Edge-Enabled Distributed Deep Learning for 5G Privacy Protection

Qibo Sun, Jinliang Xu, Xiao Ma, Ao Zhou, Ching-Hsien Hsu, and Shangguang Wang

Abstract

Due to the limited storage and computing power, edge devices at the network edge cannot train deep learning models locally. Traditional deep learning training requires users to upload a local dataset to a cloud center, and trains the data using massive computation resources of the cloud center. However, it results in two bad effects: uploading a local dataset to a centralized cloud center controlled by a third party leaves user data privacy at risk; and uploading multimedia data will consume huge bandwidth resources of mobile users and storage resources of the cloud center, resulting in low scalability in term of the number of edge devices. To deal with these two problems, we propose an edge-enabled distributed deep learning platform by dividing a general deep learning training network into a front and back subnetwork. Specifically, the front subnetwork consisting of several layers is deployed close to input data and is trained separately at each edge device using the local dataset, and the outputs of all front subnetworks are sent to the back subnetwork for later training at a cloud center; while the back subnetwork is deployed at the cloud center, and its output is sent to each front subnetwork. As no original dataset is transferred from edge devices to the cloud center, the platform can protect data privacy and has high scalability. Above that, another two measures are taken to ensure data privacy: asymmetric encryption technology is adopted to guarantee the safety and integrality of the transferred parameters between edge servers and the cloud center; and blockchain technology is used to monitor the actions of the stakeholders in this platform and thereby ensure trust among the stakeholders. Experimental results show the validation of the proposed method.

Introduction

5G mobile edge computing (5G MEC) deploys numerous small cloud centers at the network edge, and provides cloud resources in proximity to mobile users with high network bandwidth and low delay [1, 2]. Compared to the traditional cloud computing framework, 5G MEC can be considered as a distributed form of cloud computing. Coordination of cloud computing and 5G MEC can work together to provide better quality of services to mobile users. In 5G MEC, hardware or software resources can be owned by different parties, for example, telecom operators, IT content service providers, general companies that rely on cloud computing services, and even individual users. Thus, the management of 5G MEC should be decentralized, that is, all the stakeholders of 5G MEC can contribute to the global decision of the platform and no one can decide independently without the others [3].

Because of the above advantages of 5G MEC, various edge devices such as mobile phones and surveillance cameras at the network edge generate enormous data to be processed. But a single edge device cannot train a deep learning model due to the limited computation and storage resources. Thus, mobile users, that is, owners of edge devices, are required to upload local datasets to a remote cloud center in the traditional training model of deep learning applications, as shown in Fig. 1. Then, the cloud center trains a general deep learning model using the datasets from all the edge devices. This training process is centralized and energy-efficient. Moreover, the training results fit for all datasets since the training process makes use of information from all the users. However, the centralized training model may result in a data breach if the cloud center is disrupted, putting user data privacy at risk. In addition, uploading data-intensive multimedia data, such as pictures and video clips, from edge devices to the cloud center consumes enormous wireless bandwidth resources, resulting in low scalability in terms of the number of edge devices [4].

In summary, a traditional deep learning training framework consists of two parts: servers deployed at the remote cloud center, and clients at mobile users. Cloud servers need to collect datasets generated by mobile users to train various deep learning applications, thereby providing services for the target users. However, uploading data to cloud servers is undesired for mobile users due to privacy issues [5-7], resulting in a lack of data to train a traditional deep learning model in the cloud center.

To solve these problems, we propose an edge-enabled distributed deep learning platform by dividing a general deep learning training network into a front and back subnetwork. The front subnetwork consisting of several layers is deployed close to input data and is trained separately at each edge device using the local dataset. During the training process at edge devices, the output data of the former layer is the input data of the latter

Qibo Sun, Jinliang Xu, Xiao Ma, Ao Zhou, and Shangguang Wang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Ching-Hsien Hsu is with the Department of Computer Science and Information Engineering, Asia University, Taiwan; also with Guangdong-Hong Kong-Macao Joint Laboratory for Intelligent Micro-Nano Optoelectronic Technology, School of Mathematics and Big Data, Foshan University, Foshan 528000, China; also with the Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan. Ching-Hsien Hsu is the corresponding author.

