Digital Communications and Networks 10 (2024) 248–264

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

A survey on blockchain-enabled federated learning and its prospects with digital twin

Kangde Liu [a], Zheng Yan [a,b], Xueqin Liang [a,b,*], Raimo Kantola [b], Chuangyue Hu [c]

[a] State Key Lab on Integrated Services Network, School of Cyber Engineering, Xidian University, Xi'an, China
[b] Department of Communications and Networking, Aalto University, Espoo, Finland
[c] Yangtze Delta Region Blockchain Technology Research Institute, Jiaxing, China

ARTICLE INFO

Keywords:
Digital twin
Artificial intelligence
Federated learning
Blockchain

ABSTRACT

Digital Twin (DT) supports real time analysis and provides a reliable simulation platform in the Internet of Things (IoT). The creation and application of DT hinges on amounts of data, which poses pressure on the application of Artificial Intelligence (AI) for DT descriptions and intelligent decision-making. Federated Learning (FL) is a cutting-edge technology that enables geographically dispersed devices to collaboratively train a shared global model locally rather than relying on a data center to perform model training. Therefore, DT can benefit by combining with FL, successfully solving the "data island" problem in traditional AI. However, FL still faces serious challenges, such as enduring single-point failures, suffering from poison attacks, lacking effective incentive mechanisms. Before the successful deployment of DT, we should tackle the issues caused by FL. Researchers from industry and academia have recognized the potential of introducing Blockchain Technology (BT) into FL to overcome the challenges faced by FL, where BT acting as a distributed and immutable ledger, can store data in a secure, traceable, and trusted manner. However, to the best of our knowledge, a comprehensive literature review on this topic is still missing. In this paper, we review existing works about blockchain-enabled FL and visualize their prospects with DT. To this end, we first propose evaluation requirements with respect to security, fault-tolerance, fairness, efficiency, cost-saving, profitability, and support for heterogeneity. Then, we classify existing literature according to the functionalities of BT in FL and analyze their advantages and disadvantages based on the proposed evaluation requirements. Finally, we discuss open problems in the existing literature and the future of DT supported by blockchain-enabled FL, based on which we further propose some directions for future research.

1. Introduction

The prosperity of the Internet of Things (IoT) interconnects amounts of devices and enables them to generate huge quantities of data, including user preferences, usage frequency, failure reports, and so on. These data can be further analyzed by service providers to improve their Quality of Service (QoS). However, the dynamic nature of the environment results in data transmission delays that prevent service providers from obtaining the most up-to-date information to make timely and accurate decisions to provide better QoS. In addition, the development of 5G and computing paradigms prompts large-scale networks and brings non-negligible challenges in new technology development and network management. Specifically, the current network infrastructure is not suitable for new technology development, given the high operational risks and high deployment costs. In addition, the need for cost savings requires more flexible network management tools.

As an emerging digitalization technology, Digital Twin (DT) offers a feasible solution to overcome the above-mentioned challenges. DT projects a physical object to a virtual object and the virtual object can constantly learn and update itself according to the changes of its corresponding physical object [1]. Thus, DT can provide real time data analysis and a reliable experiment platform. Specifically, DT analyzes the data generated by IoT devices to derive the best decisions to improve QoS in real time. The operation risks and deployment costs of new technologies in the current network infrastructure are no longer significant constraints since these new technologies can be verified on the virtual

* Corresponding author. State Key Lab on Integrated Services Network, School of Cyber Engineering, Xidian University, Xi'an, China.
E-mail addresses: kangde9730L@163.com (K. Liu), zheng.yan@aalto.fi (Z. Yan), dearliangxq@126.com (X. Liang), raimo.kantola@aalto.fi (R. Kantola), huchuangyue@gmail.com (C. Hu).

https://doi.org/10.1016/j.dcan.2022.08.001
Received 26 August 2021; Received in revised form 5 July 2022; Accepted 2 August 2022
Available online 7 August 2022

object economically. Similarly, all the attempts related to network management can be performed on DT to find the best optimization strategy.

Both DT modeling and DT-based decision-making strategies and operations require numerous data, which increases the deployment difficulty of DT. Specifically, when it comes to large-scale data input, it is impractical to copy all the transformed raw data into the DT model, and it is also tricky to extract all the useful underlying information from all the data and analyze the data patterns. Besides, how to analyze data and make full use of them in DT applications is another serious issue.

Artificial Intelligence (AI) can efficiently discover the patterns and associations of numerous data, thus facilitating the practical deployment of DT. Traditional AI employs a centralized server to collect data and train models. However, this data is rarely used effectively by AI companies due to data owners' fear of sensitive information leakage when sharing this data with companies. In addition, the complexity and size of the models grow over time due to the huge amount of training data and the complexity and variety of training tasks, which creates a huge challenge for AI companies to implement model training with an acceptable time and resource consumption. Fortunately, the emergence of Federated Learning (FL) [2] tackles the above-mentioned issues. FL enables model trainers (i.e., data owners) to locally preserve their data and only upload their local model updates rather than raw data to a central server for global model training. Meanwhile, FL offloads training tasks from the central server to its subordinates, while the central server is only responsible for aggregating local model updates. Therefore, FL relieves the computation pressure on the central server and improves the time efficiency of model training.

However, although FL eliminates raw data exchange during training and protects data privacy to a certain extent, FL still faces some non-negligible problems. First, FL still relies on a central server to aggregate local model updates, which can be subject to single-point failures. If the central server is attacked or collapsed, the whole training process would be aborted immediately, which wastes the resources of trainers and reduces the enthusiasm of the trainers in joining FL tasks. Second, in the absence of a strong monitoring mechanism, the central server may perform maliciously, which can severely reduce trainers' motivation to join and hinder the progress of FL. For example, it can deliberately tamper with the records of the trainers for pursuing its maximum revenue. It may abandon some local model updates as they wish to disturb the whole training process or lower the contributions of the corresponding trainers.

Blockchain Technology (BT), which is originally introduced in 2008, is a promising candidate to overcome the above-mentioned problems of FL. Such a promising technology has attracted significant attention with its characteristics of decentralization, anonymity, and immutability. Researchers from industry and academia have recognized the potential of introducing BT into FL to overcome the challenges faced by FL. In blockchain-enabled FL, the trainers submit local model updates to miners rather than to a central server for local model verification and global model aggregation. All miners in the blockchain calculate a global model independently. Only global models generated by miners who meet certain requirements can be published, and other miners validate this global model before attaching it to the blockchain. The blockchain-enabled FL benefits the research from the following aspects. First of all, BT overcomes single-point failures, thus enhancing the robustness of the training process. Second, a malicious global model could be mostly rejected due to the consensus mechanism of blockchain. Third, all data records are tamper-proof on the blockchain, preventing malicious servers from corrupting model training and enhancing the trustworthiness of the final aggregated model. Therefore, BT helps to achieve the initial goals of FL.

However, a comprehensive survey of blockchain-enabled FL is missing. Nguyen et al. [3] classified current articles from the perspective of problems to be solved in blockchain-enabled FL, however, they narrowed down the application scenarios to edge computing. Ali et al. [4] individually elaborated current applications of FL and BT in IoT, but they did not explore blockchain-based FL. Moreover, neither of these two papers considers the integration of DT and FL. Thus, the literature still lacks a survey about how DT and FL can make each other better and how BT can improve FL.

In this paper, we review existing works about blockchain-enabled FL and evaluate them with a series of evaluation requirements. Specifically, we first conclude a general structure of DT supported by blockchain-enabled FL according to our reviewed papers. Then, we propose several requirements from the perspective of security, fault-tolerance, fairness, efficiency, cost-saving, profitability, and support for heterogeneity. Besides, we classify the reviewed papers into four categories according to the major functions of BT in an FL system: secure and tamper-proof maintenance of data, training process coordination, introduction of incentive to trainers, and trainer behavior supervision. In addition, we evaluate these papers against our requirements, on the basis of which we further identify several unresolved issues and corresponding directions for future research. Our contributions can be summarized as follows.

