2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)

FedBC: Blockchain-based Decentralized Federated Learning

Xin Wu
Tsinghua University
Beijing, China

Zhi Wang
Tsinghua University
Beijing, China

Jian Zhao
Shenzhen Technology University
Shenzhen, China

Yan Zhang
Southern University of Science and Technology,
Peng Cheng Laboratory
Shenzhen, China

Yu Wu
Southern University of Science and Technology,
Peng Cheng Laboratory
Shenzhen, China

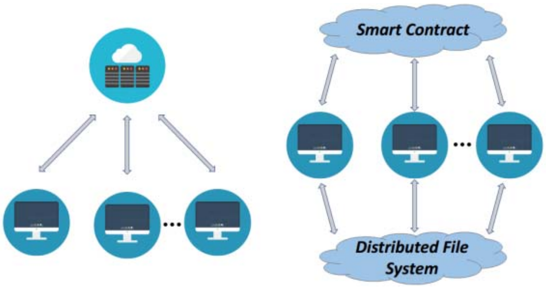
***Abstract*—Federated learning enables participants to collaborate on model training without directly exchanging raw data. Existing federated learning methods often follow the parameter server architecture, using third-party collaborators to provide aggregation and key management. In this case, the central node obtains information uploaded by other nodes. Studies have shown that with this information, the central node can infer important information, which leads to data privacy leakage. In addition, the failure on the server node can also cause the entire system to fail. We designed a completely decentralized federated learning framework based on blockchain, thereby avoiding the privacy and failure risk of the centralized structure. Moreover, we develop the corresponding model training approach. Compared with the existing methods, our framework performs better in terms of accuracy, robustness, and privacy.**

Keywords*—*federated learning, distributed computing, deep learning, blockchain

I. Introduction

Today, the importance of data is growing. For companies, data is their extremely important asset. By mining the value of data and training great models, companies can often enhance their competitiveness and gain greater profits [1]. However, model training often relies on huge amounts of data, and scarce data often makes it difficult to get a well-trained model [2]. For some small-scale companies, the amount of data they hold is very scarce, so it is very difficult to train great models on their own. In addition, raw data cannot be exchanged directly between companies due to commercial and policy reasons, leading to the generation of data isolation.

The emergence of federated learning is precisely to solve these problems [3]. It breaks down data isolation while protecting data privacy of participants, which helps participants train collaboratively to get better models [4]. The existing federated learning approaches still follow the widely used “parameter server” architecture in distributed computing [5]. As illustrated in Fig.1(a), such a design is centralized, and it relies on a centralized server. Each participating node locally calculates the gradients using local data, and uses cryptography to process the gradients, such as homomorphic encryption and differential privacy techniques. The processed information is uploaded to the server. The server aggregates them and further updates the global model parameters. The updated global model is distributed to the participating nodes to continue the next round of training. This process is repeated until the global model converges.



(a) Parameter Server (b) FedBC

Fig. 1. Comparison between different architecture.

However, existing approaches do not fully protect data privacy, even if the raw data is not transmitted between the participants and the server. Some researchers have pointed out that according to the historical gradients, important information related to the raw data can be inferred, and in some specific cases, even the raw data itself can be inferred [6]. Therefore, the original gradient information cannot be directly transmitted to the server. Shokri *et al.* [7] used differential privacy technique to protect data privacy, by adding noise to the uploaded gradients. Nevertheless, Hitaj *et al.* [8] demonstrated that a curious server can take advantage of generative adversarial network, to obtain data privacy from the gradients after differential privacy processing. Phong *et al* [9] exploited homomorphic encryption technique to prevent the curious server from gaining data privacy. They assumed that all participants are honest and not curious, but it is difficult to test whether the participants are curious in practical applications. Bonawitz *et al.* [10][11] established a secret sharing and symmetric encryption mechanism to prevent curious participants from causing privacy leakage. They assumed server and participants cannot collude at all, but this is not guaranteed in practice.

