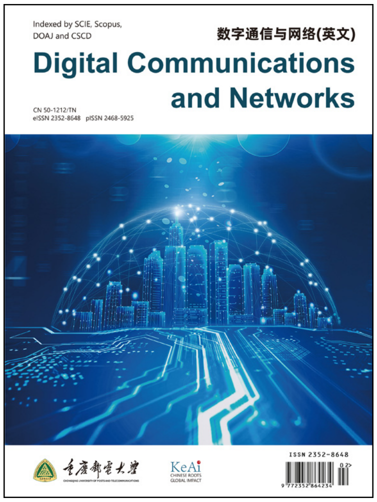

Indexed by SCIE, Scopus, DOAJ and CSCD
数字通信与网络(英文)
Digital Communications and Networks
CN 50-1212/TN
eISSN 2352-8648 pISSN 2468-5925
KeAi
ISSN 2352-8648

A hierarchical blockchain-enabled distributed federated learning system with model-contribution based rewarding

Haibo Wang, Hongwei Gao, Teng Ma, Chong Li and Tao Jing

PII: S2352-8648(24)00085-3
DOI: https://doi.org/10.1016/j.dcan.2024.07.002
Reference: DCAN 765

To appear in: *Digital Communications and Networks*

Received date: 21 June 2023
Revised date: 18 June 2024
Accepted date: 1 July 2024

Digital Communications and Networks(DCN)

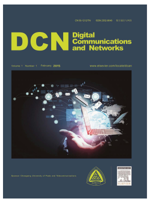
HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/dcan



A hierarchical blockchain-enabled distributed federated learning system with model-contribution based rewarding

Haibo Wang[a], Hongwei Gao*[a], Teng Ma[a], Chong Li[b],Tao Jing[a]

[a] The Research Institute of Broadband Wireless Mobile Communication, Beijing Jiaotong University, Beijing, 100044, China
[b] The Department of Electrical Engineering, Columbia University, New York, 10027, United States

Abstract

Distributed Federated Learning (DFL) technology enables participants to cooperatively train a shared model while preserving the privacy of their local data sets, making it a desirable solution for decentralized and privacy-preserving Web3 scenarios. However, DFL faces incentive and security challenges in the decentralized framework. To address these issues, this paper presents a Hierarchical Blockchain-enabled DFL (HBDFL) system, which provides a generic solution framework for the DFL-related applications. The proposed system consists of four major components, including a model contribution-based reward mechanism, a Proof of Elapsed Time and Accuracy (PoETA) consensus algorithm, a Distributed Reputation-based Verification Mechanism (DRTM) and an Accuracy-Dependent Throughput Management (ADTM) mechanism. The model contribution-based rewarding mechanism incentivizes network nodes to train models with their local datasets, while the PoETA consensus algorithm optimizes the tradeoff between the shared model accuracy and system throughput. The DRTM improves the system efficiency in consensus, and the ADTM mechanism guarantees that the throughput performance remains within a predefined range while improving the shared model accuracy. The performance of the proposed HBDFL system is evaluated by numerical simulations, which show that the system improves the accuracy of the shared model while maintaining high throughput and ensuring security.

KEYWORDS:
Blockchain, Federated learning, Consensus scheme, Accuracy dependent throughput management.

1. Introduction

Distributed Federated Learning [1] is expected to be the major Artificial Intelligence (AI) training paradigm in the decentralization-oriented internet such as Web3. DFL allows multiple parties to collaboratively train a global model without a centralized server by sharing each node's local model. However, in the absence of a centralized authority, different nodes may lack the incentive and trust to share their models. This problem arises from the potential presence of malicious nodes that could contribute low-quality models, undermining the efforts of those who share well-trained models. Blockchain, as the fusion of cryptography, public key infrastructure, and economic system, provides a decentralized solution that transparently executes transactions and automatically establishes trust in an open and trustless environment to solve the problem of reliable transmission of trust and value [2, 3, 4]. Therefore, by applying blockchain technology to DFL, it becomes possible to overcome the trust and incentive barriers inherent in the system. Both public blockchain and consortium blockchain frameworks are designed for multi-node cooperation. In the public blockchain network, all nodes are anonymous, have equal rights to access the blockchain, create and validate new blocks of data, which are truly de-

*Haibo Wang, Hongwei Gao (corresponding author), Teng Ma, Chong Li,and Tao Jing (E-mail: hbwang@bjtu.edu.cn; 22120049@bjtu.edu.cn; 19120095@bjtu.edu.cn; cl3607@columbia.edu; tjing@bjtu.edu.cn) centralized [5]. Cryptocurrency networks such as Bitcoin and Ethereum are examples of public blockchain implementations. In the consortium blockchain network, multiple entities typically form a consortium to participate in bookkeeping according to certain rules. The consortium blockchain has strict access permissions for nodes, and only authorized nodes can join. Typical representatives of consortium blockchains are Hyperledger, Corda and others. Since all nodes with a common interest are known, a high-performance consensus mechanism can be adopted but it may not be entirely decentralized or censorship-resistant [6].

Integrating blockchain with federated learning can effectively mitigate malicious activities by meticulously recording the contributions of nodes, thereby enhancing the integrity and reliability of the model training process. The approach also fosters a more robust, secure and efficient framework for the advancement of decentralized AI systems. However, neither the public blockchain nor the consortium blockchain can solve the trust and incentive problem in the DFL training process without sacrificing training efficiency. For instance, public blockchain sacrifices network throughput [1] as a result of the consensus algorithm, which requires plenty of nodes to reach an agreement in a decentralized manner [7, 8], resulting in longer delay (i.e., lower training efficiency) for larger networks.

