Computer Communications 196 (2022) 55–65

Contents lists available at ScienceDirect

Computer Communications

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

Blockchain-empowered secure federated learning system: Architecture and applications

Feng Yu [a,b], Hui Lin [a,b,*], Xiaoding Wang [a,b,**], Abdussalam Yassine [c], M. Shamim Hossain [d]

[a] *College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian, 350117, China*
[b] *Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, Fuzhou, Fujian, 350117, China*
[c] *Department of Software Engineering, Faculty of Engineering, Lakehead University, Thunder Bay, 955 Oliver Rd, ON P7B 5E1, Canada*
[d] *Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia*

ARTICLE INFO

Keywords:
Blockchain
Federated learning
Deep learning
Internet of Things
Intelligent transportation

ABSTRACT

Federated learning (FL) is a promising paradigm to realize distributed machine learning on heterogeneous clients without exposing their private data. However, there is the risk of single point failure with FL because it relies on a central server to gather the model updates from clients, moreover, malicious behaviors of some clients may lead to low-quality or even poisoned global models. Blockchain as a revolutionary distributed ledger technology can alleviate the above problems to significantly enhance the security and scalability of FL systems. Therefore, this article presents a general framework of Blockchain-based Federated Learning (BFL) system with detailed description of its key technologies and operation steps. We then review and compare the most recent representative BFL applications. And we outlook some key challenges and opportunities of the future BFL system in terms of security, cost, and scalability. Finally, we propose PoS-BFL in IoT scenarios with malicious devices. The validator voting mechanism and role switching mechanism in PoS-BFL ensure the stakes of legitimate nodes, and effectively reduce the impact of malicious nodes on the accuracy of the system model. And the experiments are conducted to demonstrate that PoS-BFL can achieve 86% accuracy, which is much higher than vanilla FL and pFedMe, and PoS-BFL is robust to some extent by adjusting the ratio of workers, validators and miners.

1. Introduction

With the advent of the era of big data, as an important intelligence resource, data has brought new opportunities and challenges to modern society, which gives us the opportunity to understand all the problems we face through a large amount of available data. In the era of big data, machine learning is considered as an efficient and intelligent data analysis tool that helps us develop better solutions to the problems we face. However, traditional machine learning requires users' local data, which may pose the risk of privacy leakage and information hijacking, resulting in a large number of users or companies being reluctant to share their data. In addition, laws and regulations, such as the General Data Protection Regulation, are becoming more stringent in protecting data privacy. Although these laws and regulations help protect data security and privacy, they also limit data flow and value creation to a certain extent, forcing data to be scattered in disconnected data silos due to factors such as security privacy or geographic location. Therefore, under the premise of ensuring data security and user privacy, how to promote data circulation and sharing, and improve the efficiency of collaboration and cooperation between institutions is a common concern in both industry and academia.

Federated Learning (FL) [1] is a popular large-scale secure multi-party machine learning paradigm. There are two roles in the learning framework, the central server and the participant. The process steps of a typical federated learning system are as follows. First, participants train the model using their own local data. Second, the trained local models or gradients are uploaded to the central server by the participants. Again, all local models uploaded by clients are aggregated into a global model by the central server. Finally, the global model is downloaded locally by each participant. The above steps are performed cyclically until the FL reaches the preset maximum number of rounds or the loss function converges. In contrast to traditional machine learning, FL does not require participants to provide raw data directly.

Although FL has shown its effectiveness in data privacy protection, it still relies heavily on a single central server [2–4]. For instance, the malicious central server can poison the model and even collect the privacy of the participant from local model updates, i.e. the malicious

[*] Corresponding author at: College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian, 350117, China.
[**] Corresponding author.
E-mail addresses: fzhiy270@163.com (F. Yu), linhui@fjnu.edu.cn (H. Lin), wangdin1982@fjnu.edu.cn (X. Wang), ayassine@lakeheadu.ca (A. Yassine), mshossain@ksu.edu.sa (M.S. Hossain).

https://doi.org/10.1016/j.comcom.2022.09.008
Received 18 June 2022; Received in revised form 26 August 2022; Accepted 4 September 2022
Available online 27 September 2022

server can determine whether an exact data record or a data record with a specific property is included in a certain participant's batch, and even expose the participant's training data through gradients. Therefore, the impartiality and security of the central server are crucial to the FL system. Simultaneously, in practical applications, the computing resources and data quality of each user are often quite different. So participants with advantageous resources and high-quality data usually lack the motivation to participate in federated learning in order to avoid the risk of privacy or maintain their industry advantages. Therefore, a reasonable and effective incentive mechanism to stimulate the enthusiasm of users to participate is indispensable in FL.

Furthermore, it is vulnerable to various attacks as participants have to upload local model updates to a central server. Namely, some participants may mislead the global model through intentional or unintentional behavior during the FL process, which means that malicious participants may launch poisoning attacks to affect the global model, leading to the failure of current collaborative learning [5]. In addition, some unconscious behaviors of participants can indirectly lead to low model quality due to participants' energy constraints and dynamic mobile network environment. At the same time, in some specific scenarios, the network bandwidth, efficiency and reliability of data sharing must be considered, especially in vertical fields with low latency requirements such as healthcare and intelligent transportation [6].

In current practical FL applications, the following three challenges need to be addressed: (1) In terms of the architecture of the FL system, the underlying network topology relies on a trusted centralized server to handle model updates for each participant. Then the whole system will be paralyzed, if there is a single point of failure problem. In addition, increasing the number of nodes participating in training will bring additional network load to the centralized server and ultimately reduce the efficiency of training. Therefore, this centralized network topology restricts the robustness and efficiency of the federated learning system. (2) In terms of participant credibility, not all participants are credible or reputable during the FL process. Untrusted participants may upload malicious local updates due to the energy constrain and dynamics of the mobile network. And it is worser that adversaries who have lower reputation can steal user private data through security attacks; (3) In terms of incentive mechanism, users participating in federated training must contribute their own private data to train the global model shared by all parties. In practical applications, the data quality and computing resources of each user are often quite different. Participants with advantaged high-quality data and resources usually lack motivation to participate in federated learning, which may lead to low-quality models. Therefore, there is a need for an incentive mechanism to increase the enthusiasm of users to participate.

As a new distributed computing and storage paradigm that integrates a variety of existing technologies, blockchain uses a peer-to-peer network for data transmission, and generates and updates data through a distributed consensus algorithm. The distributed ledger ensures that the stored data cannot be tampered with through cryptographic principles and timestamp technology, and it uses smart contracts or automated script codes to implement upper-layer application logic. Compared with the unilateral maintenance of data provided by traditional databases, the maintenance of the same data by multiple parties can be realized through the blockchain, which ensures the fairness of the business and the security of the data. The workflow of the blockchain system can be described below. Consensus process is usually the consensus of each node according to predefined mechanism to get the billing, for instance, nodes take turns to an account based on a particular order based on the work force or stakes of the competition to an account. The winning node has all of the data during the current period package, wraps it in a new block, and links to the main chain in time order. At the same time, the blockchain system may issue a certain number of tokens to reward the winning node and incentivize other nodes to continue participating in the data consensus process.

As a new paradigm of distributed computing, blockchain can improve federated learning in the following three aspects. First, the blockchain network adopts a decentralized or weakly centralized peer-to-peer network topology, which provides a suitable infrastructure for the model aggregation of federated learning, and improves the flexibility and fault tolerance of computing; Secondly, the identity authentication of the blockchain system and authority management mechanisms can improve the security of the federated learning system. Thanks to the distributed consensus protocol, the blockchain can ensure the fairness of nodes and help build trust between user nodes participating in training; Lastly, the blockchain can automatically manage the multi-round federated learning tasks of different participants through custom smart contracts, and it can also motivate more users to participate in the co-construction of the ecosystem through cryptocurrency. This can effectively mitigate challenges of FL in the real-world. At the same time, there are similar cooperation models and characteristics of trustworthiness between blockchain and FL. Specifically, they are the multi-party collaboration network architecture of blockchain and the multi-party participation structure of FL, blockchain's consensus mechanism and data verification mechanism to ensure the data is immutable and non-repudiated, and the participants' privacy is protected in the data cooperation process of FL. Therefore, this makes the combination of blockchain and FL a more complete solution that incentivize collaborative data training while ensuring data privacy and security.

Due to the characteristics of blockchain, blockchain technology has huge application prospects in various scenarios assisted by artificial intelligence. Furthermore, the foregoing content shows that the blockchain-based federated learning system can efficiently alleviate the above-mentioned traditional federated learning challenges. Therefore, many scholars have applied blockchain to enhance the research work of federated learning. However, there is still a lack of literature on the necessity, general framework, application, challenges and opportunities of Blockchain-based FL, and this article will fill this gap.

