Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.030084

REVIEW

A Survey on Blockchain-Based Federated Learning: Categorization, Application and Analysis

Yuming Tang[1,#], Yitian Zhang[2,#], Tao Niu[1], Zhen Li[2,3,*], Zijian Zhang[1,3], Huaping Chen[4] and Long Zhang[4]

[1]School of Cyberspace Science & Technology, Beijing Institute of Technology, Beijing, 100081, China

[2]School of Computer Science & Technology, Beijing Institute of Technology, Beijing, 100081, China

[3]Southeast Institute of Information Technology, Beijing Institute of Technology, Putian, 351100, China

[4]Qianxin Technology Group Co., Ltd., Beijing, 100044, China

*Corresponding Author: Zhen Li. Email: zhen.li@bit.edu.cn

[#]These authors contributed equally to this work

Received: 22 March 2023 Accepted: 26 July 2023 Published: 11 March 2024

ABSTRACT

Federated Learning (FL), as an emergent paradigm in privacy-preserving machine learning, has garnered significant interest from scholars and engineers across both academic and industrial spheres. Despite its innovative approach to model training across distributed networks, FL has its vulnerabilities; the centralized server-client architecture introduces risks of single-point failures. Moreover, the integrity of the global model—a cornerstone of FL—is susceptible to compromise through poisoning attacks by malicious actors. Such attacks and the potential for privacy leakage via inference starkly undermine FL's foundational privacy and security goals. For these reasons, some participants unwilling use their private data to train a model, which is a bottleneck in the development and industrialization of federated learning. Blockchain technology, characterized by its decentralized ledger system, offers a compelling solution to these issues. It inherently prevents single-point failures and, through its incentive mechanisms, motivates participants to contribute computing power. Thus, blockchain-based FL (BCFL) emerges as a natural progression to address FL's challenges. This study begins with concise introductions to federated learning and blockchain technologies, followed by a formal analysis of the specific problems that FL encounters. It discusses the challenges of combining the two technologies and presents an overview of the latest cryptographic solutions that prevent privacy leakage during communication and incentives in BCFL. In addition, this research examines the use of BCFL in various fields, such as the Internet of Things and the Internet of Vehicles. Finally, it assesses the effectiveness of these solutions.

KEYWORDS

Federated learning; blockchain; privacy-preserving

1 Introduction

With the significant improvement in hardware computing and storage ability, the Machine Learning (ML) and Artificial Intelligence (AI) approaches have triggered revolutions in many industries. It has been widely put into practical application in recent years [1], including E-commerce recommendation [2], NLP [3] and sentiment analysis [4,5], healthcare [6] and COVID-19 pandemic [7], etc., which facilitates people's daily life. However, it takes a tremendous amount of data to train a practical ML model and the number of parameters of the model can reach hundreds of millions, which means it is almost impossible for a single-point system to accomplish the training tasks. Hence, only several big companies or government institutions, which hold a huge amount of user data and have the ability to build multi-machine computing architectures, can use ML to complete certain tasks. In contrast, most business enterprises, research groups, and others who lack the ability to obtain large-scale data and computing resources face the dilemma of low data capacity and quality, which makes it difficult to develop well-performed ML models. Moreover, with the coming of the big data era, more problems have appeared. First of all, though billions of personally held smart devices generate unimaginable amounts of data, with the concern of personal privacy issues, it is hard for developers to obtain high-quality and legal data because of the existence of the central server in the ML design. Regardless of the privacy concern, it is also challenging to collect the data since it may consume a tremendous amount of communication resources for the data transfer from the client's end to the central server.

Thus, architectures, protocols, and other technologies enabling collaborative private data sharing and training are urgently needed. To address the above issues, in 2016, Google proposed a novel ML framework termed Federated Learning [8], allowing the participants to train an ML model collaboratively without uploading their own private dataset. The participants in FL can simply upload local model parameters instead of the private local data, which can prevent the data from leaking to others, and save a lot of communication resources meanwhile. FL has developed a lot in recent years and achieved much success in various fields [9], such as healthcare [10], visual object detection [11], drug discovery [12], Internet of Things (IoT) [13], and so on.

However, though FL is well studied in recent years, conventional FL frameworks have emerged with more and more problems with the public awareness of privacy and data security rising. First of all, the existence of a central aggregator which is designed to perform the integration of the uploaded local model updates and update the global model may be a hidden problem of the system stability since the central server is not always reliable. Once the central server is down or compromised, the whole FL system will face the single-point failure problem, leaking participants' private information or even being unavailable for an unacceptable period of time. Even if we do not consider the worst case, in the big data era, amount of participants in the FL system can sometimes be quite a large number that reach the bottleneck of the central server's network and thus bring the problem of connection delay or failure. Moreover, attack techniques aimed at FL are proposed by researchers and industries [14]: model-poisoning attack [15], which can affect the functionality of the model; membership inference attack [16,17], which can infer the information about the training data from the model updates during the process, etc. In the real-world scenario, there can also be malicious clients or data sneaking into the FL system that affect the FL in a way that is hard to detect. Though developers can bring some cryptographic techniques into FL, such as Differential Privacy (DP) [18], Secure Multiparty Computation (SMC) [19,20], Hormographic Encryption (HE) [21,22], and so on, the FL framework needs deeply customized. It cannot be suitable for all kinds of heterogeneous devices, not to mention the problem of asynchrony.

Another problem is a lack of incentives in the FL system. In conventional FL frameworks, the scenario is that participants contribute their data without repayment, indicating it is hard to encourage participants to execute predefined protocol honestly and provide reliable data. Some big companies or research groups may have the ability to get enough data by themselves because of their sufficient funds or good reputation. However, it is very hard for non-famous FL organizers to attract enough data providers to engage the system, making their work hard to proceed since the FL need multiple participants working collaboratively, especially for some data-intensive training tasks. Even if an incentive mechanism was deployed in an existing FL system, it is still hard to extend the system dynamically and easily, especially for individual data providers because of the problem of permission management, network setup, and so on.

The aforementioned deficiencies of conventional FL are preventing FL systems from working reliably and efficiently, it is critical to make essential progress to the conventional FL frameworks. Considering the challenges the FL faces, a decentralized consensus system with a fair incentive mechanism, which reminds FL developers of the blockchain, can help to solve the problems. Blockchain was first proposed in Bitcoin by Satoshi Nakamoto in 2008 [23], which is a decentralized ledger maintained by all the participants according to a predefined consensus protocol. Due to its basic design idea, it can provide several attractive features, such as decentralization, anonymity, auditability, persistency, and so on. As a technique proposed for the digital payment system, blockchain also has its own incentive mechanisms to simulate the process of currency issuance and transaction. Since its appearance, blockchain has been put into application in various fields [24–26], and many variants of the original blockchain in Bitcoin were proposed with a lot of new functional features. Hence, the concept of Blockchain-based Federated Learning is proposed. The disadvantages mentioned above can be solved easily by utilizing the blockchain in the conventional FL system. First, blockchain can make FL decentralized, which means the central aggregator can be replaced by the blockchain network and the aggregation job can be executed by the nodes. In addition, it is easier for a new data provider to join the blockchain network and extend the scale of the participants since the procedure of granting access is simplified and nearly no bottleneck of communication meanwhile. Moreover, some attacks, such as fake data or a limited range of malicious nodes can be avoided. Even though facing more complex attacks, the state-of-the-art modular blockchain platform can make it easier to design defense tools in a form of plugins, which means the cryptographic algorithms and the FL algorithms can be separated and do not interfere with each other. Furthermore, blockchain's incentive mechanisms can help FL systems to distribute rewards to nodes fairly to encourage their participation. The following summarizes the advantages of the BCFL:

- Decentralized FL can be easily achieved, and the central server is no longer needed, thus avoiding the single-point failure and making it easy to join the FL system.
- Higher attack resistance. Due to the validation mechanism and modular design of blockchain, more cryptographic algorithms can be used to defend against different types of attacks.
- Incentive mechanisms can encourage participants in the network to obey the rules and attract more data providers to join.

From the existing research, while the BCFL appears to be practical and efficient, some problems in the conventional FL remain unsolved, such as communication delays and so on. Although blockchain is not some kind of silver bullet to the challenges the FL system faces, it provides a new and practical direction to solve them.

The study contributions can be summarized as follows:

1. It compares the conventional FL with BCFL, discusses the motivation for integrating blockchain and FL, and demonstrates the challenges that BCFL can face in future research.
2. It divides the existing research of BCFL into several types based on the cryptographic tools they used and provides performance analysis.
3. It investigates the application scenarios of BCFL and analyzes its advantages.

The rest of this paper is organized as follows. Section 2 introduces the background knowledge of blockchain and FL. Section 3 discusses the motivation for integrating blockchain and FL. Section 4 demonstrates the challenges that BCFL may face in future research. Section 5 investigates the state-of-the-art BCFL research. Section 6 introduces the current application of BCFL. Section 7 presents the performance analysis of the BCFL schemes mentioned in the above sections. Section 8 draws the conclusion.

2 Background

In this section, we will briefly introduce basic knowledge and principles of Federated Learning (FL) and Blockchain.

2.1 Brief Introduction to FL

With the rapid development of computer storage capacity and progress ability, smart devices have become smaller and cheaper, which means devices, such as laptops, smartphones, tablets, and so on can be used or installed by most people [27] and can be connected to the network from anywhere at any time. This leads to an apparent consequence that a massive amount of data is being generated every day, and we can find that this data which may be privately sensitive is perfect to be fed to Machine Learning (ML) models to solve real-life tasks [28]. Though in recent years, artificial intelligence (AI) algorithms have been developed considerably, it is still a problem to process such vast amounts of non-independent and identically distributed (Non-IID) decentralized data on a centralized algorithm training facility [29–32], let alone the issue of data privacy and security. Thus, Google proposed a distributed ML framework termed FL to solve the issues mentioned above [8,33–35].

