taken into account. Specifically, edge devices with more informative data are prioritized for inclusion in the training, while those with malicious data are deemed unreliable and are not allowed to participate in the training by the reputation system. Besides, a resource allocation scheme is designed to reasonably allocate bandwidth to selected edge devices. The main contributions of this paper are summarized as follows.

- We present a novel blockchain-based FL architecture to enhance the trustworthiness and security in the metaverse. The architecture realizes parallel model learning without exposing private data from the physical world. CIFAR-100 and TinyImageNet datasets are adopted to assess the effectiveness of our proposed architecture.
- In the metaverse, a bipartite matching strategy is developed to establish shards for trustworthy and secure FL. The method accurately aligns reliable edge devices with FL models, while unreliable edge devices are not considered.
- A resource allocation scheme is proposed to reasonably allocate bandwidth to trustworthy edge devices. The optimization objective is to prioritize edge devices with more informative data. Meanwhile, the communication cost is minimized.

II. Preliminaries

A. Federated Learning

FL refers to that multiple decentralized participants (edge devices) collaboratively train a shared model without sharing their local data. Specifically, given P participants, The local dataset of each participant is denoted as $(\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_P)$. Here, $\mathcal{D}_P$ is the local dataset of the Pth participant. Each participant uses its own local dataset to train the machine learning model deployed on the participant. FL makes the participants' average loss $\mathcal{L}(\boldsymbol{w})$ converge, thus seeking the optimal global model parameters w from the parameter space $\mathbb{R}^l$. $\mathcal{L}(\boldsymbol{w})$ is denoted as

$$\min_{w\in\mathbb{R}^l} \mathcal{L}(w) = \frac{\sum_{r=1}^{P} \mathcal{L}_r(w)}{\mathcal{M}}, \text{ where } \mathcal{M} = \sum_{r=1}^{P} |\mathcal{D}_r|, \quad (1)$$

where $\mathcal{L}_r(w)$ denotes the ith participant's loss function. $\mathcal{M}$ represents as the quantity of data samples across total P participants. In each communication round of FL, a portion of the participants is first chosen to train their individual models using the local datasets of these participants. Then, the gradients of individual models are uploaded to a central server for aggregation, thus adjusting the parameters of the global model.

B. Blockchain Sharding

Blockchain sharding is treated as a technique, which enables nodes to achieve concurrent processing and multi-transaction validation. Specifically, each node arbitrarily chooses a partition of a transaction, which is validated by an externally determined constraint function. Then, nodes engaged in transaction validation need to reach a consensus.

The blockchain permits each block to conduct multiple transactions. The vector representation of transaction entries is denoted as $\mathcal{TE} = (l_1, l_2, \ldots, l_{B_s})$, where l_i is the transaction in the ith block. B_s represents the size of the block. To insert a new block into the blockchain, each transaction l requires to be verified by nodes leveraging a voting system. The voting system is expressed as

$$\begin{aligned} \mathcal{V} : \mathbb{N} \times \mathbb{T} &\to \{1, 0\} \\ (i, l) &\to \mathcal{V}(i, l), \end{aligned} \quad (2)$$

where $\mathbb{N}$ stands for the set of nodes. $\mathbb{T}$ denotes the set of all transactions. The verification process of the blockchain sharding is conducted across nodes, thus generating the set $\mathcal{S}$ of shards with k diverse shards. The ith shard is represented as $\mathcal{S}_i = l_{i,j}\,(1 \le j \le |\mathcal{S}_i|)$. Nodes can access the third-party verification mechanism. Nodes in a blockchain hold three duties, which ensure the proper work of the system. First, nodes operate transactions through attaining an agreement on

a transaction's validity. Second, transactions are confirmed by judging whether nodes meet the verification approach. This is achieved by detecting whether all concurrent nodes on the shard consent to the transaction. Third, based on the property of the transaction, nodes retain confirmed transactions either off-chain or on-chain. Due to the three duties, the blockchain becomes a transparent and decentralized ledger, which has the benefits of security and integrity. Leveraging the automatic parallelization of multiple shards' available computing capacity, blockchain sharding intends to accomplish the goal. In each epoch, the process of sharding verifies a set of transactions $\mathcal{S} = \bigcup_{i=1}^{g_i} \mathcal{S}_i$, where g_i denotes the sum of nodes that engage in the verifying process of the $\mathcal{S}_i$.

