Communication-efficient and Scalable Decentralized Federated Edge Learning

Austine Zong Han Yapp[1] , **Hong Soo Nicholas Koh**[1] , **Yan Ting Lai**[1] , **Jiawen Kang**[1] , **Xuandi Li**[1] , **Jer Shyuan Ng**[2] , **Hongchao Jiang**[2] , **Wei Yang Bryan Lim**[2] , **Zehui Xiong**[3] , **Dusit Niyato**[1]
[1]School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore
[2]Alibaba-NTU Singapore Joint Research Institute (JRI), NTU, Singapore
[3]Singapore University of Technology and Design (SUTD), Singapore
{aust005, kohh0064, laiy0028, s190068, hongchao001, limw0201}@e.ntu.edu.sg, {kavinkang, cindy.li, dniyato}@ntu.edu.sg, zehui_xiong@sutd.edu.sg

Abstract

Federated Edge Learning (FEL) is a distributed Machine Learning (ML) framework for collaborative training on edge devices. FEL improves data privacy over traditional centralized ML model training by keeping data on the devices and only sending local model updates to a central coordinator for aggregation. However, challenges still remain in existing FEL architectures where there is high communication overhead between edge devices and the coordinator. In this paper, we present a working prototype of blockchain-empowered and communication-efficient FEL framework, which enhances the security and scalability towards large-scale implementation of FEL.

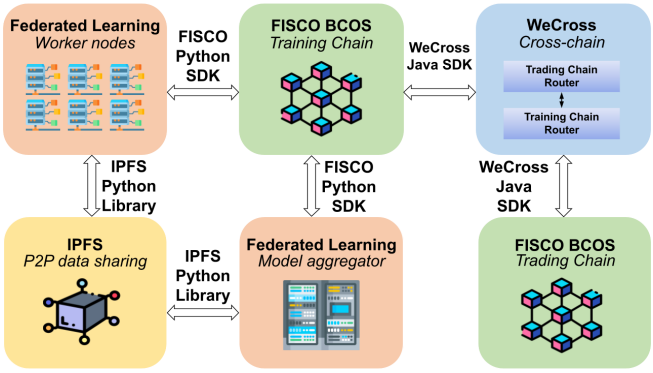

Figure 1: Overview of the System Architecture

1 Introduction

With the advancement of Artificial intelligence (AI) and the enhanced perception capabilities of the Internet of Things (IoT), the wealth of data collected at the edge of the network can be efficiently leveraged on. Traditionally, Machine Learning (ML) algorithms require the collection and collation of training data to a centralized server where the model training is then conducted. In a typical IoT setting, this involves the collection of user-specific data from the individual edge devices, which may be sensitive in nature.

In light of growing privacy concerns, a distributed ML framework known as *Federated Edge Learning* (FEL) has recently been proposed [McMahan *et al.*, 2017]. In FEL, ML model training is conducted collaboratively by the edge devices. Only the local model updates, instead of raw data, are transmitted to a central server for aggregation. Such an approach allows sensitive user data to remain on-device, therefore introducing an additional layer of data privacy. One such implementation of FEL is in the development of next-word-prediction models for Google's Gboard [Hard *et al.*, 2018].

Despite the benefits that existing implementations of FEL can provide, its applications are still limited due to two critical problems. Firstly, the central server that serves to aggregate the local model updates suffers from a single point of failure, posing a security and reliability risk. If the central server is compromised, the global model training may be misdirected or completely halted in the case of a system-wide crash. Secondly, existing implementations of FEL impose significant communication overheads during the exchange of model updates between the edge devices and the central server. This results in high demand for network bandwidth which may limit the scalability of FEL [Lim *et al.*, 2020].

To tackle these problems, we present a working prototype[1] of the blockchain-empowered FEL (BFEL) framework [Kang *et al.*, 2020]. In BFEL (Fig. 1), communication-efficient FEL is implemented at the edge of the network. The model parameters contributed by the edge devices are stored and managed in a decentralized manner by using consortium blockchains. The multi-blockchain system design mitigates training disruptions that may result from a single point of failure, and enhances the security and scalability of FEL. A model trading blockchain is also developed to facilitate a marketplace for the trading of global models (Fig. 2).

