A Blockchain-Based Decentralized Federated Learning Framework with Committee Consensus

Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, Zibin Zheng, and Qiang Yan

Abstract

Federated learning has been widely studied and applied to various scenarios, such as financial credit, medical identification, and so on. Under these settings, federated learning protects users from exposing their private data, while cooperatively training a shared machine learning algorithm model (i.e., the global model) for a variety of real-world applications. The only data exchanged is the gradient of the model or the updated model (i.e., the local model update). However, the security of federated learning is increasingly being questioned, due to the malicious clients or central servers' constant attack on the global model or user privacy data. To address these security issues, we propose a decentralized federated learning framework based on blockchain, that is, a Blockchain-based Federated Learning framework with Committee consensus (BFLC). Without a centralized server, the framework uses blockchain for the global model storage and the local model update exchange. To enable the proposed BFLC, we also devise an innovative committee consensus mechanism, which can effectively reduce the amount of consensus computing and reduce malicious attacks. We then discuss the scalability of BFLC, including theoretical security, storage optimization, and incentives. Finally, based on a FISCO blockchain system, we perform experiments using an AlexNet model on several frameworks with a real-world dataset FEMNIST. The experimental results demonstrate the effectiveness and security of the BFLC framework.

Introduction

With the introduction of GDPR (General Data Protection Regulation), both industry and academia began to pay more attention to the privacy protection of machine learning. User-generated private data should not be exposed or uploaded to a central server. Google proposed Federated Learning (FL) in 2016 to solve the problem of collaborative training for privacy protection. This framework proposes a distributed training model with two roles: the participating devices and the central server. Instead of uploading private data, nodes locally update the global model and then upload the model updates (i.e., the local gradients). The central server collects these updates and integrates them to form an updated model. Because of this privacy feature, FL has been attracting researchers' attention in recent years.

In FL settings, a server performs the central operations of update aggregation, client selection, and global model maintenance. The server needs to collect updates from numerous clients for aggregation operation, and it also needs to broadcast new global models to these clients, which puts a high demand on network bandwidth. Also, cloud-based servers are affected by the stability of cloud service providers [1]. A centralized server can skew the global model by favoring some clients. Malicious central servers can poison the model and even collect clients' privacy from updates. Therefore, the stability, fairness, and security of the central server are crucial to FL.

An intuitive idea is to remove the server and execute the central tasks on distributed clients. The blockchain, which is viewed as decentralized storage, can serve as the basis for FL training. In detail, we can design protocols to execute the aggregation task on clients. BAFFLE [2] mentions using blockchain to store and share the global model, and perform aggregation with Smart Contracts (SC). The smart contracts are computer programs triggered by the blockchain events, which are intended to automatically execute, control, or document for specified tasks. With the removal of the central server, the above challenges are avoided. However, the computation and network transmission pressure of this task are all transferred to the nodes. In particular, when all nodes have to deal with consensus tasks, the computational overhead is huge.

Zhou *et al.* [3] propose a decentralized multi-community training framework, which utilizes the blockchain to maintain a global model within each community. The communities communicate with others of the updated models following an all-reduce protocol (https://github.com/baidu-research/baidu-allreduce). Chen *et al.* [4] propose to leverage the blockchain to record the updates from nodes and the evaluation of those updates. Underrated nodes may be kicked out as a defense against malicious devices. However, maintaining multiple blockchains at the same time [3] is not conducive to model sharing, and nodes in different communities can hardly obtain models or records from other communities. If a community as a whole is malicious, it is difficult for other honest communities to detect and resist, then a trusty global detection might be needed.

Digital Object Identifier:
10.1109/MNET.011.2000263

Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, and Zibin Zheng are with Sun Yat-sen University; Qiang Yan is with WeBank Co. Ltd.