We summarize the main contributions of this paper as follows:

- We explain the three challenges of the FL system, namely, robustness or efficiency issues caused by a centralized underlying architecture, data security and privacy leakage caused by participant credibility and low-quality models due to lack of incentives.
- We briefly review the blockchain technology, and point out the similarity of blockchain and federated learning in the cooperative model, as well as credibility and the complementary characteristics of application value, which makes the combination of the two become a better solution.
- We present the general framework of Blockchain-based Federal Learning (BFL) to solve the above-mentioned problems, and it can resist most common attacks, such as privacy leakage attacks, poisoning attacks, etc. Moreover, BFL achieves decentralization, thereby enhancing the security of the FL system. And we review the latest Blockchain-based FL applications recently and point out their advantages and disadvantages. And some key challenges are also discussed.
- We propose PoS-BFL in IoT scenarios with malicious devices. The validator voting mechanism and role switching mechanism in PoS-BFL ensure the stakes of legitimate nodes, and effectively reduce the impact of malicious nodes on the accuracy of the system model. And experiments are conducted to demonstrate that PoS-BFL can achieve 86% accuracy, which is much higher than vanilla FL and pFedMe, and PoS-BFL is robust to some extent by adjusting the ratio of workers, validators and miners.

We organize the rest of this paper as follows. The general framework and core technology of Blockchain-based FL are described in Section 2. Then, the applications of Blockchain-based FL are stated in Section 3. The open issues and future research directions are given in Section 4. Our method (PoS-BFL) in IoT with malicious nodes setting is proposed and experiments are conducted in Section 5. The conclusion is presented in Section 6.

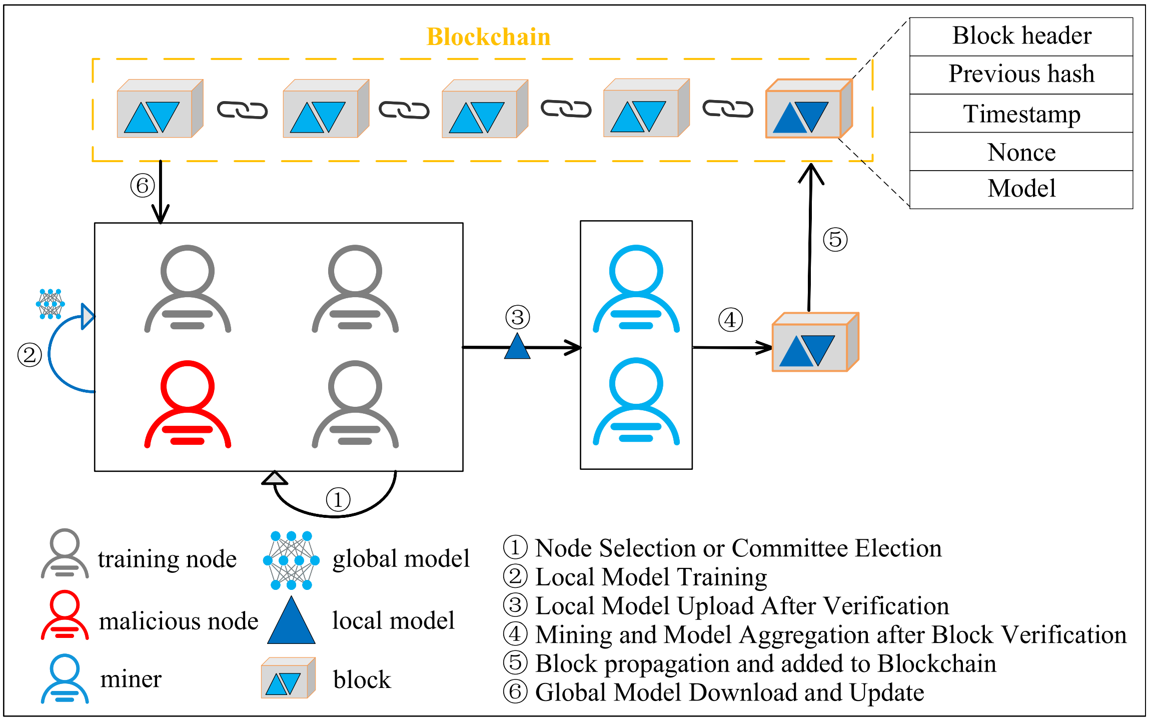
Blockchain
Block header
Previous hash
Timestamp
Nonce
Model
⑥
⑤
②
③
④
①
training node
malicious node
miner
global model
local model
block
① Node Selection or Committee Election
② Local Model Training
③ Local Model Upload After Verification
④ Mining and Model Aggregation after Block Verification
⑤ Block propagation and added to Blockchain
⑥ Global Model Download and Update

Fig. 1. A general framework of Blockchain-based FL: BFL.

Table 1
Terms and their abbreviations.

Term	Abbreviation of term
Internet of Thing	IoT
Internet of Vehicle	IoV
Distributed Hash Table	DHT
Proof of Work	PoW
Delegated Proof of Stake	DPoS
Practical Byzantine Fault Tolerance	PBFT
Proof of Federation	PoF
delegated Practical Byzantine Fault Tolerance	dPBFT
Interplanetary File System	IPFS
Proof of Information	POI
Proof of training Quality	PoQ
Reject On Negative Influence	RONI
Road Suit Units	RSUs
Base Stations	BSs
Directed Acyclic Graph	DAG
Local Differential Privacy	LDP

2. Blockchain-based FL framework: BFL

Fig. 1 illustrates the general framework of Blockchain-based FL, including six steps. Most existing work has adopted all or part of the framework. This section first introduces the six steps of the general framework, and then introduces the various core technologies used in BFL in detail. Table 1 shows related terms and abbreviations in this article.

2.1. Framework

Without loss of generality, it presents a general framework of Blockchain-based Federated Learning (denoted as BFL) in Fig. 1. The main components of the BFL system are blockchain, miners, nodes, local models and global models. The blockchain is responsible for storing the model, the miner is responsible for finding a nonce and publishing the block, the nodes are divided into training nodes and malicious nodes, they all try to get rewards by participating in FL training. The difference is that malicious nodes are kicked out due to the node selection process and cannot participate in training. The specific operation steps of BFL are described as follows:

(1) Node selection or committee selection. Participants (unless otherwise specified, participants are the same as nodes) are likely to be curious or malicious in the real world. When these nodes participate in the FL process, they may upload malicious model updates to the traditional FL aggregation server, affecting the final model quality. Therefore, it is first necessary to select some honest nodes to participate in the federated learning process. As shown in Fig. 1, the nodes marked in red are the malicious nodes to be filtered out. In addition, committee members can be elected from the training nodes to perform this task in order to verify the update of the model uploaded by the training node in the FL process. The committee election method can be to prefer nodes with high stakes or reputation. It is worth noting that the miners (the nodes marked in blue in Fig. 1) are selected from other nodes except malicious nodes and training nodes, and are responsible for the block generation and verification process. Whenever the block is verified by one miner, it is miners who generated and verified the block will be rewarded. See Section 2.2.3 of Technologies for the specific incentive mechanism.

(2) Local model training. The training node trains local data samples to update its initial global model.

(3) Local model upload after verification. The committee members will verify the local model uploaded by the training node. If most committee members promise that the model update is legal, the selected leader node will upload it to the miners.

(4) Mining and model aggregation after block verification. After the miner receives the model update from the leader node, it starts to execute the consensus algorithm, such as the classic and commonly used PoW algorithm. When the miner finds a nonce that solves the mathematical problem or receives a nonce broadcast from other miners, it stops mining, otherwise the miner will keep looking for this nonce. The miner who finds the nonce will publish a new block as the latest block, and after the block was verified, the training nodes would aggregate the model updates according to pre-set aggregation rules and stored aggregated model in the latest block.

(5) Blocks are propagated and added to the blockchain. Broadcast the new block to all other miners, and add the block storing the local model to the blockchain when the majority of miners verify and promise it is legit.

(6) Download and update the global model. Validated local models are downloaded from the blockchain by training nodes for the computation of the global model.