An increase in the shard's quantity (sub-chains) can enhance the scalability of the network. Without regard to the amount of shards, the concurrent nodes on one shard have constant communication bandwidth and computing capacity. Due to the ample sharding to validate transactions, the network has the ability to scale.

C. Poisoning Attacks

In FL, the participants' data may contain some private information such as phone numbers, addresses, and workplaces. In addition to this private information, intentionally modifying participants' data results in poisoning attacks. For instance, attackers maliciously tamper with user images that are adopted to train a facial recognition model. This leads to the adverse consequence, *i.e.*, legitimate users are replaced by others. Since participants are authorized to adjust the parameters of the model, poisoning attacks exist in the training phase of the model. Poisoning attacks occur when attackers adopt tampered data to train local models on the participants, leading to degradation of model performance. Poisoning attacks target the integrity and availability of the data. Classic poisoning attacks focus on modifying the training samples' labels or the target class of the training data while keeping other attributes of the data unchanged. In the following section, taking a poisoning attack as an example, the label-flipping attack is illustrated.

III. System Overview

To address these issues, we propose a multi-task FL architecture using blockchain sharding to prevent poisoning attacks and enable parallel training. As shown in Figure 1, the process includes releasing FL tasks, evaluating device trustworthiness via a reputation system, applying blockchain sharding for scalability, training local models on trusted devices, and uploading gradients to the central server.

Our system model consists of multiple edge devices communicating via a wireless network. Each device trains local models on private data and uploads gradients to a central server. Devices also act as blockchain miners within

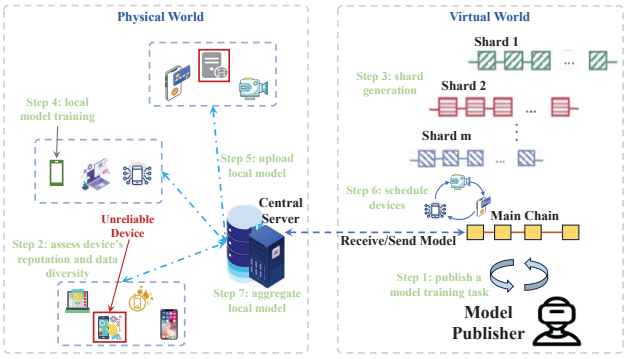

Fig. 1: The overview of our proposed architecture.

a sharding layer, which includes a main chain for global model updates and subchains for local model training and updates. Each shard has a primary node (central server) and a cluster of edge devices that collaboratively train local models until the global model meets performance targets or training rounds complete.

A. Function of System Model

According to edge devices of the information volume and reputation scores, we choose reliable edge devices to engage in model training. The function of the system model is described as follows:

1) FL model release. All blockchain miners (*e.g.*, edge devices) acquire a FL task and corresponding model performance requirements (*e.g.*, precision, and training time) from the model publisher. Edge devices that meet the demands of the model publisher are eligible to engage in the model training.

2) Assessment of the edge device's reputation and data diversity. The model publisher chooses reliable devices by assessing their reputation and data diversity. The reputation assessment relies on the precision of the local model on the test set. The data diversity is evaluated based on the degree of information in the edge device's dataset.

3) Shard generation. After evaluating the edge device's reputation and data diversity, we select the reliable devices that satisfy shard standard, *i.e.*, edge devices with excellent reputation and more informative data. The selected devices autonomously determine whether to participate in model training.

4) Local model training. After sharding generation, each shard member trains their local models through their data. The aggregation mode is weighted averaging. The node with the highest reputation is treated as the primary node.

5) Cross-shard communication. The mechanism can prioritize devices with a high reputation.

6) Block creation. After completing the training, the model publisher updates the reputation of the devices that participated in the training.

IV. Blockchain Sharding-based FL

In this section, we first detail the various function steps of the system model, especially for shard generation, choosing reliable devices, and cross-shard communication. Then, bandwidth allocation and device selection problem is introduced. Finally, we describe the solution to the problem.