0890-8044/20/$25.00 © 2020 IEEE

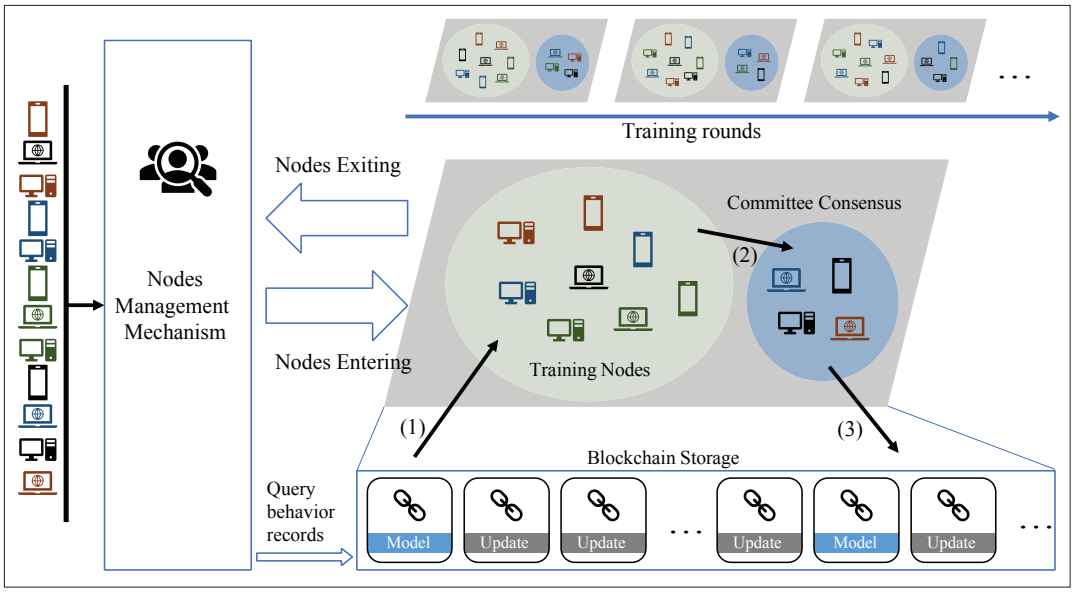

FIGURE 1. The training process of the proposed BFLC framework. (1) Training nodes acquire the newest global model and perform local training. (2) Training nodes send local updates to committee. (3) Committee validate the updates and record new models or updates onto the blockchain.

Through the literature review, it would be an effective way for blockchain to serve as decentralized storage and replace the central FL servers. However, the efficiency of consensus is in urgent need of improvement. Although storing models and updates in blockchain brings many security advantages, it is also a huge burden of the storage capacity on blockchain nodes. Therefore, how to reduce the consumption of a blockchain-based FL is also a key challenge.

In this article, we propose a decentralized, autonomous blockchain-based FL architecture to address these challenges (Fig. 1). The architecture based on the alliance chain provides the node permission control without a centralized server. In terms of storage, we design the storage pattern on the chain for models and updates, by which the nodes can quickly get the latest model. Each validated update is recorded and kept untampered on the blockchain. Considering the huge storage consumption on the blockchain, partial nodes can abandon the historical blocks to release the storage space. In terms of the block consensus mechanism, a novel *committee consensus* mechanism is proposed, which only increases a few validation consumptions and achieves more stability under malicious attacks. In each round of FL, updates are validated and packaged by a small number of nodes (i.e., the *committee*). The mechanism allows most honest nodes to reinforce each other and continuously improve the global model. A small number of incorrect or malicious updates will be ignored to avoid damaging the model. In the meantime, the BFLC training community is flexible, where the nodes can join or leave at any time without damaging the training process. Combined with an effective incentive mechanism, the nodes who contribute can gain actual rewards, thus promoting the development of the whole training community in a virtuous circle.

Our contributions are summarized as follows:

- We propose a blockchain-based FL framework BFLC, which defines the model storage patterns, the training process, and a novel committee consensus in detail.
- We technically discuss the scalability of BFLC, including the node management in the community, the analysis of malicious node attacks, and the storage optimization.
- We demonstrate the effectiveness of BFLC by experiments on a real-world FL dataset. We also verify the security by simulating malicious attacks.

Related Work

Konecný *et al*. proposed Federated Learning, whose goal is to train a high-quality centralized model while training data remains distributed over a large number of clients [1]. The network situation of FL is unreliable and relatively slow, and the clients are not always online. In these years, FL is applied in many scenarios like video analysis, information inspection and classification, and credit card fraud detection, while keeping the personal data sensitivity safe. The theoretical studies of convergence, network latency, or malicious attacks on FL are also active fields.

The centralized federated server has been challenged and questioned increasingly in these years. It is a natural thought to keep the concept of server at a minimum or even avoiding it completely. The study in [5] assumed that the data remains at the edge devices, but it requires no aggregation server or any central component. Hu *et al*. [6] proposed a segmented gossip approach, which makes full utilization of node-to-node bandwidth then can achieve a convergence efficiently.

