FLchain: Federated Learning via MEC-enabled Blockchain Network

Umer Majeed, and Choong Seon Hong
Department of Computer Science and Engineering, Kyung Hee University, 17104, Republic of Korea
Email: {umermajeed, cshong}@khu.ac.kr

***Abstract*—In this paper, we propose blockchain network based architecture called "FLchain" for enhancing security of Federated Learning (FL). We leverage the concept of channels for learning multiple global models on FLchain. Local model parameters for each global iteration are stored as a block on the channel-specific ledger. We introduce the notion of "the global model state trie" which is stored and updated on the blockchain network based on the aggregation of local model updates collected from mobile devices. Qualitative evaluation shows that FLchain is more robust than traditional FL schemes as it ensures provenance and maintains auditable aspects of FL model in an immutable manner.**

***Index Terms*—Blockchain, distributed computing, federated learning, multi-access edge computing.**

I. Introduction

Machine Learning (ML) is often applied on relevant user data to enhance underlying services. Conventional ML schemes require aggregation of training data on centralized cloud, which raises concerns for the privacy and misuse of the users personal data [1]. Federated Learning (FL) is a cooperative approach to distributed ML. The privacy of user data in FL remains intact as none of the raw data is transferred out of user device. Thus, FL is the road towards privacy preserving development of learning models.

In traditional FL, mobile devices compute their local model update based upon on-device data samples and send it to a central server. The central server aggregates the local model updates received from different devices, and updates the global model. The updated global model is fetched by mobile devices to compute their next revision of local models [2]. The cycle continues until the desired accuracy is achieved at the central server. The drawback of this approach is a complete dependency on the reliability of a central server for storage and computation of the global model update. Any malicious activity leads to flawed global model update which is detrimental for accuracy of subsequent local model updates, thereby the entire FL process becomes erroneous.

Blockchain has emerged as a chronological, decentralized, provenance-preserving, and immutable ledger technology [3]. It is an effective solution to replace the attack-prone central server in an insecure environment. To mitigate the security issues involving a central server in FL, blockchain can be integrated with FL.

The rest of the paper is organized as follows. Section II analyzes the literature review of blockchain enabled FL. In Section III, we briefly explained preliminaries related to the proposed work. The system model is devised in Section IV. Section V enumerates the detailed operational specification of FLchain. Section VI presents an evaluation of FLchain, and Section VII concludes the study.

II. Related Work

This section discusses the recent efforts made to improvise FL over blockchain networks.

Coupling of blockchain and FL ensures the privacy of the users data by proposing an ultra-practical scheme for the training of robust decentralized learning models. The trained learning model parameters can securely be stored on the blockchain in an immutable manner with the fool-proof resistance against unauthorized access and malicious actions. Moreover, the blockchain securely preserves the provenance and chronological aspects of learning models [4].

In [5], authors discussed the coalition of FL with blockchain. The collection of training data to a centralized server from geographically dispersed sites is prone to cyber-attacks, privacy leakage, and network delay. Blockchain provides a secure way to exchange learning model parameters for the FL procedure. Blockchain enables auditing of learning models for each epoch of the global model in FL. Moreover, the performance of blockchain based FL is found to be almost comparable with stand-alone FL [6].

Kim *et al.* in [7] proposed on-device FL architecture over blockchain (BlockFL). The local model updates are performed on data samples available on user devices. The local model updates are accumulated in blocks on the blockchain. The global model updates are also calculated by user devices from the latest block, thus the notion of on-device FL is established. They consider the scalability, robustness and latency minimization of the global learning model. The model assumes that all the participating devices submit local model updates to blockchain network within a specified waiting time T_{wait}. The availability of local model updates from all partaking devices in a scheduled time is practically infeasible due to user mobility, network delay, power issues, and intermittently availability problems.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea goverment(MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing). *Dr. CS Hong is the corresponding author.

III. PRELIMINARIES AND DEFINITIONS

This section briefly explains preliminaries of the proposed architecture and its operation.

A. Channel

Fabric introduces the concept of channels which are private subnet used for enabling isolated communication between atleast two peers. Only channel-associated peers are entitled to read, submit, and validate the transaction within a channel. A separate ledger is maintained for each channel. The consensus is also applied on per-channel basis [8]. In FLchain, for each global learning model, a new channel with the genesis block is created which stores channel-specific ledger. The genesis block stores the initial weights of the global learning model, dimensions of weights, hyper-parameters, activation function, and bias.