To address deal with these challenges, we propose a Hierarchical Blockchain-enabled DFL (HBDFL) system to motivate nodes with local datasets to share their models, protect nodes' private data, and eliminate the mistrust between organizations while ensuring the performance of model accuracy and system throughput. In the proposed HBDFL system, the problem of incentive is solved using the public blockchain and the problem of model security is addressed using the consortium blockchain. The key contributions of the paper are summarized as follows:

- We design a HBDFL system that includes a public blockchain layer and a consortium blockchain layer, providing a very generic framework for DFL problems. On the public chain layer, a Model-Contribution-based Rewarding Mechanism (MCRM) is proposed to allocate cryptocurrency-based reward based on the validated model accuracy and training difficulty. On the consortium blockchain layer, the PoETA consensus mechanism is proposed to guarantee the security and accuracy of DFL model training.

- A Distributed Reputation-based Verification Mechanism (DRTM) is proposed to improve the system throughput, where each transaction contains the necessary information to validate a local upload local model and reward calculation.

- We conduct extensive simulations to evaluate key performance, including training latency, accuracy of the shared model, and resilience against network attacks. The results validate the design decisions and demonstrate the effectiveness of the proposed schemes.

The rest of the paper is organized as follows: Section II discusses related works, Section III presents the system design, Section IV presents the evaluation and discussion, and Section V provides the conclusion.

2. Related works

The integration of blockchain with federated learning has emerged as a prominent area of research, with the goal of enhancing data privacy and security in distributed systems. Researchers have proposed various frameworks and architectures to combine these two technologies [9]. In [10], the authors proposed a privacy-preserving data sharing mechanism for distributed multiple parties in IIoT applications that incorporates federated learning into a permissioned blockchain, but the security threats were not analyzed. A BLADE-FL framework has been proposed in [11], which integrates the training and mining process in each client to overcome the problems of a centralized network and maintain the privacy enhancing capabilities of the FL system, but there is no incentive mechanism design to encourage clients to share their models. In [12], the authors proposed a framework that collects a small amount of data from different hospitals and trains a shared machine learning model using blockchain-enabled federated learning. In [13], the authors introduced a novel and fully decentralized blockchain-enabled federated learning framework to guarantee end-to-end trust and delay of the upcoming autonomous vehicular system. In [14], the authors proposed a blockchain and federated learning secure architecture for privacy-preserving in smart healthcare. In [15], the authors proposed a FL-Block model to solve the identified problems in fog computing, which shows superior performance in terms of privacy protection, efficiency, and resistance to the poisoning attack. However, the authors have not conducted a comprehensive assessment to determine the equilibrium between privacy preservation and system efficiency. In [16], the authors proposed an improved DPoS consensus mechanism based on the Probabilistic Linguistic Term Set (PLTS) for the smart autonomous multi-robot system to guarantee the immutability of the smart autonomous multi-robot information, thus improving the security of the smart autonomous multi-robot system. In [17], blockchain and federated learning are integrated into the Internet of Vehicles to improve computing efficiency, guarantee the reliability of shared data, and reduce the delay of heterogeneous communication. Furthermore, the authors of [18] proposed a Blockchain-enabled Feder-

[1] The throughput of a blockchain is defined as the number of transactions per second (TPS) that are confirmed in the network

ated Learning framework with Committee consensus (BFLC), which effectively reduces the amount of consensus computing and malicious attacks. However, it does not address the reduction in system throughput resulting from optimizing this framework. In [19], the concept of blockchain-enabled reputation-aware FL was proposed to design a trustworthy collaborative machine learning in mobile edge computing and ensure authenticity, traceability, and provenance, incentives and penalties of all stakeholders in the fine-grained FL environments.

The use of blockchain-enabled learning methods has the potential to enhance data security in the system. However, several inherent challenges remain unaddressed, underscoring the need for further research, including:

- *Consensus efficiency*: The consensus mechanism, which is crucial for maintaining blockchain integrity, inherently limits throughput due to its time-consuming nature, especially in the large distributed networks. Therefore, the throughput of blockchain is always compromised by the consensus mechanism.
- *Model accuracy*: The existing consensus mechanism is not directly related to the DFL model training accuracy, often resulting in stagnant or slow improvements in model performance.
- *System throughput*: To achieve anonymity in a truly distributed network, the throughput performance is sacrificed because the consensus mechanism requires multiple communication rounds between peers to reach an agreement.
- *Model security*: The system should prevent global models from being exposed to unauthorized nodes to maintain data integrity and privacy.

In the following sections, we will present our proposed methods to tackle these challenges.

3. System design

In this section, we propose a Hierarchical Blockchain-enabled DFL system. Firstly, we introduce the framework and the workflow of the system. Next, we present all proposed algorithms in detail. During the system design, we give significant considerations to key metrics such as model accuracy, system throughput and the security of the model parameters.

3.1. System model

Fig. 1 illustrates the HBDFL system, which utilizes a consortium-public blockchain structure. The system is composed of two different blockchain networks: the public blockchain network and the consortium blockchain network, wherein there are three types of nodes: ***Sponsor***, ***Validator***, and ***Committer***.

3.1.1. Sponsor

A *sponsor* is an entity or individual who wants to train a particular model. Each node in the public blockchain can act as a *sponsor* and publish its DFL tasks to blockchain platform. The *sponsor* initializes the DFL workflow by providing a model to be trained and the criteria for selecting *validator*s. The Proof of Elapsed Time and Accuracy (PoETA) consensus mechanism is used by the *validator*s to select a new block. After a block is selected, the smart contract of the *sponsor*s will be triggered and the *committer*s who make contributions to the training will receive the cryptocurrency-based reward through the public blockchain.