Meanwhile, decentralization is the most direct way to avoid the above risks. Blockchain, a distributed ledger technique, can store the historical operations and keep it tamper-resistant. With the aim of the blockchain, collaborative machine learning methods can get rid of the centralized server and improve security. Blaz *et al*. [7] proposed a machine learning-based method to fasten the transaction signing process. Many real-world

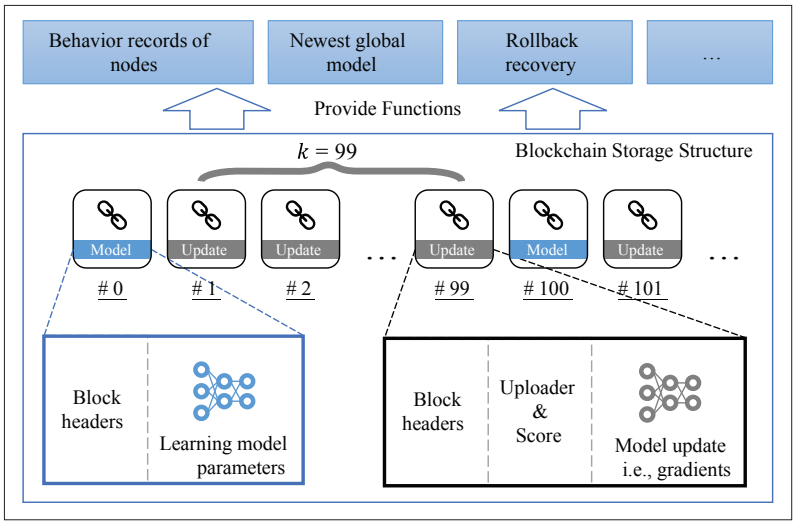

FIGURE 2. The FL storage structure on blockchain system, and the provided functions.

tasks are applying the blockchain-based FL method, such as Industrial Internet of Things, mobile edge computing, cognitive radio networks, and Internet of Vehicles.

It is reasonable to assume that the clients in FL might be malicious. Therefore, the local updates from all clients should be recorded under blockchain-based FL settings. You *et al*. [8] focused on the stability and convergence speed of FL, and proposed a blockchain-based method to address these challenges. Umer *et al*. [9] proposed a blockchain-based architecture, which can perform parallel learning for multiple global models. Bao *et al*. [10] proposed a public blockchain-based FL architecture, which provides trusty consensus based on nodes' data amount and historical performance.

These blockchain-based learning methods can effectively record the nodes' performance to reduce malicious attacks. However, there are still three main challenges.

Consensus Efficiency: It is an inevitable process for blockchain-based methods to reach a consensus for each packing block. Considering the vast amount of learning nodes, a broadcasting consensus is highly time-consuming. Therefore, reducing the consensus cost is non-trivial. One of the related works [10] selects a leader to execute the consensus. However, the criterion relies on many outer data.

Model Security: The framework should prevent the model from being exposed to unauthorized devices and from poisoning. The security of the system is rarely studied under blockchain-based FL settings.

Framework Scalability: When applying these training frameworks to real-world applications, we always need to add detailed rules to adapt to different scenarios. Therefore, the scalability of frameworks determines their scope of applications.

The Proposed Framework

Federated Learning (FL) enables the machine learning algorithms to train across multiple distributed clients without exchanging their data samples. In the original FL settings, one centralized server takes control of the training process, including client management, global model maintenance, and gradient aggregation. During each training round, the server broadcasts the current model to some participating nodes. After receiving the model, nodes locally update it with their local data and submit the update gradients to the server. The server then aggregates and applies the local gradients into the model for the next round.

The decentralized nature of blockchain can replace the central server. As aforementioned, the functions of the centralized server can be implemented by the Smart Contract (SC) instead, and be actuated by transactions on the blockchain. To tackle this vision, we propose BFLC, which is a Blockchain-based Federated Learning framework with Committee consensus. Without any centralized server, the participating nodes perform FL via blockchain, which maintains the global models and local updates. Considering the communication cost of FL, we leverage a novel delegated consensus mechanism to tackle the missions of gradient selection and block generation. In the following sub-sections, we will elaborate on the various components of the framework.

Blockchain Storage

To enable authority control, the storage of BFLC is an alliance blockchain system, and only the authorized devices can access the FL training contents. On the blockchain, we design two different blocks to store the global model and local update (Fig. 2), which are collectively known as learning information. For the sake of simplicity, we assume that only one learning information is placed in a block.

In the beginning, an initialized model was placed into the #0 block, then the 0-th round of training starts. Nodes access the current model and execute local training, and put the verified local gradients to new update blocks. When there are continuously enough update blocks, the smart contract triggers the aggregation, and a new model is generated and placed on the chain. We should note that the FL training only relies on the latest model block, and the historical block is stored for failure fallback and block verification.

We denote the number of required updates for each round as k, and denote the number of rounds as $t = 0, 1, ...$ Then we have: the # $t \times (k + 1)$ block contains the model of t-th round, which is called *model block*, and the # $[t \times (k + 1) + 1, (t + 1) \times (k + 1) - 1]$ blocks contain the updates of t-th rounds, which are called *update blocks*. From an implementation perspective, one model block should include: block headers, number of round t and global model, while one update block includes: block headers, number of round t, local update gradient, uploader address and update score.

