2022 IEEE the 14th International Conference on Wireless Communications and Signal Processing

Blockchain-enabled Edge Computing Framework for Hierarchic Cluster-based Federated Learning

Xiaoge Huang*, Yuhang Wu*, Zhi Chen*, Qianbin Chen*, Jie Zhang†
**School of Communication and Information Engineering,*
Chongqing University of Posts and Telecommunications, Chongqing 400065, China
†School of Communication and Information Engineering, University of Sheffield, United Kingdom
huangxg@cqupt.edu.cn

***Abstract*—Federated learning implements decentralized machine learning tasks without exposing users' private data. However, in practical scenarios, intelligent devices data pertain to different fields are non-independent and identically distributed (non-IID), which leads to a decrease in the accuracy of the global model. In addition, if there are untrusted devices participated in federated learning, the global model accuracy will be decreased. To address the above-mentioned issues, in this paper, we propose a blockchain-enabled hierarchic cluster-based federated learning in edge computing framework to improve the accuracy of the global model and ensure the local model credibility. Firstly, we propose the hierarchic cluster-based federated learning (HCFL) algorithm, which realizes hierarchically aggregation based on user cosine similarity to improve global model accuracy. Moreover, blockchain technology is enabled in the proposed HCFL algorithm to verify the local model gradient from IDs before global aggregation. Moreover, incentive mechanism is proposed to dynamically adjust reward of IDs for promote IDs train trusted models. Finally, simulation results demonstrate the efficiency and performance of the blockchain-enabled hierarchic cluster-based federated learning framework.**

***Index Terms*—Federated Learning, Blockchain, Edge Computing Network, Hierarchic Cluster-based**

I. Introduction

Recently, artificial intelligence has played an important role in many burgeoning applications, such as voice recognition, image classification and automatic driving, etc. With massive amount of data,the centralized-training neural networks could obtain high fitness models, but also cause data security issue. Federated learning (FL) was proposed in 2016 to solve the problem of user privacy protection, which has attracted great attention in both academic and industry [1]. In addition, FL remained the problem of non-independent identically distributed (Non-IID) of data from the different intelligent devices (IDs) [2]. Sattler et al. pointed out that Non-IID data would reduce the convergence rate in FL [3]. Zhao tested the impact of IID data and non-IID data on FL performance based on different data sets and found that non-IID data would lead to a significant decline in model accuracy [4]. Since model gradients are generated based on training local data, which reflect the different degrees of original data. It is difficult to evaluate the difference between model gradients with Euclidean distance. Huang et al. proposed to evaluate the distributional difference of model gradients based on cosine similarity [5].

Furthermore, in FL combined with edge computing network architecture, the storage and computation of the global model are completely dependent on the central server and edge server, which are vulnerable to single points of failure (SPOF) or targeted attacks. Due to its decentralized characteristics, blockchain is highly consistent with FL. After Kim et al. [6] first proposed the combination of blockchain and FL in 2018, many pieces of research have been carried out. [7] proposed that FL training data could be stored by blockchain, which can resist unauthorized access and malicious node attacks by using the immutable characteristics and key verification mechanism of blockchain. In addition, in [8], the authors discussed a FL-chain model where blockchain provided a secure way to exchange the model parameters of FL and audited changes to the global model. However, the consensus mechanism in blockchain reduces network throughput, which could be solved by optimizing network sharding [9].

In this paper, we propose the blockchain-enabled hierarchic cluster-based federated learning (BHCFL) algorithm to improve the global model accuracy while ensure trusted model update. The main contributions of this paper are outlined as follows

- Firstly, a blockchain-enabled edge computing model is proposed for federated learning, which consists of central cloud layer, consensus layer, edge layer and ID layer.
- Secondly, to improve the global model accuracy under non-IID data, we propose the hierarchic cluster-based federated learning (HCFL) algorithm, which divides IDs into different clusters and hierarchical aggregation.
- Thirdly, to delete untrusted IDs which will decrease the global model accuracy, a blockchain-enabled hierarchic cluster-based federated learning (BHCFL) algorithm is proposed, which will verify the local model gradient from IDs before global aggregation.
- Finally, the experiment results are provided to evaluate the performance of the BHCFL algorithm. WeBase platform is used to verify the local model gradient. FEMNIST dataset is implemented to validate the effectiveness of the BHCFL algorithm.