3.1.2. Validator

*Validator*s are responsible for transaction verification, model accuracy confirmation and aggregation of local models into a global model. The registeration and cancellation of each *validator* require ratification from more than 50% *sponsor*s. A *validator* generates a block in every T_p.

3.1.3. Committer

*Committer*s have the local dataset and the computing capacity for AI model training. To protect the privacy of training data, the ϵ-DP noise [20] is added to the model parameters when *committer*s perform the model training. To be specific, the *committer*s execute three steps as follows:

- *Step 1:* Download the latest global model from the consortium blockchain network.
- *Step 2:* Start training the model using its private dataset. To protect the privacy of training data, ϵ-DP noise will be adopted in the process of training.
- *Step 3:* Package the trained local model and other relevant information (e.g., model accuracy, digital signature of the committer, etc.) into a transaction. Note that due to limitation of on-chain storage, the trained model should be stored off-chain using a decentralized storage network. By doing so, only the indexing address of the model is stored on the consortium chain.

In the blockchain network, all nodes are connected to both public and consortium chains. The transactions and the block structure on the consortium chain are shown in Figure 2. A block consists of a header and a body. The header records the metadata of the block including the timestamp, the hash of the previous block, the address of the global model, and so on. The body contains a set of transactions generated by *committer*s. The transaction includes the transaction Id, the requester public key, the digital signature, etc. Note that, in addition to transactions related to FL model, there exist other types of transactions, such as

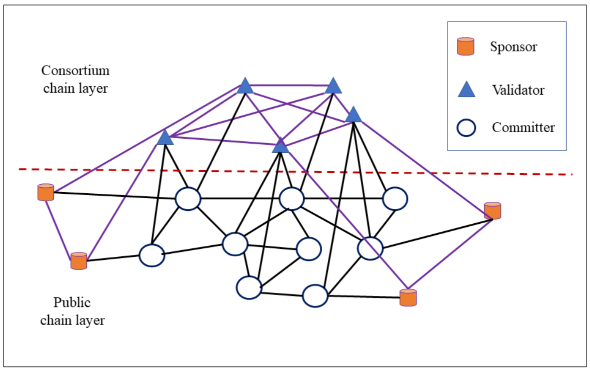


Fig. 1. HBDFL system

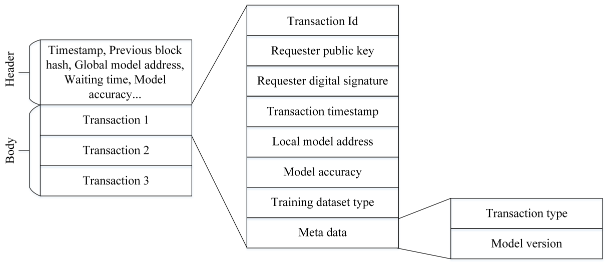


Fig. 2. transaction and block structure.

reward requests to *sponsor*s and *validator* registration and cancellation.

Only validators are the nodes that execute the consortium chain consensus algorithm in terms of block generation and validation. The consortium chain is designed to validate, aggregate and store the FL model [2]. Moreover, the consortium chain employs the proposed PoETA and DRTM to ensure system security, model accuracy and superior system throughput. The public chain is designed to well incentivize the public to join and contribute to the DFL tasks.

As shown in Fig. 3, the system workflow is described as follows:

1. The *Sponsor*s initialize the process by submitting a smart contract to the consortium blockchain. The smart contract contains the key information such as the model architecture and the type of dataset.
2. Based on the request from the *sponsor*s, the *validator*s generate a genesis block containing the initialized model to be trained and the requirements of the dataset. The initial model architecture and the *validator*s that generate the genesis block are selected offline in advance by all sponsors.
3. The *committer*s, who own local training datasets and computing resources, download the latest block (i.e., the latest global model) from the consortium chain and execute local model training.

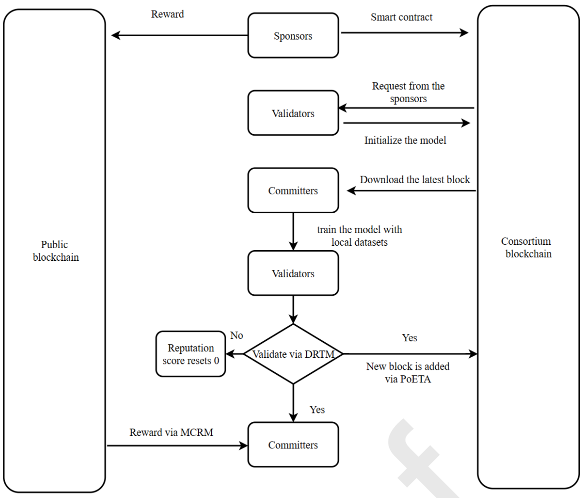


Fig. 3. System Workflow

   When the training is completed, the address of the model is packaged into a transaction that is sent to a random *validator*. Note that each *validator* has an upper transaction limit and therefore cannot be selected when the number of the included transactions reaches the upper limit.
4. Each *validator* collects and packages the received transactions into a new block after waiting a predefined waiting period T_p. A new block will be validated via DRTM and selected to be added to the consortium blockchain via the PoETA consensus algorithm. Next, the *validator* who generates this block will make model averaging (FedAvg) operation periodically among the local models in the transactions of the new block to generate a global model [23].
5. Finally, the *committer*s will be rewarded by the *sponsor*s on the public chain via the designed cryptocurrency-based incentive mechanism MCRM.