Committee Consensus Mechanism

The chain structure of blockchain guarantees immutability. Therefore, appending the correct blocks to the chain is a crucial component which is the consensus mechanisms work for. The competition-based mechanisms append blocks on the chain first, whereafter the consensus meets.

Conversely, the communication-based generate mechanisms reach an agreement before appending blocks.

Considering the computation and communication cost of consensus, we propose an efficient and secure Committee Consensus Mechanism (CCM) to validate the local gradients *before* appending it to the chain. Under this setting, a few honest nodes will constitute a *committee* in charge of verification of local gradients and block generation. In the meantime, the rest of the nodes execute local training and send the local updates to the committee. The committee then validates the updates and assigns a score on them. Only the qualified updates will be packed onto the blockchain. At the beginning of the next round, a new committee is elected based on the scores of nodes in the previous round, which means that the committee will not be re-elected. It is noteworthy that the update validation is a pivotal component of the CCM, therefore, we describe a feasible approach: the committee members validate the local updates by treating their data as a validation set, and the validation accuracy becomes the score. This is the minimized approach that acquires no further operation of the committee, but only the basic ability to run the learning model. After combining the scores from the various committee members, the median will become the score of this update.

Working with this mechanism, BFLC can achieve these advantages:

High efficiency: only a few nodes will validate the updates, rather than broadcasting to every node and reach an agreement.

K-fold cross-validation: the committee members will not participate in the local training in the round. Therefore, the local data of the committee are taken as a validation set. As the alternating of committee members at each round, the validation set changes as well. In this setting, *k*-fold cross-validation on FL is achieved.

Anti-malevolence: based on the validation scores, the corresponding nodes with better performance will be elected by the smart contract and constitute the new committee for the next training round. That means the selected local data distribution is gregarious and the node is not malicious.

Model Training

Nodes other than committees perform local training each round. For security and privacy, raw data will be kept in nodes locally, and these nodes only upload the gradients to the blockchain. There are two main challenges:

- The local data distribution might be not Independent and Identically Distributed (non-IID).
- The devices are not always available.

To address the first challenge, the committee consensus mechanism could maximize the generalization ability of the global model by validating the local updates with committee members' data distribution [1]. To address the second one, only a certain number of local updates are requisite for each round, and we design an initiative local learning progress for nodes. Nodes can actively obtain the current global model at any time and perform local training. The gradients will be sent to the committee and be validated. When eligible updates are packaged on the blockchain, as a reward, tokens can be attached to nodes. We will discuss the incentive in the next section.

As aforementioned, a certain number of valid updates are required for each round. Therefore, when the committee validates enough local updates, the aggregation process is activated. These validated updates are aggregated by the committee into a new global model. The aggregation can be performed on the local gradients [11] or the local models [12], and the network transmission consumptions of these two methods are equal. After the new global model is packed on the blockchain, the committee will be elected again, and the next training round begins.

Discussion

Node Management and Incentive

The BFLC training process depends on the mutual promotion of nodes, and node management is also a key part of BFLC. The participant nodes can not only access the global model but can also upload updates to affect the model. To control permissions, we have designated the initial nodes that constitute the training community to be responsible for node management, that is, to be the managers. Each device must be verified by the managers before joining the training community. This verification is in blacklist mode: if the device has been kicked out of the community for misconduct (e.g., submitting misleading updates), the device will be rejected.

Depending on the proposed blockchain storage structure, the latest global model can be quickly found on the chain after new nodes joined. Nodes can immediately use the model to complete their local tasks, or they can train the model with local data and gain rewards. It is noteworthy that only a certain number of valid updates are required for aggregation at each round, and only part of the nodes are online to participate as well. Therefore, as long as the nodes actively submit updates, it is likely to participate in the global model training and gain rewards.

Nodes in a community can always use the model without committing updates, so an effective incentive is required to encourage nodes to train models. To address this problem, we propose an incentive mechanism called *profit sharing by contribution*.

Permission Fee: Each device should pay for the access permission of the global model, and these fees are kept by the managers. Nodes then have unlimited access to the latest models in the community.

Profit Sharing: After aggregation of each round, the managers distribute rewards to the corresponding nodes based on the scores of their submitted updates.

As a result, frequently providing updates could earn more rewards, and the constantly updated global model will attract more nodes to participate. This incentive mechanism has high scalability to adapt to different real-world applications and is worth studying.