FL is a distributed ML technique that allows clients in the framework to contribute to the model without uploading their raw private data. The local devices can train data locally and then upload the model updates, such as the gradients. FL consists of several roles, which are shown in Fig. 1. In practical application protocol [35], there exists N clients (C_1, C_2, ..., C_n) which hold their own dataset (D_1, D_2, ..., D_n), and they are not willing to share the raw data and can not access others' data meanwhile, and there is a central server which also termed aggregator. The basic workflow of each round in a traditional FL can be summarized as below [8]:

1. **Clients Selection.** A subset of existing clients is selected, each downloading the current model from a central server.
2. **Clients Train Model Locally.** Selected clients train the model locally based on their own private data with the pre-selected algorithm.
3. **Upload Local Training Updates.** Clients upload the local model updates to the central server to aggregate.
4. **Global Model Aggregation.** The central server aggregates these uploaded models (typically by averaging) to update the global model.

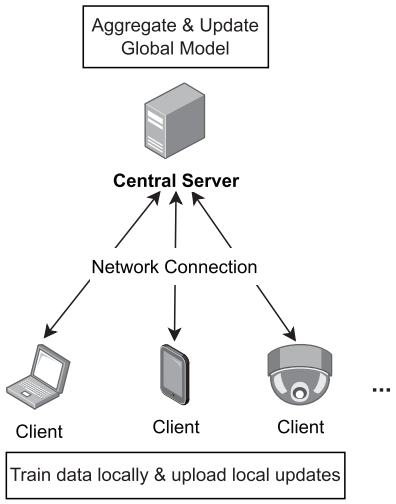

Figure 1: Traditional FL architecture

Various FL frameworks are usually divided by how data is distributed among various parties in the feature and sample ID space. The basic workflow of these frameworks can have differences due to the data distribution. They can be divided into the categories as follows, and Fig. 2 shows the difference between these three classifications [36]:

- **Horizontal FL.** In this case, clients' data have the same structure, in other words, the data sets share the same feature space but are different in samples. In horizontal FL, clients' private information may be leaked during the progress of uploading local model updates or aggregation, and there are advanced schemes using cryptographic techniques, such as Differential Privacy (DP) [37–39], Homomorphic Encryption (HE) [21,40] and so on.
- **Vertical FL.** This type of FL suits the case that the datasets share the same sample space but vary in feature space. Under this mechanism, the federated system needs to aggregate different features from these datasets and build a model with a "commonwealth" strategy in a privacy-preserving manner. Algorithms for vertical partitioned data were proposed including classification [41], gradient descent [42], secure linear regression [43–45], cooperative statistic analysis [46] and so on.
- **Federated Transfer Learning (FTL).** FTL is applied to the case that datasets differ not only in samples but also in feature space. In this case, transfer learning [47,48] can be used to overcome the lack of data or labels without slicing the data. Frameworks of FTL have been proposed in recent years to solve various problems, e.g., [49–51], and extend the typical two-party protocol to multi-party.

Though FL sounds like a pretty good technique since it can handle scenarios of different data distributions, it faces several challenges [52]. Currently presented, FL focuses on improving communication efficiency, compatibility with heterogeneous systems and networks, and privacy-preserving problems [31]. What we can not deny is that these are core challenges for FL, but there are still other concerns if FL is applicable for real-life deployment. These concerns are part of the motivation for integrating blockchain and FL, and a discussion about them will be presented in detail in Section 3.

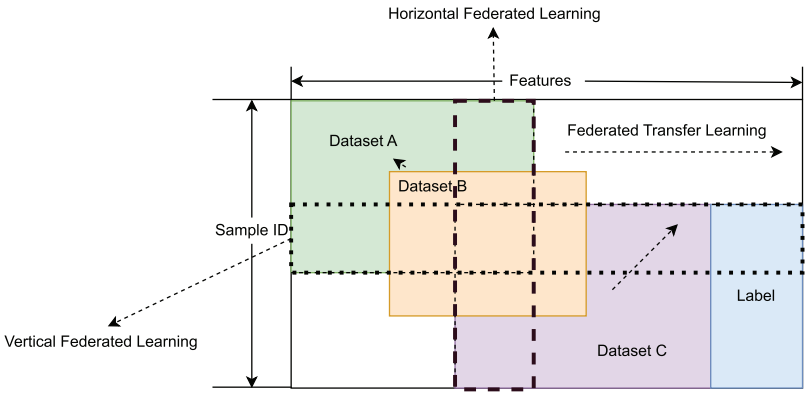

Figure 2: Categories of federated learning [36]

2.2 Brief Introduction to Blockchain

In 2008, Satoshi Nakamoto published Bitcoin [23], a peer-to-peer payment system that is totally decentralized and transparent. Today, Bitcoin is the world's largest cryptocurrency. The underlying technology blockchain has attracted great interest and has developed a lot since then. Fig. 3 shows a classic structure of a block in the blockchain.

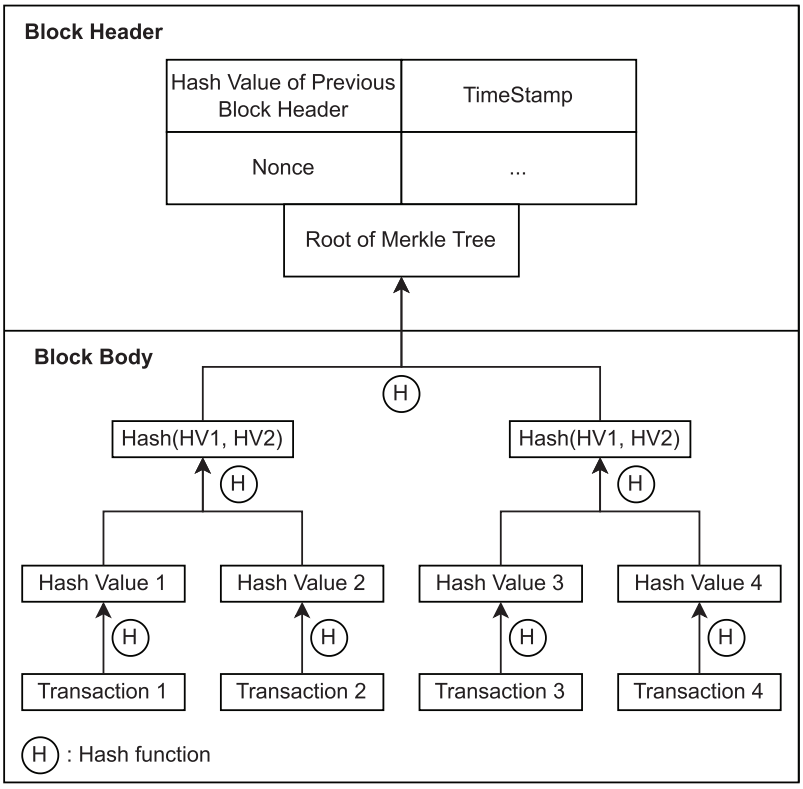

Figure 3: Classical structure of a block in blockchain

Blockchain is a distributed ledger, in which the participants who generate their Proof-of-Contribution are termed miners. Each miner keeps one replica of the whole ledger on its local device. A block consists of two parts: block header and block body. The block header contains information, such as the hash value of the previous block header, timestamp, the consensus protocol this blockchain uses, etc. The block body stores the transactions package, of which the hash values are computed in the form of a Merkle tree and the root is stored in the header as well. A blockchain starts from the genesis block where the initial parameters of this blockchain are set and stored, for example, the incentive mechanism and the total number of tokens. Then the miners compete to win the opportunity to generate a new candidate block, which is broadcast to all miners to reach a consensus according to a chosen protocol. Once the candidate block is confirmed, the block is appended to the end of the blockchain and the miner can get rewards, such as tokens if an incentive mechanism is set.

Since the advent of blockchain, many blockchain variants have been created to solve problems under different scenarios [53]. The original bitcoin organized data as a single "chain" by utilizing the block hash values, but later proposed blockchain architectures have extended to parallel chains [54,55] and graphs [56,57]. What is not changing is the characteristic of decentralization. Typically, blockchain can be categorized into 3 types [58] based on what kind of permission the user needs to join the blockchain network:

- **Public Blockchain.** Public blockchain is open to everyone without permission. It is totally decentralized and free of third-party authority institutions, every user in the blockchain network can access the ledger and participate in the progress of consensus, which means every node is equal in the network as shown in Fig. 4. In this case, there may be a large number of transactions that need proceeding while the speed of new block generation is limited, thus appearing to have a low throughput rate. Bitcoin and Ethreum [59] are two well-known public blockchain platforms.
- **Private Blockchain.** In contrast to the public blockchain, participants in the private blockchain network are under some supervision; only authorized clients can join the blockchain network and access the ledger. Inside the private blockchain, it is similar to the public blockchain. Usually, a private blockchain is a private deployment of the public blockchain, or the source code of the blockchain may be slightly customized to fulfill certain requirements. As the participants are much fewer than in the public blockchain network, it has a high performance of transaction processing speed.
- **Consortium Blockchain.** Consortium blockchain is a specific type of private blockchain. It is partially centralized, the network is controlled by several chosen participants. Only the selected participants can reach a consensus among themselves and generate new blocks. Other users can only access the ledger by the provided service interfaces, as it is shown in Fig. 5. Hyperledger [60] (proposed by IBM, hosted by the Linux Foundation) and Libra (proposed by Facebook) [61,62] are both consortium blockchain platforms.

Because of the architecture, blockchain naturally has these characteristics:

1. **Decentralization.** In a conventional distributed system, state-changing operations or transactions need to be validated by a trusted central authority. In the blockchain, due to the utilization of the P2P network, all participants are equal, and transactions can be conducted between any two peers without authentication according to the consensus protocol.
2. **Anonymity.** Generally, the participants use long bits random numbers as the address in the blockchain network, and one user can have many addresses. Moreover, no central authority monitors the transactions or asks for the users' private information, the real-world identity and the blockchain identity can be separated in most cases.