3.2. *PoETA consensus mechanism*

PoETA consensus mechanism in the consortium chain is intended for *validator*s to agree on a block, where all contributing committers to this block will be rewarded by sponsors. Proof of Elapsed Time (PoET) [24] consensus mechanism is initially introduced and utilized by Intel in the industrial sector. Targeted at blockchain networks, PoET aims to mitigate the significant energy and computational resources associated with traditional consensus mechanisms. In PoET, each *validator* generates a random waiting time before appending the new block to the blockchain, in which waiting time needs to follow a probability distribution F which is determined by the system in advance. In each consensus epoch, the *validator* with the shortest waiting time is selected.

In the proposed PoETA mechanism, we introduce “model accuracy” as a new element to PoET. By

[2] Given the limited storage capacity of a blockchain, only the address of the FL model is stored on-chain. The model parameters can be stored on decentralized storage network such as IPFS [21]

doing so, a *validator* with the short waiting time and a high accuracy model is selected to append its block to the consortium chain. Specifically, we define a block score s_i for the new block at *validator i* in each DFL training-consensus epoch. This epoch is the time interval between each new block added to the consortium chain. The block score is calculated as

$$s_i = T_i \cdot (1 - a_i), \quad i \in N, \tag{1}$$

where T_i and a_i are the random waiting time and the accuracy of the global model at *validator i*, respectively. Then, the new block with the lowest score is appended to the the consortium chain.

There may be network errors or delays that cause the forking chain problem, where multiple subchains follow the same parent block. In this case, the sum of the block scores on each subchain following their common parent block is compared, and the sub-chain with the lowest total score is selected as the main chain.

3.3. Distributed reputation-based throughput mechanism

In general, *validator*s must verify each transaction received from *committer*s before adding it to a block. Additionally, they must validate each block received from other *validator*s before appending it to the main chain. However, the time and computational cost of validating a model is considerably large when the model contains enormous parameters. We design the DRTM to determine which transactions or blocks to examine in order to reduce the cost of verification while increasing the throughput of the system.

In DRTM, a requester can be a *committer*, who needs to be verified for a submitted, and a *validator*, who needs to be verified for the generated block. Each *validator* maintains a table, in which each item records the public key and the reputation score of the requesters. We define the validation score of the jth requester node V_j, the reputation score of the jth requester node r_j and the upper limit of the reputation score c. We divide the reputation score levels from 0 to c, where a higher reputation score corresponds to a lower validation probability. The entire algorithm is summarized in Algorithm 1: First, the reputation score is initialized to zero. When a new transaction arrives at the *validator*, the *validator* obtains the reputation score of the requester from the key-value map using the requester's public key. Second, the requester's validation score is taken uniformly and randomly between 0 and c. Third, if the validation score is greater than the reputation score, the transaction or block is validated. If the transaction or block is valid, the requester's reputation score is increased by one. If an invalid transaction or block is detected, the corresponding reputation score is reset to zero. In principle, the higher the reputation score of the requester is, the lower probability it will be verified with.

Algorithm 1 DRTM

```
procedure DRTM($r_j$, $c$, $V_j$)
    Initialize $r_j = 0$, $c$
    while new transactions or blocks arrive at the validator do
        $V_j = Random(0, c)$
        if $r_j < V_j$ then
            Validate the transaction or block
            if The transaction or block is valid then
                if $r_j < c$ then
                    $r_j = r_j + 1$
                else
                    $r_j = c$
                end if
            else
                $r_j = 0$
            end if
        end if
        Update blockchain
    end while
end procedure
```

3.4. Accuracy dependent throughput mechanism

For the FL task, there exists a trade-off between the throughput performance and the accuracy of the global model. To address this, we propose an ADTM algorithm that is executed by the *validator*s to monitor the model accuracy while keeping the throughput remains within a predefined range (v_{min}, v_{max}). Our proposed ADTM algorithm, detailed in Algorithm 2, aims to improve the throughput to a certain extent while ensuring the accuracy of the system. In Algorithm 2, $a_{curr}, a_{max}, k, k_{limit}$ and T_p represent the accuracy of the current block, the maximum accuracy of previous blocks, the degree of accuracy, the system-defined degree of accuracy and the limit of packet time, respectively. At the beginning of the ADTM, the degree of accuracy k is calculated based on the trend of accuracy. Only when $|k| \geq k_{limit}$, the *validator* changes the packaging time T_p. If the throughput remains within the desirable range, the adjustment of T_p is based on trends of accuracy and distance to the predefined throughput limit. Otherwise, the adjustment of T_p only considers the performance of the throughput. Although the limit of packet time T_p is different between *validator*s, the goal of balancing the trade-off between the accuracy of the global model and throughput performance is the same.