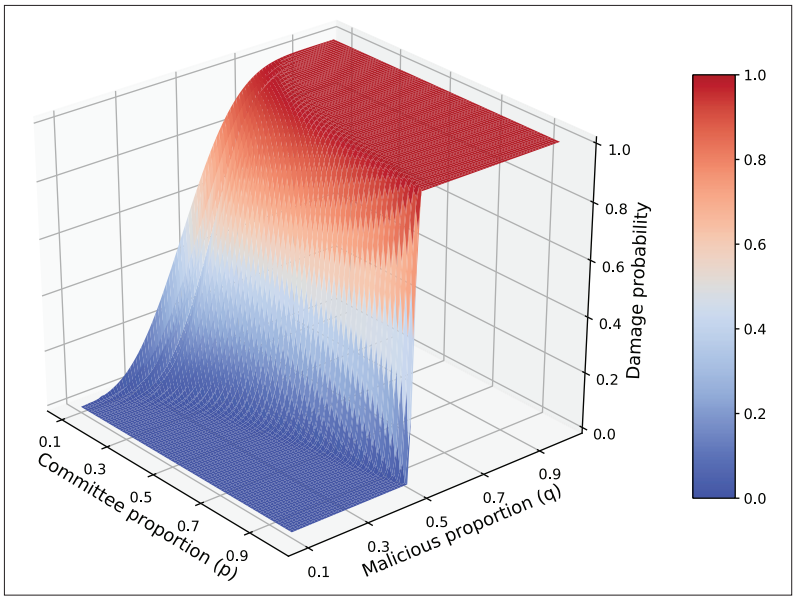

FIGURE 3. The attack success probability changing along with p and q.

Committee Election

At the end of each round, a new committee is elected from the providers of validated updates. In decentralized training settings, this election significantly affects the performance of the global model, because the committee decides which local updates will be aggregated. Committee election methods include the following categories.

Random Election: New committee members are randomly selected from validated nodes. From a machine learning perspective, this approach improves the generalization of the model and reduces overfitting. However, the resistance to malicious attacks is weak.

Election by Score: The providers with top validation scores constitute the new committee. This may exacerbate the uneven distribution of samples due to the absence of partial nodes in the committee. However, this approach significantly increases the cost of the attack and brings more security and stability.

Multi-Factor Optimization: This approach considers multiple factors of the devices (i.e., the network bandwidth) and the validation scores for optimal election. However, this optimization will bring additional computing overhead. Therefore, this approach should be applied depending on the scenarios and requirements.

Malicious Nodes

A malicious node is defined as a node submitting incorrect, malicious model updates. The original FedAvg [12] aggregates all the updates into a new global model. If there are malicious updates, the global model will be poisoned and obtains lower performance. As aforementioned, under the CCM, the updates will be verified by the committee before being aggregated. In this sub-section, we theoretically analyze the factors and the success possibility of malicious attacks.

We denote the amount of all nodes as N, in which the amount of committee members is M, and the remaining $N - M$ nodes are training nodes. Distinctly, a malicious update is accepted to the aggregation if and only if more than $M/2$ committee members are cooperating. However, the committee members are the M of the best performers at the last round, which means these malicious committee members' updates are accepted by other $M/2$ malicious nodes in the last committee. It is an infinite dependency loop, therefore, as long as there are more than $M/2$ honest nodes in the first committee, no malicious node could enter the committee and harm the global model.

Considering another extreme situation: the malicious nodes conspire together to earn the committee seats by pretending to be normal nodes. When the malicious nodes hold half of the seats, the attack begins. To analyze this attack mode, we denote the amount of participating nodes as A, the percentage of malicious nodes in A is $q \in (0, 1)$, and the percentage of the committee is $p \in (0, 1)$. The attack target is holding more than $(A \times p)/2$ seats in committee. We assume that the performance of each node is similar. Therefore, the attack success probability can be calculated as the probability of this event: extracting $A \times p$ nodes from A nodes, more than half of which come from $A \times q$. By fixing $A = 1000$, we plot the probability change along p and q in Fig. 3. We should note that only when the malicious percentage is greater than 50 percent, the attack success probability is remarkable. This conclusion is similar to the 51 percent attack in the Proof-of-Work blockchain system. In a decentralized community, the malicious nodes should hold 51 percent of the computational resources to attack the system, where the cost far outweighs the benefit. The historical models and updates are stored on the blockchain, therefore, fallback is always an option after the attack happened.

Storage Optimization

In real-world applications, storage overhead is an important factor that determines the hardware requirements for the training devices. Based on the above-mentioned blockchain storage scheme (Fig. 2), the latest model can be quickly accessed. Although historical models and updates can provide post-disaster recovery, they also occupy huge storage space. Here, we give a simple and feasible storage overhead reduction scheme: nodes with insufficient capacity can delete historical blocks locally, and only keep the latest model and updates of the current round. In this way, the problem of insufficient storage space can be eased, while the ability to recover and verify is retained on the core nodes. However, the shortcomings of this method are also obvious. The credibility of the blockchain decreases with the deletion. In a mutually distrusting community, each node may not use this scheme for security concerns.

