Hindawi
Security and Communication Networks
Volume 2022, Article ID 9428610, 8 pages
https://doi.org/10.1155/2022/9428610

Research Article

Privacy-Preserving Collaborative Computation for Human Activity Recognition

Lin Wang,[1,2] Chuan Zhao,[1,2,3] Kun Zhao,[4] Bo Zhang,[1,2] Shan Jing,[1,2] Zhenxiang Chen,[1,2] and Kuiheng Sun[1,2]

[1]*School of Information Science and Engineering, University of Jinan, Jinan 250022, China*
[2]*Shandong Provincial Key Laboratory of Network-Based Intelligent Computing, University of Jinan, Jinan 250022, China*
[3]*Shandong Provincial Key Laboratory of Software Engineering, Jinan, China*
[4]*Inspur Electronic Information Industry Co. Ltd., Beijing, China*

Correspondence should be addressed to Chuan Zhao; ise_zhaoc@ujn.edu.cn and Kun Zhao; zhaokunbj@inspur.com

Received 17 November 2021; Accepted 28 January 2022; Published 28 February 2022

Academic Editor: Yuling Chen

Human Activity Recognition (HAR) enables computer systems to assist users with their tasks and improve their quality of life in rehabilitation, daily life tracking, fitness, and cognitive disorder therapy. It is a hot topic in the field of machine learning, and HAR is gaining more attention among researchers due to its unique societal and economic advantages. This paper focuses on a collaborative computation scenario where a group of participants will securely and collaboratively train an accurate HAR model. The training process requires collecting a massive number of personal activity features and labels, which raises privacy problems. We decentralize the training process locally to each client in order to ensure the privacy of training data. Furthermore, we use an advanced secure aggregation algorithm to ensure that malicious participants cannot extract private information from the updated parameters even during the aggregation phase. Edge computing nodes have been introduced into our system to address the problem of data generation devices' insufficient computing power. We replace the traditional central server with smart contract to make the system more robust and secure. We achieve the verifiability of the packaged nodes using the publicly auditability feature of blockchain. According to the experimental data, the accuracy of the HAR model trained by our proposed framework reaches 93.24%, which meets the applicability requirements. The use of secure multiparty computation techniques unavoidably increases training time, and experimental results show that a round of iterations takes 36.4 seconds to execute, which is still acceptable.

1. Introduction

Human Activity Recognition (HAR) is a machine learning task to identify human activities through images, videos, or sensor data generated by smart wearable devices. HAR has a wide range of applications today, such as monitoring the health of individuals by recognizing their activities, or it can be used in public places such as plazas and train stations to identify unusual acts of terror in order to give an advance warning [1].

However, when using this type of data, companies or data owners often face a number of issues:

(1)Data owners are reluctant to reveal their information, whether directly or by computational inference

(2) The massive amount of data generated by IoT devices poses a huge challenge to the storage and processing capacity of central servers

(3) The network bandwidth cannot handle such an order of magnitude of data transfer

(4) The node performing the computation can be hijacked or corrupted by adversaries to perform incorrect computation operations

To address the above-mentioned issues, researchers have conducted many explorations. In order to protect privacy and save bandwidth, federated learning is proposed [2]. Federated learning enables us to keep model training procedure on local devices without transmitting data to central