The throughput v is calculated as the number of transactions stored in the main chain per second:

$$v_i = \frac{\sum_{l=1}^{L} M_i^l}{\sum_{l=1}^{L} T^l_{packet_i} + T^l_{wait_i} + T^l_{communicate_j}}, \tag{2}$$

where M_i^l is the number of transactions in the lth block, $T^l_{packet_i}$ represents the time of packaging and verifying the lth block, $T^l_{wait_i}$ is the time of waiting when executing the PoETA consensus mechanism, $T^l_{communicate_j}$ exist only when the block is proposed by

Algorithm 2 ADTM

```
procedure ADTM($v_{min}, v_{max}, a_{max}, a_{curr}, v_i, k, k_{limit}, T_p$)
    Initialize calculate $v$ according to (2), $k$
    if $a_{curr} \geq a_{max}$ then    ▷ Calculate the degree of accuracy
        if $k \leq 0$ then
            $k = 1$
        else
            $k = k + 1$
        end if
    else
        if $k \geq 0$ then
            $k = -1$
        else
            $k = k - 1$
        end if
    end if
    if $|k| \geq k_{limit}$ then    ▷ Adjust $T_p$ by percentage
        if $v_{min} \leq v_i \leq v_{max}$ then
            if $k \leq 0$ then
                $T_p = T_p \cdot (1 - \frac{v_{max}-v_i}{v_{max}-v_{min}})$
            else
                $T_p = T_p \cdot (1 + \frac{v_i-v_{min}}{v_{max}-v_{min}})$
            end if
        else if $v_i \leq v_{min}$ then
            $T_p = T_p \cdot (1 - \frac{v_{min}-v_i}{v_{min}})$
        else if $v_i \geq v_{max}$ then
            $T_p = T_p \cdot (1 + \frac{v_i-v_{max}}{v_i})$
        end if
    end if
end procedure
```

another *validator*. Due to the communication delay and asynchronous consensus mechanism, the value of v could be different between *validator*s.

Eq.(2) suggests that there are two ways to adjust v. On the one hand, the *validator* can change the block size M to adjust the v, while on the other hand, the *validator* could change the T_{packet}, which indicates the time to wait for receiving transactions. The former affects the communication delay between *validator*s, which can lead to an increase in the number of fork chains and even isolate the *validator* from the network. Instead, no matter how the *validator* modifies the T_{packet}, the throughput of the other *validator*s can't be affected.

3.5. Model-contribution-based rewarding mechanism

Since the size of private data and the computing power can vary widely among *committer*s, those with large training datasets and high computing power may be unwilling to contribute to the FL task. To attract more *committer*s to participate in the FL task, we propose the Model-Contribution-based Rewarding Mechanism (MCRM) that is executed in smart contract, where the *sponsor*s allocate cryptocurrency rewards to the *committer*s though the public blockchain layer based on the MCRM (Algorithm 3). Note that if the model accuracy is already high, or a large number of training epochs have been run, it becomes difficult for a *committer* to improve the model accuracy. Considering the fairness, a higher reward is assigned to *committer*s who improve relatively well-trained model. As illustrated in Algorithm 3, the input parameters l, a_{pre}, λ,s and ψ represent the number of previous blocks in the consortium chain, the accuracy of the global model in the previous block, a scaling coefficient , a compensation coefficient and the reward coefficient converting reward score to cryptocurrency unit, respectively. After a new block is added to the public chain, the endorsement count of each block on the public chain is increased by one. The endorsement count is the number of times that a block has been successfully validated by all *validator*s. Then, the system calculates the $score_i$. using the formula in line 7 of Algorithm 3. On the right side of the formula, the first term measures the relative improvement of model accuracy in the new transaction i (local model) comparing to a_{pre} (accuracy of the previous global model), while the second term measures the length of the consortium chain , reflecting the number of epochs for which the FL model has been trained.

Algorithm 3 MCRM

```
 1: procedure MCRM($a_{pre}, \lambda, s, l, \psi$)
 2:     Initialize $\lambda$, $s$, $\psi$
 3:     while new transactions arrive at the sequence do
 4:         if new block is added to the public chain then ▷ More than half of the nodes agree to the transaction
 5:             Aquire accuracy $a_i$
 6:             $\lambda_i = \frac{\lambda}{-\log(a_{pre})}$, $s_l = s \cdot \log l$
 7:             $score_i = \lambda_i \cdot (a_i - a_{pre}) + s_l$,
 8:             $\psi_i = \psi \cdot score_i$
 9:             sponsors send reward $\psi_i$ to its committer
10:         end if
11:     end while
12: end procedure
```

4. Performance evaluation and discussion

In this section, the system performance, security and privacy are analyzed via the numerical performance (i.e., transaction throughput, model accuracy). In this discussion, it is assumed that the adversary can be the *validator* in the consortium layer and the *committer* in the public blockchain network layer. Adversaries can discard transactions or blocks, create fake transactions or blocks, and attempt to resolve the *committers*' private training set.

4.1. Security and privacy

Table 1 summarizes the solutions that can be employed to satisfy the security requirements. Next, we will analyze some typical attacks which could happen to the *validator*s as follows.

1. In terms of system availability, Denial-of-Service (DoS) attacks are designed to prevent a system

Table 1: System Security

Requirement	Solution
Confidentiality	Asymmetric encryption is used for all communications
Integrity	Each transaction and block contains the digital signature of the owner
Availability	1) *validator*s verify all incoming transactions and blocks, thus protecting the shared model from malicious actions. 2) The PoETA consensus mechanism is used to ensure the system works well even if a few *validator*s are compromised (less than 50%). 3) The actual model parameters are stored by the IPFS technology, which reduces the communication and storage overhead in the blockchain network.
Authentication	To prevent the public key from tampering, all register and cancel operations are performed through the smart contract, which ensures the public key will be added to the main chain.

from serving clients and can occur when the attackers flood a *validator* with a large number of fake transactions, preventing it from processing any legitimate transactions. However, DoS attacks typically concentrate on a few servers and cannot scale to systems with many nodes. Due to the distributed architecture of HBDFL, even if some nodes fail, the network can still function normally and reach consensus as long as the proportion of disabled nodes is less than 50%. Therefore, the proposed HBDFL network is inherently resistant to the DoS attack. In addition, the consortium blockchain has more control over nodes than the public blockchain, where the *validator*s can prevent a particular *committer* from sending transactions to themselves.