Therefore, trusted and reliable third-party storage may be a better solution. The blockchain only maintains the network address where each model or updated file is located and records of modification operations. Other nodes interact with

the centralized storage to access the model and updates. This centralized storage will be responsible for disaster recovery backup and distributed file storage services.

Future Work

Transmission Efficiency: The storage and synchronization of the blockchain consume huge hardware resources, not only the hard disk space but also the network bandwidth. Therefore, how to reduce transmission consumption under the premise of ensuring the stability of model training is a topic worthy of study.

Public Scene: In this article, the alliance blockchain system takes care of the authentication tasks, but it also raises the threshold for joining the training community. How to establish a public community using a Proof-of-Work style consensus while resisting attacks from the malicious nodes is also an interesting topic.

Lightweight Training: For many IoT devices, the hardware conditions are usually not enough to train a deep neural network. Hence, how to reduce the complexity of model training (e.g., seeking help from edge servers), in the meantime ensuring privacy protection, is a worthy research topic.

Experimental

Settings and Normal Training

To demonstrate the effectiveness of the BFLC, we perform it on the real-world federated dataset FEMNIST [13]. This dataset contains 805263 samples and 3550 users for handwritten character image classification tasks and contains 62 different classes. Following the instruction of the dataset, we simulate 900 devices in the training community, where the local datasets are unbalanced in number and non-IID. We employ a blockchain system named FISCO (https://github.com/fisco-bcos) with PBFT consensus on an Intel Core CPU i9-9900X with a clock rate of 3.50 GHz with 10 cores and two threads per core. The SC layer was constructed by the C++ pre-compiled contracts. The learning model is written with Python 3.7.6 and Tensorflow 1.14 and is executed on Geforce RTX 2080Ti GPUs. We should mention that the 900 devices in the simulation are divided according to the original collection, in order to restore the realistic data distribution. The FISCO framework we use is an open-source blockchain system project, with convenient pre-compiled SC functions, which can quickly deploy machine learning algorithms.

We compare BFLC with the basic FL [12] framework and the stand-alone training framework as the baseline. Each framework performs the classic image classification model AlexNet [14] as the global model and fixed the hyper-parameters to ensure fairness. In terms of the experimental settings, we define the proportion of active nodes in each round as *k* percent, among which 40 percent will be elected as committee members in the next round for BFLC. The proportion of training nodes for Basic FL is also *k* percent. Meanwhile, stand-alone training will leverage the whole dataset. Under the conditions of different *k* values, we record their performance in Table 1.

Frameworks	Proportion *k* % of active nodes				
	10 %	20 %	30 %	40 %	50 %
BFLC	89.33 %	89.89 %	90.02 %	89.87 %	89.78 %
Basic FL	90.02 %	90.20 %	90.29 %	90.11 %	90.42 %
Stand-alone	91.34 %				

TABLE 1. Accuracy of BFLC, Basic FL and stand-alone on FEMINST dataset with different proportion of active nodes.

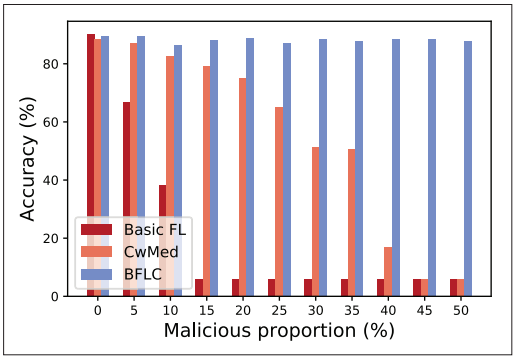

FIGURE 4. Performance of methods under malicious attacks.

In Table 1, with the proportion of active nodes increase, the performance of BFLC keeps approaching the effect of the basic FL framework and only has a slight loss compared to the stand-alone training with the intact dataset. BFLC can significantly reduce the consumption of consensus through the committee consensus mechanism rather than broadcasting. Compared with stand-alone training, BFLC also has the privacy protection of FL and requires no trusted central server to manage, which significantly reduces the risk of privacy leakage.

Under Malicious Attack

The malicious nodes in the training community will generate harmful updates, which will significantly reduce the performance of the global model if being integrated. In this sub-section, we simulate malicious node attacks to demonstrate how the proposed BFLC, basic FL, and CwMed [15] will be affected under different malicious proportions among active nodes. We assume that the attack mode of the malicious node is random perturbation with a pointwise Gaussian random noise.