The rest of the paper is organized as follows. The system

This work is supported by the National Natural Science Foundation of China (NSFC) (61831002,61401053), the European Union Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 734798, and Innovation Project of the Common Key Technology of Chongqing Science and Technology Industry (Grant no. cstc2018jcyjAX0383)

978-1-6654-5084-3/22/$31.00 ©2022 IEEE

model is presented in Section II. In Section III, we propose the HCFL algorithm to improve global model accuracy with Non-IID data. In Section IV, the BHCFL algorithm is introduced to ensure the trusted model update. Simulation results are discussed in section V. Section VI draws the conclusion.

II. System Model

The blockchain-enabled edge computing network is shown in Fig. 1, which consists of the central cloud layer, consensus layer, edge layer, and ID layer. In addition, blockchain technology is enabled in the edge-cloud collaboration network to ensure reliability of IDs and intelligent services. The details of each layer are given as the following.

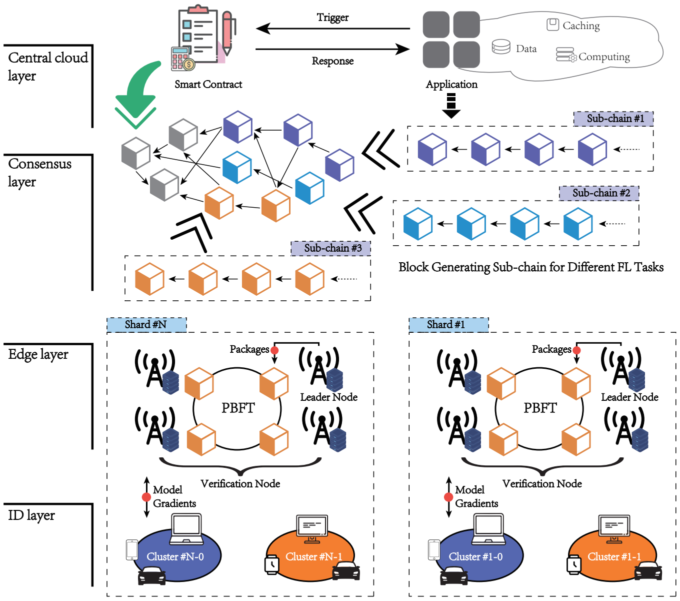


Fig. 1. Blockchain-enabled Edge Computing Network Model

Central cloud layer: It contains one central cloud, which coordinates FL task delivery, generate smart contract, achievement verification, protocol formulation, etc.

Consensus layer: It contains two layers, namely, main-chain layer and sub-chain layer. The main-chain layer is used to publish FL tasks by the cloud. The sub-chain layer consists of multiple shards which independently accomplish FL tasks. Each shard includes a leader node and lots of verification nodes, which adopts the Practical Byzantine Fault Tolerance(PBFT) consensus and updates the local model gradient information. The leader node aggregates the global model from the local model gradient of IDs. The verification nodes verify the local model gradients and storage the trusted model gradients to blockchain. After global aggregation, verification nodes will allocate reward to the trusted IDs which provide trusted model gradient.

Edge layer: It is composed of M edge servers, denoted as $\mathcal{M} = \{1, 2, \cdots, m, \cdots, M\}$, which provides data transmission, computing and storage services for the consensus layer and the ID layer. Edge servers collect the local model gradient of IDs for global aggregation, block verification and block storage.