3. **Auditability.** All transactions conducted in the blockchain are recorded to the distributed ledger and validated with a digital timestamp. Every user in the network can audit and trace every transaction, which brings transparency to the flow of the tokens.
4. **Persistency.** Due to the "chained block" design, all blocks are linked by a one-way hash function. Suppose a malicious user wants to modify any recorded information on the blockchain, which can significantly change the hash value of the block. In that case, it has to change all subsequent blocks along the chain, and this problem is considered to be extremely difficult. In addition, as a distributed system, other participants must confirm block generation so the network can easily detect data falsification. Blockchain is usually considered tamper-proof and immutable for this reason.

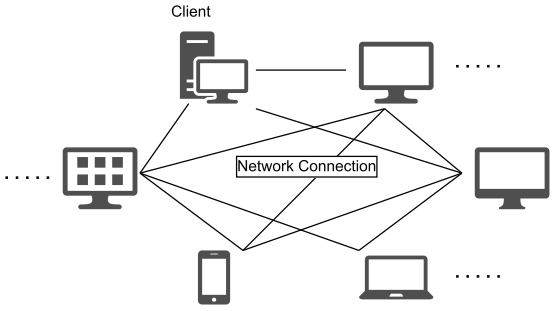

Figure 4: Network of public blockchain

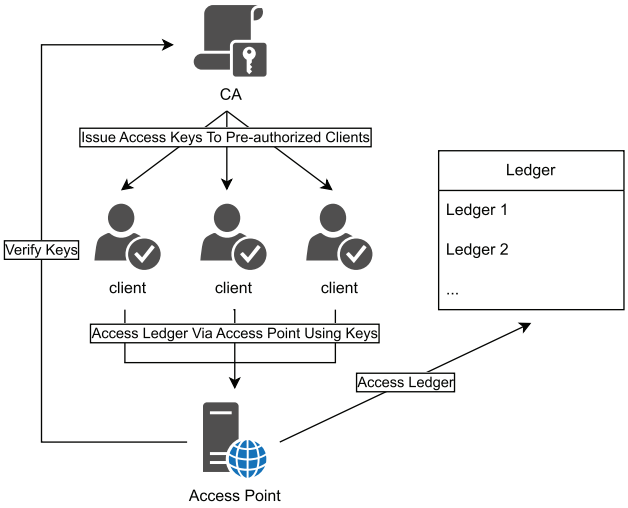

Figure 5: Network of consortium blockchain

3 Motivation

Although FL provides a new solution for machine learning, there are several issues to be addressed when applying it to practical problems.

Single Point of failure In the original version of FL, participants upload their local model updates to a central aggregator, and then the aggregator executes the predefined aggregation algorithm to update the global model. After the aggregation, participants need to download the latest global model

to update. The network communication and request processing load are heavy when there is a large number of participants thus the bottleneck of the server can be reached and affect the performance to a lower level. In a worse case, if the central aggregator fails, so will the whole FL system. Meanwhile, if the central server is compromised, the behavior of the central server is unpredictable and untrustworthy, which also fails the system or even leads to the leakage of participants' private information. So we should adapt mechanisms into the FL systems to avoid single-point failure and audit the behavior of the aggregator.

Privacy breach The main purpose of FL is to protect the privacy of participants and to avoid uploading their private data directly. However, some studies have found that the transmission of gradients can compromise private information [40,63–65]. Though cryptographic techniques, such as HE, DP, ZKP, etc., can be used to avoid privacy leakage, it costs a lot of work to customize the FL system and may make it hard to be compatible with more types of devices.

Malicious client and data Except for a compromised aggregator that can affect the FL system directly, malicious or compromised clients can also affect the global model using constructed data [15,66,67]. Though mechanisms to detect malicious clients are proposed by researchers, such as [68] and [69], they burden the system load since they use another model to check whether the data is fake or not.

Lack of incentive mechanism In conventional FL systems, there is no incentive mechanism and all the system cares about is data collection and model aggregation. It is appropriate for a self-organized FL system in which all the data provider is under the organizer's control. However, in the real-world scenario, it is nearly impossible to implement a well-performing FL model without any outer data provider. Hence, rewards need to be set as an incentive to attract more devices to participate in the system. Research has been conducted to design suitable incentive mechanisms for FL [70,71], approaches include Deep Reinforcement Learning (DRL), Stackelberg Game, etc.

With the above challenges, the FL system needs to be deeply customized which makes it hard to deploy, meanwhile raising the complexity for devices to contribute to the system. As introduced in Section 2.2, blockchain can provide with several attractive features: decentralization, anonymity, auditability, and persistency, which are indeed what the conventional FL needs. Blockchain is naturally decentralized, and by deploying FL on a blockchain platform, the FL system can also achieve decentralization easily and be free of central aggregator which avoids the single point failure problem. In a blockchain network, clients are all anonymous thus avoiding the leakage of clients' private information. Moreover, with the auditability and persistency provided by the blockchain, malicious clients and data can be detected and recorded, which can help the FL system gain attack resistance. The lack of incentive, which is the most important problem preventing FL from being applied to practical scenarios, is easily solved since the cryptocurrency based on blockchain has gained great success and developed a set of fair incentive mechanisms.

Moreover, with the massive interest in blockchain technology due to cryptocurrencies that have significantly impacted on the world economic system over the past few years, blockchain is evolving to become more and more sophisticated and practical. Techniques like blockchain-based smart contracts [72,73] and modular blockchain are developed and put into application. With these blockchain techniques, it can be fast to transplant a conventional FL system to a blockchain platform, and cryptographic tools that are used to defend against attacks can be adapted to the system as a plugin, which means developers can solve one problem at a time instead of changing the whole system. With the above-mentioned development of blockchain technology, the potential combination of blockchain and federated learning can be achieved.

4 Challenges of BCFL

Blockchain seems to provide a perfect solution for improving FL, but it still faces many challenges. An ideal practical BCFL framework should achieve relatively high-security protection ability and privacy-preserving ability while retaining the training efficiency of conventional FL frameworks. However, in real-world scenarios, we always have to make trade-offs between these two goals. The challenges BCFL faces can be summarized below.

Training Efficiency To build a practical FL framework, besides the accuracy of the training model, the indicator of training efficiency is also very important. However, in a blockchain network, many factors can delay the process of training. For example, the physical distance between clients can be extremely large which can cause a high latency of communication thus making the training process very slow. Moreover, in an untrustworthy network environment, it can cost clients a major part of time on data verification. In addition, in some frameworks that adopt cryptographic tools, the efficiency of the current implementation of these tools can be very low, especially for complex cryptographic primitives, such as zero-knowledge proof and homomorphic encryption. All these factors can make a BCFL framework become unpractical in some ways.

System Security and Privacy Though blockchain can provide some great security features naturally, problems still may lower the system's security. Firstly, in some BCFL frameworks based on public blockchain platforms, it is hard to trace the malicious clients since there are no access restrictions and there is no relevance between clients' addresses and their real-world identities. Secondly, the training data is shared on the blockchain, and all participants can access it without any permission, which may cause potential privacy leakage. The blockchain can help achieve data traceability and connect new participants to the system, but it also hinders us from managing the whole system easily.

Reasonable Incentive Mechanisms Bitcoin and Ethereum have shown us successful digital payment systems based on blockchain, but in an FL framework, we can not spend that much computation resources to generate proof of works. While applying incentive mechanisms on a blockchain network, it is essential to design reasonable strategies to distribute the incentives. Several factors must be taken into consideration when considering the characteristics of FL. For example, the amount of data that the client contributes, the training round the client participates in, the data quality of the client's training set, and so on. Algorithms need to be carefully designed since these factors can be highly sensitive.

5 State-of-the-Art BCFL

The possibility of combining blockchain and federated learning has been increasingly investigated by scholars in recent years because of the various shortcomings of federated learning and the problems encountered in solving practical problems. Blockchain is decentralized and blockchain-based applications are mainly immune to single points of failure. Ethereum [74] first implemented smart contracts on the blockchain, where users can execute Turing-complete code, which ensures that the code in question can be executed correctly. So smart contracts on the blockchain can replace the role of central aggregator in the traditional FL.

However, there are practical problems that are not addressed, such as privacy breaches, and malicious clients. In most of the existing BCFL schemes, the solution to these problems is introducing some cryptographic tools into the system. We can categorize these schemes according to the cryptographic tools they use in Section 5.1.

The lack of incentive mechanisms is also an important challenge of the traditional FL. We will make a brief introduction about the existing incentive mechanisms in BCFL schemes in Section 5.2.

5.1 Cryptographic Tools in BCFL

Many studies have applied some cryptographic tools to BCFL, and we will focus on the application of HE (Homomorphic Encryption) and ZKPK (Zero-knowledge Proof of Knowledge), in the current study.

5.1.1 Homomorphic Encryption

Homomorphic encryption is an advanced form of encryption that allows the user to manipulate the ciphertext directly without prior decryption, and the result of manipulating the ciphertext should be the same as the result of decrypting and then manipulating the plaintext before manipulating the ciphertext, which is like equation below. Partial homomorphic encryption is only supported for circuits with only one gate composition in the evaluation circuit, and fully homomorphic encryption supports circuits composed of multiple types of gates and unbound depth.

$$\mathscr{E}(m_1 + m_2) = \mathscr{E}(m_1) \cdot \mathscr{E}(m_2)$$

The most widely used homomorphic encryption method is the Paillier encryption method [75], which is used by [76–79]. The steps of homomorphic encryption can be summarized as follows:

1. $Genkey() \rightarrow \{(n_0, g), (\lambda, \mu)\}$, where (n_0, g) is the public key and (λ, μ) is the secret key. p, q are two large prime number, set $n_0 = pq$, $\lambda = lcm(p-1, q-1)$. Let $L(x) = (x-1)/n_0$, and select a base g, such that $gcd(L(g^{\lambda} \bmod n_0^2), n_0) = 1$
2. $Enc(m) \rightarrow c = g^m \cdot r^{n_0}$
3. $Dec(m) \rightarrow m = L(c^{\lambda} \bmod n_0^2)/L(g^{\lambda} \bmod n_0^2)$

In the scenario used in BCFL, partial homomorphic encryption is sufficient in most cases, only the encrypted gradient needs to be added to the encrypted gradient. If the gradients or distances of models submitted by clients to the blockchain are published, then this data can potentially be used by attackers to infer private data. A naive approach would be to encrypt the data submitted by the client in the form of homomorphic encryption and then aggregate the data in ciphertext form, with [76,77,80,81] encrypting the gradient to be submitted in homomorphic encryption and [78] encrypting the difference between the local model and the global model in homomorphic encryption.