We repeat steps (1)–(6) until the global model reaches a certain accuracy or converges. It is worth noting that in a deep learning model with a scale of millions of parameters, the storage and communication overhead of the FL process is expensive. Therefore, in order to reduce storage overhead, some research work allows the blockchain to store the fixed hash value of the model instead of the model itself. The details will be discussed in Technologies' On-chain Storage versus Off-chain Storage. It should be emphasized that in some Blockchain-based FL framework applications, although the training nodes, committee members, and miners described in this article may be not a one-to-one correspondence, the functions of roles are corresponding. For instance, in [3], miners are responsible for verifying model updates and blocks, and their functions correspond to the committee members and miners mentioned in this article.

As far as security is concerned, this article focuses on server vulnerabilities and participant vulnerabilities in terms of the threat model, and storage and communication are considered in terms of efficiency aspect. Note that the federated learning system is composed of two architectures, one is a client–server architecture, and the other is a P2P network architecture. And this paper mainly considers the development trend of the former. The core technology used in BFL is described below.

2.2. Technologies

2.2.1. Client selection based on protocol design and reinforcement learning

In [7], security threats in federated learning are discussed, including poisoning attacks, inference attacks, backdoor attacks, and adversarial network generation-based attacks. According to the appeal analysis, we know that the P2P network structure combined with blockchain can avoid the potential threat of untrusted central servers. [8] proposed a two-stage protocol, that is, the first stage uses numerical calculation to prevent malicious clients from being selected, and the client selection algorithm designed in the second stage can select the appropriate client in the model upload stage of the original FL protocol. Collected for each round of FL training to defend against malicious attackers. In addition, due to the heterogeneity of client resources participating in FL training and poor wireless channel conditions, the model upload or update time is too long. [9] first proposed a new client selection protocol, Federated Client Selection (FedCS), the key idea of FedCS is two-step client selection. The first step is for mobile edge computing (MEC) operators to randomly request a certain number of clients to participate in the current training. The client that receives the request will notify the operator of its resource information. The next step is for the MEC operator to determine, based on this resource information, which clients complete the next steps within a given deadline. This protocol reduces model training time and alleviates problems caused by limited client resources or unstable wireless network environment. The recent great success of Deep Reinforcement Learning (DRL) in real-time strategy (RTS) games such as AlphaGo and StarCraft has attracted a large number of researchers to this field of research. Deep Q-Network(DQN) [10] is applied to the client-side selection step, which aims to offset the bias introduced by non-iid data by actively selecting the best set of devices. This method promotes the improvement of verification accuracy and accelerates the convergence of FL.

2.2.2. Consensus mechanism based on blockchain

The consensus algorithm can effectively ensure the security of the blockchain, so using the correct consensus algorithm can significantly improve the performance of blockchain applications. In [2,4], the PoW consensus algorithm is adopted, and its core idea is to allocate charging rights and rewards through the competition of computing power among nodes. To ensure that the latest blocks are generated first and master the longest chain, one person needs more than 50% of the global computing power to tamper with the blockchain. Since the costs may far outweigh the benefits, the security of the blockchain can be guaranteed through PoW. However, this consensus has some limitations. For example, participants need to waste a lot of computing power to solve a meaningless mathematical problem; in order to reduce forks, the calculation time of each block cannot be too short, which will slow down the verification speed of transactions. For this, [6] adopts the DPoS consensus algorithm, which reduces the verification of nodes and speeds up the speed of block generation and transaction verification. DPoS has faster throughput and transaction verification speed than PoW and PoS, and is infinitely scalable. [11] uses the Algorand algorithm, where the Algorand algorithm is based on PoS and PBFT, so it is faster than PoS verification and more scalable than PBFT. As the core technology of blockchain, the consensus mechanism needs to select appropriate algorithms for different scenarios to make the system more efficient and maximize available resources.

In BFL, local model update verification is a key part of the consensus mechanism. [3] evaluates model updates based on the public test data set provided by the task publisher, and only model updates with an accuracy higher than the threshold accuracy given by the task publisher will be accepted. In [2,4], k-fold cross-validation is implemented through miners' exchange and model updates verification. The local computing time proportional to the data size corresponding to the generated local model is used to compare the data size of the training data to verify the reliability and authenticity of the local model update. At the same time, in order to ensure the authenticity of the local calculation time, the proof of elapsed time method is adopted under the support of Intel's SGX technology. Multi-Krum is adopted in [5,11] to verify the legitimacy of model updates. Then the selected leader node uploads the verified update to the miners. Miners inspired by the FL system reward mechanism compete to run the PoW algorithm because the first miner to calculate the nonce will get rewards from the blockchain network and then run the aggregation algorithm to store the aggregation model in the generation block [2,4]. Here, the first miner to get the nonce is the aforementioned leader node. In [3], one of the miners authorized by the global trust agency is randomly select as the leader node for the current round. This method allows the existence of malicious representatives but not more than 1/3. Next, the leader node in [2,4] broadcasts the block storing the aggregation model to other miners. If most miners verify it is legal, then add it to the blockchain. In [3], the leader node broadcasts the result of its own block verification to other miners because the delegated Practical Byzantine Fault Tolerance (dPBFT) algorithm is adopted, a new block is added to the blockchain when more than 2/3 of the employees verify that it is legal.

2.2.3. Incentive mechanism based on reputation

As mentioned earlier, an efficient incentive mechanism can promote the voluntary participation of participants and continue to provide high-quality data, which in turn makes the system form a virtuous circle. Reputation is introduced as an evaluation indicator, miners perform the Multi-Krum algorithm on the updates in the trading pool to remove malicious updates and accept most of the updates accepted in each round [11]. According to whether the miner's verification results are consistent with the facts, the reputation value is calculated and rewards or punishment are given. Through this incentive mechanism, the influence of malicious or lazy workers will be reduced, poisoning attacks will be resisted to a certain extent, and system security will be enhanced. Repeat the FL process until the end of the task, and finally improve the quality of the global model.

2.2.4. On-chain storage versus off-chain storage

Uploading, downloading, and direct storage of large-scale FL system models on the blockchain will cause great network bandwidth pressure and even increase network delay. Interplanetary File System (IPFS) is a P2P, distributed, decentralized system that connects across computer nodes through shared public files. It is based on content addressing, that is, files with the same content have the same IPFS hash address, which means that the IPFS hash address can be used to access IPFS content [12]. Moreover, IPFS is similar to the blockchain in that it does not allow files to be tampered with. The blockchain in the IPFS Blockchain-based FL system only stores the IPFS hash value of the model/parameter instead of directly transferring the model/parameter, so the FL system based on the content addressing hyperlink method greatly reduces the amount of transmitted information, which reduces the overhead, relieves the network bandwidth pressure, and further improves the transmission efficiency [13]. Specifically, the training node uploads the model to a distributed file storage system (such as IPFS) to obtain the corresponding hash value, and then stores the hash value in the block. Because IPFS is based on content addressing, that is, the same model file corresponds to a hash value, so the node downloads the hash value of the model from the blockchain and uses it to obtain the corresponding model file from the distributed file storage system. Namely, the distributed storage file system is mainly responsible for storing model parameter files, and the blockchain stores the corresponding IPFS hash value. In theory, the distributed storage system relieves the storage pressure of the blockchain, and transforms the original model transmission between the blockchain and the node into the model IPFS hash value transmission to reduce the communication overhead, thereby improving the communication efficiency. As an off-chain storage technology, the IPFS file system aims to reduce the burden on the main chain, such as transferring some complex computing tasks to the off-chain platform and executing some transactions off-chain. Therefore, there are certain advantages compared to on-chain storage.

3. Applications, advantages and limitations

The potential single point of failure risk of the central server in the traditional FL system is eliminated in the blockchain-based federated learning system, which establishes the trust between the server and the client and improves the scalability of the wireless edge network. In the Internet of Things, Intelligent Transportation, Unmanned Aerial Vehicle (Unmanned Aerial Vehicle) environment edge content caching, edge data caching [19] and healthcare, urban computing and Internet and finance, smart cities, industrial manufacturing, physical information systems and other fields There are a lot of applied research [20]. This holds promise for future large-scale industrial applications. Therefore, this section provides a brief overview and analysis of recent research efforts in IoT, intelligent transportation, and healthcare. Table 2 summarizes more specific research work.