ID layer: It contains I IDs with a certain calculation and storage capabilities, denoted as $\mathcal{I} = \{1, 2, \cdots, i, \cdots, I\}$. Since IDs pertain to different industries and institutions, their data are non-IID. IDs execute local training and transmit the local model gradient to the edge layer.

III. Hierarchic Cluster-based Federated Learning Algorithm

To handle the issue of model accuracy decrease caused by training non-IID data. In this section, we divide IDs into different clusters according to their cosine similarity. After that, the HCFL algorithm is introduced to increase the global model accuracy with the non-IID data of IDs.

A. Local Model Gradient Update

Firstly, IDs execute local training based on their data, and update the local model parameter to minimize the loss function. Denote $f_w : \mathcal{X} \to \mathcal{Y}$ and $L(f_w(x), y)$ as model function and the loss function respectively. Therefore, there exists a model parameter that could minimize the average loss function when participants have IID data. Specifically, for $\mathrm{ID_i}$, given data set $(x, y) \in \mathcal{D}_i$ and model f_{w_i} with $\mathrm{ID_i}$, the optimal model parameter can be obtained by

$$w_i = \arg\min L(f_{w_i}; \mathcal{D}_i) \tag{1}$$

$$w = \arg\min \frac{1}{M} \sum_{i=1}^{M} L(f_{w_i}; \mathcal{D}_i) \tag{2}$$

where, w is the global model parameter, w_i is the local model parameter of $\mathrm{ID_i}$. However, the above conclusion is invalid when the ID data is Non-IID. To protect user privacy, IDs only upload the local model gradient to Edges. Model gradients can be obtained by the derivative of the model function with respect to model parameters. In detail, in the t-th local training round, $\mathrm{ID_i}$ adopts the stochastic gradient descent (SGD) algorithm to calculate the small batch samples of local data $\mathcal{D}_i$ to generate a local model gradient, which can be expressed as

$$\Delta w_i^{t+1} = -\eta \nabla_w L(w_i^t, \mathcal{D}_i) \tag{3}$$

where, η is the learning rate. Since the uploaded local model gradient is a conditional probability distribution mapping of user data, the consistency of data distribution with different IDs can be obtained. Thus, the local model gradient indicates the distribution of ID data.

B. Cluster Partitioning Condition

To ensure the high accuracy of the cluster models, which improve the accuracy of the global model, the IDs with closer local model gradient distribution are divided into a cluster, denoted as $\mathcal{C} = \{c_1, \cdots, c_k, \cdots, c_K\}$, and $\bigcup_{k=1}^{K} c_k = \{1, \cdots, i, \cdots, I\}$. The cosine similarity between two vectors is evaluated by the cosine of the angle between them. The local model gradient is the vector with geometric characteristics, the cosine similarity between model gradients can be calculated by

$$\alpha_{i,j} = \frac{\langle \Delta w_i, \Delta w_j \rangle}{\|\Delta w_i\| \, \|\Delta w_j\|} \tag{4}$$

The larger the value of cosine similarity means the more similar distribution of the two model gradients. Hierarchical clustering analysis is adopted to optimize the cluster process [10]. In order to reduce the amount of calculation, we only divide the cluster with Non-IID data. If the data distribution of IDs in a cluster is inconsistent, the convergence solution ($\Delta w_i^* \to 0$) of $\mathrm{ID_i}$ is different from the target convergence solution ($\Delta w^* \to 0$) of the cluster. Therefore, a cluster will be split when the following two conditions are satisfied

- The cluster c model verges on a convergence value

$$0 \leq \left\| \sum_{i=1}^{I} \frac{D_i}{D_c} \Delta w_i^* \right\| \leq \varepsilon_1 \tag{5}$$

- The local model of $\mathrm{ID_i}$ has not reach the convergence value

$$\max_{i=1,\cdots,I} \|\Delta w_i^*\| > \varepsilon_2 \tag{6}$$

where, D_i and D_c denote the datasets sizes of $\mathrm{ID_i}$ and IDs in cluster c, ε_1 and ε_2 are condition factors. To improve the accuracy of the global model, ε_1 should be set as small as possible within the runtime constraint. According to simulation results, we set $\varepsilon_1 \approx \max ||\Delta w_c^t||/10$. The value of ε_2 is based on the number of ID and the distribution of ID data, which is set as $\varepsilon_2 \in [2\varepsilon_1, 10\varepsilon_1]$ based on the simulation results.