The root cause of data privacy risk lies in the centralized architecture. Although many researches attempt to solve existing problems through new technologies, at the same time, new problems arise. In order to address the risk of privacy leakage from the root cause, a completely decentralized architecture and approach must be designed. In distributed computing, there are also many studies on decentralization. Geng *et al.* [12] proposed to form hierarchical groups to synchronize gradients in distributed

978-1-7281-7005-3/20/$31.00 ©2020 IEEE

Dalian, China
June 27-29, 2020

manner. Wang *et al.* [13] developed a high-performance, low-cost gradient synchronization algorithm for distributed learning. But they often only focus on one of many aspects, such as topology, control, storage or key management. To establish a completely decentralized architecture for federated learning, it needs to be decentralized in all of the above aspects. It is a huge challenge to achieve decentralization in all of these aspects and make them work together as a whole. In summary, this paper makes the following contribution:

First, we design a federated learning framework called FedBC, which is fully decentralized in terms of topology, control, storage, and key management. Compared to existing methods, it avoids the risks of privacy leakage and single point of failure introduced by the centralized structure, and therefore has better security and robustness.

Second, we develop a training approach that is compatible with the architecture. It guarantees that our framework achieves the state-of-art accuracy performance. For collaborative training of participating nodes, it can make full use of the information provided by the nodes to train models with higher accuracy.

Finally, we compare our framework with existing differential privacy-based federated learning method. Experiments prove the advantages of our framework in terms of accuracy, robustness, and privacy. This shows that our framework has great application value.

II. The Proposed Framework

A. System Overview

Compared to the centralized design, the ring structure has great advantages in decentralization. However, the traditional ring structure has poor reliability. A single point of failure can cause the system to rebuild the topology and interrupt training. Therefore, we build the topology based on a decentralized blockchain, as shown in Fig.1(b). The logical topology of participants is ring, but their physical topology only needs to ensure that they are connected. The participating nodes are organized by decentralized blockchain technology, and the status of each participating node in the framework is completely equal. Failure or exit of any node will not affect the normal operation of the system.

The control and key management functions of the system are handled by smart contracts. A smart contract is a program deployed on a blockchain that performs actions based on pre-defined, open and transparent rules [14]. This process is also decentralized, and is not affected by any single node. By interacting with the smart contract, participating nodes within the system can flexibly join and exit. Failure of the participating nodes will also be handled by the smart contract and will not affect other participants to continue the training. In a round of training, the smart contract notifies the information necessary for the participating nodes to train, and processes the information uploaded by each node. Since the smart contract inherits the immutability of the blockchain technology, its code cannot be modified after deployment, ensuring stable and reliable control functions. At the same time, the transparency of the smart contract allows any node in the system to read its contents and verify its security.

For the distributed file system, we apply the IPFS (Interplanetary File System) based on blockchain technology [15]. It provides a reliable, secure, decentralized storage mechanism for files in the training process, and is seamlessly integrated with the rest of the system. It provides a unified access mechanism for intermediate files in the training process. It replaces domain-based addressing with content-based addressing compared to traditional distributed file systems. What the user is looking for is a content stored somewhere. The participating nodes read the file by verifying the hash of the content, which makes the file access faster, safer, more robust, and more durable.

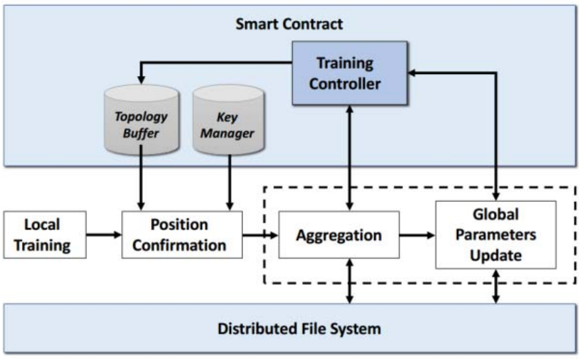


Fig. 2. The workflow of FedBC.