1. We propose several requirements that a blockchain-enabled FL should meet in seven ways: security, fault-tolerance, fairness, efficiency, cost-saving, profitability, and support for heterogeneity.
2. We classify existing blockchain-enabled FL systems according to the functionality of BT and thoroughly evaluate these systems based on the proposed requirements.
3. We discuss the prospects for blockchain-enabled FL-enabled DT and detail how DT can be combined with FL to make each other better.
4. We point out open issues in the current literature and propose several future research directions.

The rest of this paper is organized as follows. Section 2 introduces the basic knowledge of BT, FL, and DT, while at the same time presents a general structure of blockchain-enabled FL-enabled DT. Section 3 describes a series of evaluation requirements, based on which we evaluate the pros and cons of current blockchain-enabled FL systems in Section 4. The open issues and future directions are listed in Section 5. Finally, Section 6 concludes this survey paper.

2. Background

In this section, we firstly introduce BT, including its structure, how it works in the Bitcoin network, and its characteristics. Then, we present FL, including its workflow along with its pros and cons. The definition of DT and its applications with FL are presented subsequently. Finally, we formulate a general structure of blockchain-enabled FL-enabled DT with its corresponding workflow.

2.1. Blockchain Technology (BT)

Blockchain is a chain-structured peer-to-peer decentralized ledger [5]. It follows the principle that each node has a copy of the ledger so that no one can control the blockchain network completely. The ledger consists of immutable blocks with a block body and a block header, which are linked one by one in chronological order. The block body consists of a set of transactions and a Merkle tree that is built on the hash of these transactions. The root of the Merkle tree is stored in the block head along with a hash pointer to the previous block and other information. In a blockchain network, miners are responsible for verifying all transactions and packing the validated transactions into a block.

Herein, we take Bitcoin blockchain as an example to introduce the general procedure of BT. When a transaction appears in the Bitcoin network, it will be verified by miners who have abundant resources and own the transaction pool. Each miner owns its transaction pool to store transactions it received and only the validated transaction can be packed into the pool. All miners compete with each other to solve a hash puzzle and the one that solves it first would gain the right to generate a block, which is comprised of the transactions in its transaction pool. Then, it would broadcast its block and other miners would verify all transactions in the new block. This new block is attached to its blockchain network when and only when the miner acknowledges the validity of this new block. This process of verification among all miners is called a consensus mechanism which guarantees the decentralization of BT.

Notably, several miners may finish their solutions at the same time. Therefore, several blocks are generated and broadcasted in the blockchain network. When the miners choose different blocks to append, the blockchain network is subject to forking issues. The Bitcoin network applies the longest chain principle to solve the forking issues, which means that a miner would append its block to the longest chain. Herein, the longest chain is accepted by the majority of the miners.

The special structure provides BT with the following key characteristics.

- Immutability: It signifies that all transactions and blocks on-chain are permanent and tamper-proof. The hash value of a block is calculated based on its block body and block head that contains the hash value of previous blocks. Therefore, if an adversary modifies some information on a previous block, the hash value of the next block would be changed while the subsequent blocks cannot link this block. This characteristic provides strong and credible proof of what happened before and makes BT a reliable ledger.
- Decentralization: It means BT can not be controlled by any single entity. The blockchain network is maintained by the miners, each of which has a full replica of the blockchain network. Any transaction that occurs in a blockchain network needs to be verified by the miners rather than a trusted party or a single entity and the miners are requested to reach a consensus on the transactions packed in a block. Therefore, the reliability of the transactions on the blocks is guaranteed because they have been verified by all miners in the blockchain.
- Anonymity: All nodes including the miners and blockchain users are using a pseudonym in blockchain, which ensures the anonymity of BT. Blockchain applied cryptographic techniques to protect node identity. Each node can generate amounts of public-private pairs arbitrarily and thus own a large number of blockchain addresses that are created based on the public keys. Therefore, the node can separate their real identities from the identities in the blockchain network for protecting their identity privacy.
- Traceability: Traceability indicates that any past transaction can be retrieved later on since all transactions are preserved permanently on the blockchain. Thus, if a node gains permission to access the blocks, it can obtain all historical information. The combination of traceability and immutability guarantees that no node can deny its past behavior that is recorded on the blockchain. These properties indicate significant applications. For instance, Lin et al. [6] requested task offloading and processing between Virtual reality Devices (VDs) and Edge Access Points (EAPs) in a medical information sharing system. However, the operation of EAPs and the offloaded task cannot be verified by the owner of VDs, which damages the security of the system and the confidence of the VD owners. They further upgraded their system by leveraging BT to record all information with traceability [7].

According to the papers we surveyed [8–19], we find that two types of blockchain are widely applied in existing FL systems: public (i.e., permissionless) blockchain and permissioned blockchain. Public blockchain means anyone can join and leave the blockchain network in a non-contact way and play any role in the network. Furthermore, anyone can audit the transactions published on the blocks. Permissioned blockchain indicates that only authorized and pre-designated entities can join the blockchain system with constrained rights and the system is maintained by one or several organizations. It is noted that there exist many problems inherent with blockchain [20,21], e.g., low throughput, distributed denial of service attacks, privacy, and deanonymizing attacks. These problems brought by blockchain would affect the performance of FL or even threaten the security of a blockchain-enabled FL system. Fortunately, there have been several reviews regarding the problems and how to solve them in blockchain. Zheng et al. [22] gave an overview of blockchain technology as well as its problems. The authors in Refs. [23,24] explored how to solve the security and scalability in blockchain separately. Yang et al. [25] investigated how to solve blockchain issues with the assistance of edge computing and Liu et al. [26] utilized machine learning methods to overcome some problems in blockchain. Different from existing surveys, our paper aims to review the literature about applying blockchain for improving FL. Based on our serious search and review, we find that few surveys touch this topic. Therefore, our paper does not put blockchain problems and their solutions as our focus.

2.2. *Federated Learning (FL)*

Traditional machine learning employs a central server to collect data and train models, therefore, the communication overhead and the computation overhead of the central server are extremely heavy. Since the effectiveness and accuracy of machine learning are positively related to the amount of data and the capability of the central server, traditional machine learning poses great challenges to the central server. Moreover, the training data may reveal sensitive information of the data owners. Thus, the data owners would hesitate to share their data for training, especially when they are skeptical of the trustworthiness of the central server. Therefore, the data cannot be effectively utilized due to the capability restriction of the central server and the privacy concerns of the data owners. Fortunately, Google came up with FL to address the above data underutilization and privacy breaches by enabling data owners to collaboratively build global models without sharing the raw data [2]. From the perspective of networking structure, FL can be divided into two categories: centralized FL and decentralized FL [27]. The centralized FL relies on a central server to aggregate the local models while the decentralized FL enables each data owner to perform the aggregation process. Since all reviewed papers apply the centralized FL, thus we only introduce the centralized FL in the following section for simplicity.

The workflow of the FL is summarized as follows. Let us assume that there is a central server releasing an FL task and a total number of K trainers. The central server first distributes the initial global model to all trainers. The trainers train the global model based on their local data and submit the training results (called local model updates) to the central server, while keeping the original data locally. The central server then aggregates all the received local model updates and updates the global model accordingly. Subsequently, the central server again distributes the updated global model to all trainers, who further train the updated global model using their local data. The above process would continue until the global model is converged or reaches predefined training rounds.