C. Hierarchic IDs Clustering

To improve the global model accuracy, IDs are divided into different clusters and the global model is aggregated from cluster models, which mainly includes the following three steps

1) Initialize the cluster, all IDs are in one cluster, denoted as $\mathcal{C} = \{\{1, \cdots, i, \cdots, I\}\}$. Calculate cosine similarity matrix A_α of all IDs based on their local model gradients.
2) Divide IDs into two clusters $\mathcal{C}_1, \mathcal{C}_2$ based on their cosine similarity A_α from $\mathcal{C}$. The cosine similarity between IDs in a clusters should be greater than the cosine similarity between $\mathcal{C}_1$ and $\mathcal{C}_2$, that is $\alpha_{itr} > \alpha_{crs}$, then $\alpha_{itr}^{\min} > \alpha_{crs}^{\max}$. $\alpha_{crs}^{\max}$ and $\alpha_{itr}^{\min}$ are obtained by the formula (7), respectively.

$$\begin{aligned} \alpha_{crs}^{\max} &= \max_{i \in \mathcal{C}_1^*, j \in \mathcal{C}_2^*} \alpha(\Delta w_i^*, \Delta w_j^*) \\ \alpha_{itr}^{\min} &= \min_{i,j \in \mathcal{C}_k^*, k \in \{1,2\}} \alpha(\Delta w_i^*, \Delta w_j^*) \end{aligned} \tag{7}$$

3) Subsequently, split $\mathcal{C}_1$ and $\mathcal{C}_2$ until the condition (5) is satisfied and the condition (6) is not satisfied. This demonstrates that the optimal value w^* is achieved.

In conclusion, the HCFL algorithm is proposed. It is noticeable that if the model accuracy deteriorates after multiple clustering, the rollback operation is adopted to ensure the stability of the HCFL algorithm.

The HCFL algorithm can obtain the optimal clustering with high global model accuracy based on the distribution of ID data. Finally, a binary tree structure of the global model is generated according to the HCFL algorithm, where the global model is aggregated by all leaf node cluster models.

IV. Blockchain Enable Trusted Model Update

In this section, we introduce the model gradient verification and the incentive mechanisms based on blockchain technology to prevent untrusted IDs from reducing the global model accuracy.

A. Local Model Gradient Verification

To ensure the security and the efficiency of the uploaded local model gradient, the key verification mechanism and homomorphic encryption algorithm in blockchain are used. Model gradient verification is based on the Multi-Krum algorithm [11]. Assume the number of local model gradients in the transaction pool is I(i.e. all IDs will be verified) and the number of untrusted IDs is I_F.

In PBFT, the maximum value of I_F should satisfy $I \geq 3I_F + 1$. The set of trusted IDs is denoted as $\mathcal{A}$, and the set of untrusted IDs is denoted as $\mathcal{B}$. Trusted IDs will upload the trusted gradient to ensure the convergence of the global model, and the untrusted IDs will upload the untrusted gradient to damage the global model. Therefore, the model gradient between $\mathcal{A}$ and $\mathcal{B}$ satisfies the following formula

$$\left\|\Delta w_i^T - \Delta_k^T\right\|_{i,k \in \mathcal{A}} < \left\|\Delta w_i^T - \Delta_j^T\right\|_{i \in \mathcal{A}, j \in \mathcal{B}} \tag{8}$$

Accordingly, model gradient verification mainly includes the following three steps