B. System Workflows

Our system consists of several steps to complete a round of training, as illustrated in Fig.2. This process will continue until the model converges. The specific steps are as follows:

- Local Training: Participating nodes use local data for training and updates the local model. When the node performs training with several batches locally, it subtracts the initial model at the beginning of the current round from the current model, and obtains the change value of the model parameters. Unlike the training method of the parameter server architecture, we aggregate the cumulative change values after multiple iterations of the model. Gradient information has been proven to be useful for inferring information about raw data [6]. Some researchers have attempted to pass and aggregate model parameters [16]. However, Song *et al.* [17] proved that the model parameters also contain information about the original data. Carlini *et al.* [18] showed that when a deep learning based sequence generation model is trained based on text data, it unintentionally memorizes the training data information that can be extracted from the model.
- Position Confirmation: Each node obtains the logical topology of the current training from the topology buffer in the smart contract, and confirms its position in the topology. The logical topology is ring and is regenerated by the controller module in the smart contract in each round. The purpose is to make the system more flexible in responding to the joining and exiting of nodes during the training process. Subsequently, each node queries the homomorphic public key used in the current round of training and the public key of the direct successor node. These public keys are managed uniformly by the key manager in the smart contract. The homomorphic encryption key is randomly generated by the smart contract before each round, and the code for

generating and using the key is transparent. Its public key can be queried by all nodes, but the private key can only be used inside a smart contract and cannot be read.

- Information Aggregation: For the change value of the model parameters, each node encrypts it with the homomorphic public key and records the encrypted change value. The smart contract randomly selects the starting node in the ring and notifies it to start the aggregation process. The first aggregated node directly stores its own encrypted change value into IPFS to obtain a file hash index. It uses the public key of the direct successor node to encrypt the hash index, and uploads the encrypted index to the smart contract. For other nodes, the smart contract interacts with them in turn according to the topology of the ring. The smart contract notifies the current node of the encrypted index of the previous node. After the current node decrypts the index by using the local private key, the corresponding file is obtained from IPFS. The node adds the local encrypted change value to the contents in the file, generates a new file and stores it in IPFS. Then, this node encrypts the index and uploads it to the smart contract. For the last node, since its direct successor is the starting node, it just needs to upload the unencrypted file index to the smart contract. In this step, each node assumes partial aggregation responsibility, making the entire aggregation process completely decentralized.
- Parameters Update: The smart contract acquires the corresponding file content from the IPFS based on the last unencrypted file index received. It decrypts the content using the homomorphic private key, stores the decrypted content into IPFS, and notifies all nodes on the ring of the hash index of the new file. Each node obtains the aggregated information from the IPFS according to the index, and adds the information to the initial model parameters of the current round to obtain a new model. After completing this step, the node begins the next round of training.
- Exception Handling: In our system, the status of all nodes is equal, which makes any single node failure will not affect the continued operation of the system. If the starting node on the ring fails, the smart contract reselects the new starting node. During the aggregation process, if a node on the ring does not respond for a long time, the smart contract notifies the previous normal node to re-encrypt the hash index according to the public key of the successor node, thereby skipping the abnormal node. When the last node on the ring fails, the smart contract notifies the previous normal node to perform the duties of the last node.

III. Experiments

In this section, we implement our FedBC framework and carry out some experiments based on this framework. We demonstrate the advantages of the system in terms of accuracy, robustness and privacy.

A. Experiment Setup

We design a framework that aggregates the change value of the model parameters in turn, and then uses the aggregated information to update the model. Therefore, it is applicable to almost all deep learning models. The blockchain part is built by Hyperledger, and the distributed file system is implemented using IPFS. We use the MNIST dataset, the most widely tested and used in federated learning [19]. For the model in the experiment, existing federated learning-related research usually uses a simple and general model [20], so we use a deep neural network with two hidden layers. In our main experiment, we set the number of participating nodes to 10, which is often used in experiments in the existing literature [21].