A. Shard Generation Model

The procedure of the shard generation is shown in **Algorithm 1**. Evaluating a device's reputation is important to ensure a reliable training procedure. For the tth communication round in a communication period, the reputation score of the pth participating device is calculated as

$$\begin{aligned}\mathcal{R}_p^t = \mathcal{R}_p^{t-1} - \zeta\left(\mu_1\left(pre_k^{local} - \text{avg}(pre)\right) + \\ \mu_2\left(pre_p^{local} - pre_p^{test}\right),\end{aligned} \tag{3}$$

where $\zeta \in [0,1]$ denotes the reputation coefficient that reduces the reputation score. $\mu_1, \mu_2 \in [0,1]$ represents the weights. $\text{avg}(pre)$, pre_p^{test}, and pre_p^{local} is the average precision of all received local models in the communication period, the test precision of the pth participating device, and the local precision of the pth participating device, respectively. Leveraging Equation (3), a malicious device's reputation score reduces when it asserts a higher precision compared to the test precision or uploads a substandard model update.

Considering the property of multi-task FL and the scalability of blockchain, the optimization problem of blockchain shard generation can be defined as

$$\begin{aligned}&\max \sum_{j=1}^{J}\sum_{p=1}^{P_j} R_{p,j}^t x_{p,j}, \\ &s.t. \sum_{j=1}^{J} x_{p,j} \le 1, \forall p \in [1, P_j], \\ &\sum_{p=1}^{P_j} x_{p,j} \ge 1, \forall p \in [1, P_j], \\ &x_{p,j} \in \{0,1\}, \forall p \in [1, P_j],\end{aligned} \tag{4}$$

where $x_{p,j} = 0$ denotes that the pth device is not choosed during the training in the jth shard, while $x_{p,j} = 1$ represents that the pth device is choosed for the training in the jth shard. The above problem is a mixed-integer nonlinear problem, which is NP-hard. Based on the $\sum_{j=1}^{J} x_{p,j} \le 1, \forall p \in [1, P_j]$, each device is assigned to at most one shard. According to constraint $\sum_{p=1}^{P_j} x_{p,j} \ge 1, \forall p \in [1, P_j]$, each shard must contain at least one device. To address the problem, we establish the problem as an integer problem,

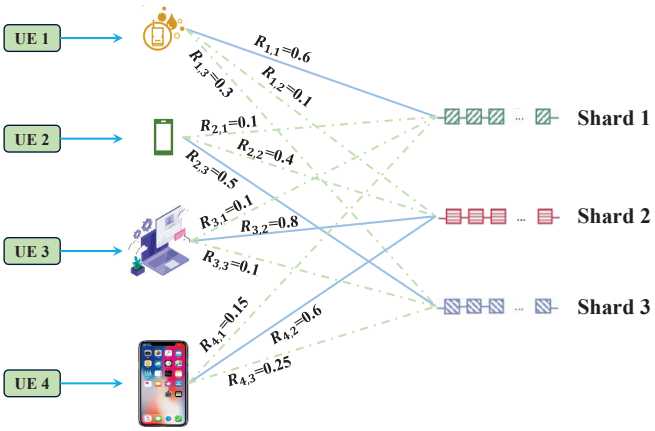

Fig. 2: Experimental results on LondonHW under different hyperparameter settings.

i.e., matching shards with devices based on the devices' reputation values. The objective is to improve the total reputation value of the selected devices. Once the model training task is announced by the model publisher, shard generation begins. Whenever receives a model training task, **Algorithm 1** is executed. Figure 2 shows the shard generation.

Algorithm 1 Shard generation

Input: M FL models preparing to train, P edge devices
Output: New shards S_m

```
S = ∅
Assessing the reputation of each device:
for m = 1, 2, ..., M do
    for p = 1, 2, ..., P do
        Leveraging Equation (3) to compute the pth device's reputation R_p for the model m
    end for
end for
Shard generation:
for m = 1, 2, ..., M do
    Create shard S_m
    for p = 1, 2, ..., P do
        if R_p = max(R_p) then
            Inserting the device to S_m
        end if
    end for
    return Shard
end for
```

B. Reliable Devices Selection

Since data from different has non-IID properties, it is necessary to consider the diversity of the data when choosing devices. Because each dataset is measured by different evaluation metrics, we design a diversity assessment function

through weighting. Specifically, in the communication round t, The diversity index $\mathcal{D}_p^t$ is calculated as

$$\mathcal{D}_p^t = \sum_i \eta_{i,p}\lambda_{i,p}, \tag{5}$$

where $\eta_{i,p}$ and $\lambda_{i,p}$ stands for the elements' diversity and dataset size of the pth device. The selection parameter is expressed as

$$S_p^t = \varphi_1 \mathcal{R}_p^t + \varphi_2 \mathcal{D}_p^t, \tag{6}$$

where φ_1 and φ_2 denotes fixed weight parameters. The benefit of the selection parameter is that it can be validated on local devices without leaking any privacy.