1) Step 1: for $\mathrm{ID_i}$, calculate the sum of Euclidean distance of its nearest k local model gradients as the risk score, which can be obtained as follows

$$s_i^T = \sum_{k: i \to j} \left\|\Delta w_i^T - \Delta w_j^T\right\| \tag{9}$$

where T indicates the T-th global epoch and $k : i \to j$ represents that Δw_j^T is one of k gradients closest to Δw_i^T.
2) Step 2: Sort the risk scores of IDs. If the number of IDs with the same risk scores is greater than k, turn to Step 3. Otherwise, the lowest risk score in the $k+1$ model gradient will be retained, and the untrusted model gradients will be deleted.
3) Step 3: Set $k = k + 1$ and return to step 1. If $k = I - I_F - 2$, the $I - I_F - 1$ model gradient with the lowest risk score is regarded as the trusted update.

B. Incentive Mechanism

FL result in resource consumption from the participant. Thus, to economize consumption costs, the untrusted IDs will upload local model gradients with low accuracy(i.e., untrusted model). In order to prompt more IDs active in FL, the incentive mechanism is proposed, which will allocation rewards to IDs according to the three indicators of IDs.

Define the data quality θ_i of $\mathrm{ID_i}$ as

$$\kappa_i = \frac{\tau_i}{\psi} \tag{10}$$

$$\theta_i = \frac{e^{\kappa_i}}{\sum_{j=1}^{I} e^{\kappa_j}} \tag{11}$$

where ψ and τ_i denote the local training time required to achieve the global model accuracy based on the standard dataset and $\mathrm{ID_i}$ dataset, respectively. In addition, denote the computing frequency of $\mathrm{ID_i}$ as f_i, the total reward as R and the expected reward of $\mathrm{ID_i}$ is given by

$$R_i = R \cdot \theta_i \cdot \frac{f_i}{\sum_{j=1}^{I} f_j} \tag{12}$$

The reward of $\mathrm{ID_i}$ is adjusted dynamically with the model gradient verification results. Define τ as the number of global epoch to adjust the reward. If $\mathrm{ID_i}$ continuously uploads the untrusted model gradient within τ global epoch, its reward will be reduced. Otherwise, the reward will be increased. The reward of $\mathrm{ID_i}$ is adjusted by

$$R_i^{T+1} = \begin{cases} R_i^T - \delta, & i \in \mathcal{H}^T \cap \mathcal{H}^{T-1} \cdots \cap \mathcal{H}^{T-\tau} \\ R_i^T + \delta, & i \in \mathcal{G}^T \cap \mathcal{G}^{T-1} \cdots \cap \mathcal{G}^{T-\tau} \\ R_i^T, & i \in other \end{cases} \tag{13}$$

where $\mathcal{H}^{T+\tau}$ and $\mathcal{G}^{T+\tau}$ denote the set of IDs with the highest risk score and the lowest risk score in the $(T+\tau)$-th global epoch, respectively. δ is the incentive factor.

To sum up, the BHCFL algorithm is proposed and shown in Algorithm 1. Where $2-5$ lines are the local training process, $6-12$ lines are the model gradient verification process and $13-26$ lines are the HCFL process. The complexity of the BHCFL algorithm is $\mathcal{O}(I^3)$.