B. Baselines

We compare our federated learning approach to the following baselines in our experiments. 1) Training alone: In this case, each node trains the local model independently of each other using local data. There is no information interaction between the nodes. 2) Training with dropout: In this case, participating nodes in federated learning may drop out during the training process. Exited nodes can no longer participate in federated learning. If a node is allowed to participate in training midway, it is unfair to the nodes participating in the entire training process. 3) Training with differential privacy: In this case, participating nodes do not use our approach, but instead use existing differential privacy-based federated learning method [20].

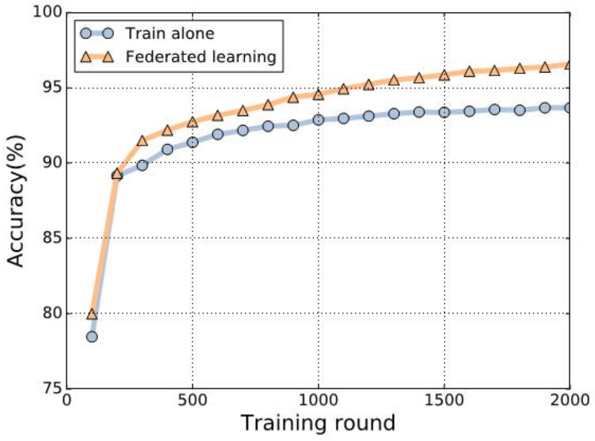


Fig. 3. Model accuracy when samples are sufficient.

C. Results

In terms of accuracy, we carry out some experiments to prove the effectiveness of our approach. We separately measure the performance of our designed federated learning approach when the sample size is sufficient or scarce. When the sample size of each node is sufficient, the model accuracy changes as shown in Fig.3. At this point, we let each node hold 5500 training samples. After 1000 rounds of training, compared with node training alone, our designed approach improves the model accuracy by 1.70%. After 2000 rounds, the gap between the two has reached 1.90%. Further, we measure the change in accuracy when the sample size was scarce, as shown in Fig.4. After 1000 rounds of training, the model accuracy increases by 6.81%. After 2000 rounds, the improvement in accuracy has reached 7.32%. Such experimental results show that, regardless of whether the sample size is sufficient or scarce, our approach can effectively help participating nodes to obtain models with higher accuracy. When the number of samples held by the node is relatively small, the improvement effect is more obvious. In addition, as training continues, the degree of improvement in model accuracy continues to increase.

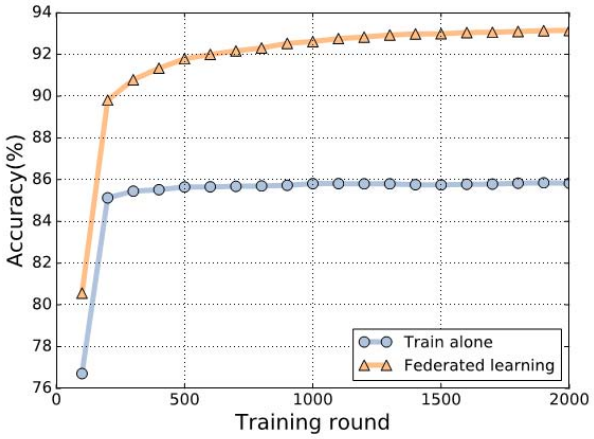


Fig. 4. Model accuracy when samples are scarce.