Digital Object Identifier: 10.1109/MNET.021.2000292

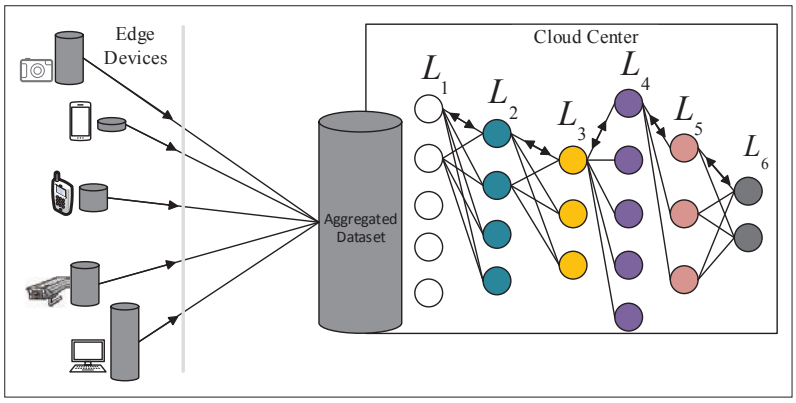

FIGURE 1. Training model of the traditional deep learning applications. Datasets generated from edge devices are sent to the remote cloud center. After collecting the datasets, the cloud center begins to train a general deep learning model. Collecting data costs time and money, and reveals privacy data of mobile users. The deep learning model consists of six layers of parameters. Each layer is in one color, and they are denoted by symbols L_1, L_2, L_3, L_4, L_5, and L_6 in this figure.

layer [8]. The back subnetwork is deployed at the cloud center. The outputs of all front subnetworks are sent to the back subnetwork for training at the cloud center. After the training at the cloud center is completed, the output is sent to each front subnetwork. In our proposed deep learning platform, datasets generated by mobile users are only transmitted from mobile devices to edge servers, which is helpful for reducing energy consumption and privacy preserving.

With the proposed deep learning framework, mobile users only need to transmit the parameters trained at edge servers to the cloud center, instead of uploading the original data. When a mobile user transfers parameter data to the cloud center through the public network, anyone can monitor the data transferring process and obtain these parameters. Then the model training process can be attacked in the way of a Sybil attack. To be specific, if node A wants to send data to node B, an attacker first behaves like node B to cheat node A so as to obtain the data. Then the attacker modifies the data, and at last sends the modified data to node B. To further guarantee the data security and integrality, we propose to adopt asymmetric encryption technology. It can not only protect the data from being seen and modified by the third party, but also inform the receiver that the data is from the expected sender. Thus, it is safe for parameter transmission during model training in our proposed deep learning framework.

As indicated in the above, the proposed edge-enabled deep learning framework can be deployed on the decentralized 5G MEC platform. It requires cooperation among several stakeholders, for example, mobile users, edge servers of the 5G MEC, and the cloud center. Mobile users send the original data to edge servers. Edge servers take user data as input and export parameters of a certain layer of the deep learning model, and provision computation resources for the initial training. The cloud center is responsible for organizing these stakeholders and processing the remaining training tasks. In summary, our contributions are as follows:

- We divide the deep learning network into two parts that are trained separately at edge servers and the cloud center. This can avoid uploading the original data from mobile users to the cloud center.
- We introduce edge servers which have sufficient storage, computing and network resources in 5G MEC to join the training process.
- We use asymmetric encryption technology to protect data safety and integrality when transmitting parameters from edge servers to the cloud center during a training process.

The rest of this work is organized as follows In the following section we introduce how to split a general deep learning model into two parts, and how it works during the training process. Then we detail how to use asymmetric encryption technology to protect the safety and integrality of the transferred parameters between edge servers and the cloud center. Following that we describe how blockchain is used to provide trust among the stakeholders. We then list the experimental results against baselines to validate the proposed method. Following that we give the existing related works. The final section concludes this work.

Edge-Enabled Distributed Deep Learning

Nowadays a network application can be divided into two parts: servers running on a remote cloud, for example, Google cloud (https://cloud.google.com/) or Aliyun (https://www.aliyun.com/), and clients at edge devices. Clients generate data all the time and store it locally. As time goes on, the old data may be deleted from the devices or most of the old data will be cleaned, because storage on mobile devices is limited, and mobile users must make a choice. The cloud server wants to collect every piece of user data to the cloud, and it can use this data to train all kinds of deep learning models, and then provide corresponding services to mobile users to make money, like targeted advisement or recommendations. To the cloud server, it is a great waste to delete the data generated by users. However, for privacy protection, a mobile user is unwilling to upload data to the cloud server. As a result, without a well-trained deep learning model, the server cannot make more money in targeted recommendations, and mobile users cannot enjoy high quality services.

We adjust the deep learning model to solve this problem. It cannot only use all of the user data to train the model globally, but also avoid uploading user data to the cloud server. We notice that a deep learning model has multiple hidden layers. The former layer of the model takes the unlabeled original user data as input, and the latter layer exports the predicted label and compares it with the real label in the original data. Then it propagates back to improve the parameters in the hidden layers to make the model perform better. Every hidden layer has many parameters, and between two layers are many parameters also. Both of them need to be trained with the original data.