2. In terms of the system integrity, the attackers could control the *committer*s or the *validator*s, and modify the model parameters. We utilize the InterPlanetary File System (IPFS) [21] to address this issue. It is a network protocol designed to create a peer-to-peer method to store and share hypermedia in a distributed file system. It is used in HBDFL to store the model parameters , and by doing so we can ensure the unique identification of a model, where the address of the model (stored on chain) will change when the model parameters are tampered with.

The privacy problem of our proposed system can be divided into three parts: the privacy of the models, the privacy of the nodes (*committer*s, *validator*s, and *sponsor*s), and the privacy of the training dataset. The degree of privacy is different between the consortium chain layer and the public chain layer. In the public blockchain network, *sponsor*s or *committer*s can prevent the disclosure of identity privacy by using asymmetric encryption. In the consortium's blockchain network, the true identity of the *validator*s must be known to the *sponsor*s, as the action of registration or cancellation must be ratified by at least 50% of the *sponsor*s. Finally, the linking attack could happen when the attackers infer the training dataset from the model according to the similar public dataset. In the proposed system, the *committer*s employ ϵ-DP technology to protect the privacy of local training data sets .

4.2. Performance evaluation

In our simulation, a three-layer Convolutional Neural Network (CNN) is adopted in each DFL node and the training dataset is taken from the MNIST dataset [25] and Fashion-MNIST dataset [26]. The CNN model is written with Python 3.7 and Tensorflow 1.x. The MNIST dataset consist of 60000 training images and 10000 test images with size of 28 × 28 pixels in 10 classes. Fashion-MNIST is a dataset comprised of 70,000 grayscale images, divided into 10 categories of clothing and accessories. Each image has a resolution of 28 × 28 pixels, which is designed to be a more sophisticated replacement for the traditional MNIST dataset of handwritten digits. For the consortium blockchain, we employed a network consisting of 30 *validator*s. For the public blockchain, 150 *committer* nodes are used. . Therefore, each *committer* is assigned with dataset of 400 training samples and 100 test samples for MNIST. The block size is limited to 50 transactions. We assume that the transactions are generated according to the Poisson process. Other important parameters are detailed in Table 2.

In the simulation part, we first evaluate the accuracy of the PoETA consensus mechanism compared to the standard PoET mechanism. Then, we study the effect of the DRTM on security and performance. Finally, we compare the accuracy of HBDFL with ADTM and the Centralized Training (CT) scheme.

4.2.1. Accuracy of PoETA

The accuracy evolution over training epoch of PoETA and PoET is presented in Fig. 4, with the minimal waiting time $T_{min} = 10s$. As shown in Fig. 4, the PoETA consensus mechanism achieves better accuracy and less variation in accuracy over time compared to PoET. The main reason is that PoETA includes an accuracy parameter a_i, when determining the main chain. On the other hand, the purpose of the PoET consensus mechanism is simply for all *validator*s to select the same shared model without considering the accuracy of the crowd-sourcing FL task running on the consortium blockchain.

Table 2: Parameters of Network

parameter	value
committer number	150
validator number	30
block size	50
T_{min} (s)	10
$T_{average}$ (s)	5
c	50
T_p (s)	5
v_{min} (transaction/s)	1
v_{max} (transaction/s)	6

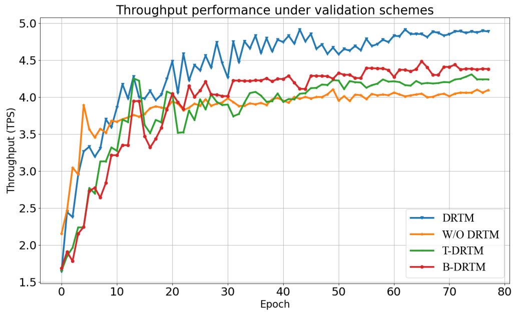


Fig. 5. Throughput under different validation schemes

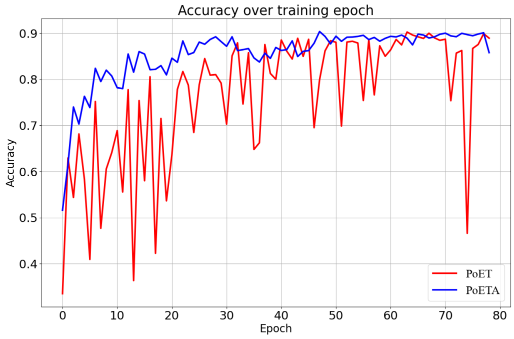


Fig. 4. Accuracy over training epoch.

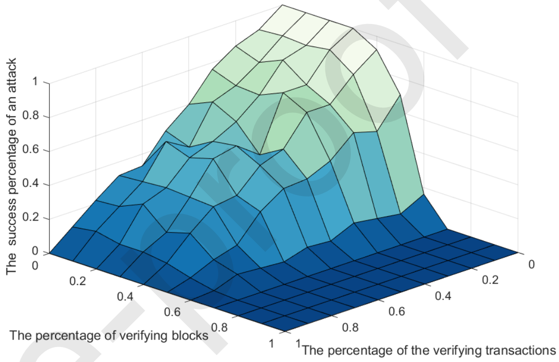


Fig. 6. Security of DRTM.