Algorithm 1 BHCFL Algorithm

```
Input: Initial global model $w_0$, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, local epoch $n$ during a global epoch
Output: Model of all IDs $w_1^*, \cdots, w_i^*, \cdots, w_I^*$
 1: repeat
 2:   for $i = 1 : I$ do
 3:     $w_i \leftarrow w_i + \Delta w_{c(i)}$
 4:     Local training $n$ epoch $\Delta w_i \leftarrow -\eta \nabla_w L(w_i, \mathcal{D}_i)$
 5:   end for
 6:   for $i = 1 : I$ do
 7:     while $k < I - I_F - 2$ do
 8:       $s_i^T = \sum_{k:i \to j} \left\| \Delta w_i^T - \Delta w_j^T \right\|$
 9:     end while
10:     According (10), (11) and (12) obtain the expected reward $R_i$ for $ID_i$
11:     Update reward by (13)
12:   end for
13:   $\mathcal{C}_{tmp} \leftarrow \mathcal{C}$
14:   for $c \in \mathcal{C}$ do
15:     $\Delta w_c \leftarrow \frac{1}{\|c\|} \sum_{i \in c} \Delta w_i$
16:     if $\|\Delta w_c\| < \varepsilon_1$ and $\max_{i \in c} \|\Delta w_i\| > \varepsilon_2$ then
17:       $\alpha_{i,j} = \frac{\langle \Delta w_i, \Delta w_j \rangle}{\|\Delta w_i\| \|\Delta w_j\|}$
18:       $c_1, c_2 \leftarrow \arg\min_{c_1 \cup c_2 = c} (\max_{i \in c_1, j \in c_2} \alpha_{i,j})$
19:       $\alpha_{cross}^{max} \leftarrow \max_{i \in c_1^*, j \in c_2^*} \alpha_{i,j}$
20:       $\alpha_{intra}^{min} \leftarrow \min_{i,j \in c_k^*, k \in \{1,2\}} \alpha_{i,j}$
21:       if $\alpha_{intra}^{min} > \alpha_{cross}^{max}$ then
22:         $\mathcal{C}_{tmp} \leftarrow (\mathcal{C}_{tmp} \backslash c) \cup c_1 \cup c_2$
23:       end if
24:     end if
25:   end for
26:   $\mathcal{C} \leftarrow \mathcal{C}_{tmp}$
27: until converged
```

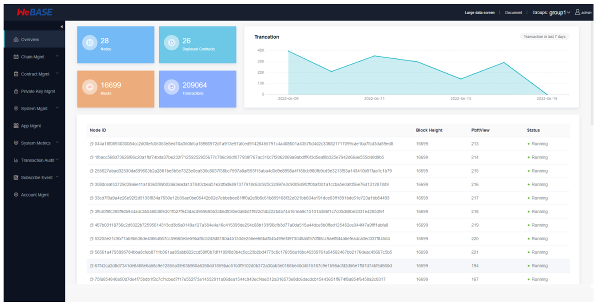
V. Simulation Results

In this section, we evaluate the numerical performances of the proposed BHCFL algorithm in various aspects through simulations.

A. Simulation Parameters

In the simulation, the FEMNIST dataset is randomly assigned to 200 IDs. Rotate part of the image to generate non-IID dataset. Fig. 2, we verify the model gradient based on the blockchain network provided by the WeBase platform, which can manage consensus nodes, view consensus status, deploy smart contracts, and audit transactions.

In addition, a three-layer standard convolutional neural network in Tensorflow is used in the model training process. Furthermore, 80% and 20% of the local data are used to train the model and test the model, respectively. Assume there are 40 edge servers in the edge computing network.