The basic FL will not perform any defense measures, and model updates generated by randomly selected active nodes will be integrated. CwMed constructs a global gradient, where each entry is the median of entries in the local gradients with the same coordinate. BFLC relies on the committee consensus mentioned above to resist the attack. Each update will obtain a median score from the committee.

In order to enhance the effectiveness of the attack, we assume that malicious nodes are collusion, that is, members of the malicious committee will give random high scores (e.g., 90 percent) to the malicious updates. The proportion of active nodes is fixed as 10 percent, and

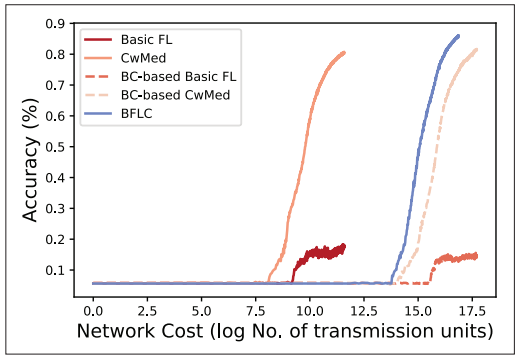

FIGURE 5. Transmission cost during the training process.

20 percent of them will be elected as the next committee. As shown in Fig. 4, the BFLC can resist a much higher malicious nodes proportion than the compared methods. This indicates the effectiveness of BFLC with the help of the committee mechanism.

Transmission Cost

Based on the experiment of malicious attacks, we calculate the cost of network transmission. Each network interaction, including the transmission of models and updates, transmits the full model size data (denoted as one *transmission unit*). In the original FL settings, only network traffic between servers and clients will occur. In decentralized settings, network transport occurs between clients (broadcasting by default).

After fixing the proportion of malicious nodes to 10 percent, we can obtain the relationship between the network cost and the model performance. In Fig. 5, we plot the logarithmic number of transmission units along the x-axis and the accuracy score along the y-axis. In summary, the decentralized FL settings increase the cost of network transmission with no doubt. However, the experimental results demonstrate the stability of BFLC by obtaining higher performance under malicious attacks. Furthermore, BFLC can also reduce part of the network cost compared with other decentralized methods, and fasten the convergence. Indeed, the decentralized federal learning method can avoid the risks from the central server, but the optimization of network transmission under the premise of ensuring training accuracy is a topic worthy of future research.

Conclusion

As aforementioned, the security of federated learning is facing challenges in many aspects, such as the model poisoning from malicious nodes and privacy leaking from a malicious server. Based on a trusted blockchain system, we propose BFLC, which is a decentralized, federated learning framework with the committee consensus. The consensus can effectively avoid the influence of malicious central servers or malicious nodes. In the experiment section, we verified the effectiveness and security of the BFLC framework by adopting a real-world dataset. We also discussed the scalability of BFLC, which has broad research prospects in security, data storage, and incentive mechanisms.

Acknowledgment

The work described in this article was supported by the National Key Research and Development Program (2016YFB1000101); the National Natural Science Foundation of China (11801595, 61722214); the Natural Science Foundation of Guangdong (2018A030310076, 2019A1515011043); and the CCF-Tencent Open Fund WeBank Special Funding. The corresponding author is Chuan Chen (chenchuan@mail.sysu.edu.cn).