C. Communication Model

Since insufficient power of the device causes delays and disconnections, we must reasonably adopt bandwidth in model transmission. In our work, Orthogonal Frequency Division Multiple Access (OFDMA) is adopted to upload local models from devices to the central server. In particular, the transmission rate $\mathcal{TR}_p$ of the pth device to a central server is defined as

$$\mathcal{TR}_p = \psi_p B \log_2\left(1 + \frac{M_p g_p}{\psi_p B N_0}\right), \quad \forall p \in [1, P], \tag{7}$$

where g_p, M_p, ψ_p, and N_0 stand for the channel gain, the transmission power, the allocated bandwidth, and the power spectral density of the Gaussian noise, respectively.

the communication round continues until the last participating device uploads its local model. Hence, the constraints of model transmission are set as follows

$$\left(ti_p^{train} + ti_p^{upload} + ti_p^{PBFT}\right) x_p \leq T, \quad \forall p \in [1, P], \tag{8}$$

where T represents the deadline. ti_p^{train}, ti_p^{PBFT}, and ti_p^{upload} denote the training time, consensus time, and uploading time, respectively.

Since the dataset and hardware of the devices affect the training time, ti_p^{train} is computed as

$$ti_p^{train} = \left|\mathcal{D}_p^t\right| \frac{\zeta_p}{\rho_p}, \tag{9}$$

where ρ_p denotes the CPU's power. ζ_p (cycles/bit) represents the number of CPU cycles required to handle one data sample.

ti_p^{upload} is represented as

$$ti_p^{upload} = \frac{si}{dr_p}, \tag{10}$$

where si denotes the size of the model. dr_p stands for the actual data transmission rate of the pth device.

ti_p^{PBFT} is represented as

$$ti_p^{PBFT} = ti_v + \frac{B_S \log(m)}{dr_p} + \frac{2H_S \log(m)}{dr_p} + c_a, \tag{11}$$

where ti_v and c_a denote the round-by-round verification time and block cost, respectively. B_S and H_S are block and header sizes, respectively.

D. Bandwidth Allocation and Device Scheduling

The optimization problem of bandwidth allocation and device scheduling is formulated as

$$\max \sum_p R_p x_p, \tag{12}$$

$$s.t. \sum_p b_p \leq 1, \tag{12a}$$

$$\left(ti_p^{train} + ti_p^{upload} + ti_p^{PBFT}\right) x_p \leq T, \forall p \in [1, P], \tag{12b}$$

$$x_p \in \{0, 1\}, \forall p \in [1, P], \tag{12c}$$

$$0 \leq b_p \leq 1, \forall p \in [1, P], \tag{12d}$$

where b_p represents the percentage of bandwidth assigned to the pth device. Constraint (12a) limits that the total bandwidth allocated to all shards is less than the available bandwidth. Based on the constraint (12a), model training and uploading of selected devices is finished before the deadline T set by the model publisher. According to the constraint (12c), the x_p is defined as a binary value. Constraint (12d) defines the range of the bandwidth allocation ratio for a device. If the optimization problem of bandwidth allocation and device selection only includes constraints (12a), (12c), and (12d), the problem is a knapsack problem, which is NP-hard. The greedy approach is leveraged to solve the optimization problem while taking knapsack constraints and the constraint (12b) into account.

V. Experiments

A. Experimental Setting

In our simulations, we use CIFAR-100 and TinyImageNet for image classification. CIFAR-100 is split into 48,000 training and 12,000 test samples, while TinyImageNet has 8,000 training and 2,000 test samples. The TensorFlow ML module is used, with each communication round set to 250 seconds and the model size fixed at 75KB. Device training time is estimated based on our experimental setup.

In the training phase, we simulate non-IID data across heterogeneous devices using a label-based partitioning method. CIFAR-100 and TinyImageNet are divided into 100 and 200 label-specific clusters, respectively, with each CIFAR-100 cluster containing 480 samples and each TinyImageNet cluster containing 40. Devices are randomly assigned 1–30 clusters at the start of training.