It is obvious that FL is free of the traditional data collection tasks and offloads some training workloads at the server side to the data owners that are also the trainers in FL. FL retains the privacy of the trainers as they only need to upload model updates, while sensitive and personal data remains stored locally. Furthermore, FL is equipped with high communication efficiency since the communication overheads of raw data uploading at both the server side and the trainer side are considerably heavier than those of the local model updates uploading. However, FL still faces serious problems, such as single point-failures and the lack of a comprehensive oversight mechanism for the central server. Fortunately, researchers have explored the application of BT to deal with these issues. Since BT is inherently decentralized, it is effective to avoid single point-failures in FL. Furthermore, the immutability and traceability of BT can guarantee the transparency of the FL training process, which also provides FL with a robust supervision mechanism.

2.3. *Digital Twin (DT)*

DT is an emerging technique in the last couple of years that can project a physical object to a virtual model, i.e., object. It allows to perform network theoretical analysis and experiments directly on the virtual model without changing the current network infrastructure. In addition, DT guarantees real-time data analysis, helping us to dynamically perceive physical objects to make intelligent decisions. The combination of FL and DT realizes two-way closed-loop benefits circulation.

On the one hand, AI boosts the development and deployment of DT based on its powerful data analysis ability. For instance, Mostafa et al. [28] utilized AI models to extract useful data from a large number of inputs during the process of DT creation and provided AI-assisted suggestions in manufacturing plants. Rathore et al. [29] presented the applications of AI-based DT in several scenarios, e.g., prognostics, power and energy, smart manufacturing, etc. FL facilitates the development of DT by breaking many constraints and limitations in AI, such as “data island” and lack of enough training resources. In addition, the creation of DTs requires the transfer of all operational data from IoT devices, which not only generates a significant communication load but also leads to data privacy issues. Without an appropriate solution, the willingness of data owners to invest in their data is inhibited. FL only requires data owners to transmit local model updates, aggregating all local model updates to form a better virtual model that successfully protects user privacy, therefore, enriching the functionality of FL is a key step to improve DT.

On the other hand, DT also promotes the development of FL. Since DT shows excellence in running states awareness, it has been applied to analyze the running states including the computing and communication capabilities of trainers in FL timely. Lu et al. [1] utilized DT to map the devices of trainers to virtual models for analyzing the running states of its corresponding trainer, which are composed of computing and communication capabilities. Their method achieves dynamic resource allocation by arranging more resources for the trainers with poor running states for offsetting their negative influences on training performances. Sun et al. [30] leveraged DT to devise an optimal trainer selection method in practical and large-scale FL cases and explore a dynamic incentive mechanism for pursuing high global model accuracy and high energy utilization. Song et al. [31] devised an adaptive FL model to minimize training costs by dynamically determining an optimal global aggregation frequency with the assistance of DT. In Ref. [32], DT facilitates the training process in FL to be handled by a third party for efficiency improvement. Specifically, the model trainers are projected in the third party for direct model training, which avoids the influence of inferior trainers. If the third party is honest and it possesses considerable resources, the FL task can be completed in a highly efficient way.

2.4. *DT supported by blockchain-enabled FL*

Based on the articles we reviewed, we conclude a general structure of DT that is supported by blockchain-enabled FL in Fig. 1. Several roles are presented in this structure, i.e., virtual objects, task publishers, trainers, Consensus Nodes (CNs), and miners. The virtual objects are comprehensive digital representations of the trainers. The task publishers are responsible for issuing an FL task with some specific training requirements. The trainers are the data owners that train their local data for the FL task. The miners verify the local model updates and aggregate the validated ones. The CNs are the nodes involving in the block consensus process and are usually elected from the miners.

The general workflow is described as follows. 1) The task publishers release their individual FL task and requirements on the blockchain, including the initial global model, the type of training data, the minimum requirement of CPU frequency of the trainers, the maximum waiting uploading time, etc. The maximum waiting uploading time of each publisher requires all related trainers to submit their local model updates in time. 2) The trainers download the global model of the FL task that they would like to execute from the blockchain. 3) The trainers train the global model with local data and produce local model updates. 4) Virtual objects are constructed according to the information from their corresponding physical objects. Usually, the information includes the local model updates, their running states, etc. The constructed virtual objects are associated with different miners. 5) The trainers upload their local model updates on the blockchain and the miners can obtain the latest information from the virtual objects to assist miners in trainer selection, resource allocation, etc. 6) The miners verify the local model updates

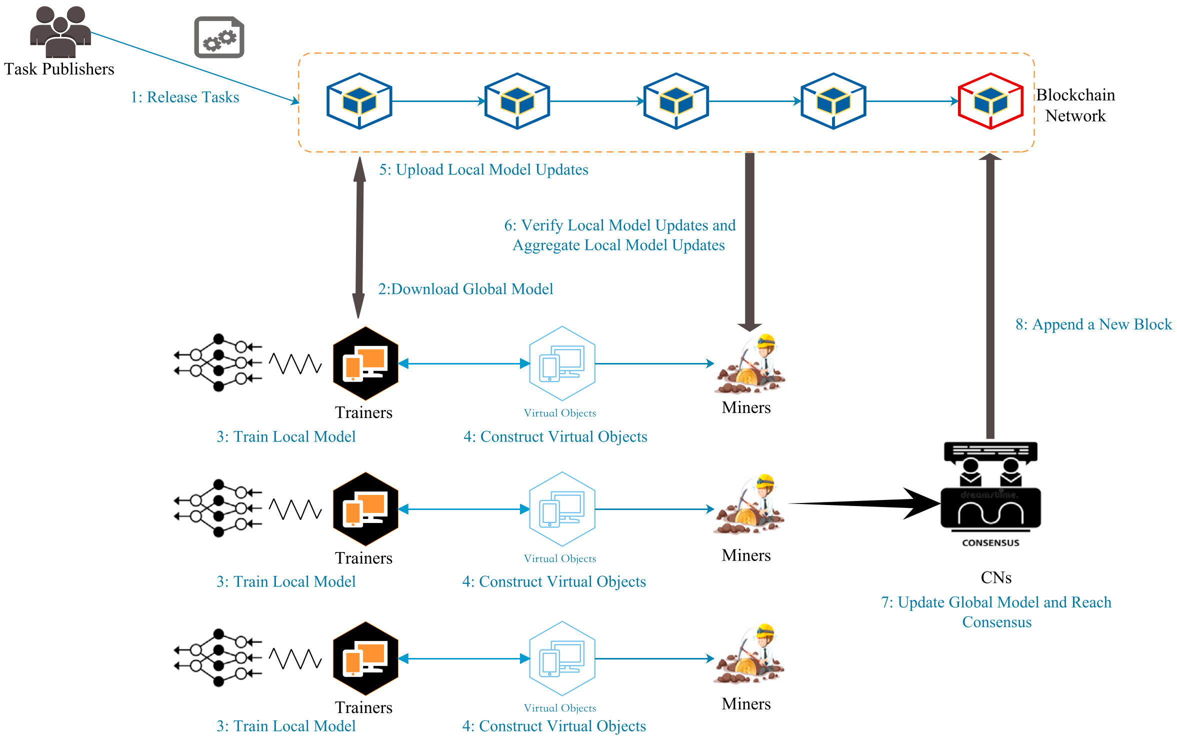

Fig. 1. Structure of DT supported by blockchain-enabled FL.

from their covering trainers and aggregate received local model updates after the specified maximum waiting uploading time. Meanwhile, the miners exchange their aggregated local model updates with other miners. 7) In this step, a temporary leader will be selected from the CNs to update the global model by aggregating local model updates from all miners and generating a block containing the newly aggregated local model updates and the current global model while the CNs reaches consensus with this block. 8) This newly generated block would be appended into the blockchain network if it passes the consensus process. The above process would continue until the task publishers terminate the training process or the pre-defined training rounds have been reached.

Herein, according to the two types of blockchain we find in surveyed papers, we list the following differences between a public blockchain-enabled FL and a permissioned blockchain-enabled FL regarding node roles and node rights. In a public blockchain-enabled FL, each node can work with arbitrary roles and quit the FL task if it is a miner or a trainer. All nodes can perform any activities without restrictions. However, in permissioned blockchain-enabled FL, the role of the node has been pre-assigned and prescribed with constrained rights. Therefore, the node can only perform activities under certain rules and cannot leave the FL task arbitrarily.