However, the use of homomorphic encryption in BCFL needs to solve a problem, that is, the aggregated gradient value is in the form of ciphertext, which needs to be decrypted before it can be updated to the global model. Decrypting the aggregated gradient requires a private key, which can neither be stored in the general client nor on the blockchain. Reference [77] used secret sharing technology to distribute the private key to all clients. After the aggregation is completed, all clients need to jointly decrypt the aggregated gradient (at least t clients are required to participate) as shown in Fig. 6. As another solution, Wang et al. [78] stored the private key on the CA node. Although this solves the problem, if the CA node fails, the entire network will not work. At the same time, if the CA is malicious, then it can decrypt all gradient values, which may lead to privacy leakage. Even though fully homomorphic encryption is not practical due to its computational and expansion ciphertext, novelty work [81] still adopts it to aggregate gradients. In [81], the authors used the CKKS scheme [82] to encrypt the local gradient in parallel in order to ameliorate the computational efficiency, and they proved that fully homomorphic encryption is more efficient and more suitable than Paillier in their circumstances.

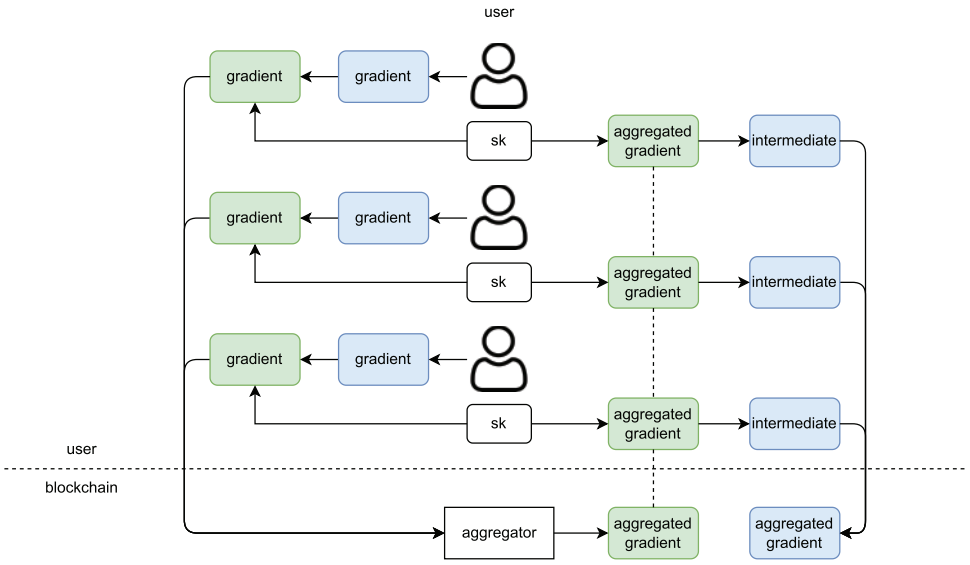

Figure 6: Homomorphic encryption with secret sharing

In addition to safely aggregating gradient values, homomorphic encryption can also be used to build two-party secure computations for neural networks. Reference [83] proposed a method to securely make a label prediction for neural networks that do not require the model owner to reveal the weight value of the model and the sample owner to reveal the value of the sample. Every node in a neural network can be represented as $f(\mathbf{x}) = \sigma(z), z = \mathbf{x}^T\mathbf{w} + b$, where $\sigma(\cdot)$ is a nonlinear activation function and $\mathbf{x}$, b are weight and bias of this node. The label prediction process is performed interactively, layer by layer. The label prediction process is executed interactively layer by layer, in each layer, the data owner sends $\mathscr{E}(\mathbf{x})$ to the model holder, and the model holder computes $\mathscr{E}(z)$ homomorphically and sends it to the data holder. The data holder computes $\sigma(z)$ after decrypting it, thus obtaining the output of this neural network node. The above process is shown in Fig. 7.

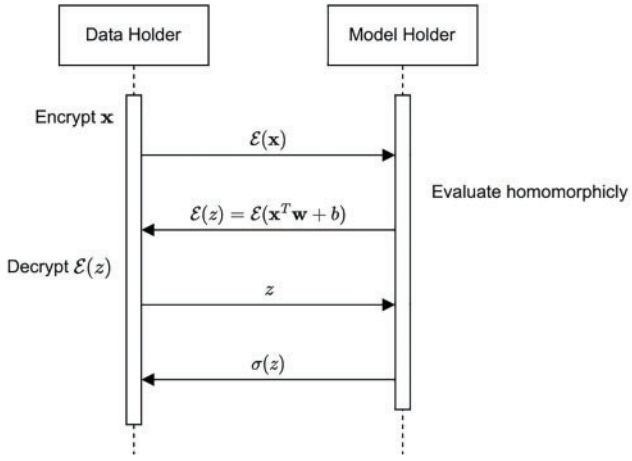

Figure 7: Neural network evaluation in [84]

5.1.2 Zero-Knowledge Proof

In cryptography, Zero-knowledge Proof of Knowledge is a method that allows one party (the prover) to prove to the other party (the verifier) that a statement is correct, without revealing any information other than that the statement is correct.

1. **Completeness** If the statement is true, an honest prover will convince the honest verifier with a very high possibility.
2. **Soundness** If the statement is false, no malicious prover can convince the honest verifier in polynomial time.
3. **Zero-knowledge** A verifier can learn nothing else besides the statement being true.

In recent years, more and more non-interactive zero-knowledge proof frameworks have been proposed, including Bulletproof [85], STARKs [86], and domain-specific languages, such as Circom and Zokrates have emerged that can compile programs into arithmetic circuits. In 2016, Ben Sasson [87] first introduced zk-SNARKs to the blockchain, a technology that greatly protects the privacy of traders. From there, zero-knowledge proof technology is a natural fit with blockchain technology because all data is public in a blockchain, and if zero-knowledge proofs are used, both participants' privacy is protected. All other participants can verify the generated proofs.

According to [88], a malicious client may interfere with the accuracy of the global model or plant a backdoor in the model by submitting a false gradient. In this case, we need to prove without revealing the real samples that the gradients submitted by the client were indeed generated by the real samples through correct computation.

In this case, many BCFLs choose to use zero-knowledge proofs to prove that the gradient submitted by the client was indeed obtained by the sample through the correct algorithm, a process mostly similar to the one shown in Fig. 8.

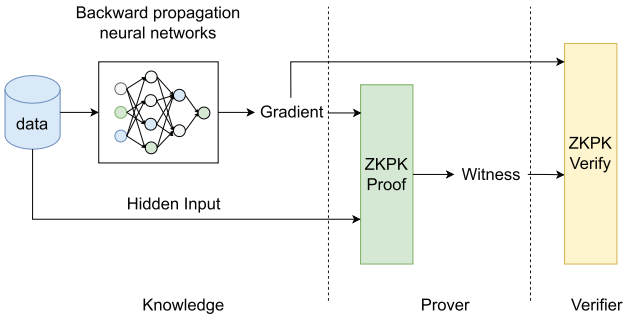

Figure 8: Zero-knowledge proof of knowledge in FL

Both references [89–91] used zero-knowledge proof methods to ensure that the submitted gradient values are from the real dataset, reference [90] used zero-knowledge proof techniques to construct a zk-Trainer to output gradient values as well as proofs simultaneously; reference [91] mentioned many computational details on this basis, including how to efficiently compute matrix operations in the gradient solution process, how to introduce auxiliary variables to simplify the computational process, and how to trade-off computational accuracy with computational complexity. Both solutions mention the problem that needs to be solved when applying zk-SNARK to BCFL. zk-SNARK's arithmetic operations are performed in a finite field, which means that the numbers involved in the operations must be positive integers, but the numbers handled in machine learning are generally floating-point

numbers. The general solution is to scale the floating point number by a specific multiple and then round it to simulate a floating point number. Also, a boolean variable is employed to indicate the positive or negative of the value.

The existing schemes only guarantee that the generated gradient is honestly generated according to the predefined machine learning algorithm on a confidential dataset, which ensures that the client cannot carry out a model poisoning attack. But this can not prevent data poisoning attacks, which require proof that the source of the dataset is trustworthy. Reference [91] mentioned that this can be achieved by further proving that the data comes from a certified sensor.

5.1.3 Differential Privacy

Differential privacy is a commonly used method to protect privacy in federation learning, where the derived local differential privacy is a random perturbation added to the gradient submitted by the client to the central node, which can effectively protect the client's privacy from being compromised. If when a random function $\mathscr{M}$, its input value x, and output value v^* that is a perturbed value for x satisfy:

$$\Pr[\mathscr{M}(x) = v^*] \leq e^{\varepsilon} \cdot \Pr[\mathscr{M}(x') = v^*] + \delta$$

where $\mathscr{M}$ is a random algorithm $\mathscr{M} : \mathbb{X} \to \mathbb{V}$ which domain is $\mathbb{X}$ and range $\mathbb{V} \subseteq \mathbb{X}$, two inputs $x, x' \in \mathbb{X}$, output $v^* \in \mathbb{V}$, $\Pr[\cdot]$ is the conditional probability density function, ε represents the privacy budget, and δ is a positive small number, then this random function $\mathscr{M}$ is an (ε, δ)-LDP. The lower the value of the privacy budget, the better the privacy protection and the less accuracy is retained.

In the application scenario of BCFL, if cryptographic methods, such as HE are not applied to protect the gradients, the gradients submitted by all participants are exposed on the blockchain. In this case, if an attacker applies inference attacks on these exposed gradients, there is an obvious risk of participants' privacy leakage.