B. Global Model State Trie

Similar to the "Account State Trie" for pursuing the state of accounts in Ethereum [9], we propose "Global Model State Trie" for pursuing the weights of the global learning model in FLchain. Each channel has its own global model state trie in the form of Merkle Patricia tree. The global model state trie stores the weights in key-value pair where the key is weight location (subscripts indicating indexes of weight) and value is actual weight coefficients. The weight coefficients are updated concurrently with the generation of a block in FLchain. After the consensus, the global model state trie provides the updated weight coefficients for global learning model.

IV. SYSTEM MODEL

In this section, we proposed a system model for FL via integrated Multi-access edge computing (MEC) and blockchain network. The physical infrastructure of FLchain as shown in Fig. 1 consists of mobile and edge devices. Mobile devices compute the local model updates with on-device data samples. The edge devices serve two purposes. First, they provide network resources to the resource-constraint mobile devices. Second, they serve as nodes in the blockchain network of FLchain.

Each global model M_j is trained on separate channel. The set of available channels at FLchian is denoted by $C \triangleq \{1, 2, 3, \ldots, C_n\}$. Where, C_n are total number of available channels on FLchain and D_j indicates the number of devices registered at channel $j \in C$. The blockchain network, which consists of edge devices, stores the local model updates from user devices in the form of blocks on separate blockchain for a specific channel. The blockchain network also computes and securely stores global model updates in Merkle Patricia Tree on channel-specific ledger. Fig. 2 shows the simplified architecture of blockchain for FL for a given channel. The underlying blockchain platform for FLchain should be custom developed having features from Hyperledger Fabric [10] and Ethereum [11].

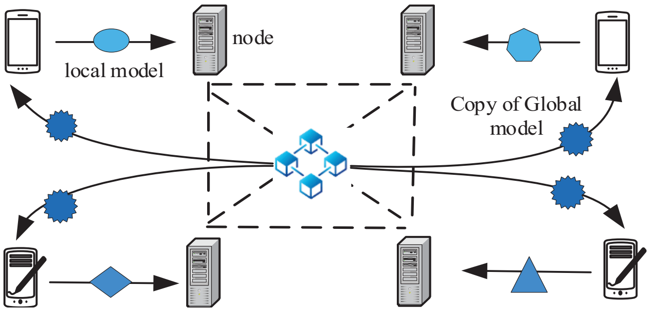

Fig. 1. Federated Learning via MEC-enabled Blockchain network

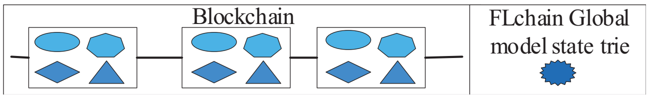

Fig. 2. FLchain: Simplified Blockchain Architecture for Federated Learning

V. BLOCKCHAIN OPERATIONS IN FLCHAIN

In this section, we describe the operation of FLchain. Algorithm 1 shows the overall procedure of FLchain for channel j. Fig. 3 shows the sequence diagram of operations of FLchain for device i and channel j.

A. Initialization

A new channel is created for training of each new global learning model. The initial weight parameters and other necessary configurations are set and stored in genesis block. Let j be the channel for a new global learning model M_j under consideration.

FLchain can be applied to any learning model. However, we consider global learning model M_j as a linear regression problem. Let D_j be the set of the relevant devices for channel j. $S_{i,j}$ is the set of data sample at device i for channel j. Then, $S_j = \bigcup_{i \in D_j} S_{i,j}$ with $|S_j| = N_{S,j}$. The learning objective is to minimize the loss function $L(w)$ over all the data samples $s_z \in S_j$ with $s_z = \{x_z, y_z\}$, where $x_z \in \Re^d$ and $y_z \in \Re$. The loss function is minimized by finding optimal weight parameters w_j^*, where $w_j \in \Re^d$ is the d-dimensional column vector and indicates global model weight vector for channel j.