3. Evaluation requirements

We propose the following evaluation requirements to evaluate the effectiveness of applying BT in FL from seven aspects: security, fault-tolerance, fairness, efficiency, cost-saving, profitability, and support for heterogeneity. A structure of all the evaluation requirements is listed in Fig. 2.

3.1. Security (Se)

Considering the diversity of attacks in blockchain-enabled FL, we evaluate the security of existing works by assessing the effectiveness of defending several attacks. Note that the attacks listed below are main flow attacks in blockchain-enabled FL according to our survey. For other attacks, like the back door attack that is a common attack for traditional AI, we leave them behind our discussion. Specifically, we believe that FL can resist the back door attack to some extent inherently since FL alleviates the negative effect of this attack through local model aggregation.

3.1.1. Poison Attack Resistance (PAR)

A poison attack in FL refers to a situation that the trainers submit wrong or well-modified local model updates to deteriorate the accuracy of the global model updates. Such an attack adversely affects the accuracy of the global model, thus further poses an influence on the convergence rate of the global model. Moreover, the poison attack increases resource consumption on all entities including the trainers, the miners, and the task publishers. Therefore, it is necessary to consider the ability to resist poison attacks when assessing the effectiveness of BL applications in FL.

3.1.2. Verifier Compromise Attack Resistance (VCAR)

Verifiers play an important role in blockchain technology and are responsible for authenticating the validity of the transactions. An adversary with a Verifier Compromise Attack (VCA) in the FL scenario can compromise the verifiers and the compromised verifiers will not fulfill their obligation honestly. Consequently, the global model may not converge in a predefined time, which incurs extra communication and computation rounds. Therefore, the model accuracy and the communicational and computational efficiency will be severely influenced. The verifiers are normally the miners in blockchain [8,9,11,12,32–35]. Therefore, we interchangeably use these two terms in the following section.

3.1.3. Trainer Compromise Attack Resistance (TCAR)

The trainers are another critical role in FL, which are responsible for training the local models with local data. When a Trainer Compromise Attack (TCA) is launched against a trainer, the infected trainer will submit false local model updates or delay the submission process, and the accuracy of the global model will deteriorate to the detriment of the task publisher, so it is also necessary to prevent this attack.

An intuitive solution to achieve the VCAR/TCAR is to achieve miner/trainer unlinkability. The unlinkability preserves the real identities of the miners/trainers; therefore, an adversary cannot launch a bribery transaction on-chain. According to the degree of supporting VCAR/TCAR, we classify the fulfillment of the security requirement concerning VCAR/TCAR into three categories: low, medium, and high. Specifically, the low fulfillment refers to the situation that the miners/trainers are fixed in the whole training process. The reason is that if they are fixed then the adversary can analyze the behavior pattern of a trainer/miner and reveal the real identity of the trainer/miner with a high probability. The

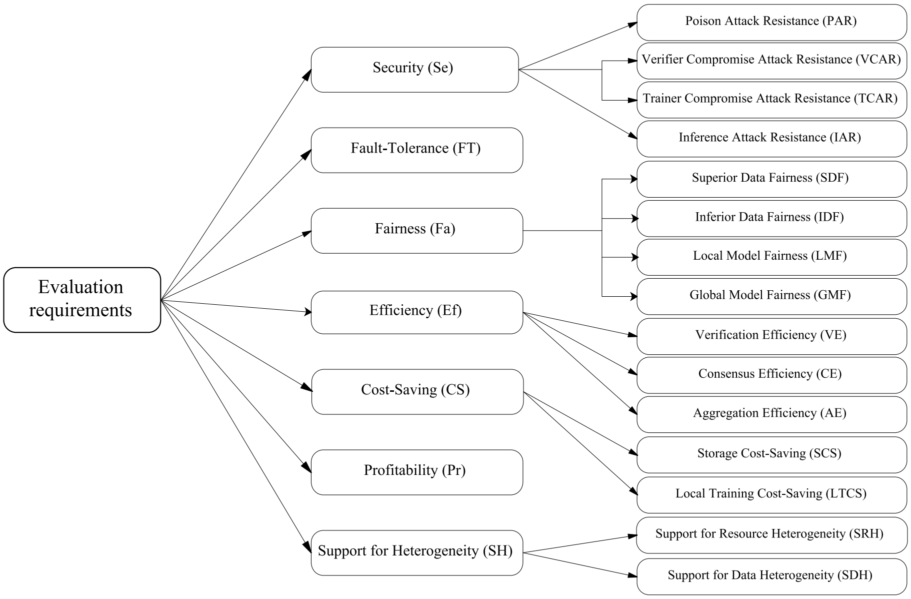
Evaluation requirements
Security (Se)
Fault-Tolerance (FT)
Fairness (Fa)
Efficiency (Ef)
Cost-Saving (CS)
Profitability (Pr)
Support for Heterogeneity (SH)
Poison Attack Resistance (PAR)
Verifier Compromise Attack Resistance (VCAR)
Trainer Compromise Attack Resistance (TCAR)
Inference Attack Resistance (IAR)
Superior Data Fairness (SDF)
Inferior Data Fairness (IDF)
Local Model Fairness (LMF)
Global Model Fairness (GMF)
Verification Efficiency (VE)
Consensus Efficiency (CE)
Aggregation Efficiency (AE)
Storage Cost-Saving (SCS)
Local Training Cost-Saving (LTCS)
Support for Resource Heterogeneity (SRH)
Support for Data Heterogeneity (SDH)

Fig. 2. Structure of evaluation requirements.

medium fulfillment means that although the miners/trainers are ambulatory, the adversaries can probabilistically predict whether a miner/trainer would be involved in the next round. For those miners/trainers that have been predicted successfully, the adversary can combine task requirements and their behavior to deanonymize them and may finally discover their real identities. If the miners/trainers are randomly chosen in each round and the adversaries can hardly predict the probability, we say that the fulfillment of security in supporting VCAR/TCAR is high because there would be little useful evidence to support the adversary to predict the real identity of a trainer/miner.

3.1.4. Inference Attack Resistance (IAR)

In FL, local model updates have a strong association with personal information. Melis et al. [36] have found that the local model updates could disclose the sensitive information of the trainers. An inference attack is an attack in which an attacker uses local model updates to compromise or even violate the privacy of the trainer. Without privacy preservation, the sensitive trainers would hesitate to join the training process. Therefore, we also use the ability to implement an IAR as a requirement to assess whether a blockchain-enabled FL system can protect the privacy of trainers.

3.2. Fault-Tolerance (FT)

Fault-tolerance refers to the ability of a Proof of Work (PoW)-based blockchain system in resisting the forking issues. Since blocks usually contain updated information about the global model and the forking issues can cause the trainer to download a different global model, which can negatively affect the model convergence or even the accuracy of the global model, we consider FT as one of our evaluation requirements. Notably, we only consider this requirement in a PoW-based system, since forking issues occur rarely when other consensus mechanisms are applied.

3.3. Fairness (Fa)

Fairness can not only guarantee individual benefits but also build a harmonious competitive environment for promoting sustainable system development. A well-designed blockchain system normally provides the miners and the CNs with fair mining rewards and transaction fees. We mainly assess whether a blockchain-enabled FL system can ensure fairness for the trainers and the task publishers. Specifically, we propose superior data fairness, inferior data fairness, and local model fairness requirements for the trainers and a global model fairness requirement for the task publishers.

3.3.1. Superior Data Fairness (SDF)

SDF means that the trainers can obtain payments according to their data quality or volume. Since different trainers hold different local training data in practice, the fulfillment of the SDF requirement guarantees a fair incentive mechanism that can attract the trainers with high-quality data or high-volume data.