Many schemes, such as references [86,91–94] used LDP to add noise to the submitted gradients, where references [84,91,92] added Laplace noise to the gradients and references [93,94] added Gaussian noise to the gradients. Fig. 9 depicts the DP-based federal learning scheme. The overall scheme consists of five steps, two of which are optional and are shown as dashed lines. According to the two optional steps, LDP and central DP are two classical methods. The former aims to protect the gradient of the local models, while the latter tends to protect the gradient of the global model. Since the core principles and ideas are very similar for both methods, and the main difference is merely the location where the perturbation is added in, we take LDP as an exmaple to discuss the problem in DP. The problem that needs to be solved with LDP is how to determine the privacy budget, which is a trade-off between privacy protection and data accuracy. To determine the privacy budget, an adaptive LDP algorithm was implemented using the RMSProp optimization algorithm [94]; the WGAN algorithm was used to determine the appropriate privacy budget [92].

5.2 Incentive in BCFL

The earliest blockchain to be proposed was Bitcoin [23], and incentives were an important factor in Bitcoin's ability to succeed; the probability that an honest participant would be rewarded was higher than the probability that they would be able to get by faking. All proposed public chains need corresponding incentives to reward honest participants so that the majority of participants are honest and thus the blockchain system can function properly. If federated learning is applied to the public chain, then we also need to address the incentive problem of the participants. On the one hand, only by

giving enough incentives to honest participants can we encourage them to continue contributing to FL; on the other hand, we need corresponding penalties to punish falsifying and dishonest participants.

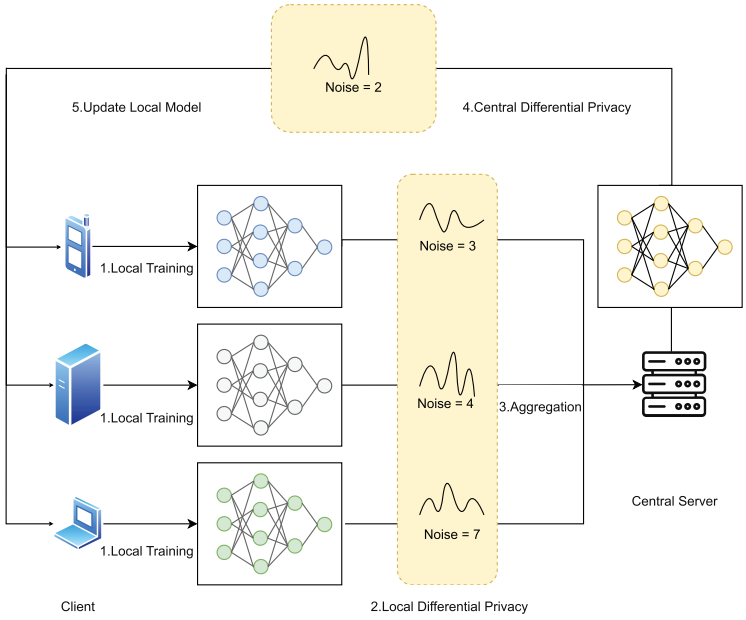

Figure 9: Differential privacy in federated learning

In blockchains, such as Bitcoin, the method of verifying client behavior is deterministic and only requires checking that the hash in the block is legitimate and that the execution of business logic, such as Bitcoin scripts and smart contracts contained within it is correct. But for BCFL, we do not have a natural way to determine whether the client is honest or not. So how to determine whether a client is honest or not, and how to compare the differences in contributions between different clients will be crucial issues that need to be addressed.

Reference [84] combined multi-KRUM [95] and reputation-based incentive protocols [96] to propose incentive protocols that can resist poisoning attacks and motivate honest participants. In the protocol, the multi-KRUM algorithm is first executed on all submitted ladders, thus proposing malicious ladders, then each participant is scored based on the Euclidean distance between the ladder submitted by each participant and the ladders submitted by others, and finally the reputation of each participant is adjusted based on the scoring results. Reference [97] proposed a mechanism design, which is a method of designing economic mechanisms or incentives to achieve desired goals in a strategic environment where participants act rationally. A competitive model updating approach is introduced so that any rational worker follows the agreement and maximizes their profits. In each round, participants select the best one of the models submitted by all participants in the previous round, update it based on this model, and submit it. The benefit to the participant will be determined by the votes received for the submitted model in the next round. If an honest participant wants to receive the highest benefit, he or she will honestly vote for the best model and honestly update the model to receive the next round of voting. Reference [98] used a simpler strategy, only checking whether the wrong masking gradient has been uploaded or other dishonest behavior has been performed. In [83], all participants will form mining pools, each pool maintains its own model, and all participants in each pool update and optimize this model through FL. After each training round, the models trained

by all pools will be evaluated, and the pool to which the best model belongs will be incentivized. To this end, model validation without exposing the model and validation dataset was also proposed in [83] outside the FL process. Moreover, different mining pools in [84] are able to auction their models to obtain rewards.

6 Application of BCFL

Since the idea of BCFL was proposed by researchers, the joint technique has attracted interest from various fields and has been applied to real-life applications. In this section, we will introduce the applications of BCFL in some fields according to the current research. To the best of our knowledge, all schemes are designed to function under certain circumstances and there is no general framework for the BCFL yet.

6.1 BCFL in Internet of Things (IoT)

Nowadays, IoT devices can be found everywhere ranging from road monitors to smart furniture. By 2030, according to a prediction, the number of IoT devices may be about 125 million [99]. Billions of people interact with IoT devices every day and a tremendous amount of data is generated, which is a perfect scenario for FL. However, the conventional FL has several disadvantages, and the BCFL can make the data more secure. Most of the BCFL schemes in IoT fields focus on the issues of privacy, resource allocation, communication efficiency, and failure detection.

Reference [100] proposed a blockchain-enabled FL scheme for fog computing, which aims to address the privacy and communication cost issues of the existing works. In [101], it proposed the digital twin edge networks (DITENs) which integrate the digital twins with edge networks to connect the physical edge networks and digital system. It then proposed a blockchain-based FL scheme in DITEN to improve privacy, data protection, and communication efficiency. The work in [102] proposed a scheme for FL in IoT settings based on a chameleon hash scheme with a changeable trapdoor, and instantiate the scheme as a redactable medical blockchain. Reference [103] proposed a blockchain-based FL system for failure detection in the IoT industry, and to solve problems, such as data heterogeneity, it designs algorithms to calculate the distance between data from different clients. This scheme implements failure detection while ensuring the clients' privacy [104].

In addition, federated learning faces the poisoning attack [105]; several works focus on analyzing participants' behavior using blockchain. Reference [106] introduced a blockchain-based hierarchical federated learning for cyber-physical systems, which employs the blockchain to verify and validate the trained models on the edge. Similarly, Al Mallah et al. [107] devised a BCFL scheme, in which the miner in blockchain not only exchange local model update but also monitoring the behavior of all the participants to select reliabel devices. There are many other jobs like this, such as [108,109], etc.

6.2 BCFL in Internet of Vehicle (IoV)

IoV is a key part of IoT, which is composed of lots of in-vehicle sensors and road infrastructure. As a real-time system, security and reliability are the most important features, otherwise, people's lives would be in great danger or even cause death. The data generated in IoV is also valuable and it can help us improve daily traffic, making it safer and more convenient for people to travel, thus schemes proposed in the form of BCFL are prevalent in this field.

Reference [110] proposed a secure FL framework termed SFAC for Unmanned Aerial Vehicles (UAVs) assisting Mobile Crowdsensing (MCS), and designed privacy-preserving algorithms to protect the UAV's privacy while maintaining the model accuracy. Reference [111] highlighted the problem of

data sharing among vehicles, and to relieve the communication load and meanwhile protect individual privacy, a new architecture was proposed based on federated learning, in which blockchain technique was adopted to enhance security and reliability of the data. In the work of [112], a BCFL framework with features of privacy-aware and efficiency was proposed, and researchers performed a detailed simulation and analysis on the framework to find out the advantages and challenges. Reference [113] proposed a heterogeneous blockchain-based hierarchical trust evaluation strategy named BHTE, which could utilize federated learning for 5G-enabled intelligent transporting systems. Many similar frameworks or schemes were proposed for IoV to solve the problem of data sharing and aggregation problem, such as [114–117], etc.

6.3 BCFL in Healthcare

FL has been widely used in healthcare fields in recent years for disease diagnosis, pandemic prediction, and so on. The data of patients are sensitive because the leakage of private information may cause potential discrimination issues. Hence, the idea of BCFL is quite welcomed in healthcare applications, since it can make use of data while protecting patients' privacy.

Reference [118] proposed a blockchain-enabled privacy-preserving FL architecture for smart healthcare, in which users could obtain a well-performing model without uploading their data to a central server. There is also some similar research, such as [119–123] and so on. Reference [124] proposed an architecture based on blockchain and FL for multi-agent systems and provided a new model of agents that could be implemented as a real-time medical data processing system. Lakhan et al. [125] designed a framework termed FL-BETS, which was a BCFL-enabled task scheduling framework, to identify fraud of data and protect privacy at a low resource cost. In [126], BCFL was used to diagnose COVID-19 while protecting patients' privacy, and it could also deal with heterogeneous data.

6.4 BCFL in Finance

As blockchain was first proposed for Bitcoin, a digital payment system, it was originally designed to solve finance-related problems. In recent years, there emerges a lot of types of cryptocurrency, and the trading market of cryptocurrency has become very large. Except for pure blockchain-enabled payment systems, there are also BCFL schemes. For example, Liu et al. [127] proposed FedCoin, which is a P2P blockchain-based payment system for FL. It enables a practical profit distribution solution based on Shapley Value (SV).

BCFL can also be used to solve trading problems in the market [128]. Moreover, BCFL can be applied to address trust issues when several financial institutions need to work collaboratively. In the scenarios of financial investment, customers' information is usually confidential and companies do not want to share it with others. Thus, since BCFL can train models without disclosing the raw data, it can be used in this case and make it easier for the companies to work together.