server. At present, federated learning has a mature application in the industrial field [3]. However, with in-depth research, it is found that there are still some problems in federated learning. For example, although federated learning only transmits model update parameters, the updated parameters will still disclose sensitive information to the third party or the central server [4, 5]. Commonly used methods, including secure multiparty computation and differential privacy, aim to resist privacy disclosure in the learning process[4, 6]. However, these approaches are often accompanied by a loss of model efficiency or an increase in training time. Blockchain also has many studies that combine it with federated learning due to its decentralized nature. Kumar et al. use blockchain to first validate the data and then use federated learning to train a deep learning model globally to improve recognition rates against CT images of COVID-19 patients [7]. Qi et al. use a blockchain-based federated learning framework for predicting traffic flow, the model will be verified by miners, the noise will be added to the model to enhance privacy safeguards, and the scheme can effectively prevent poisoning attacks, but there will be some sacrifice in model effectiveness [8]. Edge computing is also commonly used in cutting-edge research in machine learning, where the use of edge nodes to offload computational and storage tasks from a central server can effectively improve training efficiency. Khelifi et al. explored the applicability of deep learning models (i.e., convolutional neural networks, recurrent neural networks, and augmented learning) with IoT devices. The study sought to assess the future trends of deep learning plus edge computing in the future. The study points out that convolutional neural models can be used in the IoT domain and that reliable machine learning models can be trained even with data from complex environments [9]. Secure multiparty computing often plays an important role in this as well. Sangaiah et al. proposed an approach using edge computing plus machine learning to protect the confidentiality of certain location-based services. The approach uses Hidden Markov Models by combining decision trees and k-means algorithms. The benefits of the mobile edge service strategy are location confidentiality and low latency. Both network and computing services are located near the user as a way to achieve lower latency [10, 11].

In this paper, we adopt the idea of federated learning, where users train models locally and optimize the model jointly by uploading parameters instead of uploading data to a central server [2]. We also use a secure aggregation algorithm to eliminate the possibility of the server inferring information via gradients [12, 13]. Furthermore, we consider edge computing and blockchain in our framework. To be specific, we replace the traditional central server with a blockchain. The properties of blockchain make our proposed framework possess a series of security features such as transparency, auditable, and tamper-proof [14]. Edge nodes are introduced in our scheme to relieve the computational pressure and bandwidth pressure on the system [15]. The specific framework structure and the implementation will be presented in Section 3.

In general, our contributions can be summarized as follows:

(1) We consider federated learning and edge computing scenario to keep private data local instead of being uploaded to the central server, which helps to protect users' privacy. By doing so, we also achieved alleviating the load of the central server, making the computation tasks be processed faster.

(2) We implement an advanced secure aggregation algorithm that aggregates exactly the results we want, the same as the computed result under plaintext. Also, the whole aggregation and transmission process is in the form of shares, which ensures that the adversary cannot steal information by observing these intermediate shares.

(3) We deploy smart contracts to replace the traditional central server, avoiding the occurrence of a single point failure of the central server. Moreover, the public auditability feature of blockchain also allows other nodes to verify the aggregation results, thus preventing dishonest behaviors of aggregation nodes.

2. Preliminaries

In this section, we briefly introduce basic tools and corresponding techniques needed in this paper.

2.1. Edge Computing. Edge computing is a distributed computing architecture that refers to distributing computation and storage tasks to edge nodes that are logically closer to users and data sources for processing. This architecture can effectively reduce network latency caused by data transmission, significantly improve the response time of network services, and enhance data security for a better user experience.

In the 1990s, to improve network quality, a research group at MIT proposed CDN (Content Delivery Network) to enable network sites close to users to acquire and cache network content and reduce the footprint on users' broadband. This architecture is widely used in various Internet scenarios [15]. On the other hand, cloud computing was created to cope with the increasing amount of data and computing. The rapid growth and evolution of cloud computing have led to dramatic changes in the way society works and business models [16], but along with development, cloud computing has also revealed many drawbacks. For example, the increasing volume of computation and data not only increases the computational burden on servers but also increases the bandwidth burden on cloud computing centers. This may prolong data processing time and reduce data processing speed and transmission speed. This is fatal for applications such as the Internet of Things, which has a huge amount of data and is latency-sensitive [17].

Edge computing can be seen as a combination of CDN and cloud computing. Due to the advancement of

technology, the performance of devices as edge nodes is also improving, which enables the tasks of edge nodes to be no longer limited to storage but also includes data processing and computing operations such as machine learning. With the development of IoT, edge computing is widely used to process IoT data, which makes edge computing technology have a broader development prospect. The usual architecture of edge computing is shown in Figure 1.