As is shown in Fig. 2, we cut a deep learning model into two parts at a certain hidden layer. The left part is executed at every mobile device of mobile users. It is easy to know that the left part is many entities, and we call it the client part. The right part is only one entity, and is running on the

cloud center. We call it the cloud part. The entities of the left part run in parallel, and output the value of the hidden layer parameters. The server part receives these parameters, integrates them together using methods like weighted summarizing, and then send them to the next hidden layer until the output. When getting the output, the cloud part compares the output with the real label of the datasets, and tells the parameters to improve themselves. The parameter improve process begins from the last layer to the first layer. This is a converse direction against the first training step. As the cloud can only get the parameters of a certain hidden layer instead of the original user data, the proposed method can protect user privacy.

This kind of distributed training is quite different from the traditional distributed deep learning model. It is fit for the case of mobile Internet. In the traditional distributed deep learning framework, every node has a unique dataset, which is owned by itself, but each node has a whole copy of the deep learning network and parameters. During the training process, these nodes share the parameters with each other in some way. TensorFlow is a very famous deep learning framework, and it supports this type of distributed training framework. It also provides two kinds of parameter sharing method, namely synchronous and asynchronous. Traditional distributed training can also protect user data to some extent, but due to the limited computing power and battery capacity, it is difficult to train a deep learning model on a single mobile device. Moreover, network bandwidth and delay of a mobile device are also not fit for parameter sharing in a peer to peer way.

In a 5G MEC environment, edge servers can protect mobile devices from executing the logic of the client part. As we know, all kinds of resources needed for distributed training are very limited, so even the client part is very difficult for them, and the mobile devices can offload the task of the client part to an edge server. As the distance of the network connection between a mobile device and an edge server is very small, the network bandwidth and delay is very good for mobile users. At the same time, mobile users can do other work using their mobile devices. Edge servers are safer than the remote cloud center, because personals and companies can deploy their own edge servers. The users have complete control over the edge servers and the data stored on them, while in a remote cloud, things become quite different. Operation details of a remote cloud are not accessible for people, and none of the mobile users can control it.

Asymmetric Encryption for Safety and Integrality

In the last section, we proposed to transfer parameters between edge servers and the remote cloud center. That is to say, it is needed to transfer parameters between two nodes. Suppose node A is going to transfer data to node B. There exists the following safety problem in this process:

- For node B, how to verify that the data is from node A is important to avoid the denial of node A.
- For node B, how to confirm that the data has not been modified by malicious node.

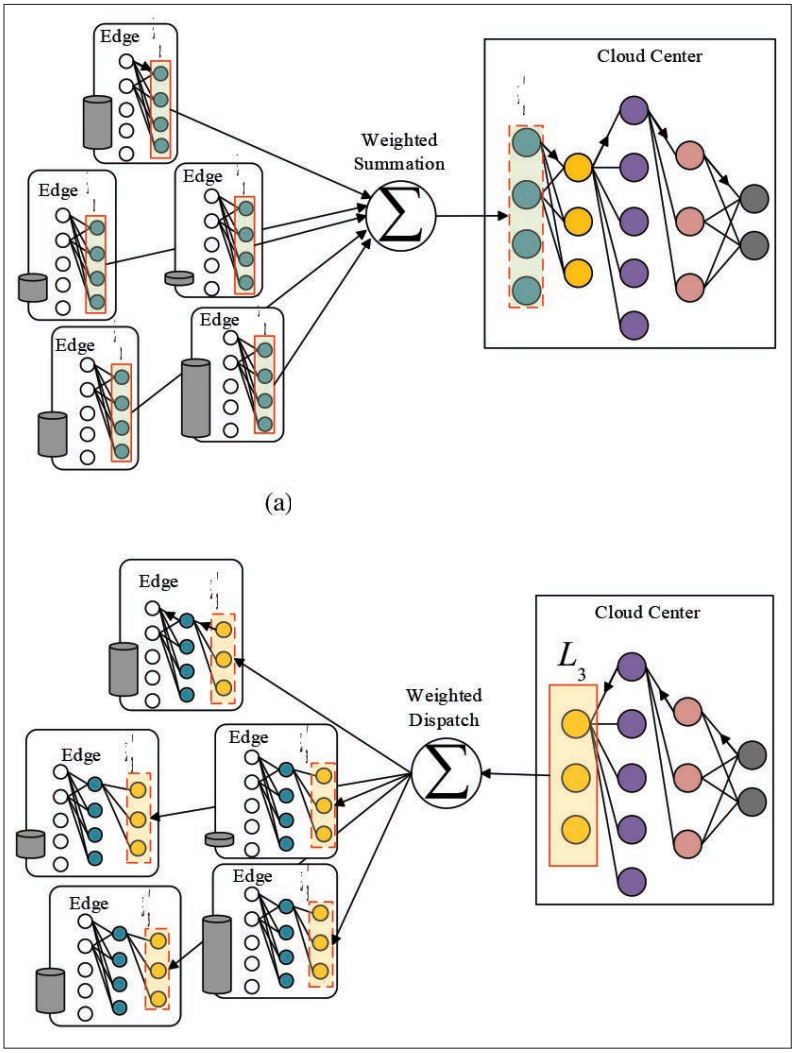

FIGURE 2. Parameter learning of the proposed edge-enabled deep learning framework. In the forward propagation, parameters from all edge servers or mobile devices are united with a weighted sum method to output new parameters for the next hidden layer. While in the back propagation, parameters from the server part are copied to numerous edge servers or mobile devices: a) forward propagation; b) back propagation.