To simulate label-flipping attacks, we define source and target label pairs for CIFAR-100 and TinyImageNet. In CIFAR-100, (21, 4) is the easiest and (12, 7) the hardest

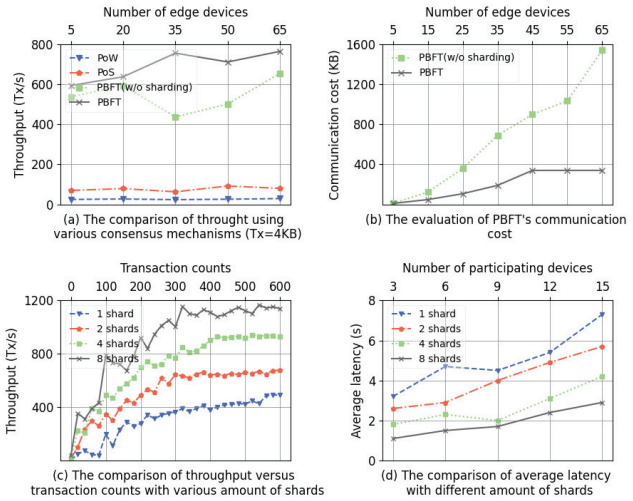

Fig. 3: The evaluation of blockchain sharding.

to classify. In TinyImageNet, the source and target pairs are (24, 5) and (15, 11), respectively.

To simulate a real FL environment, a square cellular network with a side length of 400 meters is applied. The network is composed of 63 edge devices and 1 central server. 63 edge devices are randomly deployed in the cellular network. Through Rayleigh distribution's fading coefficients, the channel gain between edge devices and the central server is denoted as

$$|\Omega_p|^2 = \Theta_p^{-\alpha} |\Upsilon_p|^2, \tag{13}$$

where Θ_p represents the distance between the central server and an edge device. Υ_p stands for the Rayleigh random variable.

B. The Effectiveness of Blockchain Sharding

To evaluate consensus mechanisms, we compare the throughput of PoS, PoW, and PBFT (with and without sharding), as shown in Figure 3(a). PBFT achieves the highest throughput due to its ability to validate parallel transactions across edge devices. Unlike PoS and PoW, whose throughput remains constant, PBFT's throughput increases with the number of edge servers, as PoS and PoW limit transaction volume regardless of device count.

Figure 3(b) shows that PBFT with sharding reduces communication cost compared to PBFT without sharding, as only a subset of trusted edge devices validate transactions. As a result, its communication cost remains stable despite an increasing number of edge devices.

Figure 3(c) shows how shard count affects transaction volume and throughput. Throughput stabilizes after 300 transactions, with single-shard systems stabilizing fastest. Increasing the number of shards further improves overall throughput.

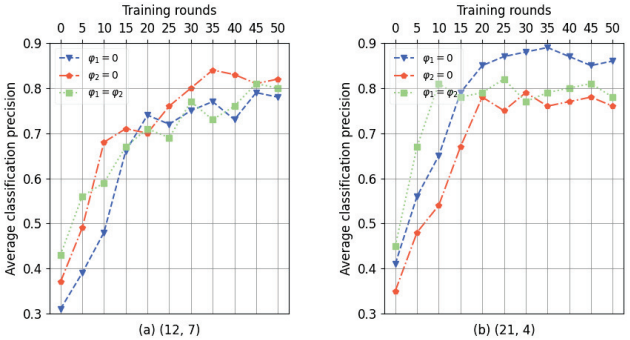

Fig. 4: The average classification precision in relation to the source tuple and the target tuple on dataset CIFAR-100.

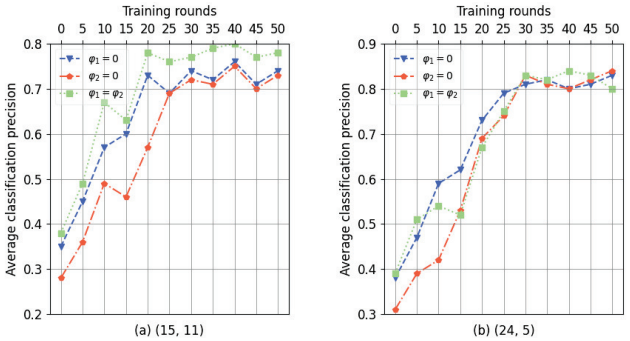

Fig. 5: The average classification precision in relation to the source tuple and the target tuple on dataset TinyImageNet.