3.1. Internet of Things

Traditional FL needs to upload models or gradients to a central server, which may threaten the security of the system if the central server is malicious. Therefore, BlockFL proposed by [2] utilizes the blockchain consensus mechanism to ensure machine learning without any centralized training data and collaboration. Its main advantage is that the removal of the central server avoids the risk of malicious servers, and the verification before updating the storage model reduces the risk of waste of block resources. However, directly storing model updates in the blockchain can easily cause storage pressure, increase network bandwidth pressure, and reduce network transmission efficiency. Only giving rewards based on the size of the sample data will inevitably ignore the sample quality, an important factor that can affect the global model. In the context of fog computing, the FL-Block proposed by [4] has made some improvements on the basis of [2]. The blockchain only stores model updates, and the model data storage is implemented through a distributed hash table., reducing storage pressure and block generation rate. FL-Block realizes distributed privacy protection through comprehensive verification, hybrid identity, off-chain data storage and retrieval, and access control. At the same time, poisoning attacks are eliminated from fog servers. However, how to balance privacy protection and efficiency deserves further study. In [11], home appliance manufacturers train machine learning models based on customer data based on FL using the reputation mechanism. In addition, differential privacy is used to protect the privacy of extracted features. In order to motivate participants to actively participate in the FL process, an incentive mechanism is designed in [11], that is, miners perform Multi-Krum algorithm on updates in the transaction pool to remove malicious updates and accept updates in each round. In most updates, the reputation value is calculated and rewards or penalties are given based on whether the miner's verification result is consistent with the facts. Similarly, considering the storage limitations of the blockchain, the representative distributed file storage system IPFS is used to realize off-chain storage. GFL proposed by [13] also uses IPFS to reduce the pressure of storage and communication. Among them, Ring Decentralized Federated Learning (RFDL) uses a similar Ring-allreduce model parameter synchronization method to reduce communication congestion problems and make full use of network bandwidth. The method of knowledge distillation and dynamic polymerization ratio improves the generalization of GFL. However, uploading a model with more parameter models to the IPFS process will consume a lot of time, and model compression methods can be considered to alleviate this problem. The BFL system architecture diagram in IoT is shown in Fig. 2.

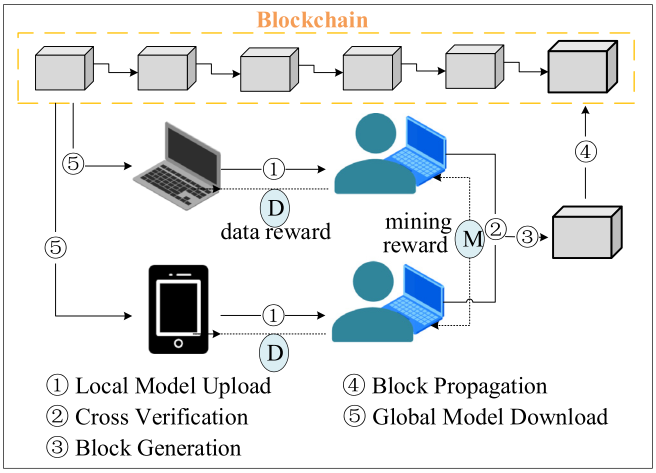

Fig. 2. A case of Blockchain-based Federated Learning System in IoT.

3.2. Intelligent transportation

Designing a safe and timely traffic flow forecasting procedure is extremely important in the field of intelligent transportation. In [3], a FL-based method is proposed for the modification of the Gate Recurrent Unit (GRU) neural network model. Miners validate model updates for distributed vehicles. A set of trusted consensus nodes replaces the central server and manages all local model updates. Furthermore, location privacy protection is achieved using local differential privacy techniques, preventing attackers from gathering information from participants using membership inference attacks. However, the expensive communication overhead in FL results in low accuracy and efficiency of the GRU model. In order to alleviate the problems of reliable communication, data sharing security, and training efficiency in the Internet of Vehicles environment, and to improve data security,

Table 2
Applications and detail analysis based on Blockchain-based FL system.

Ref	Scenes	Advantages	Limitations
[2]	IoT	• Remove the central server • Avoid unnecessary block waste.	• Reward clients only based on dataset size.
[4]	Fog Computing	• Reduce storage pressure with DHT • Protect data privacy by Encrypting uploaded models.	• Cannot completely eliminate forks • Lack of scalability.
[14]	IoT	• Two types of on-chain storage models: model and update • Effective incentive mechanism to promote a virtuous circle.	• Local models are directly stored in the blockchain.
[11]	IoT	• Incentive mechanism for customers to participate combined with Multi-Krum and reputation.	• Customers may delay the overall crowdsourcing progress and cause unnecessary delays.
[15]	Industrial IoT	• System architecture made from permissioned blockchain component and federated leaning component.	• The limited resource of devices in IIoT • New intelligent mechanisms are required to improve data utility.
[6]	IoV	• DPOS instead of PoW • The asynchronous FL scheme based on node selection, local asynchronous aggregation and global synchronous aggregation algorithms.	• Lack of scalability.
[3]	IoV	• One third of the malicious nodes are allowed within dPBFT • Provide location privacy for participating vehicles with LDP.	• The low accuracy and efficiency of GRU model.
[16]	Healthcare	• Chained digest creation approach to achieve the integrity • Session-based scheme to achieve the integrity and security.	• Data move and access delay problem.
[17]	Healthcare	• POI consensus algorithm • Decentralized privacy-preserving healthcare predictive modeling framework.	• Lack of efficiency and scalability due to POI consensus.
[18]	Mobile Networks	• Worker reputation safety management • A reputation-based scheme to reach rapid consensus for selecting credible workers.	• Lack of Trade-off of optimization between leaning performance and resource expenditure.
[5]	P2P System	• Open source code • The first P2P ML system provides privacy protection • Can be deployed in hundreds Node's WAN environment • Client data poisoning attacks and privacy attacks are solved.	• Lack of scalability • Vulnerable to privacy disclosure attacks.
[13]	Decentralized System	• Open source code • Reduce storage pressure and communication pressure through IPFS and model parameter synchronization to • Improve generalization of the model with knowledge distillation and dynamic aggregation ratio methods.	• May attain low-quality models because data amount is regarded as the only choice of quality • The process of uploading models with more parameters to IPFS System take lots of time.

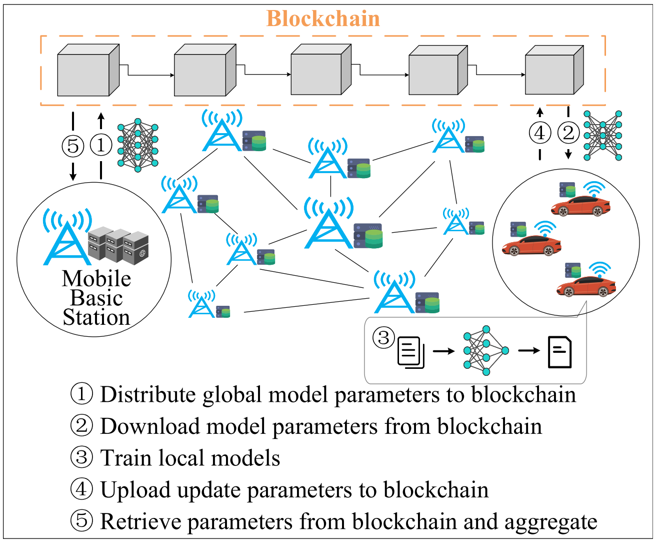
Blockchain
⑤
①
④
②
Mobile Basic Station
③
① Distribute global model parameters to blockchain
② Download model parameters from blockchain
③ Train local models
④ Upload update parameters to blockchain
⑤ Retrieve parameters from blockchain and aggregate

Fig. 3. Blockchain-based federated learning system in intelligent transportation.

training efficiency and accuracy, a hybrid blockchain consisting of a permissioned blockchain and a local directed acyclic graph is proposed. (PermiDAG, PermiDirected Acyclic Graph) architecture is proposed [6] for synchronous global aggregation and asynchronous local training. Additionally, PermiDAG allows vehicles to store only local DAGs and allows RSUs to store permissioned blockchains to improve storage policies and reduce storage pressure. In addition, PermiDAG's partition tolerance allows certain nodes of this system to run the blockchain efficiently as well. Whereas, Markov decision process is used to model the quality of learning (QoL) problem of node selection and applies Deep Deterministic Policy Gradient (DDPG) to find the optimal node. However, the DDPG-based node selection algorithm takes more rounds to find the optimal solution when the number of vehicle nodes increases, so the algorithm needs to be improved within the scope of scalability. The architecture diagram of the BFL system in intelligent transportation is shown in Fig. 3.