References

[1] J. Konecný *et al.*, "Federated Learning: Strategies for Improving Communication Efficiency," CoRR, vol. abs/1610.05492, 2016, available: http://arxiv.org/abs/1610.05492.
[2] P. Ramanan, K. Nakayama, and R. Sharma, "BAFFLE : Blockchain Based Aggregator Free Federated Learning," CoRR, vol. abs/1909.07452, 2019, available: http://arxiv.org/abs/1909.07452.
[3] S. Zhou *et al.*, "PIRATE: A Blockchain-Based Secure Framework of Distributed Machine Learning in 5G Networks," CoRR, vol. abs/1912.07860, 2019, available: http://arxiv.org/abs/1912.07860.
[4] X. Chen *et al.*, "When Machine Learning Meets Blockchain: A Decentralized, Privacy-Preserving and Secure Design," *Proc. 2018 IEEE Int'l. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1178–87.
[5] I. Hegedüs, G. Danner, and M. Jelasity, "Gossip Learning as a Decentralized Alternative to Federated Learning," *Proc. 2019 Int'l. Conf. Distributed Applications and Interoperable Systems — 19th IFIP WG 6.1*, DAIS 2019, held as part of the 14th Int'l Federated Conf. Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17–21, 2019, pp. 74–90, available: https://doi.org/10.1007/978-3-030-22496-7_5.
[6] C. Hu, J. Jiang, and Z. Wang, "Decentralized Federated Learning: A Segmented Gossip Approach," CoRR, vol. abs/1908.07782, 2019, available: http://arxiv.org/abs/1908.07782.
[7] B. Podgorelec, M. Turkanovic, and S. Karakatic, "A Machine Learning-Based Method for Automated Blockchain Transaction Signing Including Personalized Anomaly Detection," *Sensors*, vol. 20, no. 1, 2020, available: https://doi.org/10.3390/s20010147, p. 147.
[8] Y. J. Kim and C. S. Hong, "Blockchain-Based Node-Aware Dynamic Weighting Methods for Improving Federated Learning Performance," *Proc. 20th Asia-Pacific Network Operations and Management Symposium*, APNOMS 2019, Matsue, Japan, Sept. 18–20, IEEE, 2019, pp. 1–4, available: https://doi.org/10.23919/APNOMS.2019.8893114.
[9] U. Majeed and C. S. Hong, "FLchain: Federated Learning via MEC-Enabled Blockchain Network," *Proc. 20th Asia-Pacific Network Operations and Management Symposium*, APNOMS 2019, Matsue, Japan, Sept. 18–20, IEEE, 2019, pp. 1–4, available: https://doi.org/10.23919/APNOMS.2019.8892848.
[10] X. Bao *et al.*, "FLchain: A Blockchain for Auditable Federated Learning with Trust and Incentive," *Proc. 2019 5th Int'l. Conf. Big Data Computing and Communications*, BIGCOM 2019, QingDao, China, Aug. 9–11, 2019, pp. 151–59, available: https://doi.org/10.1109/BIGCOM.2019.00030.
[11] R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," *Proc. 22nd ACM SIGSAC Conf. Computer and Commun. Security*, Denver, CO, USA, Oct. 12–16, 2015, I. Ray, N. Li, and C. Kruegel, Eds. ACM, 2015, pp. 1310–21, available: https://doi.org/10.1145/2810103.2813687.
[12] B. McMahan *et al.*, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proc. 20th Int'l. Conf. Artificial Intelligence and Statistics*, AISTATS 2017, 20–22 April 2017, Fort Lauderdale, FL, USA, ser. Proc. Machine Learning Research, A. Singh and X. J. Zhu, Eds., vol. 54. PMLR, 2017, pp. 1273–82, available: http://proceedings.mlr.press/v54/mcmahan17a.html.
[13] S. Caldas *et al.*, "LEAF: A Benchmark for Federated Settings," CoRR, vol. abs/1812.01097, 2018, available: http://arxiv.org/abs/1812.01097.
[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, no. 6, 2017, pp. 84–90, available: http://doi.acm.org/10.1145/3065386.

[15] D. Yin *et al.*, "Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates," *Proc. 35th Int'l. Conf. Machine Learning*, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018, ser. *Proc. Machine Learning Research*, J. G. Dy and A. Krause, Eds., vol. 80, PMLR, 2018, pp. 5636–45, available: http://proceedings.mlr.press/v80/yin18a.html

Biographies

YUZHENG LI received the B.S. degree from the Sun Yat-sen University, Guangdong, China, in 2018. He is currently pursuing an M.S. degree with the School of Data and Computer Science, Sun Yat-sen University. His current research interests include federated learning, blockchain, statistical machine learning, multi-view learning, and optimization.

CHUAN CHEN received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2012, and the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2016. He is currently an associate research fellow with the School of Data and Computer Science, Sun Yat-sen University. His current research interests include blockchain, machine learning, numerical linear algebra, and numerical optimization.

NAN LIU received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2019. He is currently pursuing an M.S. degree with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His research interests include federated learning, blockchain, and network representation learning.

HUAWEI HUANG received his Ph.D. in computer science and engineering from the University of Aizu, Japan. He is currently an associate professor with the School of Data and Computer Science, Sun Yat-Sen University, China. His research interests mainly include distributed learning and blockchain. He has served as a visiting scholar with the Hong Kong Polytechnic University (2017–2018); a post-doctoral research fellow of JSPS (2016–2018); and an assistant professor with Kyoto University, Japan (2018–2019).

ZIBING ZHENG received his Ph.D. degree from the Chinese University of Hong Kong in 2011. He is currently a professor in the School of Data and Computer Science at Sun Yat-sen University, China. He has published over 120 international journal and conference papers, including three ESI highly cited papers. According to Google Scholar, his papers have more than 9,100 citations, with an II-index of 46. His research interests include blockchain services computing, software engineering, and financial big data.

QIANG YAN is currently the blockchain scientist at WeBank. He received his Ph.D. degree in information systems from Singapore Management University in 2013. Before joining WeBank, he was the tech lead of the privacy infrastructure team at Google Switzerland. His research interests include applied security and privacy for blockchain, AI, mobile systems, social networks, human factors in security system design, and applied cryptography.