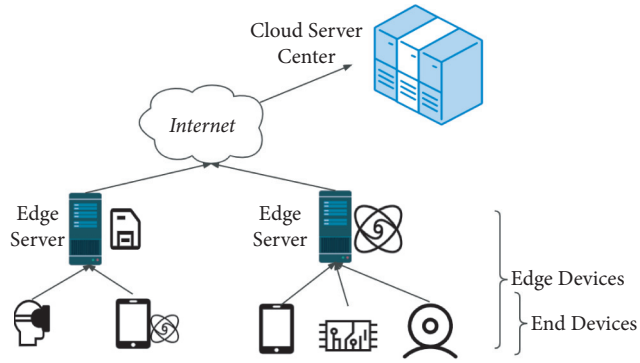

Figure 1: Edge computing.

2.2. Secret Sharing. Secret sharing refers to schemes for distributing a secret among a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number of shares are combined. Individual shares are of no use on their own. In this paper, we use Shamir's Secret Sharing, which is formulated by Adi Shamir [12]. Shamir's Secret Sharing is an ideal and perfect (t, n)-threshold scheme. In such a scheme, the aim is to divide a secret s into n pieces of data $s_1, \ldots, s_n$ (known as shares) in such a way that

(1) Knowledge of any t or more s_i pieces makes s easily computable. That is, the complete secret s can be reconstructed from any combination of t pieces of data

(2) Knowledge of any $t-1$ or fewer s_i pieces leaves s completely undetermined, in the sense that the possible values for s seem as likely as with knowledge of 0 pieces. The secret s cannot be reconstructed with fewer than t pieces.

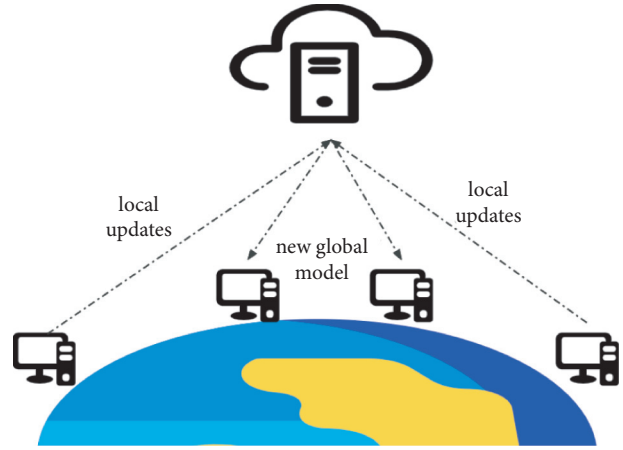

Figure 2: Federated learning.

2.3. Federated Learning. Federated learning is a distributed machine learning framework proposed by Google that allows multiple users to collaborate on training a global model while maintaining user's privacy, as shown in Figure 2[18]. In recent years, various countries have established legal restrictions to preserve the privacy of personal information, which makes direct access to user data for machine learning training impossible. Google proposes federated learning, which trains data locally rather than uploading data to a traditional central server to address this issue. This distributed architecture ensures the confidentiality of user data while also optimizing the usage of computing resources on local devices. The central server is only responsible for the coordination, which decreases the server's processing load. However, some issues must be addressed before federated learning can be used in practice, such as communication issues with a large number of participating devices and system compatibility issues caused by the diversity of participating devices [19]. However, with the increased emphasis on privacy protection, federated learning has become a very promising technology [20].

Definition 1 (federated learning).

Federated learning refers to training a global model using data stored in millions of remote devices, a task that can be represented by the following objective function:

$$\min F(w), \text{ where } F(w) \coloneqq \sum_{n=1}^{m} p_k F_k(w), \tag{1}$$

where m represents the total number of devices, F_k is the local objective function of the k-th device, p_k is defined as the influence weight of the corresponding device, p_k has the following properties, $p_k \geq 0$, and $\sum_{k=1}^{m} p_k = 1$.