4.2.2. *Performance of the DRTM*

We evaluate the performance of the DRTM on the system throughput and security from two perspectives.

First, we evaluate the system throughput.

- *DRTM:* DRTM is applied at both the transaction and block level, calculating whether a block or transaction needs to be validated based on the reputation scores of its *committers*.
- *Transaction-level DRTM (T-DRTM):* DRTM is applied to transactions, while each transaction must be validated.
- *Block-level DRTM (B-DRTM):* DRTM is applied to blocks, and within the block for validation, each block must be validated.
- *Without DRTM (W/O DRTM):* Every transaction and block must be validated.

In the simulation, the upper bound of the reputation score $c = 50$, and the waiting period $T_p = 5s$, respectively. As shown in Fig. 5, taking the *W/O DRTM* as the baseline scheme, the final throughput of *T-DRTM*, *B-DRTM* and *DRTM* is about 0.15, 0.26 and 0.85 TPS higher than *W/O DRTM*, which means that the system's performance is improved by nearly 20% using DRTM. This result is expected because with reputation-based validation probabilities in DRTM, the fewer transactions requiring verification within T_p, the more transactions can be packed into a new block; and the fewer blocks requiring verification, the earlier the *validator*s can start packing the next new block. Therefore, the throughput of the system will increase as the system grows.

Second, although the DRTM improves the system's throughput performance, it can mitigate the security of consortium tier. Since the DRTM is applied to both transaction and block levels, the attack on the proposed federated learning system could come from either the malicious *committer*s or the malicious *validator*s. Intuitively, more verifications by validators can reduce the probability of a successful attack, but at the cost of the system throughput. To study the trade-off between the system throughput and security, we vary the percentage of verification from 100% to 0% as illustrated in Fig. 6. Each verification setting is simulated for 100 times, and the success of an attack is defined as that a false transaction or block not being detected by more than 50% of the *validator*s. As shown in Fig. 6, the probability of a successful attack can be reduced to almost zero when the percentage of verifying blocks are greater than 70%. Because the high reputation score of *committer*s is based on the fact that the *committer*s have consistently committed honest models in the past, so the probability that these *committer*s will commit honest models in the fu-

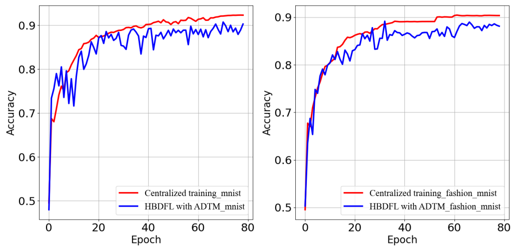


Fig. 7. Accuracy performance of HBDFL with ADTM and Centralized training

ture is extremely high. Thus the system will reduce the probability of verifying the models they submit in order to increase throughput. When the percentage of verification blocks is low, the probability of a successful attack is very high. For example, when the percentage of verification blocks is between 20% and 30% and the percentage of verification transaction is between 20% and 30%, the success percentage of attack is about 70% to 80%. Because the system has a high probability of not validating new transactions and blocks, which leads to the inability to effectively evaluate new transactions and is very prone to receive incorrect models. In summary, if the minimum percentage of block validation is 70% and the minimum percentage of validated transactions is 0%, the security performance of the system is guaranteed.

4.2.3. Performance of the ADTM mechanism

To better illustrate the performance of the ADTM mechanism, we compare the training accuracy of HBDFL adopting ADTM with the accuracy of centralized training based on all *committers*' datasets in Fig. 7. In this simulation, the packet time limit of each *validator* $T_p = 5s$, the minimal percentage of verifying blocks and the minimum percentage of verifying transactions are both 20%, respectively. As shown in the left graph of Fig. 7, which presents data obtained from the MNIST dataset, we observe that the HBDFL-ADTM methodology achieves a final model accuracy of 0.9073, while the centralized training, as the benchmark, achieves 0.923. This comparison shows that the HBDFL-ADTM model can achieve approximately 98.3% of the accuracy achieved by the centralized training approach. Transitioning to the right graph of Fig. 7, which is based on the Fashion-MNIST dataset, it can be seen that the accuracy figures for HBDFL-ADTM and centralized training are identical to those obtained with the MNIST dataset, with HBDFL-ADTM reaching an accuracy of 0.883 and centralized training reaching 0.903. This consistency across different datasets verifies the robustness of the proposed HBDFL-ADTM, confirming that it can achieve the training accuracy very close to the centralized scheme, which is significant for distributed learning models that prioritize privacy and cost-efficiency. This demonstrates the potential of HBDFL-ADTM as a viable alternative to centralized training, providing a balance between high accuracy, privacy preservation, and cost-effectiveness in distributed learning environments.

5. Conclusion

In this work, a Hierarchical Blockchain-enabled Distributed Federated Learning system is proposed to motivate dataset owners to participate in the DFL training process. We design the MCRM which calculates the rewards to the *committer* according to the model training difficulty, model accuracy, and length of the main chain they contribute. To improve the system's throughput and the efficiency of consensus algorithms, we propose the PoETA, consensus scheme, and the distributed reputation-based verification. Simulation results show that the PoETA consensus can improve the model training accuracy, DRTM can simplify the transaction verification process with the security of the system guaranteed, while ADTM can dynamically maintain the system throughput with its accuracy very close to the centralized training (benchmark) scheme. In the future work, we will endeavor to mathematically prove the security performance of our system, consider more complex and broader application scenarios, and investigate further the impact of different model aggregation algorithms and local model training algorithms.