If these two problems cannot be solved, edge servers will not trust each other in training a deep learning model.

We propose to use asymmetric encryption to solve the above-mentioned problems. Suppose the private key of node A is prv_A, and its public key is denoted by pub_A. The private key of node B is prv_B, and its public key is pub_B. The public keys are known to everyone, and the private keys are kept by themselves. Before sending data out, node A should do as shown in Fig. 3a:

- Use its private key prv_A to make a signature for the data, and obtain the signature denoted by $sig_{A,data}$.
- Pack data and $sig_{A,data}$ together, denote it as data2.
- Use the public key of node B pub_B to encrypt data2, and obtain the data after encryption as D.

After all of these steps, node A sends D to node B just as what it should do before. When node B receives data, it should do the following steps as shown in Fig. 3b:

- Node B first uses its own private key prv_B to decrypt data D, and obtains data2.
- Then it uses the public key of node A to verify whether $sig_{A,data}$ is the signature of data generated by the private key of node A prv_A.
- If yes, the verification is successful, and node B knows that the data is from node A. Node A cannot deny it. And the data has not been modified by anyone during the transferring process.

Now we discuss why asymmetric encryption can be used to meet our demands to solve the above-mentioned two safety problems. The reason that node B knows that the data is sent to itself instead of others is that the data can be decrypted with the private key of node B. It knows that the data is from node A for the fact that the signature is generated by A. The reason that the data has not been modified by others is that once the data is modified, the decrypt and verification will not be successful. What is more, when the data is transferred in the network, the data has been encrypted, so no one knows its contents [9].

Here we use two kinds of algorithm in the area of digital encryption:

- The private key and public key are symmetrical. That is to say, the data encrypted by the private key can be decrypted by the public key, and the private key can decrypt the data that is encrypted by the public key.
- The message generating algorithm can transform arbitrary data into a hash string with the same length. If we use a private key to encrypt the string, we get the signature of the original data.

Blockchain as Management Service

According to the analysis in the last section, there is no trust between an edge server and cloud center, or between two edge servers. That is because anyone can provide or remove an edge server as they wish. It is not beneficial for the cooperation between edge servers and the cloud center. We propose to use blockchain to serve as a third-party guarantee, and provide trust for them. Blockchain can save logs or information into a distributed ledger that no one can modify. If the actions of any node are saved to the blockchain, everyone can check its history and assess its trust level. If the node's history is very good, more people will cooperate with it. Otherwise, its life in the distributed network for deep learning training will be finished. That is how the blockchain can provide management service.

As is shown in Fig. 4, the blockchain platform is decentralized. Its nodes run on numerous edge servers in 5G MEC, just as the training nodes of the deep learning model. The blockchain together with smart contract is helpful for the cooperation of nodes, reducing the burden of management, and providing a more flexible way to protect the data in transferring on the network.

The intrinsic attributes of the 5G MEC environment determine that blockchain management is fit for the devices in the Internet of Things. As we know, there will be millions of various devices in 5G MEC, such as sensors, cameras, intelligent vehicles, and mobile phones. If the communications between these devices must go through the core net or cloud center, the long network delay will be very terrible, and the bandwidth resources will be insufficient. Under the monitor of the blockchain, any two devices can communicate with each other without a third party serving as the guarantee, and no data or message is needed to be transferred to the core net or cloud center.

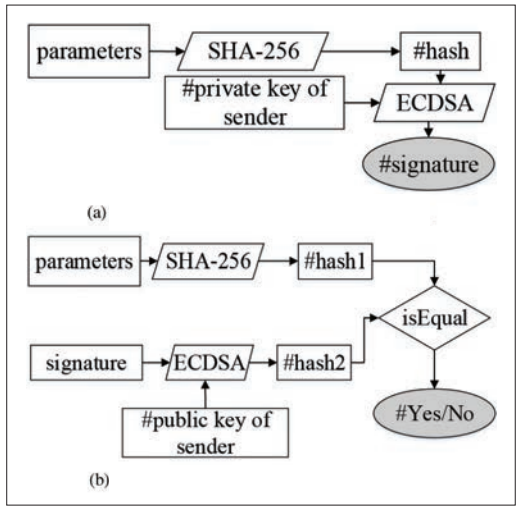

FIGURE 3. Signature of transferred data at the sender and verification at the receiver: a) signature of the parameters before sending; b) verification of signature when receiving data.