2.4. Smart Contract. For the first time, Nick Szabo proposed the concept of smart contract in 1995 [21]. Smart contracts are a set of digital contracts that are automatically executed between committed parties. Smart contract is more secure and has lower transaction costs than regular contract. However, due to technological limitations, smart contract could not be executed until the practical implementation of blockchain technology. Blockchain is built on mutually trusted nodes, allowing us fairly and securely to run contracts. There are numerous stable and well-known applications, such as Ether on the public chain and Fabric and Quorum on the nonpublic chain. Smart contract can be developed to extend the functionalities of blockchain beyond digital currency such as Bitcoin, allowing it to be widely used in banking, copyright, and many other industries. However, there are still issues in smart contract that must be addressed, such as unusual programming languages and a lack of

debugging tools, which pose security concerns to smart contract [22].

3. Assumptions and Threats

Our framework is an open machine learning system that allows each node to join or depart at any time. We assume that all nodes want in order to collaborate to train a machine learning model but do not want their data to be utilized or observed by others.

3.1. Design Assumptions

3.1.1. System Topology. We assume that the edge nodes are smart hardware in the home with enough processing power for local training, such as smart gateways, smart routers, or personal computers. Each edge node connects all of the smart devices in the home (e.g., camera, smartphone, and smartwatch). We anticipate that no malicious attacks will be initiated among family members. Thus, we can transmit plaintext between smart devices and edge node without considering encryption. We assume that each edge node can communicate with a subset of other edge nodes, allowing messages to be broadcast from any edge node to all edge nodes.

3.1.2. Machine Learning. We assume that the Genesis Block propagates all training information to all edge nodes. The initial model, hyperparameters, optimization strategies, and learning objectives are all part of this. The edge nodes want to keep the local dataset private during the training phase. We use stochastic gradient descent (SGD) as the optimization algorithm in the local training phase. SGD is a universal optimization technique that may be used to train a wide range of models, including deep neural networks [23].

3.2. Threats. We analyzed possible threats during the training process as follows:

(1) Users' data can be maliciously analyzed and abused. We must prevent exposing users' plaintext data

(2) An adversary can deduce user information by seeing updates. User updates should not be directly observed

(3) Corrupted edge nodes may perform incorrect calculations and submit invalid global models

When data need to be stored on a cloud server, encryption of the data is often an option to prevent the cloud server from stealing the data. However, we assume that an edge node is only responsible for collecting and processing information from family members. Therefore, we do not consider encryption between smart devices and edge devices.

4. Framework Design

4.1. Framework Overview. Our proposed framework's main goals are as follows:

(1) Data owners collaborate to train an efficient Human Activity Recognition model

(2) Accelerate the training process by introducing edge computing architecture

(3) Prevent leakage of user information during the aggregation phase by using secure aggregation algorithms

(4) Use blockchain public verifiability and tamper-evident to oversee the behavior of packaged nodes

Each node on the blockchain network collects data generated by the smart devices. Each block includes the information generated after one iteration round. Figure 3 shows the process of one iteration.

(1) *Preparation*. Smart devices collect data on human behaviors using built-in sensors. When certain criteria are satisfied (power and network connection, no other tasks, and sufficient data), smart devices will transmit this data to the associated edge node. Before training begins, all edge nodes on this blockchain network receive an initial random global model from the Genesis Block, used for the first update. This process is shown in Steps 1, 2, and 3 of Figure 3.

(2) *Local Training*. A local model is calculated using the latest global model and local data. This is Step 4 of Figure 3.

(3) *Model Aggregation*. With the secret sharing algorithm, each node divides its update into n secret shares (n specifies the number of edge nodes in the distributed ledger) and distributes them to other nodes (Step 5). Step 6 requires all nodes to aggregate the shares they receive and then broadcast the results in Step 7. The first node that receives enough results will reconstruct the global model (Step 8).