References

[1] A. Hard, K. Rao, and R. Mathews, Federated learning for mobile keyboard prediction, *arXiv preprint arXiv:1811.03604*, 2018.

[2] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, A blockchain-based decentralized federated learning framework with committee consensus, *IEEE Network,* 2021, vol. 35, no. 1, pp. 234-241.

[3] Y. Wang, Z. Su, J. Ni, N. Zhang, and X. Shen, Blockchain-empowered spacce-air-ground intergrated networks: Opportunities, challenges, and solutions, *IEEE Communications Surveys & Tutorials,* 2022, vol. 24, no. 1, pp. 160-209.

[4] M. Nofer, P. Gomber, O. Hinz, Blockchain, *Business & Information Systems Engineering,* 2017, vol. 59, no. 3, pp. 183-187.

[5] M. Scherer, Performance and scalability of blockchain networks and smart contracts, *Dissertation,* 2017.

[6] M. S. Ali, M. Vecchio, and M. Pincheira, Applications of blockchains in the Internet of Things: A comprehensive survey, *IEEE Communications Surveys & Tutorials,* 2018, vol. 21, no. 2, pp. 1676-1717.

[7] I. Martinez, S. Francis, and A. S. Hafid, Record and reward federated learning contributions with blockchain, *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE*, 2019, pp. 50-57.

[8] R. Zhang and B. Preneel, Lay Down the Common Metrics: Evaluating Proof-of-Work Consensus Protocols' Security, *2019 IEEE Symposium on Security and Privacy (SP),* San Francisco, CA, USA, 2019, pp. 175-192.

[9] Y. Zhou, L. Liu, L. Wang, N. Hui, X. Cui, J. Wu, Y. Peng, Y. Qi, and C. Xing, Service-aware 6G: An intelligent and open network based on the convergence of communication, computing and caching, *Digital Communications and Networks*, 2020, vol.6, pp. 253-260.

[10] Y. Lu, X. Huang, and Y. Dai, Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, *IEEE Transactions on Industrial Informatics*, 2019, vol. 16, no. 6, pp. 4177-4186.

[11] J. Li, Y. Shao, and K. Wei, Blockchain Assisted Decentralized Federated Learning (BLADE-FL): Performance Analysis and Resource Allocation, *arXiv preprint arXiv:2101.06905,* 2021.

[12] R. Kumar, A. A. Khan, and J. Kumar, Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging, *IEEE Sensors Journal,* 2021, vol. 21, no. 14, pp. 16301-16314.

[13] S. R. Pokhrel, and J. Choi, Federated learning with blockchain for autonomous vehicles: Analysis and design challenges, *IEEE Transactions on Communications*, 2020, vol. 68, no. 8, pp. 4734-4746.

[14] S. Singh, S. Rathore, O. Alfarraj, A. Tolba, and B. Yoon, A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology, *Future Generation Computer Systems*, 2022, vol. 129, pp. 380-388.

[15] Y. Qu, L. Gao, T. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, Decentralized Privacy Using Blockchain-Enabled Federated Learning in Fog Computing, *IEEE Internet of Things Journal*, 2020, vol. 7, no. 6, pp. 5171-5183.

[16] Jun Liu, Mingyue Xie, Shuyu Chen, Chuang Ma, Qianhong Gong, An improved DPoS consensus mechanism in blockchain based on PLTS for the smart autonomous multi-robot system, *Information Sciences*, 2021, vol. 575, pp. 528-541,

[17] Y. Lu, X. Huang, and K. Zhang, Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles, *IEEE Transactions on Vehicular Technology*, 2020, vol. 69, no. 4, pp. 4298-4311.

[18] Y. Li, C. Chen, and N. Liu, A blockchain-based decentralized federated learning framework with committee consensus, *IEEE Network*, 2020, vol. 35, no. 1, pp. 234-241.

[19] M. H. ur. Rehman, K. Salah, and E. Damiani, Towards blockchain-based reputation-aware federated learning, *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE*, 2020, pp. 183-188.

[20] C. Dwork and A. Roth, The algorithmic foundations of differential privacy, *Found. Trends Theor. Comput. Sci.,* 2014, vol. 9, no. 3-4, pp. 211-407.

[21] S. Muralidharan and H. Ko, An Inter Planetary file system (IPFS) based IoT framework, *IEEE international conference on consumer electronics (ICCE), IEEE*, 2019, pp. 1-2.

[22] C. Xing, Y. Jing, S. Wang, S. Ma, and H. V. Poor, New Viewpoint and Algorithms for Water-Filling Solutions in Wireless Communications, in *IEEE Transactions on Signal Processing,* 2020, vol. 68, pp. 1618-1634.

[23] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B.Arcas, Communication-efficient learning of deep networks from decentralized data, in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282

[24] L. Chen, L. Xu, and N. Shah, On security analysis of proof-of-elapsed-time (poet), *International Symposium on Stabilization, Safety, and Security of Distributed Systems,* Springer, Cham, 2017, pp. 282-297.

[25] G. Cohen, S. Afshar, and J. Tapson, EMNIST: Extending MNIST to handwritten letters, *IEEE international joint conference on neural networks (IJCNN), IEEE*, 2017, pp. 2921-2926.

[26] Kadam, Shivam S., Amol C. Adamuthe, and Ashwini B. Patil, CNN model for image classification on MNIST and fashion-MNIST dataset, *Journal of scientific research* 2020, pp. 374-384.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

HaiboWang is an editorial board editor for Digital Communications and Networks and was not involved in the editorial review or the decision to publish this article.