Figure 3(d) shows that average latency increases with the number of devices per shard. This is due to more complex intra-shard communication and higher transaction volume, leading to increased network and processing delays.

C. The Evaluation of Data Diversity and Reputation

By limiting the number of selected devices, we evaluate the impact of reputation and data diversity on model performance, with seven adversarial devices flipping labels randomly. Figures 4 and 5 show accuracy under varying φ_1 and φ_2 on CIFAR-100 and TinyImageNet. Figure 4 confirms that more local computations improve accuracy, and Figure 4(b) highlights the importance of reputation and data diversity, especially for label pair (21, 4).

Figure 5 shows that more training rounds improve accuracy. Early on, the reputation factor has a greater impact than data diversity due to its dominant role in device selection. Over time, their influence balances out, with accuracy converging for $\varphi_1 = 0$ and $\varphi_2 = 0$. Notably, $\varphi_1 = \varphi_2$ consistently yields the highest accuracy, highlighting the benefit of jointly considering reputation and data diversity.

VI. Conclusion

In this paper, we have presented a FL architecture based on blockchain to achieve metaverse's trustworthiness and security. The architecture intends to accomplish the parallel training through the blockchain sharding technique. We have split the blockchain into multiple shards. FL models have performed training tasks on these shards, each of which contains some reliable edge devices. We have proposed a bipartite matching strategy to create blockchain shards. The strategy has ensured that the FL model can select reliable edge devices to execute training tasks. Finally, We have adopted CIFAR-100 and TinyImaggeNet datasets to conduct simulation experiments, which evaluate the effectiveness of our proposed solution in the metaverse.

VII. ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China under Grant No. 2022YFB2702800.

References

[1] W. Zeng, J. Zheng, L. Gao, J. Niu, J. Ren, H. Wang, R. Cao, and S. Ji, "Generative ai-aided multimodal parallel offloading for aigc metaverse service in iot networks," *IEEE Internet of Things Journal*, 2025.

[2] L. Zhang, X. Wu, Y. Ma, and H. Kan, "Data exchange for the metaverse with accountable decentralized ttps and incentive mechanisms," *IEEE Transactions on Big Data*, 2025.

[3] A. Qayyum, M. A. Butt, H. Ali, M. Usman, O. Halabi, A. Al-Fuqaha, Q. H. Abbasi, M. A. Imran, and J. Qadir, "Secure and trustworthy artificial intelligence-extended reality (ai-xr) for metaverses," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–38, 2024.

[4] Y. Rong, Y. Mao, H. Cui, X. He, and M. Chen, "Edge computing enabled large-scale traffic flow prediction with gpt in intelligent autonomous transport system for 6g network," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[5] H. Guo, H.-N. Dai, X. Luo, Z. Zheng, G. Xu, and F. He, "An empirical study on oculus virtual reality applications: Security and privacy perspectives," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.

[6] Z. Zhang, C.-H. Lung, X. Wei, M. Chen, S. Chatterjee, and Z. Zhang, "In-network caching for icn-based iot (icn-iot): A comprehensive survey," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 595–14 620, 2023.

[7] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning: Concepts, advances, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3615–3634, 2024.

[8] Z. Niu, H. Dong, A. K. Qin, and T. Gu, "Flrce: Resource-efficient federated learning with early-stopping strategy," *IEEE Transactions on Mobile Computing*, 2024.

[9] Z. Zhu, Y. Shi, P. Fan, C. Peng, and K. B. Letaief, "Isfl: Federated learning for non-iid data with local importance sampling," *IEEE Internet of Things Journal*, 2024.

[10] X. Hou, J. Wang, C. Jiang, Z. Meng, J. Chen, and Y. Ren, "Efficient federated learning for metaverse via dynamic user selection, gradient quantization and resource allocation," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 4, pp. 850–866, 2023.

[11] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–35, 2022.

[12] V. T. Truong and L. B. Le, "Security for the metaverse: Blockchain and machine learning techniques for intrusion detection," *IEEE Network*, vol. 38, no. 5, pp. 204–212, 2024.

[13] X. Zhou, C. Liu, and J. Zhao, "Resource allocation of federated learning for the metaverse with mobile augmented reality," *IEEE Transactions on Wireless Communications*, 2023.

[14] M. H. ur Rehman, K. Salah, E. Damiani, and D. Svetinovic, "Towards blockchain-based reputation-aware federated learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 183–188.