3.3. Healthcare

In the healthcare field, the medical data is the privacy of patients. Use sensors and mobile applications to detect the patient's physical condition, and then share the collected data with laboratories and institutions for diagnosis and further research, this method is too rigid

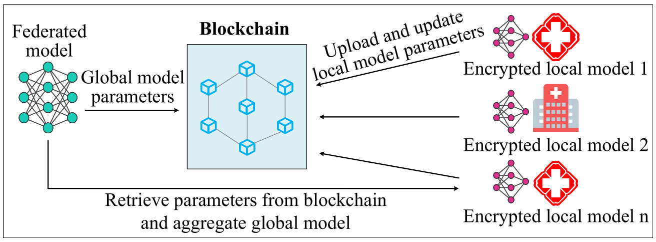

Fig. 4. Blockchain federated learning in healthcare.

to effectively support metadata changes. Therefore, it is urgent to design an efficient data sharing solution to solve the above problems. Therefore in [16], the MedChain data sharing framework, can flexibly manage different types of messages from healthcare data, is proposed, and a session-based data sharing scheme, which makes data sharing more flexible and meets the security requirements of data sharing, is adopted. And the ModelChain framework which can flexibly manage variable and immutable health care data is proposed in [17]. It uses a chained summary creation method and a session-based scheme to achieve the integrity and security of the summary respectively. The process of BFL system in Healthcare is shown in Fig. 4 [21].

4. Opportunities and challenges

Although the support of blockchain technology enables federated learning to work in a more realistic decentralized setting, the research on Blockchain-based FL is still in its infancy. We give the following possible future research directions.

4.1. Client selection and scheduling in FL

Customer selection and scheduling strategies are very important in FL training. The research work of [9] proposed the optimization method FedCS, which adopts two-step client selection, and [22] proposed TiFL, which adaptively selects clients with similar training time in each round of training. In the case of affecting the accuracy, the problem of client data heterogeneity is alleviated. [10] proposed Favor, which can intelligently select client devices, offset the bias introduced by non-iid data, and speed up the convergence rate. Although these works start from different angles to speed up the convergence, there is still a need to provide a standardized method for FL. This remains a challenging task due to the trade-offs between heterogeneous systems of different clients and any particular client and the utility of statistical models in client selection.

4.2. Computation and communication cost

In the FL training process, efficient communication protocols or model compression are used to reduce communication overhead [23], but model compression usually increases the error of the training target. Therefore, the choice of the number of magnitudes to balance error and communication is a key challenge. The reputation indicator can be introduced. Customers with high reputation will be given priority to participate in training. Employing Deep Reinforcement Learning (DRL) for client selection can reduce unnecessary computational and communication overhead. Recently, the In-Edge AI framework [24] reduces the system communication load by intelligently exchanging learning parameters between devices and edge nodes. In-Edge AI performs fine-grained collaborative scheduling of AI tasks on edge nodes and mobile devices to provide differentiated support for various services [25], which is convenient and can almost achieve real-time response.

4.3. Trade-off optimization between privacy protection enhancement and cost

Enhancing privacy protection in FL at the expense of efficiency and accuracy is well-represented in current research work. However, there is less research on the appropriate level of encryption and the amount of noise added for Secure Multi-Party Computation (SMC). If the encryption level is high and the noise is large, the accuracy of the FL model will be low, and on the contrary, the privacy of participants will not be protected. In the Blockchain-based FL system, the PoW consensus process of the blockchain brings additional delay to FL, which can be alleviated by introducing edge computing. However, the different computing capabilities of devices or channel conditions pose a challenge to how to rationally arrange edge server resources. An interesting method is to introduce DRL to arrange channel allocation and optimize client device resource utilization. Another promising approach is to focus on lightweight mining design or apply other non-linear distributed ledger technologies, such as Tangle, a DAG-based distributed ledger, which will greatly reduce FL waiting time. However, in vertical fields with low-latency requirements [26], such as the intelligent transportation and healthcare, to design low-latency, high-reliability and high-efficiency systems is also a key issue.

4.4. Scalability of system

Bitcoin is known to suffer from low throughput and high transaction latency, and other PoW-based protocols also inherit this shortcoming, resulting in scalability issues for blockchains. In [27], it can be roughly divided into three categories, namely Layer-1, Layer-2, and Layer-0, which are dedicated to solving the scalability problem of blockchain. Among them, the most concerned shard technology is regarded as the future of achieving blockchain scalability [28]. At the same time, the shard technology faces two key issues: (1) How to put transactions in different shards; (2) How to improve the efficiency of cross-shard transactions. In addition, most of the current blockchain-based research focuses on single-chain technology. However, its performance cannot meet the high efficiency and high scalability requirements of the FL system in IoT.

5. Casy study: BFL in IoT

In this section, we consider a practical application of BFL in an IoT scenario. The application scenarios, our method, and simulation experiments and numerical analyses are described in detail below.

5.1. Scenarios

Here we consider a FL System with a C/S architecture, as described in [29], the system is prone to collapse due to a single point of failure because it depends on the way that the central server aggregates local models to construct a global model. Therefore, the decentralized structure of the blockchain in BFL can solve this problem well, and it becomes a promising solution. At the same time, we consider that some IoT devices participating in the FL process are malicious, that is, participating client devices may upload malicious local model updates. And the adversary of privacy theft is one worker during the learning process. In this case, vanilla FL cannot work well. As shown in Fig. 8, in the presence of 15% malicious devices, the accuracy of the global model in the MNIST classification application has dropped to around 12%. Obviously, this is unacceptable in practical applications. And in our experiments we assume device d is malicious if Gaussian noise are added to the dataset, and when a malicious device d became a worker w, it would inject Gaussian noise of variance 1 to its legitimately learned local model parameters. In our approach, the better the data contributed by the participant, the more rewards it will receive in the learning process.

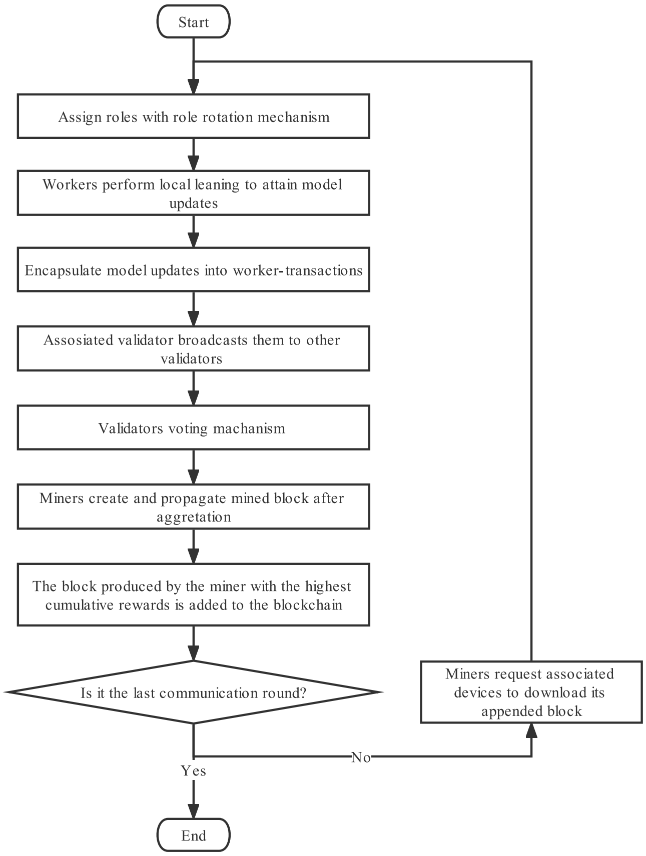
Start
Assign roles with role rotation mechanism
Workers perform local leaning to attain model updates
Encapsulate model updates into worker-transactions
Assosiated validator broadcasts them to other validators
Validators voting machanism
Miners create and propagate mined block after aggretation
The block produced by the miner with the highest cumulative rewards is added to the blockchain
Is it the last communication round?
Miners request associated devices to download its appended block
No
Yes
End

Fig. 5. The flow chart of PoS-BFL.

5.2. Security model

In federated machine learning, we can usually build three security models according to the trustworthiness of the parties: ideal model, semi-honest model, and malicious model. The semi-honest model is also known as the honest and curious model.

- Ideal model: During the execution of the protocol, each party is trusted, and after one party sends its information to the other party, the other party will not view the information and will only continue to execute the process according to the agreed protocol.
- Semi-honest model: During the execution of the protocol, the parties follow the process specified in the protocol, but it can derive additional information based on input or intermediate results from other parties.
- Malicious model: During the execution of the protocol, a malicious attacker may not implement the protocol honestly. It can analyze the privacy information of an honest participant by means of illegal input or malicious tampering of input from the participant under its control, or even terminate the protocol by means of premature termination or refusal to participate.