2 System Design

The system architecture (Fig. 1) gives an overview of BFEL. The architecture has a multi-blockchain framework, in which the FEL worker nodes and model aggregators for each task are able to communicate with the FISCO BCOS consortium blockchain using the FISCO Python software development kit (SDK), and with the InterPlanetary File System (IPFS) for data storage using the IPFS Python library [Naz *et al.*, 2019].

[1]Video demo: www.youtube.com/watch?v=zbzVqvIrmrE

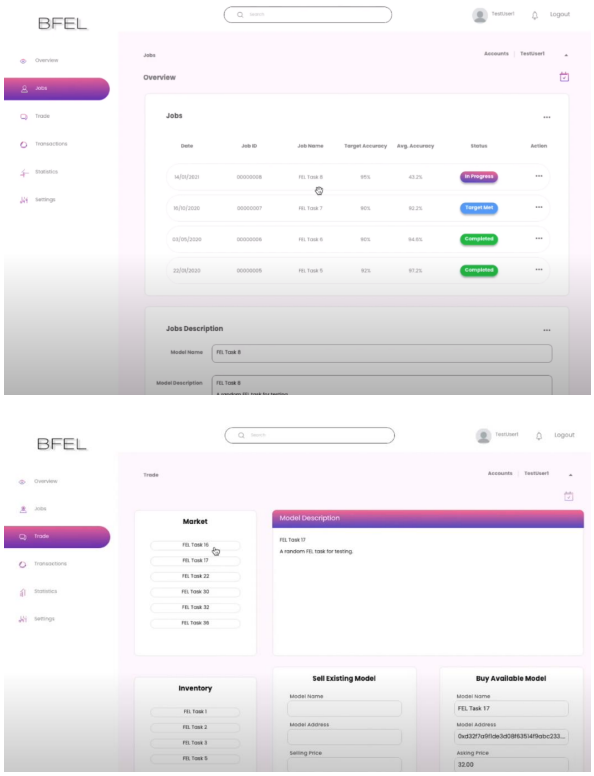

Figure 2: BFEL User Interface (UI)

The cross-chain platform named WeCross is used to bridge task data communication from the model training chain to the model trading chain. The three functions are:

1. **Communication Efficient FEL**: In the distributed edge computing system, a task publisher aims to train a FEL model. The task requirement is broadcasted to edge devices with suitable data. The edge devices may then join the FEL task as workers to collaboratively train a global FEL model using their local datasets. To increase communication efficiency, a gradient compression scheme is introduced between each iteration of training to reduce the memory footprint of each localized model update. The compressed model update with sparse gradients are transmitted to the model training blockchain for model update and verification. At the beginning of each training iteration, the workers in each model training blockchain read their updates and calculates a new global model after searching the model updates in their task groups. Given the new global FEL model, the workers perform training on their local datasets again. A number of iterations are performed until the accuracy desired by the task publisher is achieved. Upon completion of the FEL training, the final FEL global model is sent to the task publisher (i.e., model owner), which rewards the participating workers.
2. **Model Training Blockchain**: Instead of a traditional central aggregating server, a blockchain serves as the decentralized ledger to which model updates from edge devices are uploaded, aggregated, and stored on. Each task publisher initializes a high-performance consortium blockchain-based model training blockchain using the open-source FISCO BCOS platform to manage model updates from its workers and model aggregator. To improve the storage performance of the training blockchains and overcome the problem of "state bloat" where the size of the chain increases in an unsustainable manner, the full global model and worker gradient updates are stored in an off-chain manner using IPFS.
3. **Model Trading Blockchain**: The finalized ML models may be exchanged and shared among various entities. For example, a map company (acting as a task publisher) launches a traffic flow prediction task among vehicles (workers) and sells the trained model to other companies that have navigation demands (Fig. 2). Each trading record will be added into the main trading blockchain supported by data input from the model training blockchains.