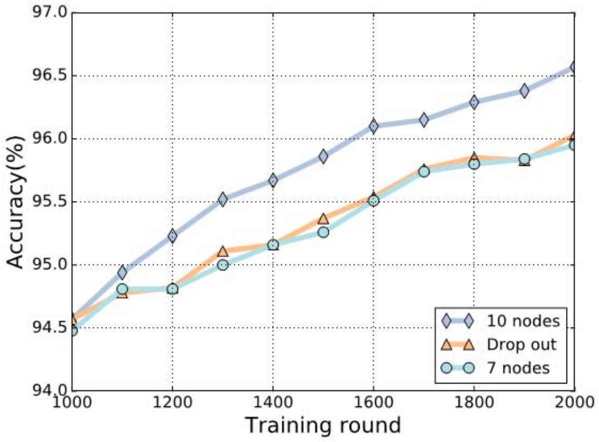


Fig. 5. Model accuracy when a part of nodes drop out.

Our framework is extremely robust compared to existing frameworks. During the training process, even if some nodes drop out, other nodes can continue to train and obtain good training results. Compared with the traditional centralized federated learning framework, our blockchain-based framework is decentralized, so there is no single point of failure. We perform experiments to show the impact on accuracy when some nodes drop out. In Fig.5, we show the change in accuracy from the 1000th round to the 2000th round. We first make 10 nodes participate in the first 1000 rounds of training, then about a third of the nodes drop out, and the remaining 7 nodes continue to train. At this time, although the dropout of some nodes makes the model accuracy lower than the original normal training level, the accuracy of the model still keeps increasing, which shows that our approach is effective in the face of the dropout of nodes. In addition, compared with the case of training 2000 rounds with only 7 nodes, the accuracy of the model obtained after the dropout is still slightly higher.

Finally, we compare with existing widely used federated learning method with differential privacy. The experimental results are shown in Fig.6. This shows that our approach performs better and help participating nodes to obtain higher model accuracy. Moreover, in our framework, the data privacy of participating nodes is guaranteed through our carefully designed architecture and its workflow. In this decentralized architecture, each node only communicates with its direct predecessor node and direct successor node. Only when the direct predecessor node and the direct successor node collude, the encrypted model change value of the corresponding node will be stolen. This encrypted information must rely on the private key of the last node in this round to decrypt it. Therefore, it needs at least three nodes to collude with each other to steal the data privacy of one node. In the traditional federated learning method, only two nodes need to collude in order to steal data privacy, namely the parameter server node and the key management node. This shows that our framework also has great advantages in terms of privacy protection.

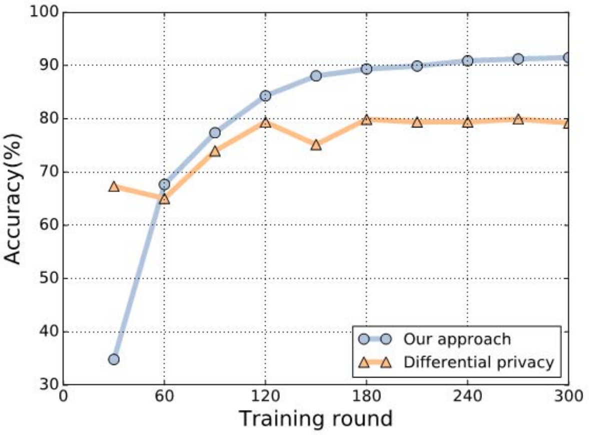


Fig. 6. Model accuracy compared with differential privacy.

In summary, our framework can not only help participating nodes improve the accuracy of the model, but also ensure the robustness that is not available in existing federated learning methods, while providing more secure privacy protection for participating nodes.

IV. Conclusion

In some privacy-sensitive scenarios, with the generation of data isolation and the urgent need for data volumes, federated learning has become an emerging solution. Existing federated learning usually relies on a parametric server architecture that leverages third-party collaborators to provide aggregation and key management. However, this centralized structure brings a lot of potential privacy and single point of failure risks, and the existing methods still cannot solve them well.