Node ID	Block Height	PbftView	Status
044a18f0893030084cc2d00efc05353e8ed10a350bfca18966972d1a913e97afced91426455791c4a408601a43576d4d2c336821717099cae1ba7fcd3da89ed8	16699	213	● Running
1fbacc568d73626f66c20a1fbf74bda37be232f71259252955677c786c90df077938f767ac310c7f2062069a9abdfff6f3d5eaf8b325e7942d66ae555d40d9b5	16699	214	● Running
255627abad32530da699663b2a2881be5b5e7323e0ea530c8037f38bc7597a8af593f15aba4d3d9e8998a4f188cb980fb9cd9e3213f93a143410b97faa1c1b79	16699	215	● Running
308dceaf43729c09a6e11a18365f89b02a63eada137840cbea51e2dfad6897377918c63c925c2c997e3c900949fcff0baf001a1ccba5e0afd56e7641312878d9	16699	216	● Running
33cd7f3a9a4e26e92f2d5135f834a7650e12b35ae0be05442b02e7ebbebee81fff0a2e9b8c61b859168f32e021bb604a191dce63ff1891bdc61e723a1bb84493	16699	217	● Running
3f640f9fc395f9db944a4c3b546838fe301fb27f943dacd959695b33b6dfc95e0a8bd1f922cf4b222bda74a161ea9c15101a386f1c7c00d68be3331e42853fef	16699	218	● Running
467b03118736c2d5022b72595614313cd3b5a0149a127a384e4a16c415395de254c68b133f96cfb3d77a0da515a44dce5b0ffed125482ce344f47a9fff1abfa8	16699	219	● Running
53255e21c9b77ab9b636de40664667cc59660e5e59baf8c52d8d8180a4b153de256ee66baf54b499efd973048a5f57df88cc9aef8d4a6efea4ca0ec037f845d4	16699	220	● Running
56561a4793995784bba8cfeb8711b361aa85abb822ccd59ff0b7df1199f8d3b4c5cc23b2bd4773c8c17635da18bc46339761a54582467bb2176deac45667c5b5	16699	221	● Running
67f42ca2d8d7341de8468e6a58c9e12855a3fe63b960a5268dd1859bac51b3f9102d0b372a30ab3e0168be40d4515167c9e1b98acfd2d56e1ffd7d74bf5d660d	16699	194	● Running
75fa854840a50d7de4f75bdb1f2c7cf1cbed7f17e552f73a14552911a06dea1544c943ecf4ae512a316573e8dc6dacdcb15443651ff674f8a854f6458a2c8317	16699	167	● Running

Fig. 2. WeBase Platform

B. Evaluation Results and Analysis

Fig. 3 and Fig. 4 show the relationship between the accuracy and number of the global epoch for different algorithms. One global epoch indicates that the IDs has completed 10 rounds of local training and the edges has completed one global aggregation. Fig. 3 shows that the BHCFL algorithm could achieve higher accuracy by clustering, and the deviation between the local model gradient and the global model gradient is reduced.

In Fig. 4, we illustrate the performance of the FedAvg al-

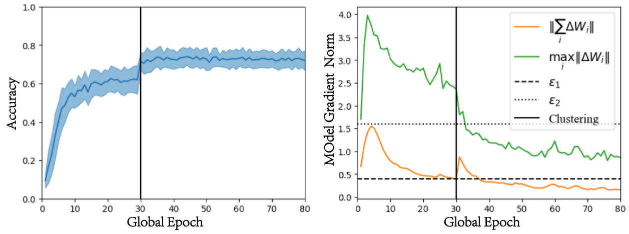


Fig. 3. Performance of the BHCFL algorithm

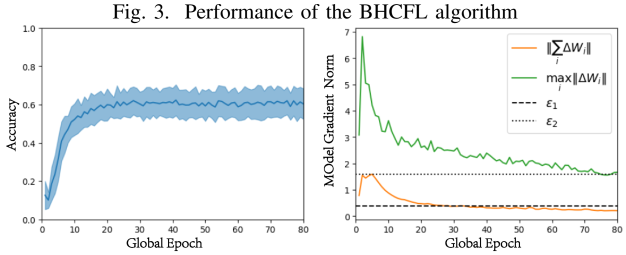


Fig. 4. Performance of the FedAvg algorithm

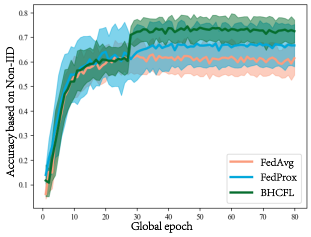


Fig. 5. Accuracy versus number of global epoch for different algorithms without untrusted IDs