3 AI and Blockchain Engine

In order to minimize the overall communication overhead between edge devices and the model training blockchain for each training iteration, the Deep Gradient Compression (DGC) [Lin *et al.*, 2017] scheme is adopted. The DGC scheme sends only gradient updates that are larger than a magnitude threshold through sparse updates, while accumulating the rest of the gradients locally until they are large enough to be transmitted. We improve the performance of DGC using several techniques: (i) momentum correction, (ii) local gradient clipping, (iii) momentum factor masking, and (iv) warm-up training [He *et al.*, 2019]. Momentum correction and local gradient clipping improves the local gradient accumulation while momentum factor masking and warm-up training reduces the staleness of delayed gradient updates. By incorporating these techniques, we able to improve compression ratio by more than 50$\times$ without any significant loss of accuracy.

Coupled with the DGC scheme, the model training blockchain is implemented on the FISCO BCOS platform to benefit from enhanced performance and privacy through access rights controls. In order to improve the training performance and reduce the total amount of data stored on the model training blockchain, IPFS is used to store intermediary model updates from worker nodes in an offline manner. The content identifier hash and worker metadata for each update are published on the blockchain via smart contracts for reference and retrieval purposes. An overview is shown in Fig. 3(a). After initialization of the global model and task parameters by the task publisher, the worker nodes will download the initial global model from IPFS and begin the iterative model training process by repeating steps 1 - 8 as illustrated in Figs. 3(b) and 3(c).

The FISCO BCOS platform is also natively supported by the WeCross platform, which allows direct model trading through cross-chain communication. To allow cross-chain communication to be possible, WeCross uses the concept of abstraction to make a unified "language" for blockchains. The abstract system has 4 layers: governance, transaction,

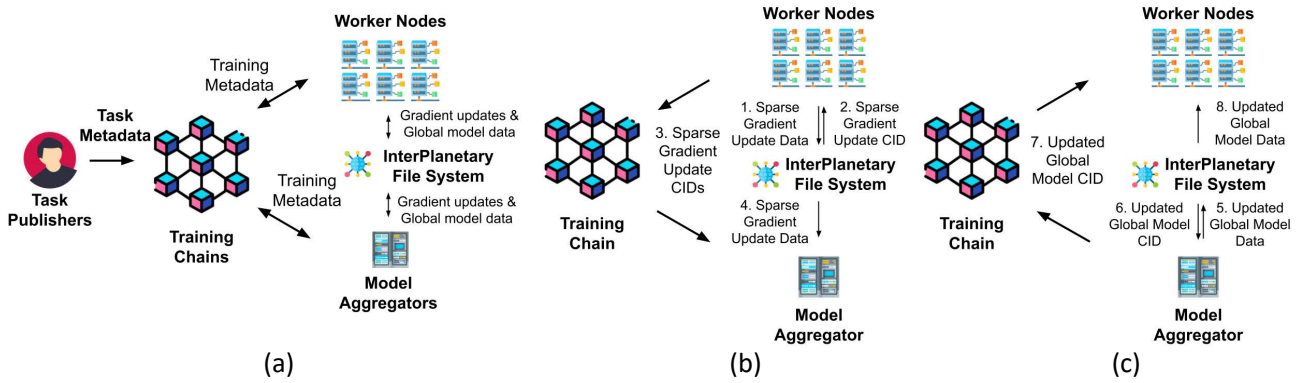

Figure 3: BFEL Training Chain + IPFS System Architecture and Workflow

interaction and data layer, as shown in the Fig. 4.