(4) *Submit Block*. Finally, the first node to reconstruct the global model will combine the essential data into a new block and upload this block to the blockchain (Step 9).

4.2. Preparation for Training. Steps 1, 2, and 3 in Figure 3 represent the preparation phase. In this phase, we mainly focus on data collection, data transmission, and creating and distributing Genesis Block.

First, the smart device will collect information about people's activities. When certain conditions are met (e.g., the volume of data is sufficient; power and network are connected), the smart devices will send the data to the associated edge node.

The initial training information will be added to the Genesis Block. We anticipate that a trusted institution will generate the Genesis Block and broadcast it to all edge nodes to begin the training process. The trusted institution is only trusted at this phase, and it will not be involved in the following training process. The Genesis Block provides the model's initial state w_0 and the predicted number of iterations T. There are also public keys P_K for each user i used to generate the commitment to each node's update (detailed in Section 4.5).

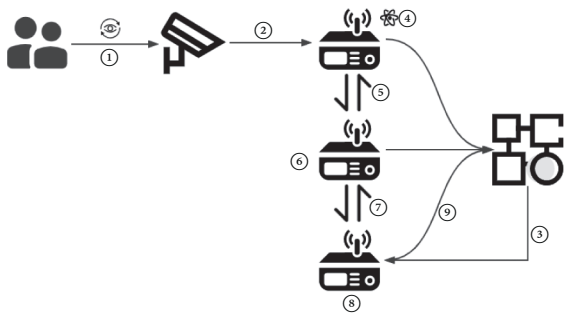

Figure 3: Overview.

4.3. Local Training. In the t – th iteration, the global model w_t is downloaded locally from the blockchain by each edge node. Each node has n_k samples, where k is the index of the node. n_k varies among nodes depending on the number of local smart devices and the amount of people's activities on that day. Each edge node computes a local gradient g_k on the current model w_t using its local data n_k. For a given learning rate ε, the local model w_{t+1}^k is given by

$$w_t - \varepsilon g_k \longrightarrow w_{t+1}^k. \tag{2}$$

The hyperparameters required for the computation process, such as the learning rate ε and the client training batch size B_t, are specified by the Genesis Block.

4.4. Aggregation Protocol. Edge nodes use a secret sharing approach to broadcast their local updates to other nodes in the blockchain network. For the following step of verification, they also broadcast the commitment $\mathrm{COMM}(\Delta w_i)_{\mathrm{sign}}$ of their update simultaneously, which carries their signature so that others cannot forge it. The entire aggregation process is described in the following.

The optimization algorithm in our proposed framework is stochastic gradient descent (SGD). Each node computes updates using the latest global model downloaded from blockchain and local data, and all updates are aggregated into a new global model. In the i – th iteration, the following equation is used to update the model parameter w:

$$w_{t+1} = w_t - \eta_t \left(\lambda w_t + \frac{1}{b} \sum_{(x_i, y_i) \in B_t} \Delta l\left(w_t, x_i, y_i\right) \right). \tag{3}$$

η_t is the learning rate, λ denotes the regularization parameter, which is used to prevent overfitting, B_t denotes the batch of one training sample of size b, and Δl denotes the gradient of the loss function.

The aggregation protocol requires all edge nodes to collaborate in order to aggregate their local updates into a new global model, and this protocol uses secret sharing to ensure that each node's private data and model updates cannot be seen or inferred by any node other than itself. Algorithm 1 shows the secure aggregation algorithm.

Assume that m edge nodes representing m families collaborate to train a model, and the update i.update for each node i will be encoded as a d-polynomial. This polynomial will be divided into n secret shares $(n = 2 * (d + 1))$. These n shares are distributed equally among all m nodes, and it takes $(d + 1)$ shares to reconstruct this model, indicating that at least $m/2$ nodes must collaborate to obtain the private data of a specific node. Each edge node i that accumulates enough shares (usually a minimum number u) aggregates those shares and then broadcasts the aggregation_result$[i]$ to all nodes once again. After receiving the aggregated $d + 1$ shares from at least half of the nodes, a node can reconstruct the sum of all local node updates $\sum_{i=1}^{u} \Delta w_j$. Eventually, the aggregated results of all nodes $\sum_{i=1}^{u} \Delta w_j$, the latest global model w_t, and all update commitments will be stored in a new block.