7 Performance Analysis of Existing BCFL Schemes

We choose some existing BCFL schemes which provide their source code in the works we mentioned in the previous sections and transform their result according to a uniform standard so that we can make comparisons more clearly.

Most of the papers on BCFL do not contain source code, and the source code that we can run successfully contains only simulation experiments, which only shows the results generated by the federated learning models, such as model accuracy and so on, but do not provide the running efficiency of the schemes in the real applications.

Of course, many solutions also demonstrate capabilities beyond the completion of federated learning, such as combating poisoning attacks, discovering and disabling malicious users, preventing single points of failure, and so on. But on the one hand, not all solutions provide the ability to detect malicious users, and on the other hand, they use different methods of poisoning attacks and different means to counteract them. Therefore, it is difficult to compare the other capabilities of these schemes horizontally, and this section only compares the accuracy of the derived models under different schemes.

We want to test all schemes using the same criteria as much as possible. We use the EMNIST [129] dataset for our tests, which is a dataset consisting of $28 * 28$ pixel grayscale images of handwritten characters, extended from the classic dataset MNIST [130], and the LEAF project [131], which is an FL benchmark project. For each scheme, we train the model for 40 rounds and output the accuracy of the model for each round. For each round, 20 participants are added to the training (in some schemes these participants may be divided into different roles). For parameters, such as model, learning rate, etc., we use the default settings in the code of each scheme, and we consider these parameters to be the best configurations obtained by the authors, which are part of the scheme, all with integrity.

For the experiments using CPU for training, an 8-core Intel(R) Xeon(R) Gold 6133 CPU is used, and for the experiments using GPU for training, a Tesla T4 is employed. This study employs Ubuntu 22.02 as the host operating system, and docker is utilized if necessary to simulate the software environment for the experiments.

This study compares VBFL [132] (VBFL in the legend) and a DAG-based FL [133] ("FL Dag" in the legend) and depicts the accuracy of the models (Fig. 10) produced by each of these two schemes over 40 rounds of training. In addition, since FL-Dag is Dag-based, there is no global model, but only a model in each node, and we show the distribution of model accuracy of the nodes in FL-Dag in each training round in Fig. 11. This research records the running time of these two schemes, which is shown in Fig. 12. We calculate the average running time of each training round, it is 90.963 s per round for VBFL, and 19.796 s per round for FL-Dag. The result shows that the chosen schemes can accomplish a relatively high level of accuracy, which means they are practical in real-world scenarios.

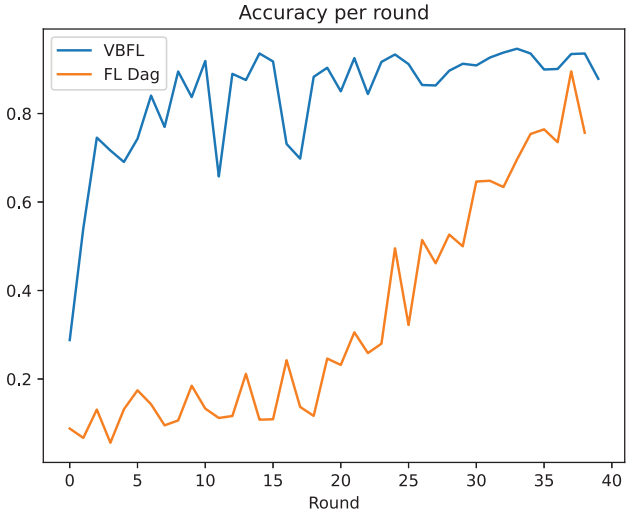

Figure 10: Accuracy per round

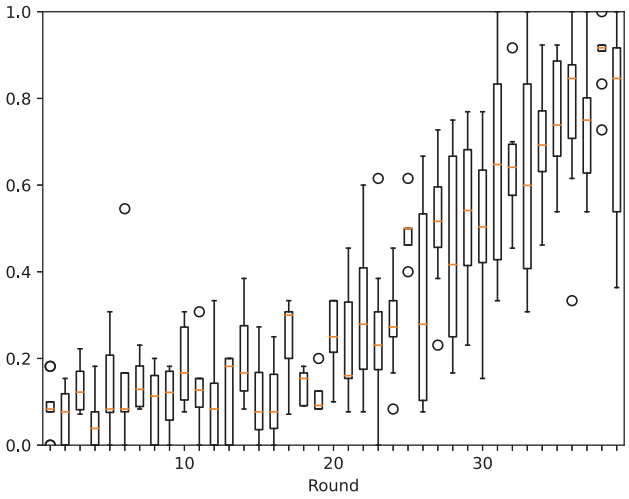

Figure 11: Accuracy per round of [133]

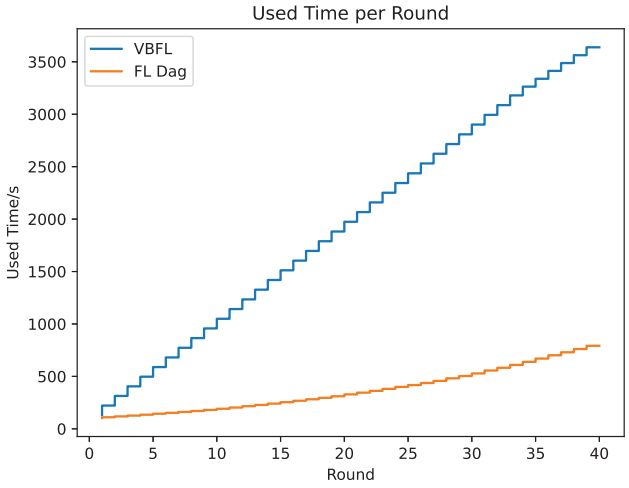

Figure 12: Running time

8 Conclusions

In this paper, we first introduce the basic principles and features of FL and blockchain technology and explain what problems of FL can be solved by combining blockchain technology with it. And we also analyze the problems that still have to be solved in BCFL. We then summarize the state-of-the-art BCFL according to the cryptographic tools it used to utilize solving these problems and their incentive mechanisms. Moreover, we briefly describe real-world applications of BCFLs and conclude with a brief evaluation of some BCFLs.

As a promising collaborative ML training technique, FL is quite promising for the features mentioned in the previous sections, and many developers have already been working collaboratively using FL to solve real-life problems. Blockchain and FL are both recently emerging technologies, and the combination of these two has become a hot research topic today. We believe this technology will become even more promising as people become more concerned about privacy and digital security. We hope this article will serve as a reference for other researchers in this field.

Acknowledgement: The authors would like to acknowledge that this work would not have been possible without the support of Qianxin Technology Group Co., Ltd.

Funding Statement: This paper is supported by High-performance Reliable Multi-Party Secure Computing Technology and Product Project for Industrial Internet No. TC220H056.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization: Z.Z.; Literature review: Y.Z., Y.T. and Z.L.; Formal analysis: Y.Z., Y.T.; Funding acquisition: H.C. and L.Z.; Methodology: Z.Z., T.N.; Supervision: Z.Z.; Validation: Y.T. and Z.L.; Visualization: Z.L.; Writing of draft: Y.Z., Y.T. and Z.L.; Writing of review and editing, Z.Z., N.T., H.C. and L.Z. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: This paper is predominantly a review that synthesizes existing methods and literature findings. This investigation utilized only data obtained from publicly accessible sources. These data are accessible via the sources listed in the References section of this paper.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science, 2(3),* 160.
2. Marchand, A., Marx, P. (2020). Automated product recommendations with preference-based explanations. *Journal of Retailing, 96(3),* 328–343.
3. Gao, H., Huang, J., Tao, Y., Hussain, W., Huang, Y. (2022). The joint method of triple attention and novel loss function for entity relation extraction in small data-driven computational social systems. *IEEE Transactions on Computational Social Systems, 9(6),* 1725–1735.
4. Otter, D. W., Medina, J. R., Kalita, J. K. (2021). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems, 32(2),* 604–624.
5. Gao, H., Dai, B., Miao, H., Yang, X., Barroso, R. J. D. et al. (2023). A novel gapg approach to automatic property generation for formal verification: The GAN perspective. *ACM Transactions on Multimedia Computing, Communications and Applications, 19(1),* 1–22.
6. Fatima, M., Pasha, M. et al. (2017). Survey of machine learning algorithms for disease diagnostic. *Journal of Intelligent Learning Systems and Applications, 9(1),* 1.
7. Lalmuanawma, S., Hussain, J., Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for COVID-19 (SARS-CoV-2) pandemic: A review. *Chaos, Solitons & Fractals, 139,* 110059.
8. Konečny, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T. et al. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.
9. Aledhari, M., Razzak, R., Parizi, R. M., Saeed, F. (2020). Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access, 8,* 140699–140725.
10. Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R. et al. (2020). The future of digital health with federated learning. *npj Digital Medicine, 3(1),* 119.
11. Liu, Y., Huang, A., Luo, Y., Huang, H., Liu, Y. et al. (2020). FedVision: An online visual object detection platform powered by federated learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 8.
12. Chen, S., Xue, D., Chuai, G., Yang, Q., Liu, Q. (2021). FL-QSAR: A federated learning-based QSAR prototype for collaborative drug discovery. *Bioinformatics, 36(22–23),* 5492–5498.

13. Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J. et al. (2021). Federated learning for Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials, 23(3),* 1622–1658.
14. Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q. et al. (2019). Beyond inferring class representatives: User-level privacy leakage from federated learning. *IEEE INFOCOM 2019–IEEE Conference on Computer Communications*, Paris, France.
15. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V. (2020). How to backdoor federated learning. *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, vol. 108, pp. 2938–2948.
16. Nasr, M., Shokri, R., Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA.
17. Zhang, J., Zhang, J., Chen, J., Yu, S. (2020). Gan enhanced membership inference: A passive local attack in federated learning. *ICC 2020–2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland.
18. Ouadrhiri, A. E., Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey. *IEEE Access, 10,* 22359–22380.
19. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H. et al. (2019). A hybrid approach to privacy-preserving federated learning. *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3338501.3357370
20. Li, Y., Zhou, Y., Jolfaei, A., Yu, D., Xu, G. et al. (2021). Privacy-preserving federated learning framework based on chained secure multiparty computing. *IEEE Internet of Things Journal, 8(8),* 6178–6186.
21. Zhang, C., Li, S., Xia, J., Wang, W., Yan, F. et al. (2020). BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning. *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020)*.
22. Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G. et al. (2017). Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. arXiv preprint arXiv:1711.10677.
23. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review,* 21260.
24. Ng, W. Y., Tan, T. E., Movva, P. V. H., Fang, A. H. S., Yeo, K. K. et al. (2021). Blockchain applications in health care for COVID-19 and beyond: A systematic review. *The Lancet Digital Health, 3(12),* e819–e829.
25. Pournader, M., Shi, Y., Seuring, S., Koh, S. L. (2020). Blockchain applications in supply chains, transport and logistics: A systematic review of the literature. *International Journal of Production Research, 58(7),* 2063–2081.
26. Alladi, T., Chamola, V., Parizi, R. M., Choo, K. K. R. (2019). Blockchain applications for Industry 4.0 and industrial IoT: A review. *IEEE Access, 7,* 176935–176951.
27. Poushter, J. (2016). Smartphone ownership and Internet usage continues to climb in emerging economies. *Pew Research Center, 22(1),* 1–44.
28. Gao, H., Xiao, J., Yin, Y., Liu, T., Shi, J. (2022). A mutually supervised graph attention network for few-shot segmentation: The perspective of fully utilizing limited samples. *IEEE Transactions on Neural Networks and Learning Systems*, 35286269.
29. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D. et al. (2018). Federated learning with non-iid data. arXiv preprint arXiv:1806.00582.
30. Yang, K., Jiang, T., Shi, Y., Ding, Z. (2020). Federated learning via over-the-air computation. *IEEE Transactions on Wireless Communications, 19(3),* 2022–2035.

31. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M. et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning, 14(1–2),* 1–210.
32. Yu, F., Lin, H., Wang, X., Yassine, A., Hossain, M. S. (2022). Blockchain-empowered secure federated learning system: Architecture and applications. *Computer Communications, 196,* 55–65.
33. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B. A. Y. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282.
34. Konečny, J., McMahan, H. B., Ramage, D., Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527.
35. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B. et al. (2017). Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3133956.3133982
36. Yang, Q., Liu, Y., Chen, T., Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST), 10(2),* 1–19.
37. Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H. et al. (2020). Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security, 15,* 3454–3469.
38. Truex, S., Liu, L., Chow, K. H., Gursoy, M. E., Wei, W. (2020). LDP-Fed: Federated learning with local differential privacy. *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pp. 61–66.
39. Triastcyn, A., Faltings, B. (2019). Federated learning with bayesian differential privacy. *2019 IEEE International Conference on Big Data (Big Data)*, IEEE.
40. Trieu Phong, L., Aono, Y., Hayashi, T., Wang, L., Moriai, S. (2017). Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security, 13(5),* 1333–1345.
41. Du, W., Han, Y. S., Chen, S. (2004). Privacy-preserving multivariate statistical analysis: Linear regression and classification. *Proceedings of the 2004 SIAM International Conference on Data Mining*, SIAM.
42. Wan, L., Ng, W. K., Han, S., Lee, V. C. S. (2007). Privacy-preservation for gradient descent methods. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/1281192.1281275
43. Gascón, A., Schoppmann, P., Balle, B., Raykova, M., Doerner, J. et al. (2016). Secure linear regression on vertically partitioned datasets. *IACR Cryptology ePrint Archive, 2016,* 892.
44. Karr, A. F., Lin, X., Sanil, A. P., Reiter, J. P. (2009). Privacy-preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics, 25(1),* 125.
45. Sanil, A. P., Karr, A. F., Lin, X., Reiter, J. P. (2004). Privacy preserving regression modelling via distributed computation. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/1014052.1014139
46. Du, W., Atallah, M. (2001). Privacy-preserving cooperative statistical analysis. *Proceedings of the 17th Annual Computer Security Applications Conference*, USA, IEEE Computer Society.
47. Pan, S. J., Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, 22(10),* 1345–1359.
48. Chen, J., Ying, H., Liu, X., Gu, J., Feng, R. et al. (2020). A transfer learning based super-resolution microscopy for biopsy slice images: The joint methods perspective. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 18(1),* 103–113.

49. Liu, Y., Kang, Y., Xing, C., Chen, T., Yang, Q. (2020). A secure federated transfer learning framework. *IEEE Intelligent Systems, 35(4),* 70–82.
50. Gao, D., Liu, Y., Huang, A., Ju, C., Yu, H. et al. (2019). Privacy-preserving heterogeneous federated transfer learning. *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA.
51. Chen, Y., Qin, X., Wang, J., Yu, C., Gao, W. (2020). FedHealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems, 35(4),* 83–93.
52. Li, T., Sahu, A. K., Talwalkar, A., Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine, 37(3),* 50–60.
53. Natoli, C., Yu, J., Gramoli, V., Esteves-Verissimo, P. (2019). Deconstructing blockchains: A comprehensive survey on consensus, membership and structure. arXiv preprint arXiv:1908.08316.
54. Kokoris-Kogias, E., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L. et al. (2016). Enhancing bitcoin security and performance with strong consistency via collective signing. arXiv preprint arXiv:1602.06997.
55. Yu, J., Kozhaya, D., Decouchant, J., Esteves-Verissimo, P. (2019). Repucoin: Your reputation is your power. *IEEE Transactions on Computers, 68(8),* 1225–1237.
56. Baird, L. (2016). The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance. *Swirlds Tech Reports SWIRLDS-TR-2016-01*.
57. Popov, S. (2018). The tangle. *White Paper, 1(3),* 30.
58. Wang, J., Wu, P., Wang, X., Shou, W. (2017). The outlook of blockchain technology for construction engineering management. *Frontiers of Engineering Management, 1,* 71–79.
59. Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper, 151,* 1–32.
60. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K. et al. (2018). Hyperledger fabric: A distributed operating system for permissioned blockchains. *Proceedings of the Thirteenth EuroSys Conference*, New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3190508.3190538
61. Brühl, V. (2020). Libra—A differentiated view on facebook's virtual currency project. *Intereconomics, 55(1),* 54–61.
62. Baudet, M., Ching, A., Chursin, A., Danezis, G., Garillot, F. et al. (2019). State machine replication in the libra blockchain. *Technical Report*.
63. Su, L., Xu, J. (2018). Securing distributed machine learning in high dimensions. arXiv preprint arXiv:1804.10140, 1536–1233.
64. Melis, L., Song, C., de Cristofaro, E., Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. *2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, IEEE.
65. Zhu, L., Liu, Z., Han, S. (2019). Deep leakage from gradients. In: *Advances in neural information processing systems*.
66. Fang, M., Cao, X., Jia, J., Gong, N. Z. (2020). Local model poisoning attacks to byzantine-robust federated learning. *Proceedings of the 29th USENIX Conference on Security Symposium*.
67. Xie, C., Koyejo, O., Gupta, I. (2020). Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation. *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*. vol. 115, pp. 261–270. https://proceedings.mlr.press/v115/xie20a.html
68. Li, S., Cheng, Y., Wang, W., Liu, Y., Chen, T. (2020). Learning to detect malicious clients for robust federated learning. arXiv preprint arXiv:2002.00211.
69. Zhang, Z., Cao, X., Jia, J., Gong, N. Z. (2022). FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. *Proceedings of the 28th ACM SIGKDD Conference on*

Knowledge Discovery and Data Mining, New York, NY, USA, Association for Computing Machinery. https://doi.org/10.1145/3534678.3539231

70. Zhan, Y., Zhang, J., Hong, Z., Wu, L., Li, P. et al. (2022). A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing, 10(2),* 1035–1044.
71. Khan, L. U., Pandey, S. R., Tran, N. H., Saad, W., Han, Z. et al. (2020). Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Communications Magazine, 58(10),* 88–93.
72. Hewa, T., Ylianttila, M., Liyanage, M. (2021). Survey on blockchain based smart contracts: Applications, opportunities and challenges. *Journal of Network and Computer Applications, 177,* 102857.
73. Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X. et al. (2019). Blockchain-enabled smart contracts: Architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(11),* 2266–2277.
74. Buterin, V. (2013). Ethereum white paper: A next generation smart contract & decentralized application platform. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper (accessed on 03/06/2020).
75. Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. *Advances in Cryptology-EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques*, Prague, Czech Republic, Springer.
76. Jia, B., Zhang, X., Liu, J., Zhang, Y., Huang, K. et al. (2022). Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT. *IEEE Transactions on Industrial Informatics, 18(6),* 4049–4058.
77. Sun, J., Wu, Y., Wang, S., Fu, Y., Chang, X. (2022). Permissioned blockchain frame for secure federated learning. *IEEE Communications Letters, 26(1),* 13–17.
78. Wang, N., Yang, W., Guan, Z., Du, X., Guizani, M. (2021). BPFL: A blockchain based privacy-preserving federated learning scheme. *2021 IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain.
79. Fan, M., Yu, H., Sun, G. (2021). Privacy-preserving aggregation scheme for blockchained federated learning in IoT. *2021 International Conference on UK-China Emerging Technologies (UCET)*, Chengdu, China.
80. Qi, M., Wang, Z., Wu, F., Hanson, R., Chen, S. et al. (2021). A blockchain-enabled federated learning model for privacy preservation: System design. In: Baek, J., Ruj, S. (Eds.), *Information security and privacy*. Cham: Springer International Publishing.
81. Miao, Y., Liu, Z., Li, H., Choo, K. K. R., Deng, R. H. (2022). Privacy-preserving byzantine-robust federated learning via blockchain systems. *IEEE Transactions on Information Forensics and Security, 17,* 2848–2861.
82. Cheon, J. H., Kim, A., Kim, M., Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security*, Hong Kong, China, Springer.
83. Qu, X., Wang, S., Hu, Q., Cheng, X. (2021). Proof of federated learning: A novel energy-recycling consensus algorithm. *IEEE Transactions on Parallel and Distributed Systems, 32(8),* 2074–2085.
84. Zhao, Y., Zhao, J., Jiang, L., Tan, R., Niyato, D. et al. (2021). Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet of Things Journal, 8(3),* 1817–1829.
85. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P. et al. (2018). Bulletproofs: Short proofs for confidential transactions and more. *2018 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, IEEE.
86. Gennaro, R., Gentry, C., Parno, B., Raykova, M. (2013). Quadratic span programs and succinct nizks without PCPs. *Advances in Cryptology–EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Athens, Greece, Springer.