3.3.2. Inferior Data Fairness (IDF)

It is non-trivial to punish malicious trainers for the sake of system security. IDF indicates that a trainer will be punished if its malicious behavior is detected. Therefore, the IDF requirement can reduce the probability of the trainers to behave maliciously from an economic perspective.

3.3.3. Local Model Fairness (LMF)

Since the local model updates are public in the blockchain network and can be accessed by all system entities, some malicious trainers could plagiarize the published local model updates and claim them as theirs, which inhibits the willingness of honest trainers to participate or forces trainers to submit local model updates as late as possible to avoid being copied, thus affecting the whole training process of FL. If LMF is satisfied, the miners are capable of distinguishing similar or even the same local model updates and identifying the plagiarism behavior. Hence, the interests of honest trainers are guaranteed.

3.3.4. Global Model Fairness (GMF)

The trainers can access the global model of the FL task they would like to participate in and enjoy the rewards from the task publishers. Dishonest or malicious trainers can make illegal profits by selling the global model, which will certainly harm the task publisher, especially when the training process is coming to an end and the global model tends to converge. When a blockchain-enabled FL system satisfies the GMF requirement, then the task publishers can be compensated for the malicious selling of their global models. Therefore, GMF is an important requirement that can ensure the participation willingness of the task publishers.

3.4. Efficiency (Ef)

An efficient blockchain-enabled FL system is attractive to both the system users and the system participants. We evaluate the efficiency of consensus and verification for BT and the efficiency of aggregation for FL.

3.4.1. Verification Efficiency (VE)

Numerous transactions need to be verified in a successful large-scale system, bringing a heavy verification burden on the miners. Meanwhile, the scalability of the blockchain is subject to the ability to process transactions. Thus, realizing the requirement of verification efficiency can improve the throughput capacity of the blockchain-enabled FL system and enhance the practicality of the system. We propose the VE requirement to evaluate the system's ability to verify the transaction expeditiously.

3.4.2. Consensus Efficiency (CE)

It is a time-consuming process for all the CNs to reach a consensus on the validity of a block. Consensus efficiency can be improved if we limit the number of CNs needed to reach consensus, which can be achieved by CN selection. Meanwhile, the forking issues in PoW-based blockchain would increase the difficulty of reaching the consensus and increase the processing latency of transactions. CE means that a system can reach a consensus as soon as possible, which could promote the progress of FL training and enhance the whole system efficiency.

3.4.3. Aggregation Efficiency (AE)

The aggregation of local model updates will not be activated until all the local model updates have been uploaded or the maximum waiting uploading time has passed. Therefore, aggregating all the local model updates is inefficient. If a blockchain-enabled FL system allows the miners to calculate the global models without waiting until receiving all the local model updates, we say that this system achieves the requirement of AE, and vice versa.

3.5. Cost-Saving (CS)

In a practical scenario, the resources of the CNs, miners, and trainers are limited. Therefore, the high block verification costs of the CNs, the high transaction verification costs of the miners, and the high storage costs, as well as training costs of the trainers, suppress their enthusiasm in participating in the blockchain-enabled FL systems. Considering the fulfillment of VE and CE contributes to saving the costs in block verification and transaction verification, we mainly consider the storage costs and the training costs.

The storage costs refer to the costs for storing the local model updates and the global models. The training costs contain the computational costs and the communication costs in terms of calculating and uploading the local models. Any approach that can reduce these costs can definitely encourage the participation of more system entities. Furthermore, they can participate in more tasks with the same restricted resources for obtaining more profits. It is worth mentioning that we only consider the communication latency in local model updates uploading without considering it in global updates downloading. This consideration originates from two reasons. The first reason is that the uplink speed is much lower in internet connection when comparing with the downlink speed. The other one is that the low uplink speed is the major bottleneck of enhancing communication efficiency [37], especially when the communication cost is unaffordable for the trainers. Therefore, in order to evaluate the ability of the blockchain-enabled FL in saving these costs, we propose two requirements that are a Storage Cost-Saving (SCS) requirement and a Local Training Cost-Saving (LTCS) requirement. Specifically, SCS indicates the ability to relieve the storage pressure for the trainers. We divide the degree of fulfillment of SCS into three levels: high, medium, and low. If the hash value of local model updates and global models are preserved in a block, or the trainers have no need to store the whole blockchain, then the SCS requirement is highly fulfilled. Medium fulfillment indicates the global models are stored in a block and all local model updates are preserved in a third party or elsewhere. As long as local model updates are presented in the blockchain, we say such a system achieves low SCS. We evaluate the fulfillment of LTCS by considering whether a system can save the local training costs for the trainers.

3.6. Profitability (Pr)

Considering the lack of regulation in blockchain-enabled FL systems and the public characteristics, incentive mechanisms are widely applied to guarantee the high participation level and the cooperative behavior of the system entities. The task accomplishment of the task publishers requires the participation of trainers while the miners and CNs are also important to the operation of BT. Therefore, existing incentive mechanisms are mainly designed from the perspective of the trainers' utilities for motivating them to participate and BT inherently brings incentives to the miners and CNs. However, the task publishers are vital in the sustainable development of the systems since the payments and rewards to the trainers, the CNs, and the miners are from the task publishers. A blockchain-enabled FL system without any task publisher cannot survive. Therefore, the utility of the task publishers should be considered in the incentive mechanism design. We employ Pr as the requirement to evaluate whether a blockchain-enabled FL system can guarantee the profits of the task publishers.

3.7. Support for Heterogeneity (SH)

The characteristics of openness and decentralization bring the blockchain-enabled FL systems with heterogenous trainers in terms of resources and data. Ignoring the influence of heterogeneity will affect the system performance to some extent. Specifically, a system with a large number of poor trainers, the data of which are with Non-Independent and Identically Distributed (Non-IID), will suffer from a long training completion time. Moreover, the global model is difficult to converge. Thus, a system designer should consider these factors to ensure quick convergence in the global model training.

3.7.1. Support for Resource Heterogeneity (SRH)

Resource heterogeneity means the computation and communication resources vary with each trainer. A trainer with few resources theoretically performs poorly in submitting their local model updates. The time for the model aggregation process is determined by the slowest trainer [19]. SRH indicates that the system designers consider the heterogeneity of resources because shortening the resources gap among the trainers would accelerate the model convergence rate and decrease the learning completion time. In other words, those trainers with poor resources should be banned to join the FL tasks for improving the system efficiency.

3.7.2. Support for Data Heterogeneity (SDH)

Data heterogeneity indicates that the trainer is equipped with Non-IID datasets, the existence of which influences the convergence of the global model [38], and damages the interest of the task publishers. Without enough task publishers, the system cannot sustain itself in the long run. Therefore, eliminating the influence of the Non-IID data and guaranteeing the model convergence can attract the task publishers. SDH refers to the ability of the system in tackling the difficulty of convergence brought by the Non-IID data.

4. Blockchain-enabled FL

In this section, we review the state-of-the-art papers regarding the functionality of BT played in FL and comment on the pros and cons of each paper based on the proposed evaluation requirements in Section 3. According to the main purpose for BT in FL, we classify the application scenarios into four categories: secure and tamper-proof maintenance of data, training process coordination, introduction of incentives to trainers, and trainer behavior supervision. We show the taxonomy regarding these four categories in Fig. 3. Table 1 briefly concludes our evaluation results.

4.1. Secure and tamper-proof maintenance of data

Traditional FL relies on a central server to store all training-related data and DT-based virtual models are designed as the representations of trainers for real time analysis. However, the server may not be reliable and honest, because when the central server controls access to the model updates or virtual models, it can change the updates or information in the virtual models as needed to gain additional illegal benefits. In addition, the modified virtual models do not accurately reflect the trainer's information, which reduces the efficiency of the whole training process, so it is necessary and urgent to secure the training-related data.