$$w_j^* = \underset{w_j \in \Re^d}{\operatorname{argmin}} L(w_j), \tag{1}$$

$$\text{where} \quad L(w_j) \overset{\text{def}}{=} \frac{1}{N_{S,j}} \sum_{i \in D_j} \sum_{s_z \in S_{i,j}} l_z(w_j), \tag{2}$$

$$\text{and} \quad l_z(w_j) \triangleq l_z(w_j, x_z, y_z) = \frac{1}{2}\|y_z - w_j^T x_z\|^2. \tag{3}$$

The preliminary weight parameters at global iteration $t = 0$ are randomly chosen from pre-selected range. The global weight $w_j(0)$ and local weight parameters $w_{i,j}(0)$ for device i: $w_j(0), w_{i,j}(0) \in [0, w_{j,max}]$ and global gradient $\nabla l(w_j(0)) \in (0, 1]$. The global gradient for loss function is defined as:

$$\nabla L(w_j) = \frac{1}{N_{S,j}} \sum_{i \in D_j} \sum_{s_z \in S_{i,j}} \nabla l_z(w_j). \tag{4}$$

B. Channel Inquiry

When a device i wants to join the FLchain for a specific channel, first it carries out the channel inquiry; subsequently, the list of available channels C is sent to the mobile device by blockchain network.

C. Channel selection

When a device i wants to contribute to FL for a particular global model, It performs channel inquiry. Once, the list of accessible channels C is available, It selects the relevant channel c_i for that particular global model. Let the channel selected by device i be $c_i = j$.

D. Device Registration

If the device i is not already registered, the device need to register itself for its apriori selected channel. After the registration, the user device is assigned the private & public keys through which it can submit its revised local model weights to the channel. $Pri_{i,j}$ and $Pub_{i,j}$ denote the assigned private key and public key for device i and channel j, respectively.

E. Local Model Update

After completing the device registration procedure, or receiving notification to compute next update of local model, the device downloads the most recent global model parameter $w_j(t-1)$ from blockchain network through its associated edge node. Whereas, t is the the current global model iteration which needs to be computed at channel j. For each global model iteration t, the local model at device D_i is updated for V epochs. At epoch v of local model, the local model for device i is updated by stochastic variance reduced gradient (SVRG) as [12]:

$$w_{i,j}^{v}(t) = w_{i,j}^{v-1}(t) - \eta\nabla\Phi, \tag{5}$$

$$\nabla\Phi = \Big[\nabla l_z(w_{i,j}^{v-1}(t)) - \nabla l_z(w_j(t)) + \nabla l(w_j(t))\Big], \tag{6}$$

where $\eta > 0$ is step-size and after V local epochs we have $w_{i,j}(t) = w_{i,j}^{V}(t)$. The local model update $w_{i,j}(t)$ is determined by the device i and is forwarded to blockchain network in form of transaction. $trans_{i,j}(t)$ is the transaction generated by device i for channel j at iteration t and signed by the device private key $Pri_{i,j}$. The transaction data consists of $\Big(w_{i,j}(t), \{\nabla l_z(w_j(t))\}_{s_z \in S_{i,j}}\Big)$.

F. Transaction Pool

The submitted transactions are accumulated in the transaction pool (mempool). In particular, each node in blockchain network keeps its own channel-specific mempool. The transactions are validated, verified and authenticated. The peers on the channel j wait for time $T_{wait,j}$ for accumulation of transactions in mempool for every global model iteration.

There maybe transactions in mempool which are belated due to network latency. The transactions which were originally intended for being ensemble in previous global model updates are not usable for computation of next global model updates, thereby these transactions are discarded.

Algorithm 1 : FLchain operation for channel j

```
Setup channel $j$ for global model $M_j$
initialization: $t = 0$; $w_j(0), w_{i,j}(0) \in [0, w_{j,max}]$;
for all  $i \in D_j$ do in parallel
    Inquire available channels
    Select channel $c_i = j \in\ C \triangleq \{1, 2, 3, \ldots, C_n\}$
    Register device $i$ to channel $j$
end for
while $\|w_j(t) - w_j(t-1)\|_2 \le \varepsilon_{threshold,j}$ do
    for all  $i \in D_j$ do in parallel
        Download $w_j(t)$ from channel $j$ to device $i$
        $t \leftarrow t + 1$; $w_{i,j}^{0}(t) = w_j(t)$;
        for $v = 1, ..., V$ do
            $w_{i,j}^{v}(t) = w_{i,j}^{v-1}(t) - \eta\nabla\Phi$, and Eq. (6)
        end for
        $w_{i,j}(t) = w_{i,j}^{V}(t)$, Generate $trans_{i,j}(t)$ and forward to blockchain network
        Wait for notification from channel $j$
    end for
    Calculate $w_j(t)$ using Eq. (7)
    global model state trie updation, block generation and consensus
end while
```