Workers in the BFL system are semi-honest during the learning process. And workers train in compliance with the agreement but are curious about model updates from other parties, so some sensitive information can be inferred from transactions on the blockchain. Even though the data is not explicitly shared in the original format, it is still possible for curious workers to steal the training data from the gradient and roughly reconstruct the original data. At the same time, malicious workers may contribute noisy data, which can seriously affect the performance of the federated model. So the security model that we build here is the second one, the semi-honest model. And our goal is to minimize the impact of malicious workers on model accuracy in BFL system as much as possible, and to avoid the disclosure of workers' privacy information including data.

5.3. Method: PoS-BFL

First, as mentioned above, in order to solve the single point of failure problem, we introduced Blockchain to form the BFL system; secondly, considering the participation of malicious devices, we added validators to the system. And each device in the BFL system can be one of the following roles: (1) a training node (namely worker w) updating its local model during FL procedure; (2) a model validator v to validate and vote on the validity of received model updates; (3) a minner m attempt to store the corresponding voting results and local models in the next newly generated consensus block. Specifically, the validator broadcasts received local model updates to other validators, then validators will use the global model of the previous round and $w's$ local model of the current round to make predictions on their own test set, respectively. If the accuracy of the illegal model drops significantly, then the model is maliciously distorted. By exploiting this phenomenon, each validator v casts its own vote on the legitimacy of the model, and based on the cumulative votes of multiple validators, malicious devices associated with illegal models will be added to the blacklist. The PoS consensus mechanism is that in each round of communication, each role will receive a reward corresponding to its contribution, and the device with the most accumulated reward will be the miner. Due to the verification mechanism, malicious devices are less likely to receive rewards from workers frequently, which reduces their cumulative reward and thus reduces the likelihood of them being selected as winner-miner. And in order to reduce the possibility of malicious workers being selected consecutively as miners, we use a role rotation mechanism so that miners would be re-selected in each round. Therefore, PoS-BFL can effectively resist the illegal behavior of malicious devices theoretically.

5.3.1. Procedure of PoS-BFL

Assuming that there is a set of IoT devices $D = \{d_1, d_2, \dots, d_n\}$, similar to vanilla FL, carries out a learning process of $R = \{R_1, R_2, \dots, R_j, \dots\}$ communication rounds. In each round, each device is assigned a role: worker $w \in \mathcal{W}$, validator $v \in \mathcal{V}$ and miner $m \in \mathcal{M}$, where $|\mathcal{W}| + |\mathcal{V}| + |\mathcal{M}| = |D|$ and they perform the corresponding function according the role. The device id is used as its public key to verify the signature of the generated transaction or block. In the communication round R_j, worker w performs local learning on the global model of the previous communication round $j-1$, and generates a local model update, denoted as L_j^w. L_j^w will be used to construct the global model G_j after the verification process of all the validators. In addition, worker w calculates the expected reward r_j^w for learning L_j^w through the PoS-based reward mechanism. Then worker w encapsulates L_j^w and r_j^w into a worker-transaction tx_j^w signed by the private key of the worker w and sends it to a randomly associated validator. Then each validator v gets the worker-transaction tx_j^w from all its associated workers and broadcasts it to all other validators. So, For all workers, each validator v will have tx_j^w, so v can vote for each L_j^w. And r_j^w will be rewarded to the worker w if the aggregated voting result of L_j^w in the legal block (denoted as $block_j$) recorded in the communication round R_j is *positive*.

Validator v will receive a verification reward r_j^{v-veri} after verifying the signature of a worker-transaction tx_j^w. If the signature of transaction tx_j^w is verified, v would extract L_j^w from tx_j^w and vote for it, denoted as $vt^v(L_j^w)$, its value is either *positive* or *negative*. Since the validator can only obtain L_j^w, and the accuracy of the uploaded global model and local model update is not trusted, the validator uses G_{j-1}^w and L_j^w to make predictions on its test datasets, and the accuracy is $A^v(G_{j-1}^w)$ and $A^v(L_j^w)$ respectively. If $A^v(G_{j-1}^w) - A^v(L_j^w) > vh_j^v$, it means that the accuracy drop exceeds the threshold that vh_j^v can tolerate, v will mark L_j^w and worker w as potentially malicious, then the vote for L_j^w is *negative*; otherwise, v treats w as legitimate and votes *positive*. And the voting mechanism algorithm is given in Algorithm 1.

Table 3
Symbols and their meanings in PoS-BFL.

Meanings	Symbols
Device	d
Worker	w
Model validator	v
Miner	m
Set of IoT devices	D
Communication rounds	R
Local model update of w in R_j	L_j^w
Global model in R_j	G_j
Worker transaction	tx_j^w
Expected reward of the worker w in R_j	r_j^w
Verification reward of v in R_j	r_j^{v-veri}
Validate reward of the validator v in R_j	r_j^{v-vali}
Verification reward of m in G_j	r_j^{m-veri}
Vote of v for L_j^w, *positive* or *negative*	$vt^v(L_j^w)$
Number of *positive* votes for L_j^w	$N_+(L_j^w)$
Number of *negative* votes for L_j^w	$N_-(L_j^w)$
Prediction accuracy of validator for L_j^w	$A^v(L_j^w)$
Prediction accuracy of validator for G_j^w	$A^v(G_j^w)$
Threshold of the accuracy drop	vh_j^v
Candidate block of miner m	$block_j^m$

Algorithm 1. Validators voting mechanism. DT_t are the validators' test datasets.

```
Input: $G_{j-1}^w, L_j^w, DT_t, v \in \mathcal{V}, w \in \mathcal{W}$
$A^v(G_{j-1}^w) \leftarrow Evaluate(G_{j-1}^w, DT_t)$;
$r_j^{v-vali}, A^v(L_j^w) \leftarrow Evaluate(L_j^w, DT_t)$;
if $A^v(G_{j-1}^w) - A^v(L_j^w) > vh_j^v$ then
    $vt^v(L_j^w) \leftarrow negative$;
else
    $vt^v(L_j^w) \leftarrow positive$;
return $r_j^{v-vali}, vt^v(L_j^w)$
```

Miner m received a verification reward r_j^{m-veri} after verifying a signature of $tx_j^v(L_j^w)$. If the signature is verified, m would extract $vt^v(L_j^w)$ from $tx_j^v(L_j^w)$. For all extracted $vt^v(L_j^w)$, m will aggregate the voting results of each validator $v \in V$ for the same L_j^w, denoted as $vt^{m,V}(L_j^w)$. Then, for all workers w, all aggregated voting results are put into a secretly constructed candidate block, denoted $block_j^m$. All expected rewards, namely $r_j^w, r_j^{v-veri}, r_j^{v-vali}$ and r_j^{m-veri}, are included in the candidate block. Then everything is hashed by miner m according to PoS consensus and signed using the worker private key.

The block $block_j^m$ mined by the miner m is propagated to all other miners in the network. After receiving all the blocks propagated in the network, m records all the miner with the highest cumulative reward among miners produces a block as a legal block. And only this legal block can extract the reward and punishment records and the corresponding model update voting results. Legitimate blocks will be added to their own chain by each miner m, requesting the w and v associated with it. After adding a block to device $d(m, v, or\ \ w)$, two tasks are executed by d to process the additional block, i.e. compute G_j using the local model, where the number of positive votes is not less than negative votes and stakes for each device are updated by accumulating recorded legal rewards.The flow chart is shown in Fig. 5. And the symbols used in this section and their corresponding meanings is given in Table 3.

5.3.2. PoS consensus mechanism

Both legally learned local model updates and global model updates stored on the blockchain can be protected by the PoS consensus mechanism. Since it is the work of miners to aggregate votes and record voting results in the blockchain, the calculation of the global model will be destroyed when miners are malicious devices. Therefore, for a robust BFL, it is critical to avoid the selection of blocks mined by malicious devices [30].

Suppose r denote a unit reward. According to the description in Section 5.3.1, there are three types of rewards: (1) worker rewards, (2) validator rewards, and (3) miner rewards.

(1) Worker rewards. The reward of a worker $w \in R_j$ is proportional to $|train_w|$ and le_j^w, which represent the number of data samples in $train_w$ and the number of local training epochs in R_j respectively, only if $N_+(L_j^w)$, i.e., *positive* votes of L_j^w, is not less than $N_-(L_j^w)$, i.e., *negative* votes of L_j^w, by V in R_j. Then, we calculate the total reward of worker $w \in R_j$ as:

$$r_j^w = le_j^w * |train_w| * r \;\; if\ N_+(L_j^w) \geq N_-(L_j^w)\ else\ 0. \tag{1}$$

(2) Validator rewards: A validator $v \in R_j$ is rewarded because of generating $\{vt^v(L_j^w)\}$, i.e., voting for $\{L_j^w\}$ extracted from the signature-verified $\{tx_j^w\}$ and verifying the signatures of the received worker-transactions $\{tx_j^w\}$. Thus, we can calculate the total rewards of validator $v \in R_j$ by:

$$r_j^v = |\{r_j^{v-veri}\}| + |\{r_j^{v-vali}\}| \;\; = |\{tx_j^w\}| * r + |\{vt^v(L_j^w)\}| * r. \tag{2}$$

It is worth mentioning that v will not vote on L_j^w encapsulated in tx_j^w if the signature of tx_j^w is not verified. That means that $|\{tx_j^w\}|$ is not necessarily equal to $|vt^v(L_j^w)|$.