Experiments

In the above sections, we have demonstrated theoretically why the proposed method can work well. In this section, we present quantifiable outcomes of the proposed method against the baseline, and analyze the reasonability of the results.

We use a classical deep convolutional neural network named Le-Net5, which uses MNIST handwritten number photos as training data [10]. This data contains 60,000 photos as the test set, and 10,000 photos as the training set. The S2 layer is responsible for connecting the cloud part and the middle hidden layer, and has 1176 parameters. In the front propagation process, each mobile device sends 1,176 values to the cloud server, while in the back propagation, the cloud server sends 1,776 parameters to mobile devices. If we transmit the connection parameters between the two hidden layers, the number of parameters to be sent is 14146, which is much larger than the former one. As a result, sharing the hidden layer can reduce the transmission data volume better than sharing the connection parameters. Figure 5 compares the scalability of different methods. Specifically, the proposed method sends out 1176 parameter values while the baseline sends out 14146 parameter values. When the scaling size becomes two, the number of parameter values to be sent by the proposed method becomes 2352, which is approximately double 1176, while the baseline sends out 56584 parameter values, which is approximately four times 14146. When the scaling size becomes four, the number of parameter values to be sent by the proposed method becomes 4704, which is approximately four times 1176, while the baseline sends out 226326 parameter values, which is approximately

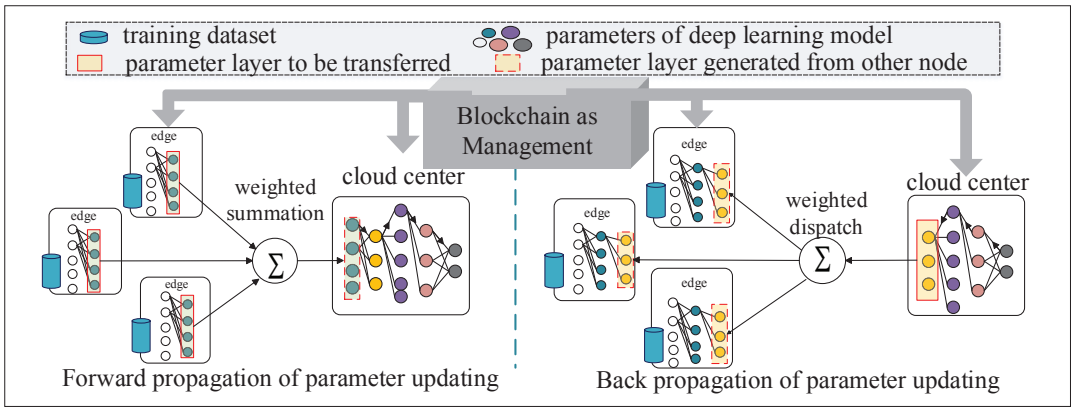

FIGURE 4. Distributed deep learning model training under the management of a Blockchain platform. The blockchain node and training node both work on an edge server in 5G MEC. The work of blockchain is to supervise all activities of edge clouds and remote cloud center.

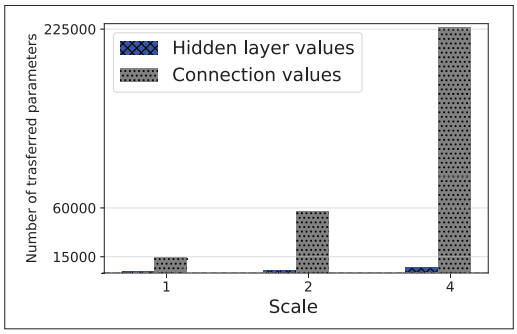

FIGURE 5. The comparison in scalability between the proposed and baseline method. The proposed just send out values of hidden layer. While the baseline communicates the connection parameters between two nodes.

16 times 14146. It is not hard to know that the data to be transferred by the proposed method is linear to the scaling size, while the baseline must send out the second power of the number of parameter values. As a result, we can conclude that the proposed method scales better than the baseline, which is more applicable for 5G MEC applications.

To validate the proposed framework, we further develop a new blockchain platform for 5G MEC applications, especially for the distributed deep learning applications. As shown in Fig. 6, we add several segments in the transaction of the blockchain network, that is, serviceId, postId, actionId and data. With these segments, the interactions between any pair of devices and the cloud center can be recorded in the blockchain, and anyone can check it at any time. So it is beneficial for building trust among devices in 5G MEC.