Zhao et al. [11] established a permissioned blockchain-enabled FL system for supporting smart home appliance manufacturers to provide suitable services to their customers, while BT is introduced to record model updates honestly. They adopted Algorand, which is based on Proof of Stake (PoS) [39], and Byzantine Fault Tolerance (BFT), as the consensus mechanism. Specifically, they utilized the Verifiable Random Functions (VRF) to randomly select CNs from miners to verify the validity of a new block. Since both trainers (i.e., customers) and miners are settled throughout the training process, satisfaction with VCAR and TCAR requirements is low and consensus is only required among a subset of miners (i.e., CNs), thus reducing validation costs. Therefore, the proposed system satisfies the requirement of CE. Furthermore, Differential Privacy (DP) techniques are leveraged to preserve data privacy, which realizes the requirement of IAR. The miners would verify the validity of received local model update from the trainers, filter out vicious or inferior updates through multi-krum [40] techniques, and renew the reputation values of the trainers. Hence, the PAR requirement is satisfied. Considering the computing power constraints of the trainers, the proposed system enables to offload training tasks to Mobile Edge Computing (MEC) servers. Although the local computation costs are decreased by task offloading, the communication costs between the trainers and the MEC servers increase. Thus, LTCS is hard to evaluate. It is worth noting that the trainers upload local models to InterPlanetary File System (IPFS) [41] while only store the hash values of the local models on the blockchain, thus relieving the storage overheads. However, the satisfaction of the SCS requirement is only medium since the global models are still preserved on the blockchain. The authors also designed a reputation-based incentive mechanism to reward honest nodes according to their data quality, which achieves the goal of SDF. Unfortunately, other requirements were not considered.

Lu et al. [19] applied a permissioned blockchain-enabled FL system to relieve frequent communication rounds between miners and trainers in the Internet of Vehicles (IoVs), where blockchain is used to record and store model parameters reliably. Road Side Units (RSUs) maintained the

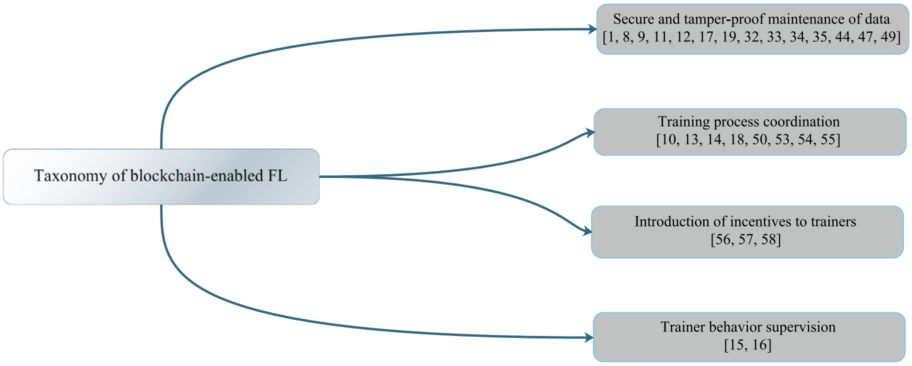
Taxonomy of blockchain-enabled FL
Secure and tamper-proof maintenance of data [1, 8, 9, 11, 12, 17, 19, 32, 33, 34, 35, 44, 47, 49]
Training process coordination [10, 13, 14, 18, 50, 53, 54, 55]
Introduction of incentives to trainers [56, 57, 58]
Trainer behavior supervision [15, 16]

Fig. 3. Taxonomy of blockchain-enabled FL.

Table 1
Summary on application of blockchain-enabled FL.

Ref	AS	BlT	Se				FT	Fa				Ef			CS		Pr	SH	
			PA	VC	TC	IA		SD	ID	LM	GM				SC	LT		SR	SD
			R	AR	AR	R		F	F	F	F	VE	CE	AE	S	CS		H	H
[1]	MEC	Pe	N	L	L	N	/	–	–	–	–	Y	Y	Y	L	–	–	Y	–
[8]	MEC	Pe	Y	H	M	Y	/	–	–	–	–	–	N	–	L	–	–	Y	Y
[9]	Others	Pe	Y	M	L	N	/	Y	–	–	Y	–	N	Y	M	–	–	–	–
[10]	Others	Pu	Y	L	Y	N	Y	Y	–	–	–	–	Y	–	–	–	–	–	–
[11]	MEC	Pe	Y	L	L	Y	/	Y	–	–	–	–	Y	–	M	–	–	–	–
[12]	MEC	Pe	Y	L	L	Y	/	Y	–	–	–	–	N	–	L	–	–	–	Y
[13]	MEC	Pu	N	L	H	N	Y	Y	–	–	–	–	Y	–	L	–	–	–	Y
[14]	Others	Pu	Y	M	H	Y	/	Y	–	Y	–	–	Y	–	L	–	–	–	–
[15]	MEC	Pe	Y	L	L	N	/	Y	–	–	–	–	Y	–	L	–	Y	Y	Y
[16]	MEC	Pe	Y	L	L	N	/	Y	–	–	–	–	N	–	L	–	Y	Y	Y
[17]	Others	Pe	Y	L	H	N	–	Y	–	–	–	–	N	–	–	–	–	–	Y
[18]	MEC	Pe	Y	H	H	Y	N	Y	–	–	–	–	Y	–	H	–	Y	Y	Y
[19]	MEC	Pe	Y	M	M	N	/	Y	–	–	–	–	Y	–	L	–	–	Y	Y
[32]	MEC	Pe	Y	L	L	N	/	–	–	–	–	Y	Y	–	L	Y	–	/	–
[33]	MEC	Pu	Y	L	H	Y	/	–	–	–	–	–	–	–	L	–	–	–	Y
[34]	MEC	Pe	Y	H	L	N	/	/	Y	–	–	–	N	–	H	Y	–	–	–
[35]	Others	NM	Y	L	L	Y	–	Y	–	–	–	–	Y	–	L	–	–	–	Y
[44]	MEC	Pe	Y	L	L	N	/	Y	–	–	–	Y	Y	–	L	–	Y	Y	Y
[47]	MEC	Pe	Y	L	M	N	Y	Y	–	–	–	–	Y	Y	L	Y	–	–	Y
[49]	MEC	NM	Y	L	L	N	/	Y	–	–	–	Y	Y	–	L	–	Y	Y	Y
[50]	FC	Pe	N	L	L	N	Y	Y	–	–	–	–	Y	–	L	–	–	–	–
[53]	Others	Pu	N	L	H	N	Y	Y	–	–	–	–	Y	–	L	–	–	–	–
[54]	Others	Pe	Y	L/M	L/L	Y	/	–	–	–	–	–	N/Y	–	L	–	–	–	–
[55]	MEC	Pe	Y	L	L	N	–	Y	–	–	–	–	N	–	H	–	–	–	–
[56]	Others	Pu	Y	H	H	Y	/	Y	–	–	–	–	–	–	L	–	Y	Y	Y
[57]	Others	Pe	N	H	L	Y	/	Y	Y	–	–	–	–	–	L	–	–	–	–
[58]	MEC	Pe/Pu	Y	L	M	N	/	Y	–	–	–	Y	–	–	–	–	Y	Y	Y

Notation.
Ref: Reference; AS: Application Scenario; BlT: Blockchain Type; Pe: Permissioned; Pu: Public; NM: Not Mentioned; FC: Fog Computing.
N: Not satisfied; Y: Satisfied; -: Not considered;/: No need to consider; L: Low; M: Medium; H: High.
Se: Security; PAR: Poison Attack Resistance; VCAR: Verifier Compromise Attack Resistance; TCAR: Trainer Compromise Attack Resistance.
IAR: Inference Attack Resistance; FT: Fault-Tolerance; Fa: Fairness; IDF: Inferior Data Fairness; SDF: Superior Data Fairness.
LMF: Local Model Fairness; GMF: Global Model Fairness; Ef: Efficiency; VE: Verification Efficiency; CE: Consensus Efficiency.
AE: Aggregation Efficiency; CS: Cost-Saving; SCS: Storage Cost-Saving; LTCS: Local Training Cost-Saving; Pr: Profitability.
SH: Support for Heterogeneity; SRH: Support for Resource Heterogeneity; SDH: Support for Data Heterogeneity.