G. Global Model Update

When the waiting time $T_{wait,j}$ is surpassed, Peers (Edge nodes) in channel j compete to generate the next block by bundling transactions for iteration t from their own mempool. Global model state trie is determined which securely stores the global model parameters $w_j(t)$. The root of global model state trie is added in block header of block with block-height t. The global model weights are updated using distributed approximate Newton-type Method (DANE) [13] as:

$$w_j(t) = w_j(t-1) + \sum_{k \in D_{j,t}} \frac{N_{k,t,j}}{N_{S,t,j}} \Big(w_{k,j}(t) - w_j(t-1)\Big), \tag{7}$$

where, $D_{j,t}$ are set of devices whose transactions are received by the winner miner and included in block with block-height t. $N_{k,t,j}$ are number of data samples contributed by device $k \in D_{j,t}$ at iteration t. Moreover, $S_{t,j} = \bigcup_{i \in D_{j,t}} S_{i,t,j}$ with $\big|S_{t,j}\big| = N_{S,t,j}$. Where, $S_{i,t,j}$ is set of samples contributed by device i at iteration t for channel j.

There maybe devices which could not report their local model updates to winner miner in specified time . The FL must have protocol to handle these straggling devices while updating the global model [14].

H. Consensus Protocol

After the latest blocks are broadcasted by miners, the peers in the blockchain must validate the block transactions and check the correctness of the updated global model state trie. The peers compute their own global model state trie from transactions within the block and verify the root of the global model state trie against the broadcasted block. If the block is found to be valid, the blockchain network must reach consensus for blockchain upto respective block and the

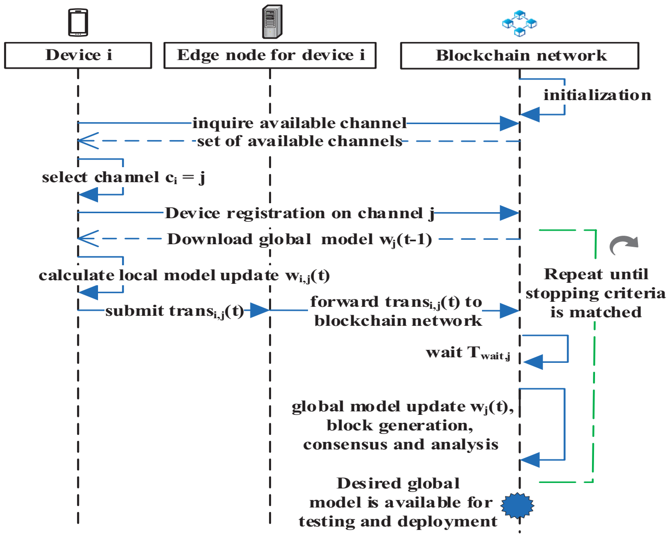
Device i
Edge node for device i
Blockchain network
initialization
inquire available channel
set of available channels
select channel $c_i = j$
Device registration on channel j
Download global model $w_j(t-1)$
calculate local model update $w_{i,j}(t)$
submit $trans_{i,j}(t)$
forward $trans_{i,j}(t)$ to blockchain network
Repeat until stopping criteria is matched
wait $T_{wait,j}$
global model update $w_j(t)$, block generation, consensus and analysis
Desired global model is available for testing and deployment

Fig. 3. FLchain: Sequence diagram for operations of FL via blockchain

corresponding miner is regarded as a winner miner. However, if the block is invalid, the block is rejected. The latest block broadcasted by winner miner is appended on channel-specific ledger during consensus. Since, every blockchain node computes, verifies and validates the global model state trie for consensus, blockchain-based FL is more reliable and robust than the typical FL. The underlying consensus protocols can be modified version of Practical Byzantine Fault Tolerance (pBFT) and Proof-of-Work (PoW) as new block-generation needs to be terminated once the stopping criteria is met by the global model.