87. Ben Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I. et al. (2014). Zerocash: Decentralized anonymous payments from bitcoin. *2014 IEEE Symposium on Security and Privacy*.
88. Lyu, L., Yu, H., Yang, Q. (2020). Threats to federated learning: A survey. arXiv preprint arXiv:2003.02133.
89. Mahmood, Z., Jusas, V. (2021). Implementation framework for a blockchain-based federated learning model for classification problems. *Symmetry, 13(7)*.
90. Heiss, J., Grünewald, E., Tai, S., Haimerl, N., Schulte, S. (2022). Advancing blockchain-based federated learning through verifiable off-chain computations. *2022 IEEE International Conference on Blockchain (Blockchain)*, Espoo, Finland.
91. Rückel, T., Sedlmeir, J., Hofmann, P. (2022). Fairness, integrity, and privacy in a scalable blockchain-based federated learning system. *Computer Networks, 202,* 108621.
92. Wan, Y., Qu, Y., Gao, L., Xiang, Y. (2022). Privacy-preserving blockchain-enabled federated learning for B5G-driven edge computing. *Computer Networks, 204,* 108671.
93. Qi, Y., Hossain, M. S., Nie, J., Li, X. (2021). Privacy-preserving blockchain-based federated learning for traffic flow prediction. *Future Generation Computer Systems, 117,* 328–337.
94. Chang, Y., Fang, C., Sun, W. (2021). A blockchain-based federated learning method for smart healthcare. *Computational Intelligence and Neuroscience, 2021,* 4376418.
95. Blanchard, P., El Mhamdi, E. M., Guerraoui, R., Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems, 30*.
96. Zhang, Y., Van der Schaar, M. (2012). Reputation-based incentive protocols in crowdsourcing applications. *2012 Proceedings IEEE INFOCOM*, Orlando, FL, USA, IEEE.
97. Toyoda, K., Zhang, A. N. (2019). Mechanism design for an incentive-aware blockchain-enabled federated learning platform. *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA.
98. Bao, X., Su, C., Xiong, Y., Huang, W., Hu, Y. (2019). Flchain: A blockchain for auditable federated learning with trust and incentive. *2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*, Qingdao, China.
99. Campbell, M. (2019). Smart edge: The effects of shifting the center of data gravity out of the cloud. *Computer, 52(12),* 99–102.
100. Qu, Y., Gao, L., Luan, T. H., Xiang, Y., Yu, S. et al. (2020). Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet of Things Journal, 7(6),* 5171–5183.
101. Lu, Y., Huang, X., Zhang, K., Maharjan, S., Zhang, Y. (2021). Communication-efficient federated learning and permissioned blockchain for digital twin edge networks. *IEEE Internet of Things Journal, 8(4),* 2276–2288.
102. Wei, J., Zhu, Q., Li, Q., Nie, L., Shen, Z. et al. (2022). A redactable blockchain framework for secure federated learning in Industrial Internet of Things. *IEEE Internet of Things Journal, 9(18),* 17901–17911.
103. Zhang, W., Lu, Q., Yu, Q., Li, Z., Liu, Y. et al. (2021). Blockchain-based federated learning for device failure detection in industrial IoT. *IEEE Internet of Things Journal, 8(7),* 5926–5937.
104. Gao, H., Qiu, B., Barroso, R. J. D., Hussain, W., Xu, Y. et al. (2022). TSMAE: A novel anomaly detection approach for Internet of Things time series data using memory-augmented autoencoder. *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 5, pp. 2978–2990.
105. Tolpegin, V., Truex, S., Gursoy, M. E., Liu, L. (2020). Data poisoning attacks against federated learning systems. *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020*, Guildford, UK, Springer.
106. Aloqaily, M., Al Ridhawi, I., Karray, F., Guizani, M. (2022). Towards blockchain-based hierarchical federated learning for cyber-physical systems. *2022 International Balkan Conference on Communications and Networking (BalkanCom)*, Sarajevo, Bosnia and Herzegovina.

107. Al Mallah, R., López, D., Halabi, T. (2023). Blockchain-enabled efficient and secure federated learning in iot and edge computing networks. *2023 International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, IEEE.
108. Moudoud, H., Cherkaoui, S. (2022). Toward secure and private federated learning for iot using blockchain. *GLOBECOM 2022–2022 IEEE Global Communications Conference*, Rio de Janeiro, Brazil.
109. Revathy, S., Priya, S. S. (2022). Security enhanced federated learning approach using blockchain. *2022 International Conference on Computer, Power and Communications (ICCPC)*, Chennai, India.
110. Wang, Y., Su, Z., Zhang, N., Benslimane, A. (2021). Learning in the air: Secure federated learning for uav-assisted crowdsensing. *IEEE Transactions on Network Science and Engineering, 8(2),* 1055–1069.
111. Lu, Y., Huang, X., Zhang, K., Maharjan, S., Zhang, Y. (2020). Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Transactions on Vehicular Technology, 69(4),* 4298–4311.
112. Pokhrel, S. R., Choi, J. (2020). Federated learning with blockchain for autonomous vehicles: Analysis and design challenges. *IEEE Transactions on Communications, 68(8),* 4734–4746.
113. Wang, X., Garg, S., Lin, H., Kaddoum, G., Hu, J. et al. (2023). Heterogeneous blockchain and AI-driven hierarchical trust evaluation for 5G-enabled intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems, 24(2),* 2074–2083.
114. Chai, H., Leng, S., Chen, Y., Zhang, K. (2021). A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems, 22(7),* 3975–3986.
115. Pokhrel, S. R., Choi, J. (2020). A decentralized federated learning approach for connected autonomous vehicles. *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Seoul, Korea (South).
116. Li, A., Chang, X., Ma, J., Sun, S., Yu, Y. (2023). VTFL: A blockchain based vehicular trustworthy federated learning framework. *2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 6, Chongqing, China.
117. Abdel-Basset, M., Moustafa, N., Hawash, H., Razzak, I., Sallam, K. M. et al. (2022). Federated intrusion detection in blockchain-based smart transportation systems. *IEEE Transactions on Intelligent Transportation Systems, 23(3),* 2523–2537.
118. Singh, S., Rathore, S., Alfarraj, O., Tolba, A., Yoon, B. (2022). A framework for privacy-preservation of IoT healthcare data using federated learning and blockchain technology. *Future Generation Computer Systems, 129,* 380–388.
119. Aich, S., Sinai, N. K., Kumar, S., Ali, M., Choi, Y. R. et al. (2022). Protecting personal healthcare record using blockchain & federated learning technologies. *2022 24th International Conference on Advanced Communication Technology (ICACT)*, PyeongChang Kwangwoon_Do, Korea.
120. Alzubi, J. A., Alzubi, O. A., Singh, A., Ramachandran, M. (2023). Cloud-IIoT-based electronic health record privacy-preserving by cnn and blockchain-enabled federated learning. *IEEE Transactions on Industrial Informatics, 19(1),* 1080–1087.
121. Liu, Y., Yu, W., Ai, Z., Xu, G., Zhao, L. et al. (2022). A blockchain-empowered federated learning in healthcare-based cyber physical systems. *IEEE Transactions on Network Science and Engineering,* vol. 10, no. 5, pp. 2685–2696.
122. Anusuya, R., Karthika, R. D., Oviya, S., Sangavi, R. (2023). Secured data sharing of medical images for disease diagnosis using deep learning models and federated learning framework. *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*.
123. Salim, M. M., Park, L., Park, J. H. (2022). A machine learning based scalable blockchain architecture for a secure healthcare system. *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea.

124. Połap, D., Srivastava, G., Yu, K. (2021). Agent architecture of an intelligent medical system based on federated learning and blockchain technology. *Journal of Information Security and Applications, 58,* 102748.
125. Lakhan, A., Mohammed, M. A., Nedoma, J., Martinek, R., Tiwari, P. et al. (2023). Federated-learning based privacy preservation and fraud-enabled blockchain IoMT system for healthcare. *IEEE Journal of Biomedical and Health Informatics, 27(2),* 664–672.
126. Durga, R., Poovammal, E. (2022). Fled-block: Federated learning ensembled deep learning blockchain model for COVID-19 prediction. *Frontiers in Public Health, 10*.
127. Liu, Y., Ai, Z., Sun, S., Zhang, S., Liu, Z. et al. (2020). Fedcoin: A peer-to-peer payment system for federated learning. In: *Federated learning: Privacy and incentive*, pp. 125–138. Springer.
128. Xiao, B., Xu, Q., He, C., Lin, J. (2022). Blockchain and federated learning based bidding applications in power markets. *Procedia Computer Science, 202,* 21–26.
129. Cohen, G., Afshar, S., Tapson, J., van Schaik, A. (2017). EMNIST: An extension of mnist to handwritten letters. *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA.
130. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86(11),* 2278–2324.
131. Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečny, J. et al. (2018). LEAF: A benchmark for federated settings. arXiv preprint arXiv:1812.01097.
132. Chen, H., Asif, S. A., Park, J., Shen, C. C., Bennis, M. (2021). Robust blockchained federated learning with model validation and proof-of-stake inspired consensus. *arXiv preprint arXiv:2101.03300*.
133. Beilharz, J., Pfitzner, B., Schmid, R., Geppert, P., Arnrich, B. et al. (2021). Implicit model specialization through dag-based decentralized federated learning. *Proceedings of the 22nd International Middleware Conference*, pp. 310–322. https://doi.org/10.48550/arXiv.2111.01257