permissioned blockchain as miners and vehicles (i.e., trainers) only store local Directed Acyclic Graph (DAG), which saves all local models. Thus, their method can only satisfy the requirement of low SCS. The authors adopted Delegated Proof of Stake (DPoS) as the consensus mechanism and satisfied the requirement of CE since the CNs are selected from only a few miners (i.e., RSUs). Note that the miners here are not responsible for verifying local models but for voting for CNs to generate blocks and reach the consensus. They also leveraged Reinforcement Learning (RL) [42] to select well-suitable trainers from all trainers according to their communication and computation capabilities as well as the accuracy of their training results to improve training efficiency. Thus, the proposed system satisfies the requirement of SRH. In addition, the above node selection method filters out unreliable trainers and selects the trainers with higher local model accuracy. Generally, the local model is trained within limited epochs prescribed by a task publisher. The higher local model accuracy means the deviation between the local model and the global model is much smaller, which indicates the local training data have a more similar distribution with the global training data. Thus, this node selection achieves the requirement of SDH. However, since the system enables the trainers to verify local models and when the reputation and resource-based trainer selection method is applied, an attacker can easily predict the selection results with prior knowledge. Thus, the satisfaction of TCAR and VCAR is medium. Vehicles share local model information with nearby vehicles and update their local DAGs after filtering out low-precision information, which achieves the requirements of PAR, but cannot support the requirements of IAR because privacy-preserving techniques are not used in sharing local models. The RSUs would collect current local model updates that have been verified by the trainers and aggregate them to generate a global model. A trainer who contributes with the high-quality local model update would gain a high reputation, which means the trainer would have a high probability of being selected for the next round of training and obtaining the rewards. Hence, the system satisfies the requirement of SDF. However, the rest requirements were not discussed in this paper.

Li et al. [9] proposed a novel consensus mechanism in permissioned blockchain-enabled FL to enhance the security of FL and reduce consensus costs of BT, where the blockchain manages model updates in case of any malicious behavior from a centralized server. They devised a novel consensus mechanism in which miners are elected from trainers based on their performance in the previous training process and the number of elected miners is relatively small when comparing with the trainer number. Herein, the miners are identical to CNs, which means all miners will participate in the consensus mechanism. Hence, this paper satisfies the requirement of medium VCAR but fails to meet the CE requirement. The membership of trainers is fixed and therefore the TCAR requirements are less likely to be met. Submitted local model updates will be further validated by miners, and only qualified updates will be aggregated each round, supporting the PAR requirement. Once miners receive enough local model updates, the process of local model update aggregation is activated, which saves the time of waiting for slow trainers and, therefore, satisfies the AE. The miners verify each unencrypted local model update and calculate a score for the update based on their dataset. The median of all calculated scores is assigned to the local model update as its final score. The higher the final score is, the more rewards the corresponding trainer can earn, which indicates that the requirement of SDF is satisfied but IAR is not supported. Besides, the system can endure some trainers to abandon historical blocks for alleviating storage pressure, which provides the system with high SCS. The GMF requirement is satisfied because the trainers are required to pay access fees to the task publishers. However, the proposed system does not consider other requirements.

Lu et al. [8] proposed a permissioned blockchain-enabled FL system in 5G beyond for enhancing the security of FL, wherein BT is applied to store model parameters. The same as [9], they also adopted DPoS [43] as the consensus mechanism. Different from the fixed trainer members in Ref. [9], the authors selected the trainers according to their resources and data quality in each round through Deep Reinforcement Learning (DRL). Therefore, the impact of heterogeneous devices is reduced. According to our discussion about the systems proposed in Refs. [9,19], this paper satisfies the requirement of high VCAR, medium TCAR, SRH, and SDH. The authors enabled the trainers to add noise to their local model updates for protecting privacy, which provides the system with IAR. The miners (i.e., CNs) verify the local model updates by calculating loss functions based on their datasets and filter out fake and low-quality updates. Thus, this paper satisfies the requirement of PAR without supporting CE. Since both the local model updates and global models are stored in the block, the satisfaction of SCS is low. Moreover, the authors did not consider other requirements.

Chai et al. [44] proposed a hierarchical and permissioned blockchain-enabled FL system for knowledge sharing in IoVs to reduce computation on global models and the blockchain stores the knowledge securely. Trainers (i.e., vehicles) send local model updates to RSUs for local model updates aggregation and the RSUs send aggregated local model updates to Base Stations (BSs) for further aggregation. The hierarchical blockchain contains a Ground Chain Layer (GC) and a Top Chain Layer (TC). In GC, the vehicles act as the trainers to send local model updates to the RSUs that work as miners and CNs for local model updates aggregation. The RSUs serve as the trainers in TC by sending the aggregated local model updates to the BSs that act as the miners and CNs. The authors applied the same novel consensus mechanism in both layers, Proof of Knowledge (PoK). We take GC as an example, where the RSUs need to reach a consensus on the accuracy of the aggregated local model updates. The RSUs purchase local model updates from the trainers, validate the local model accuracy and pay the trainers with rewards that are proportional to their model accuracy. Thus, the system realizes the requirements of SDF and PAR. Then, the RSU with the highest accuracy obtains the right to publish its block. All the RSUs and the trainers are divided into different clusters based on their location and the RSUs only verify those local model updates in the same cluster, which improves the throughput of the system and satisfies the fulfillment of VE. Also, the consensus process is executed among CNs in the same cluster, which realizes the requirement of CE. However, the local model updates transmitted between the RSUs and the trainers are without any protection., which fails to support IAR. Since the miners and the trainers are fixed in the system, thus the fulfillment of VCAR and TCAR is low. The RSUs can be regarded as task publishers in this paper since they are responsible for compensating the trainers. A multi-leader and multi-player Stackelberg game is built to model the interactions among the trainer and RSUs. Consequently, an algorithm is proposed to solve this game for maximizing the utilities of the trainers and the RSUs, which guarantees the requirement of Pr. However, the fulfillment of SCS is low since all local model updates are recorded on-chain. Unfortunately, other requirements were not concerned in this paper.

Lu et al. [12] designed a blockchain-enabled FL structure for secure data sharing among multiple participants in Industrial IoT, where BT stores local model updates and global models permanently. They adopted a permissioned blockchain with a novel consensus mechanism called Proof of Training Quality (PoQ), which selects a temporary leader from miners to publish a block based on model performance. The trainers (i.e., IoT devices) are divided into different communities according to the categories and distribution of their data. The trainers with similar data are linked through a local retrieval table. Therefore, the requirement of SDH is satisfied since similar training data will enhance the global model accuracy for FL tasks and balance the data distribution. When a task publisher issues an FL task, a trainer first calculates a local model update with the DP technique and informs the other trainers located in its retrieval table to participate in this task. The application of the DP technique guarantees the fulfillment of the IAR requirement. Similar to Refs. [8,9,19], the miners are also elected from the trainers and they also take the role of CNs. Specifically, the miners verify the quality of local model updates and aggregate the qualified local model updates. Hence, PAR is also satisfied in this paper. The validity of the block generated by the temporary leader miner is verified by the other miners, which fails to support CE. Notably, both the miners and the trainers are settled during the whole training process, thus the satisfaction of the requirements VCAR and TCAR are low. In addition, the payments to the trainers are calculated according to their contributions, which are proportional to their data sizes. Hence, the requirement of SDF is also satisfied. However, this paper does not consider other requirements.