4.5. Block Structure. Each block contains a hash pointing to the previous block in order to link to it. Furthermore, malicious edge nodes may perform the aggregation process incorrectly to damage the model. Each block should include a new global model w_t as well as the aggregation results of all node updates $\sum_u \Delta w$ to validate the edge node aggregation procedure. This allows us to test whether the global model is correctly generated by

$$w_t = w_{t-1} + \frac{1}{u} * \left(\sum_u \Delta w \right). \tag{4}$$

Figure 4 shows the blockchain structure that we designed. To ensure that the aggregated results are generated from each node's local update, we keep each node's commitment to their submitted updates in the block as well. Then, the homomorphic nature of the commitment allows us to check if the edge node honestly aggregated the model [24].

$$\mathrm{COMM}\left(\sum \Delta w_i\right) = \prod_i \mathrm{COMM}\left(\Delta w_i\right). \tag{5}$$

5. Experiment

We use virtual machines to simulate PCs capable of collecting personal data from smartphones and training local models using the Long Short-Term Memory (LSTM) algorithm. We used three virtual machines for deep learning training, each with 4 GB of RAM and a GTX2080TI GPU. Figure 5 shows the training effect of our proposed framework compared to the training effect of the algorithm using differential privacy. Differential privacy is another prominent strategy in federated learning for protecting personal information. However, using differential privacy often results in decreased accuracy. The results show that our model meets the usability requirement and outperforms the model using differential privacy.

In addition, as shown in Figure 6, we tested the running duration of each component. Because the computation of the security aggregation algorithm is substantially more significant than that of plaintext, a cycle of iteration takes 36.4 seconds, with the process of secure aggregation counting for 72.26% of the overall time. However, this is still acceptable.

for each client$[i] \in m$ **do**
  d – polynomial$\Leftarrow i$.update
  n **shares** $\Leftarrow d$ – polynomial
  n **shares** are equally distributed among m nodes
end for
for each client$[i] \in m$ **do**
  if client[i] received u shares **then**
    aggregation_results · Share$[i] \Leftarrow$
      Aggregate(client_update[1].share$[i], \cdots$ client_update$[u]$share$[i]$)
    Broadcast the share of aggregation results
  end if
end for
for each client$[i] \in m$ **do**
  if client$[i]$ received $d+1$ shares of aggregation results **then**
    Reconstructing out aggregated results
end if
end for

Algorithm 1: Secure aggregation algorithm.

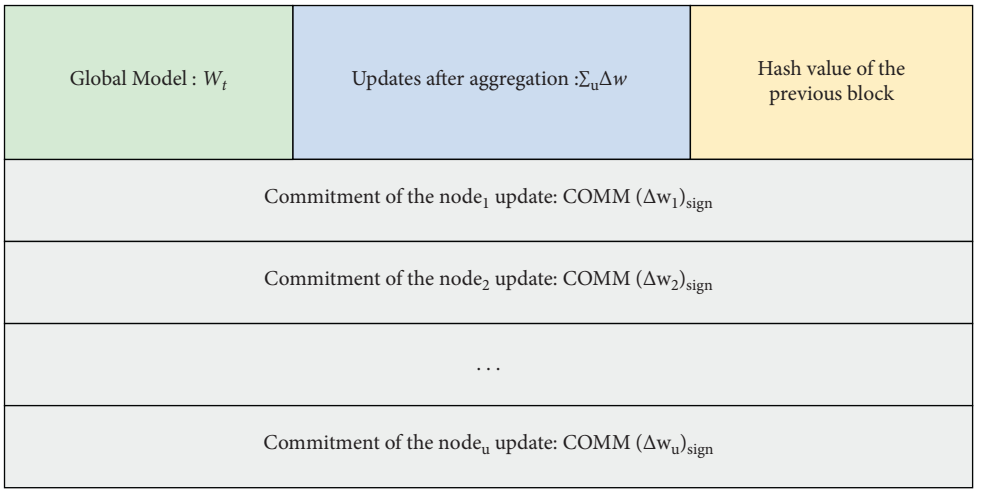

Figure 4: Block structure.