(3) Miner rewards: For verifying the signatures of received validator-transactions, denoted by $\{tx_j^v(L_j^w)\}$, a miner $m \in R_j$ is rewarded using the following formula

$$r_j^m = |\{r_j^{m-veri}\}| = |\{tx_j^v(L_j^w)\}| * r. \tag{3}$$

5.3.3. FL in PoS-BFL

The FL process in PoS-BFL is similar to standard FL. Therefore, the corresponding learning objectives can be expressed in the form of finite sum objectives:

$$\min_{\theta \in \mathbb{R}^d} f(\theta) \overset{def}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(\theta). \tag{4}$$

For federated learning problems, we usually take $f_i(\theta) = \mathcal{L}(x_i, y_i; \theta)$. That is, the prediction loss on the example (x_i, y_i) is constructed with model parameters w. We assume that there are K clients partitioning the data, with P_k as the index set of data points on clients k and $n_k = |P_k|$. Therefore, we can rewrite the objective (4) as

$$f(\theta) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\theta) \;\; where\ F_k(\theta) \overset{def}{=} \frac{1}{n_k} \sum_{i \in P_k} f_i(\theta). \tag{5}$$

If the partition P_k is formed by randomly and uniformly distributing the training samples across the clients, then we will have $\mathbb{E}_{P_k}[F_k(\theta)] = f(\theta)$. This is the IID assumption typically made by distributed optimization algorithms, and for the case where it does not hold (i.e. F_k may be an arbitrary false approximation of f) we call it the non-IID setting.

5.4. Experiments and numerical analyses

We use PyTorch to verify the proposed scheme. The experiment is performed on a computer equipped with Ubuntu system. The machine is configured with a CPU with a frequency of 2.6GHZ and a graphics card GeForce RTX 3090. We assume that both vanilla FL and PoS-BFL involve 20 devices and that their training sets are allocated equal-sized parts of the entire MINIST training set with random shards, with no overlap. And assuming that all signatures are verified. We train the CNN models in vanilla FL and PoS-BFL [31] using a batch size of 10, a learning rate of 0.01, and 5 local training sessions per round of communication. And vh^v and unit reward r was preset to 0.08 and 1, respectively.

In this experiments, to ensure the number of w and v is sufficient for each round of training, and to maximize the robustness of PoS-BFL. We assume that there are 12 workers, 5 validators, and 3 miners.

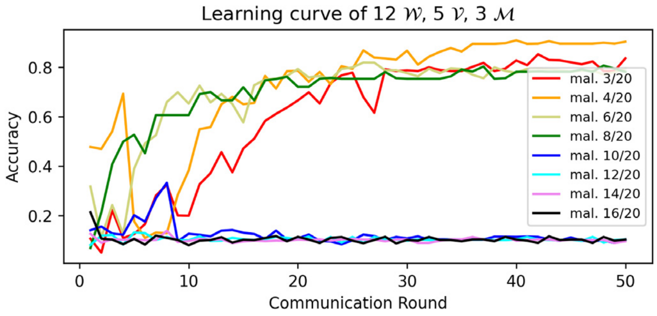

Fig. 6. Accuracy v.s. Communication Round with different malicious nodes, 12 $\mathcal{W}$, 5 $\mathcal{V}$, 3 $\mathcal{M}$.

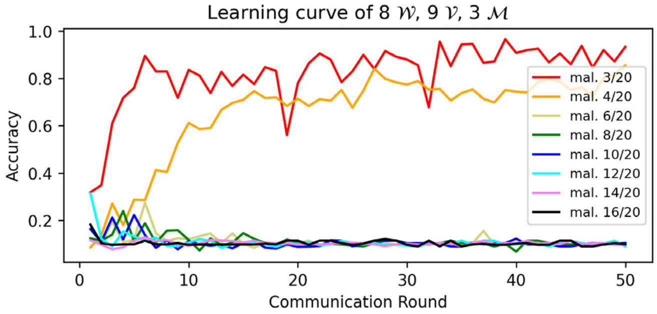

Fig. 7. Accuracy v.s. Communication Round with different malicious nodes, 8 $\mathcal{W}$, 9 $\mathcal{V}$, 3 $\mathcal{M}$.

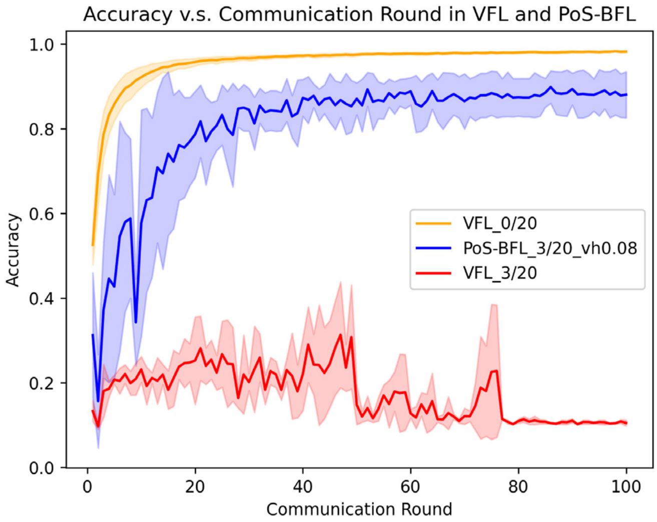

Fig. 8. Effectiveness of validation mechanism in PoS-BFL.

Table 4
Comparison of FL strategies for different device settings.

Device setting	VFL	pFedMe	PoS-BFL
12 $\mathcal{W}$, 5 $\mathcal{V}$, 3 $\mathcal{M}$	11%	13%	**86%**
8 $\mathcal{W}$, 9 $\mathcal{V}$, 3 $\mathcal{M}$	10%	11%	**95%**

5.4.1. Robustness of PoS-BFL

In this experiment, we set the number of malicious nodes by parameters (i.e. 3, 4, 6, 8, 10, 12, 14, 16). As shown in Fig. 6, there are 3 miners, 5 validators, and 12 workers. And it can be observed that a high accuracy rate ([77%, 90%]) is maintained by PoS-BFL, when there are less than 10 malicious nodes with Gaussian noises injected. This shows that PoS-BFL can resist Gaussian noise attack by 50% of malicious nodes in this setting. Further, we change the number of workers and validators to 8 and 9, respectively, and keep other setting unchanged. As shown in Fig. 7, there are 8 workers, 9 validators and 3 miners. And in this setting, the accuracy of PoS-BFL can maintain a high accuracy rate ([80%, 95%]) only when there are less or equal to 4 malicious nodes. This also shows that PoS-BFL is robust, and the robustness capability is affected by the ratio of workers and validators.

5.4.2. Effectiveness of PoS-BFL

We conduct experiments according to the following three settings, namely, vanilla FL of 20 legitimate learning devices, denoted as $VFL\_0/20$, and 3 of the 20 devices are malicious vanilla FL, denoted as $VFL\_3/20$, PoS-BFL with malicious ratio of 3/20 and fixed validator threshold of 0.08, denoted as $PoS - FL\_3/20\_vh0.08$.

In the figure, each experiment requires 3 rounds of 100 communications, and the model accuracy for all devices is recorded at the end of each round of communications. The solid line represents the average accuracy of these three experiments. Comparing the orange and red solid lines in Fig. 8, the accuracy of VFL plummets to about 12% when there are 15% malicious nodes. Comparing the blue and red solid lines, PoS-BFL can achieve an accuracy of 86% in the same 15% malicious node setting, although there is a 10% gap with the accuracy of the orange solid, which is already 7.2x higher than VFL. This is enough to denote that PoS-BFL is effective and efficient in FL settings with malicious nodes. Moreover, we compare the test accuracy of POS-BFL with VFL and pFedMe [32] in device Settings with different proportions of roles, the result of which is given in Table 4, and find that the accuracy of POS-BFL is much higher than the latter two, which further confirms the effectiveness of POS-BFL in the presence of 15% malicious nodes.