As we have mentioned, we design and develop a new blockchain for the distributed deep learning training framework (https://github.com/EdgeIntelligenceChain/EdgenceChain). The performance of the blockchain has an important influence on the training process. Table 1 shows the improvements of the proposed blockchain against the baselines. The baselines are two of the most popular blockchains, namely Bitcoin and Ethereum. As we can see, the blockchain of the proposed blockchain has the largest block volume, that is, the proposed blockchain can pack

	Block volume (bytes)	TPS (1)	Confirmation time (seconds)	Average fee ($)
Bitcoin	2M	7	3600	1
Ethereum	1M	35	18	0.15
Proposed	8M	20,000	3	0

TABLE 1. Improvements of the proposed blockchain against baselines.

the most transactions in a block. The confirmation of the proposed blockchain is the least, because the proposed blockchain adopts the proof of masternodes as its proof strategy. While Bitcoin and Ethereum use proof of working, which is very cumbersome. As a result, the number of transactions confirmed per second (TPS) of the proposed blockchain is the largest. In addition, the average fee of the proposed blockchain is the least, which can lower the threshold for using the blockchain to develop 5G MEC applications.

Related Works

In 5G MEC, a cellular base station can be integrated with an edge server. So the base station can not only provide communication services, but also provide storage services, computing services for task offloading, and even content services. With the help of Evolved Packet Core (EPC), 5G signal can be transformed into WiFi signal. This kind of WiFi signal can coexist and even work together with other WiFi services, and serve people who stay indoors and in other closed regions. Compared with 4G today, 5G MEC can provide very high network bandwidth and improve the quality of experiences of mobile users [11, 12].

With cloud computing as the enabling technology, more and more mobile applications can leverage the rich computing resources in the cloud server and offloading heavy computation tasks to the remote cloud server, which has been studied for over a decade [1, 13]. Although such offloading could significantly expand the capability of mobile devices, a long network transmission delay gets in the way inevitably since the remote clouds are usually far away from the mobile users. Such long latency may severely downgrade the user's experience especially for delay-sensitive applications. To mitigate the problem, 5G MEC

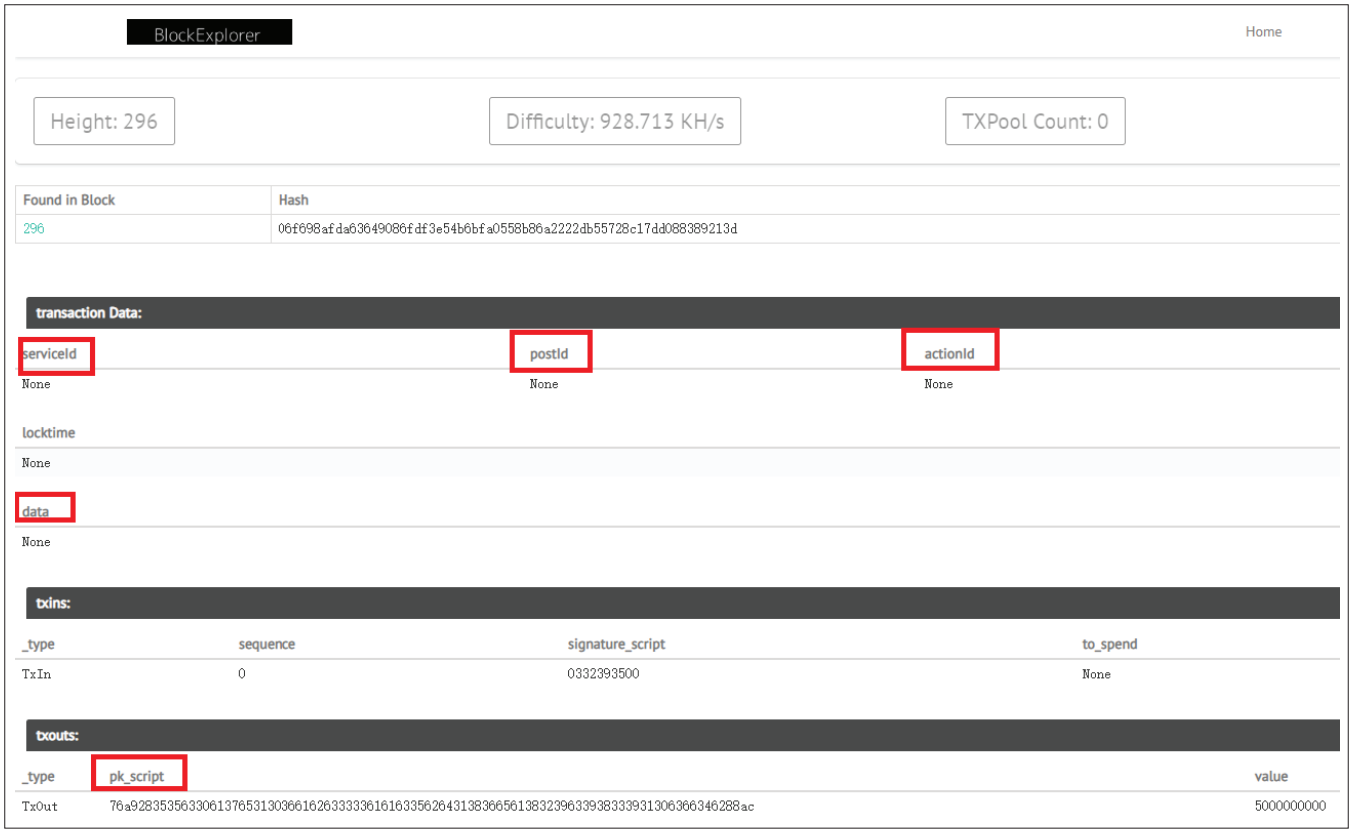

FIGURE 6. The structure of a transaction in the blockchain platform. Different from the general blockchain platform, it is designed for 5G MEC use, like distributed deep learning training in this work.