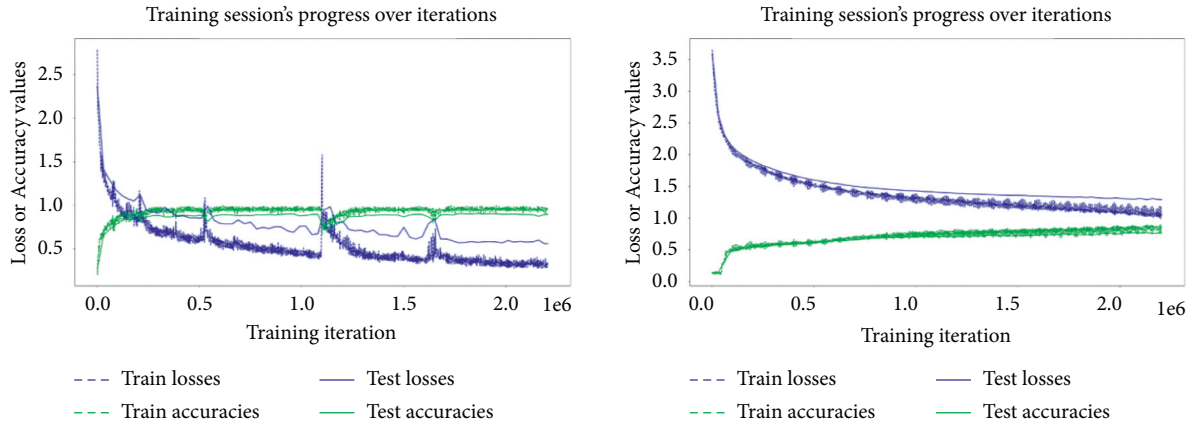

Figure 5: Comparison of the two algorithms.

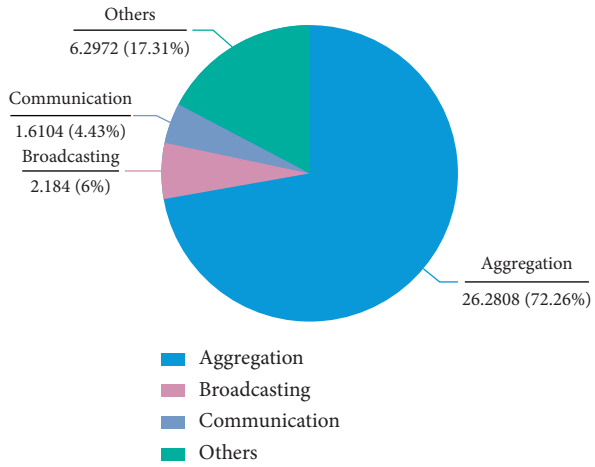

Figure 6: Running time of each part.

6. Conclusion

In this paper, we proposed a privacy-preserving collaborative machine learning framework. We combined edge computing architecture with distributed computing to ensure that data are kept local, which ensures that private data are not compromised. In addition, we used a secure aggregation algorithm to ensure that personal information does not leak even throughout the aggregation process. We tested this framework on the HAR dataset and compared the performance of our proposed framework to other popular methods. Our framework can be used for a wide range of different machine learning tasks that require privacy protection.