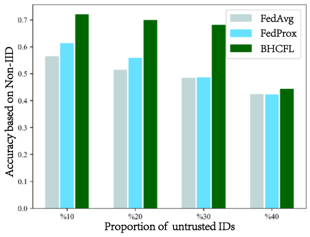


Fig. 6. Accuracy versus proportion of untrusted IDs for different algorithms

gorithm, which converges in the 30th epoch, and the accuracy of both the FedAvg algorithm and the BHCFL algorithm is 61.2%. When the global model gradient norm is less than or equal to ε_1 and the local model gradient norm is greater than ε_2. Based on the cosine similarity matrix, IDs in the cluster are further divided into several clusters. In Fig. 3, the simulation results imply that the accuracy increases from 61.2% to 72.6% based on the new clustering. Meanwhile, compare with the FedAvg algorithm, the difference between the global model gradient and the local model gradient of the BHCFL algorithm is smaller.

Fig. 5 shows the accuracy of the FedAvg, FedProx and BHCFL algorithms without untrusted IDs. In the 80th global epoch, the accuracy from low to high is FedAvg, FedProx and BHCFL respectively. In the first 30 epochs, the performance of the three algorithms is similar. Then, the FedAvg algorithm converges at the 25th epoch and its accuracy is maintained 62%. However, the accuracy of the FedProx and the BHCFL algorithm increases with the increase of epoch times, reaching 66% and 72% at the 37th and 31th epoch, respectively. In addition, the variance of global model accuracy of the BHCFL algorithm is less than the FedProx algorithm. That is because the BHCFL algorithm will divide IDs into different clusters based on the distribution of data to obtain a high accuracy global model.

The accuracy of the three algorithms with untrusted IDs is shown in Fig. 6. With the increasing number of untrusted IDs, the accuracy of the global model by the FedAvg and the FedProx algorithm decreases sharply. The BHCFL algorithm can effectively reduce the influence of untrusted IDs on the global model through model gradient verification. However, when the proportion of untrusted IDs exceeds 30%, the accuracy of the BHCFL algorithm will decrease rapidly. This is due to the PBFT consensus is only efficient when the number of the untrusted IDs is less than $\frac{1}{3}$.

VI. Conclusion

In this work, we have studied the blockchain-enabled edge computing framework for hierarchic cluster-based federated learning. First, by multi-ply dividing IDs into different clusters according to the cosine similarity of their model gradient, the global model is generated by cluster models, which are leaf nodes of the global model binary tree. Then, to ensure the trusted model before the global model aggregate, blockchain technology is used to verify the local model gradients of IDs. Meanwhile, the incentive mechanism has proposed to dynamically adjust IDs reward, which promoted IDs train trusted models. Finally, simulation results have been provided, which show that the performance of the blockchain-enabled hierarchic cluster-based federated learning framework is increased under non-IID data and untrusted IDs.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] D. Prestiadi, Maisyaroh, I. Arifin, and A. N. Bhayangkara, "Meta-analysis of online learning implementation in learning effectiveness," in *2020 6th International Conference on Education and Technology (ICET)*, 2020, pp. 109–114.

[3] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[4] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[5] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data." in *AAAI*, 2021, pp. 7865–7873.

[6] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "On-device federated learning via blockchain and its latency analysis," *arXiv preprint arXiv:1808.03949*, 2018.

[7] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for ai: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10 127–10 149, 2019.

[8] D. N. Dillenberger, P. Novotny, Q. Zhang, P. Jayachandran, H. Gupta, S. Hans, D. Verma, S. Chakraborty, J. Thomas, M. Walli *et al.*, "Blockchain analytics and artificial intelligence," *IBM Journal of Research and Development*, vol. 63, no. 2/3, pp. 5–1, 2019.

[9] X. Huang, Y. Wang, Q. Chen, and J. Zhang, "Security analyze with malicious nodes in sharding blockchain based fog computing networks," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021, pp. 1–5.

[10] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[11] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.