has been proposed, which places a number of small scale servers at the network edge and they can be reached by nearby mobile users via wireless connections. The core idea of mobile edge computing is dispersing data storage, processing, and applications on devices located at the network edge rather than implemented almost entirely in the remote cloud. It can make full use of the computing power and storage capability of the network edge devices and reduce network stress. Compared with cloud computing, the computing power and data storage are closer to the mobile devices and the amount of network transmission and delay are notably reduced. Based on the idea of mobile edge computing, Soyata *et al.* [14] proposed a mobile cloudlet cloud architecture and aimed at performing task load among cloud servers to minimize the response time given diverse communication latencies and server compute powers. Hu *et al.* [15] proposed a face identification and resolution scheme based on fog computing, which can reduce the amount of network transmission by preprocessing the AI tasks at the fog nodes. However, these tasks do not focus on the cooperation of different devices. They just consider them under the management of one owner, and make an integrated plan for all of them [16].

However, there exist big risks in these 5G MEC enabled distributed applications. In fact, many infrastructures in 5G MEC are owned by different people. No one can make plans for others. In this situation, cooperation between different resources is very important [3]. However, there is no mutual trust between them. As a result, building trust for different resources in 5G MEC is needed before any practical application based on edge servers are open to public. Blockchain uses smartcontract technology with multi-party authentication, a very useful tool to solve this problem [17]. Any action on it, everyone can check it with small cost. If most people consider it wrong or not needed, this action is a failure. In the proposed edge-enabled distributed deep learning framework, we build it on the top of a blockchain network. Specifically, the framework uses blockchain as a tool for authentication, and then cooperate with each other. The work in [18] aims to solve the problem of poor network latency and improve the privacy protection level of IoT, without focusing on the layered deep learning model.

While the proposed deep learning framework is distributed, its different parts, that is, mobile users, edge servers and cloud center, must exchange their parameters to continue the learning process. These data will be revealed to the public, so how to protect these data is important. Most existing works such as [14, 16] do not focus on this problem. The work in [19] puts each part at an agent, and the agent knows how to cooperate with each other, when to communicate with others, and how to protect the transferred data. However, it is not customized for a deep learning model, so it is not very valid in the training of a deep learning process. What is more, it is too complex to protect the parameter data in the situation of this work. As a result, we propose to use asymmetric encryption to protect the safety and integrality of the data. It helps to avoid leakage of parameters, and the deny of the sender of the

data. More important, it is very simple in comparison to the agent technology.

Conclusion

In this work, we introduced a new distributed deep learning training framework to protect user data. It runs on the network of edge servers and uses blockchain to monitor the actions of every stakeholder in the community. To protect the transmitted parameters, we adopt asymmetric encryption technology to ensure safety and integrality. This work provides a new way for devices in 5G MEC environments to cooperate with each other. In future work, we will focus on improving the proposed edge-enabled distributed deep learning framework, and pushing the platform to a point where anyone can take advantage of asymmetric encryption in protecting data privacy.

This work provides a new way for devices in 5G MEC environments to cooperate with each other. In future work, we will focus on improving the proposed edge-enabled distributed deep learning framework, and pushing the platform to a point where anyone can take advantage of asymmetric encryption in protecting data privacy.