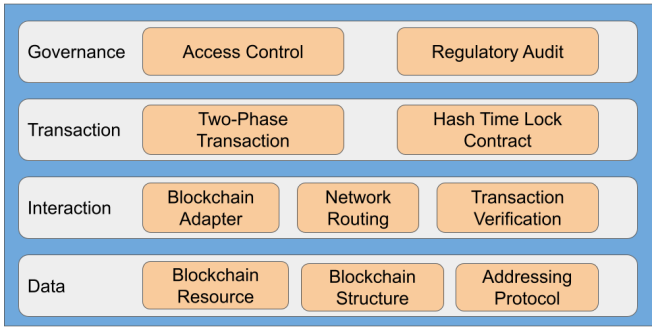

Figure 4: WeCross Abstract System Architecture

The governance layer handles the overall management of the WeCross server, including the access control and information on the blockchain. The transaction layer enacts the protocols for cross-chain transactions in the server. For this framework, 2PC is used to allow corrective actions as a transaction that can be rolled-back after execution. 2PC also allows multiple transactions to be done simultaneously.

The interaction layer handles the routing of transactions and universal blockchain adaptation for the blockchain. Transactions are also verified in this layer. As WeCross has a Universal Blockchain Interface (UBI), routing is done easily by using unique identifiers such as blockchain name and resource name to determine target location. The data layer handles information, such as the resources in the blockchain, the structure of the blockchain and the addressing protocol to be used. As most blockchains have common transaction block structure, abstract blocks can use these information to simplify the creation of transaction blocks.

Fig. 5 illustrates the WeCross system overview of the BFEL framework. Each router in the WeCross Server is connected to a blockchain and an Account Manager server is connected with the WeCross server for easy account management. The users can also connect to any router and access any blockchain from the router that they are connected to. This

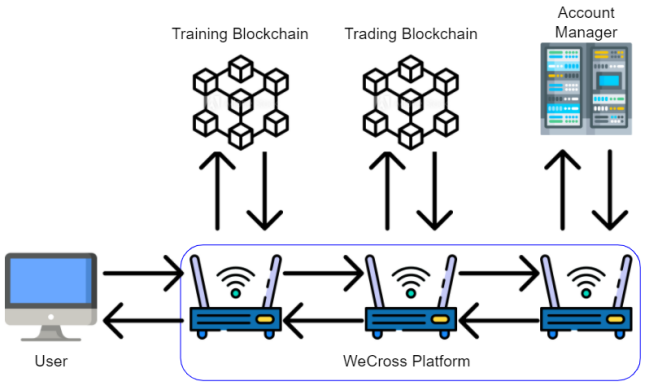

Figure 5: WeCross System Overview for BFEL Framework

is possible as WeCross uses a Heterogeneous Chain Inter-Protocol (HIP), which allows a universal access paradigm and follows a cross-chain interaction model that controls the routing of transactions. WeCross also has Java SDK support which helps in developing the BFEL framework.

4 Conclusions and Future Work

The BFEL framework is a scalable implementation of distributed edge computing system which mitigates the challenges of FEL. User access is carefully managed through the use of a consortium blockchain which keeps the training models secure and protected from privacy attacks. In addition, the cross-chain communication in the BFEL framework allows greater flexibility and scalability by eliminating the need to store all the training and trading data on a single chain.

Acknowledgments

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore JRI, National Research Foundation, Singapore, under AI Singapore Programme (AISG Award No: AISG-GC-2019-003), WASP/NTU grant M4082187(4080), SUTD SRG-ISTD-2021-165, and Ministry of Education (MOE) Tier 1 (RG16/20).

References

[Hard *et al.*, 2018] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[He *et al.*, 2019] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019.

[Kang *et al.*, 2020] Jiawen Kang, Zehui Xiong, Chunxiao Jiang, Yi Liu, Song Guo, Yang Zhang, Dusit Niyato, Cyril Leung, and Chunyan Miao. Scalable and communication-efficient decentralized federated edge learning with multi-blockchain framework. In *International Conference on Blockchain and Trustworthy Systems*, pages 152–165. Springer, 2020.

[Lim *et al.*, 2020] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

[Lin *et al.*, 2017] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[Naz *et al.*, 2019] Muqaddas Naz, Fahad A Al-zahrani, Rabiya Khalid, Nadeem Javaid, Ali Mustafa Qamar, Muhammad Khalil Afzal, and Muhammad Shafiq. A secure data sharing platform using blockchain and interplanetary file system. *Sustainability*, 11(24):7054, 2019.