This framework can be improved in two ways in the future. Firstly, as the number of nodes in the network grows, the effectiveness of our consensus protocol rapidly decreases due to network fluctuations and differences in processing capacity across users. As a result, in the future, we will provide a new consensus mechanism based on consistency hash and proof of stake (PoS). It can also prevent malicious computing nodes from poisoning the model, enhance the efficiency of the consensus process, and reduce energy usage. On the other hand, the edge computing nodes considered in this paper will only cover smart wearable devices from the same family. The edge computing nodes will be home PCs or smart gateways. So, we will overlook data theft and data poisoning at this point. However, we are aware that data and model poisoning attacks can still be carried out between family members. Thus, we will strive to apply anomaly detection methods to identify poisoned data and models in future work.

Data Availability

The dataset can be found on the UCI Machine Learning Repository: Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21st European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61702218 and 61672262), Shandong Provincial Key Research and Development Project (nos. 2019GGX101028 and 2018CXGC0706), Shandong Provincial Natural Science Foundation (nos. ZR2021LZH007 and ZR2019LZH015), Shandong Province Higher Educational Science and Technology Program (no. J18KA349), and Project of Independent Cultivated Innovation Team of Jinan City (no. 2018GXRC002).

References

[1] Y. Zhao, R. Yang, G. Chevalier, and M. Gong, "Deep residual bidir-lstm for human activity recognition using wearable sensors," *CoRR, abs*, vol. 1708, Article ID 08989, 2017.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and y. A. Blaise Aguera, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference On Artificial Intelligence And Statistics*, A. Singh and J. Zhu, Eds., vol. 54, pp. 1273–1282, PMLR, Lauderdale, FL, USA, April 2017.

[3] M. J. Sheller, G. Anthony Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: a feasibility study on brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi, S. Bakas, H. Kuijf, F. Keyvan, M. Reyes, and T. van Walsum, Eds., Springer International Publishing, Berlin, Germnay, pp. 92–104, 2019.

[4] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," 2017, https://arxiv.org/abs/1710.06963.

[5] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, vol. 37, 2021.

[6] B. Keith, V. Ivanov, B. Kreuter et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, Dallas, TX, USA, October 2017.

[7] R. Kumar, A. A. Khan, J. Kumar et al., "Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging," *IEEE Sensors Journal*, vol. 21, no. 14, pp. 16301–16314, 2021.

[8] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021.

[9] H. Khelifi, S. Luo, B. Nour et al., "Bringing deep learning at the edge of information-centric internet of things," *IEEE Communications Letters*, vol. 23, no. 1, pp. 52–55, 2018.

[10] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge

computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.

[11] C. Zhao, S. Zhao, M. Zhao et al., "Secure multi-party computation: theory, practice and applications," *Information Sciences*, vol. 476, pp. 357–372, 2019.

[12] "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[13] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, "Semi-selfish mining based on hidden Markov decision process," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3596–3612, 2021.

[14] G. Wood, P. Ardoin, D. M. Brink et al., "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[15] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *IEEE Internet Computing*, vol. 6, no. 5, pp. 50–58, 2002.

[16] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[17] M. Mohammed Sadeeq, N. M. Abdulkareem, S. R. Zeebaree et al., "Iot and cloud computing issues, challenges and opportunities: a review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 1–7, 2021.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and y. A. Blaise Aguera, "Communication-efficient learning of deep networks from decentralized data," pp. 1273–1282, PMLR, 2017, https://arxiv.org/abs/1602.05629.

[19] Li Tian, A. Kumar Sahu, A. Talwalkar, and V. Smith, "Federated learning: challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[20] T. Li, Z. Wang, Y. Chen, C. Li, Y. Jia, and Y. Yang, "Is semi-selfish mining available without being detected?" *International Journal of Intelligent Systems*, 2021.

[21] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, 1997.

[22] W. Zou, D. Lo, P. Singh Kochhar et al., "Smart contract development: challenges and opportunities," *IEEE Transactions on Software Engineering*, vol. 47, 2019.

[23] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, Paris, France, August 2010.

[24] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 177–194, Springer, Singapore, December 6–10, 2021.