References

[1] A. Alnoman *et al.*, "Emerging Edge Computing Technologies for Distributed IoT Systems," *IEEE Network*, vol. 33, no. 6, 2019, pp. 140–47.
[2] J. Xu *et al.*, "Path Selection for Seamless Service Migration in Vehicular Edge Computing," *IEEE Internet of Things J.*, vol. 7, no. 9, 2020, pp. 9040–49.
[3] J. Xu *et al.*, "A Blockchain-Enabled Trustless Crowd-Intelligence Ecosystem on Mobile Edge Computing," *IEEE Trans. Industrial Informatics*, vol. 15, no. 6, 2019, pp. 3538–47.
[4] M. Wang *et al.*, "Minerva: A Scalable and Highly Efficient Training Platform for Deep Learning," *Nips Workshop Distributed Machine Learning & Matrix Computations*, 2014.
[5] S. Chang and C. Li, "Privacy in Neural Network Learning: Threats and Countermeasures," *IEEE Network*, vol. 32, no. 4, 2018, pp. 61–67.
[6] M. S. Riazi and F. Koushanfar, "Privacy-Preserving Deep Learning and Inference," *The International Conference*, 2018.
[7] L. Song *et al.*, "PPD-DL: Privacy-Preserving Decentralized Deep Learning," *Proc. Int'l. Conf. Artificial Intelligence and Security*, 2019.
[8] A. Jindal *et al.*, "Sedative: SDN-Enabled Deep Learning Architecture for Network Traffic Control in Vehicular Cyber-Physical Systems," *IEEE Network*, vol. 32, no. 6, 2018, pp. 66–73.
[9] N. Rajesh and A. A. L. Selvakumar, "Association Rules and Deep Learning for Cryptographic Algorithm in Privacy Preserving Data Mining," *Cluster Computing*, vol. 22, no. 4, 2018, pp. 119–31.
[10] Y. LeCun *et al.*, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, 1998, pp. 2278–2324.
[11] K. Xiong *et al.*, "Smart Network Slicing for Vehicular Fog-RANs," *IEEE Trans. Vehicular Technology*, vol. 68, no. 4, 2019, pp. 3075–85.
[12] G. Qiao *et al.*, "Deep Reinforcement Learning for Cooperative Content Caching in Vehicular Edge Computing and Networks," *IEEE Internet of Things J.*, vol. 7, no. 1, 2020, pp. 247–257.
[13] S. Wang *et al.*, "A Survey on Service Migration in Mobile Edge Computing," *IEEE Access*, vol. 6, 2018, pp. 23511–28.
[14] T. Soyata *et al.*, "Cloud-Vision: Real-Time Face Recognition Using a Mobile-Cloudlet-Cloud Acceleration Architecture," *Proc. 2012 IEEE Symposium on Computers and Commun. (ISCC)*, IEEE, 2012, pp. 000059–66.
[15] P. Hu *et al.*, "Fog Computing Based Face Identification and Resolution Scheme in Internet of Things," *IEEE Trans. Industrial Informatics*, vol. 13, no. 4, 2016, pp. 1910–20.
[16] J. Dean *et al.*, "Large Scale Distributed Deep Networks," *Advances in Neural Information Processing Systems*, 2012, pp. 1223–31.
[17] B. Cao *et al.*, "When Internet of Things Meets Blockchain: Challenges in Distributed Consensus," *IEEE Network*, vol. 33, no. 6, 2019, pp. 133–39.
[18] T. Liu *et al.*, "Privacy Protection Based on Stream Cipher for Spatio-Temporal Data in IoT," *IEEE Internet of Things J.*, 2020, pp. 1–13.
[19] P. Angin *et al.*, "A Self-Protecting Agents Based Model for High-Performance Mobile-Cloud Computing," *Computers & Security*, vol. 77, 2018, pp. 380–96.

Biographies

Qibo Sun (qbsun@bupt.edu.cn) received his Ph.D. degree in communication and electronic systems from Beijing University of Posts and Telecommunication in 2002. He is currently an associate professor at Beijing University of Posts and Telecommunication in China. He is a member of the China computer federation. His research interests include services computing, Internet of Things, and next generation network intelligence.

Jinliang Xu (jlxu@bupt.edu.cn) received the bachelor degree in electronic information science and technology from Beijing University of Posts and Telecommunications in 2014. Currently, he is a Ph.D. candidate in computer science at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests include mobile cloud computing, blockchain, AI, and crowdsourcing.

Xiao Ma (maxiao18@bupt.edu.cn) received her Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2018. She is currently a postdoctoral fellow at the State Key Laboratory of Networking and Switching Technology, BUPT. From October 2016 to April 2017, she visited the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Her research interests include mobile cloud computing and mobile edge computing.

Ao Zhou (aozhou@bupt.edu.cn) is an associate professor at the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. She received her Ph.D. degree in computer science at Beijing University of Posts and Telecommunications of China in 2015. Her research interests include cloud computing and service reliability.

Ching-Hsien Hsu (robertchh@gmail.com) is Chair Professor in the Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan. His research includes high performance computing, cloud computing, parallel and distributed systems, big data analytics and intelligence. He has published 200 papers in these areas, in top journals such as IEEE TPDS, IEEE TSC, IEEE TCC, IEEE TETC, IEEE T-SUSC, *IEEE Systems*, *IEEE Network*, *IEEE Communications Magazine*, and ACM TOMM.

Shangguang Wang (sgwang@bupt.edu.cn) received his Ph.D. degree in computer science from Beijing University of Posts and Telecommunications in 2011. He is a professor and Vice-Director at the State Key Laboratory of Networking and Switching Technology (BUPT). He has published more than 150 papers in recent years, and played a key role at many international conferences, such as general chair and PC chair. His research interests include service computing, cloud computing, and mobile edge computing. He is a senior member of the IEEE, and Editor-in-Chief of the *International Journal of Web Science*.