6. Conclusions

In response to the requirements of data privacy and security in the era of big data, we have introduced federated learning. Based on the existing FL system's three challenges of the FL system, namely, robustness or efficiency issues caused by a centralized underlying architecture, data security and privacy leakage caused by participant credibility and low-quality models due to lack of incentives, the important features of Blockchain (such as immutability, traceability, and decentralized consensus), and the motivation for the combination of Blockchain and FL are introduced. We point out several important challenges of FL system, namely security, credibility, incentive mechanism and high efficiency. Furthermore, we present the general framework of blockchain based federated learning, BFL, and describe it in details in combination with practical applications. Then, we summarize representative research works of Blockchain-based FL applied in some recent popular fields, and analyze its advantages and disadvantages in detail. And combining the aforementioned applications with the current research progress, we present an analysis of the current Blockchain-based FL system security privacy, system scalability, low-latency requirements, privacy protection and cost optimization and other important challenges, and then point out worth further study directions. Finally, we propose our method PoS-BFL in IoT with malicious nodes. The validator voting mechanism and role switching mechanism in PoS-BFL ensure the stakes of legitimate nodes, and the impact of malicious nodes is effectively reduced on the accuracy of the system model. We also find that POS-BFL can achieve an accuracy of 86% in the presence of 15% malicious devices in the experiments, which is much higher than Vanilla FL and other federal learning strategies. At the same time, by adjusting the ratio of workers, validators and miners, the accuracy of the model does not plummet even when the number of malicious devices increases. It is proved that POS-BFL is robust to some extent.

As mentioned above, current POS-BFL can perform well in the presence of malicious nodes, but there are still some important research topics that deserve further investigation. For instance, the unit reward and threshold may affect the cumulative reward of workers, but we adopt a preset value to simplify the experiment in this paper. Therefore, we will delve into the potential relationship between them and model performance. In addition, the consensus adopted by the blockchain will affect the process of FL, because malicious actors may provide malicious models and may also affect the consensus process. Our future research directions include how to design consensus algorithms to improve the performance of system models.

CRediT authorship contribution statement

Feng Yu: Investigation, Data curation, Formal analysis, Writing – original draft, Writing – review & editing. **Hui Lin:** Conceptualization, Methodology, Supervision. **Xiaoding Wang:** Conceptualization, Methodology, Writing – review & editing. **Abdussalam Yassine:** Supervision, Reviewing and Editing. **M. Shamim Hossain:** Supervision, Reviewing and Editing.

Declaration of competing interest

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.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant No. U1905211 and 61702103, Natural Science Foundation of Fujian Province, China under Grant No. 2020J01167 and 2020J01169, the Researchers Supporting Project number (RSP-2021/32), King Saud University, Riyadh, Saudi Arabia. All authors read and contributed to the manuscript

References

[1] J. Konečný, H.B. McMahan, D. Ramage, P. Richtárik, Federated optimization: Distributed Machine Learning for on-device intelligence, 2016, arXiv:1610.02527 [Cs].

[2] H. Kim, J. Park, M. Bennis, S. Kim, Blockchained on-device federated learning, IEEE Commun. Lett. 24 (6) (2020) 1279–1283.

[3] Y. Qi, M.S. Hossain, J. Nie, X. Li, Privacy-preserving blockchain-based federated learning for traffic flow prediction, Future Gener. Comput. Syst. 117 (2021) 328–337.

[4] Y. Qu, L. Gao, T.H. Luan, Y. Xiang, S. Yu, B. Li, G. Zheng, Decentralized privacy using blockchain-enabled federated learning in fog computing, Ieee Internet Things J. 7 (6) (2020) 5171–5183.

[5] M. Shayan, C. Fung, C.J.M. Yoon, I. Beschastnikh, Biscotti: A blockchain system for private and secure federated learning, IEEE Trans. Parallel Distrib. Syst. 32 (7) (2021) 1513–1525.

[6] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Blockchain empowered asynchronous federated learning for secure data sharing in Internet of Vehicles, IEEE Trans. Veh. Technol. 69 (4) (2020) 4298–4311.

[7] V. Mothukuri, R.M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Gener. Comput. Syst. 115 (2021) 619–640.

[8] K. Zhang, H. Huang, S. Guo, X. Zhou, Blockchain-based participant selection for federated learning, in: Z. Zheng, H.-N. Dai, X. Fu, B. Chen (Eds.), Blockchain and Trustworthy Systems, in: Communications in Computer and Information Science, Springer, Singapore, 2020, pp. 112–125.

[9] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in: ICC 2019 - 2019 IEEE International Conference on Communications, ICC, 2019, pp. 1–7.

[10] H. Wang, Z. Kaplan, D. Niu, B. Li, Optimizing federated learning on non-IID data with reinforcement learning, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, IEEE, Toronto, ON, Canada, 2020, pp. 1698–1707.

[11] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, Y. Liu, Privacy-preserving blockchain-based federated learning for IoT devices, IEEE Internet Things J. 8 (3) (2021) 1817–1829.

[12] J. Benet, IPFS - content addressed, versioned, P2P file system, 2014, arXiv:1407.3561 [Cs].

[13] Y. Hu, W. Xia, J. Xiao, C. Wu, GFL: A decentralized federated learning framework based on blockchain, 2020, arXiv:2010.10996 [Cs].

[14] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, Q. Yan, A blockchain-based decentralized federated learning framework with committee consensus, IEEE Network 35 (1) (2021) 234–241.

[15] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, Ieee Trans. Ind. Inform. 16 (6) (2020) 4177–4186.

[16] B. Shen, J. Guo, Y. Yang, MedChain: Efficient healthcare data sharing via blockchain, Appl. Sci. 9 (6) (2019) 1207.

[17] T.-T. Kuo, L. Ohno-Machado, Modelchain: decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks, 2018, arXiv preprint arXiv:1802.01746.

[18] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, M. Guizani, Reliable federated learning for mobile networks, IEEE Wirel. Commun. 27 (2) (2020) 72–80.

[19] D.C. Nguyen, M. Ding, Q.-V. Pham, P.N. Pathirana, L.B. Le, A. Seneviratne, J. Li, D. Niyato, H.V. Poor, Federated learning meets blockchain in edge computing: opportunities and challenges, IEEE Internet Things J. (2021) 1.

[20] Y. Liu, L. Zhang, N. Ge, G. Li, A systematic literature review on federated learning: From a model quality perspective, 2020, arXiv preprint arXiv:2012.01973.

[21] R. Kumar, W. Wang, C. Yuan, J. Kumar, Zakria, H. Qing, T. Yang, A.A. Khan, Blockchain based privacy-preserved federated learning for medical images: A case study of COVID-19 CT scans, 2021, arXiv:2104.10903 [Cs, Eess].

[22] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan, Y. Cheng, TiFL: A tier-based federated learning system, in: Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing, in: HPDC '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 125–136.

[23] J. Mills, J. Hu, G. Min, Communication-efficient federated learning for wireless edge intelligence in IoT, IEEE Internet Things J. 7 (7) (2020) 5986–5994.

[24] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, M. Chen, In-edge Ai: intelligentizing mobile edge computing, caching and communication by federated learning, IEEE Netw. 33 (5) (2019) 156–165.

[25] J. Mills, J. Hu, G. Min, Multi-task federated learning for personalised deep neural networks in edge computing, IEEE Trans. Parallel Distrib. Syst. 33 (3) (2022) 630–641.

[26] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, M.S. Hossain, Mobility-aware proactive edge caching for connected vehicles using federated learning, IEEE Trans. Intell. Transp. Syst. 22 (8) (2021) 5341–5351.

[27] Q. Zhou, H. Huang, Z. Zheng, J. Bian, Solutions to scalability of blockchain: A survey, IEEE Access 8 (2020) 16440–16455.

[28] S. Yuan, B. Cao, Y. Sun, M. Peng, Secure and efficient federated learning through layering and sharding blockchain, 2021, arXiv:2104.13130 [Cs, Math].

[29] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H.B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, J. Roselander, Towards federated learning at scale: System design, 2019, arXiv:1902.01046 [Cs, Stat].

[30] H. Chen, S.A. Asif, J. Park, C.-C. Shen, M. Bennis, Robust blockchained federated learning with model validation and proof-of-stake inspired consensus, 2021, AAAI, arXiv:2101.03300 [Cs].

[31] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, 2017, arXiv:1602.05629 [Cs].

[32] C.T. Dinh, N.H. Tran, T.D. Nguyen, Personalized federated learning with Moreau envelopes, 2020, arXiv:2006.08848 [Cs, Stat].