In this paper, we design FedBC, a blockchain-based decentralized federated learning framework. It achieves decentralization in all aspects of the system, thereby avoiding the privacy leakage caused by the existing centralized structure and the risk of system crash caused by a single point of failure. At the same time, we also develop the corresponding training approach. Experiments show that our approach can enable participating nodes to obtain models with higher accuracy than when they are trained separately. More importantly, even if some participating nodes drop out, the framework can still effectively help the remaining nodes to continue training to obtain a better model. Comparative experiments with existing method demonstrate the advantages of our approach in terms of accuracy.

ACKNOWLEDGMENT

This work is supported in part by NSFC under Grant No. 61872215, Grant No. 61702098, SZSTI under Grant No. JCYJ20180306174057899, Shenzhen Nanshan District Ling-Hang Team under Grant No. LHTD20170005, Natural Science Foundation of Top Talent of SZTU under Grant No. 2018010801008, and PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications under Grant No. LZC0019.

REFERENCES

[1] Alon Halevy, Peter Norvig, and Fernando Pereira, “The unreasonable effectiveness of data,” 2009.

[2] Banko, Michele, and Eric Brill. "Scaling to very very large corpora for natural language disambiguation." Proceedings of the 39th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2001.

[3] McMahan, Brendan, and Daniel Ramage. "Federated learning: Collaborative machine learning without centralized training data." Google Research Blog 3 (2017).

[4] Yang, Qiang, et al. "Federated machine learning: Concept and applications." ACM Transactions on Intelligent Systems and Technology (TIST) 10.2 (2019): 1-19.

[5] Li, Mu, et al. "Scaling distributed machine learning with the parameter server." 11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14). 2014.

[6] Aono, Yoshinori, et al. "Privacy-preserving deep learning: Revisited and enhanced." International Conference on Applications and Techniques in Information Security. Springer, Singapore, 2017..

[7] Shokri, Reza, and Vitaly Shmatikov. "Privacy-preserving deep learning." Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. 2015.

[8] Hitaj, Briland, Giuseppe Ateniese, and Fernando Perez-Cruz. "Deep models under the GAN: information leakage from collaborative deep learning." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

[9] Aono, Yoshinori, et al. "Privacy-preserving deep learning via additively homomorphic encryption." IEEE Transactions on Information Forensics and Security 13.5 (2017): 1333-1345.

[10] Bonawitz, Keith, et al. "Practical secure aggregation for privacy-preserving machine learning." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

[11] Bonawitz, Keith, et al. "Towards federated learning at scale: System design." arXiv preprint arXiv:1902.01046 (2019).

[12] Bonawitz, Keith, et al. "Towards federated learning at scale: System design." arXiv preprint arXiv:1902.01046 (2019).

[13] Wang, Songtao, et al. "Bml: A high-performance, low-cost gradient synchronization algorithm for dml training." Advances in Neural Information Processing Systems. 2018.

[14] Buterin, Vitalik. "A next-generation smart contract and decentralized application platform." white paper 3.37 (2014).

[15] Nyaletey, Emmanuel, et al. "BlockIPFS-Blockchain-enabled Interplanetary File System for Forensic and Trusted Data Traceability." 2019 IEEE International Conference on Blockchain (Blockchain). IEEE, 2019.

[16] Li, Xiang, et al. "On the convergence of fedavg on non-iid data." arXiv preprint arXiv:1907.02189 (2019).

[17] Song, Congzheng, Thomas Ristenpart, and Vitaly Shmatikov. "Machine learning models that remember too much." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017.

[18] Carlini, Nicholas, et al. "The secret sharer: Measuring unintended neural network memorization & extracting secrets." arXiv preprint arXiv:1802.08232 (2018).

[19] Truex, Stacey, et al. "A hybrid approach to privacy-preserving federated learning." Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security. 2019.

[20] Abadi, Martin, et al. "Deep learning with differential privacy." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016.

[21] Weng, Jiasi, et al. "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive." IEEE Transactions on Dependable and Secure Computing (2019).