I. Analysis

After each global model iteration, analysis is performed to check if the FL has achieved desired results or needs more rounds to be performed. The stopping criteria can be defined on per FL task basis.

For M_j, the FL process continues for global iterations T until $\|w_j(T) - w_j(T-1)\|_2 \leq \varepsilon_{threshold,j}$ for a pre-defined constant $\varepsilon_{threshold,j} > 0$. When desired criteria is achieved, the global model is available for testing and deployment.

VI. EVALUATION

FLchain provides a suitable platform for FL over the blockchain network. The main merits of FLchain are as follows:

- FLchain provides an individual channel for the learning of each global model. The consensus and ledger for storing the local model updates are channel-specific. The global model state trie is also maintained on per-channel basis.
- Global model state trie can securely and unblemished store the global model weights in a Merkle Patricia Tree. The global model state trie can be regenerated and verified at any iteration from the genesis block upto the top block in the blockchain of a particular channel.
- In FLchain, the global model updates are computed, validated, verified and stored by the blockchain network rather than a single central server. Thus, it is more robust than the typical FL.

VII. CONCLUSION AND FUTURE WORK

In this paper, we devised an architecture for FL through the blockchain network composed of edge devices. We established that a separate channel can be assigned for learning of each global model in the blockchain network. We presented the notion of the global model state trie to securely store the global model as a Merkle Patricia Tree. The limitation of the proposed approach is that the user-devices depend on the integrity of their corresponding edge devices for forwarding of transactions to the blockchain network. In the future, we aim to optimize FLchain with respect to latency, computing and storage requirements. In addition, we will devise a reward mechanism for user devices and miner nodes partaking in FLchain.

REFERENCES

[1] J. Zhao, Y. Chen, and W. Zhang, “Differential Privacy Preservation in Deep Learning: Challenges, Opportunities and Solutions,” *IEEE Access*, vol. 7, pp. 48 901–48 911, 2019.

[2] A. Zappone, M. Di Renzo, and M. Debbah, “Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?, In Press,” *IEEE Transactions on Communications*, 2019.

[3] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[4] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, “Blockchain for AI: Review and Open Research Challenges,” *IEEE Access*, vol. 7, pp. 10 127–10 149, 2019.

[5] D. Dillenberger, P. Novotny, Q. Zhang, P. Jayachandran, H. Gupta, S. Mehta, S. Hans, S. Chakraborty, M. Walli, J. Thomas, R. Vaculin, K. Sarpatwar, and D. Verma, “Blockchain analytics and artificial intelligence,” *IBM Journal of Research and Development*, 2019.

[6] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, “Chained anomaly detection models for federated learning: An intrusion detection case study,” *Applied Sciences*, vol. 8, no. 12, p. 2663, 2018.

[7] H. Kim, J. Park, M. Bennis, and S. Kim, “Blockchained On-Device Federated Learning, In Press,” *IEEE Communications Letters*, 2019.

[8] P. Thakkar, S. Nathan, and B. Viswanathan, “Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform,” in *26th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Milwaukee, USA, Sep. 2018, pp. 264–276.

[9] V. Buterin, “Merkling in ethereum,” *published on Ethereum blog*, 2015, accessed: 2019-05-15. [Online]. Available: https://blog. ethereum.org/2015/11/15/merkling-in-ethereum/

[10] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, “Hyperledger fabric: a distributed operating system for permissioned blockchains,” in *Proceedings of the Thirteenth EuroSys Conference*. Porto, Portugal: ACM, 2018.

[11] G. Wood *et al.*, “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[12] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” *arXiv preprint arXiv:1610.02527*, 2016.

[13] O. Shamir, N. Srebro, and T. Zhang, “Communication-efficient distributed optimization using an approximate newton-type method,” in *Proc. of the 31st International conference on machine learning*, Beijing, China, 2014, pp. 1000–1008.

[14] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan *et al.*, “Towards federated learning at scale: System design,” *arXiv preprint arXiv:1902.01046*, 2019.