Liu et al. [33] integrated BT and FL to devise Intrusion Detection Systems (IDSs) in the vehicular network in order to detect attacks, where BT is leveraged for tamper-proof data storage. The trainers in this paper are the vehicles that mask their local model updates with a secret through Shamir secret sharing technology. The RSUs work as miners to distribute global models and aggregate the received local model updates after subtracting the secret. The miners are fixed in the system while the trainers can decide whether to join dynamically. Hence, this paper achieves low VCAR and high TCAR. The miners adopt a comparison method to evaluate whether a trainer is malicious or not. Specifically, the miners compare the aggregation results when a certain local model update is included or not. If the difference value exceeds a predefined threshold, the corresponding trainer is considered malicious and the local model updates are excluded from the final aggregation results, which supports the PAR requirement. Note that the trainers would receive masked local model updates from adjacent trainers and aggregate their local model updates when training their local models to improve the quality of local model updates. Thus, this paper satisfies the requirement of IAR. Specifically, the requirement of SDH is also realized since sharing local model updates with adjacent trainers alleviates the impact of Non-IID data. The miners verify each others' models and the verification results are applied to evaluate the trust of the miners. The miners can obtain payments that are proportional to their trust values. However, the incentive to the trainers is hardly investigated in this paper, thus we cannot evaluate the fulfillment of SDF. At last, their method does not consider other requirements.

Cui et al. [34] devised a permissioned blockchain-enabled FL to improve cache hit rate in edge computing, where BT is exploited to secure the hash of local models. The authors adopted PoS as the consensus mechanism, where miners are elected from trainers (i.e., edge nodes) based on reputation and random numbers. Specifically, the miners only occupy a small part of trainers in each round and they are identical to CNs, thus satisfying the requirement of high VCAR but failing to support the requirement of CE. The trainers are fixed in the system, which meets the requirement of low TCAR. In this system, the IoT device asks the trainer to train the local model based on the data sent by the IoT device. The requested data is not encrypted before sending, and the trainer can easily infer the personal information of the IoT device, which cannot satisfy the requirements of IAR. Moreover, the local models are compressed using gradient compression method and verified by miners, thus reducing the communication cost of trainers and resisting poison attacks, and their approach satisfies the requirements of LTCS and PAR. If a local model update is considered invalid, the corresponding trainer is penalized, which provides support for IDF. The authors apply a reputation mechanism to reward miners who perform honestly during the validation process but lack incentive analysis of the trainers. Only the hash of the local model is kept in the blockchain and their approach allows for high SCS, however, their approach does not involve meeting other requirements.

Liu et al. [35] proposed a blockchain-enabled FL system to protect FL from unreliable and malicious participants, where BT implemented on Ethereum [45] is used to secure local and global model updates. They allowed each task publisher to appoint one miner to verify and aggregate the local model updates from the trainers. The consensus results are decided by this miner, which realizes the requirement of CE. The trainers are mobile devices and they are locked during the training process. Thus, their system satisfies the requirements of low TCAR and low VCAR. The trainers would add noise to their local model updates before submitting them to the miner, thus fulfilling the requirement of IAR. The miner would evaluate the quality of received local model updates with the testing dataset provided by the task publishers, which satisfies the requirement of PAR. The authors utilized Earth Mover's Distance (EMD) [46] as a metric to measure the Non-IID degree of the trainers' data and their experiment shows that their method can resist the impact of Non-IID data successfully; therefore, the requirement of SDH is satisfied. The trainers receive payments that are proportional to the quality of their local model updates. Thus, the system realizes the requirement of SDF. However, the rest requirements were not discussed in this paper.

Lu et al. [32] pioneered the introduction of digital twins into blockchain-enabled FL to reduce the heavy communication costs between trainers (i.e., end-users) and miners (i.e., BSs) in the 6th generation mobile networks, where BT is used to store local and global models reliably. They adopted DPoS as the consensus mechanism and elected several CNs from the miners to verify blocks. Thus, their method satisfies the requirement of CE. Each trainer would be mapped into a digital twin maintained by a miner and the association of the miners and the trainers is stored in a permissioned blockchain to enhance their mutual trust. The digital twin enables miners to verify only the transactions of their mapped trainers, which supports the VE implementation. Since the miners and the trainers are immovable, their method satisfies the requirements of low TCAR and low VCAR. After training the local data and aggregating all local model updates of mapped trainers, the miners submit the well-trained local model updates to an aggregator for global aggregation. The local training process is offloaded from the trainers to the miners, thus relieving the computation costs of the trainers. Therefore, this paper achieves the requirements of LTCS for the trainers. Unfortunately, the requirement of IAR is not satisfied since the miners are aware of the local training data collected from the trainers. Each miner would broadcast its local model updates to the other miners for verification, which achieves the requirement of PAR. Combining the data size of the training data, the state of the communication channel, and the computational power of the miners, the authors further use DRL to match digital twins with suitable miners and optimize the bandwidth allocation between miners and aggregators, thereby improving resource utilization. Unfortunately, the authors did not consider other requirements.

Lu et al. [1] continued to explore the application of DT in blockchain-enabled FL for edge networks based on [32] while enhancing the FL communication efficiency and simulating the dynamic environment of edge devices. They leveraged permissioned blockchain to store local model updates and global models for enhancing mutual trust among unfamiliar trainers. They adopted DPoS as the consensus mechanism. Since all local model updates and global models are preserved in blockchain, their method can only satisfy the requirement of low SCS. In their work, the trainers are the edge devices and the miners are BSs. CNs are elected from the miners to execute the consensus mechanism, thus achieving the requirement of CE. A miner connects to several trainers and constructs the digital twins of its associated trainers. Each miner takes charge of verifying the transactions under its coverage, which fulfilling the requirement of VE. The trainers upload their local model updates to the miners for global model aggregation without quality verification, which fails to support the requirement of PAR. Besides, the trainers and the miners are fixed in the whole training process, which means their method can only satisfy the requirements of low TCAR and low VCAR. No privacy-preserving technique is applied in their work, thus the requirement of IAR is not satisfied. Digital twins can reflect the running state of the trainers. The authors combined RL and the digital twins to design a novel transmission scheduling policy, which offloads the communication tasks of uploading local model updates to a reliable trainer with high communication capability. Thus, their method can achieve the requirement of SRH. To further alleviate the adverse effect caused by slow trainers, the authors proposed an asynchronous FL method, wherein a trainer that has uploaded its local model update can obtain a former global model to perform the next training round and a miner can activate the global aggregation process after several training rounds, which fulfills the requirement of AE. However, the rest requirements were not discussed in this paper.

Feng et al. [47] proposed an asynchronous blockchain-enabled FL solution to solve the Poison Attack (PA) and data unreliability issues in FL for IoT and to accelerate global model aggregation, where BT guarantees the immutability of local models and global models. They adopted PoW as the consensus mechanism and solved the forking issues through an Acknowledge Code (ACK), which is utilized to terminate block broadcast when forking issues occur. Thus, their solution satisfies the requirement of FT. The miners are identical to CNs in this paper and they are preassigned and fixed during the training process. Thus, their method satisfies the requirement of low VCAR. The trainers (i.e., edge devices) are selected each round based on their scores, which are calculated through an entropy-based method based on their contributions to global models. Thus, their solution satisfies the requirements of SDH and medium TCAR. The miners will verify their received local model with their datasets to resist poison attacks; however, the local models are submitted by the trainers directly without any privacy preservation. Therefore, this paper satisfies the requirement of PAR but cannot support IAR. Note that, unlike the conventional FL, this blockchain-enabled FL only requires one local model in calculating the global model at one time. The aggregation process can be activated as long as a trainer uploads its local models to its associated miner without waiting until all trainers have uploaded the new local models. Hence, this paper satisfies the requirements of AE. Each trainer downloads the current global model and the local model from their associated miners to generate a new global model and calculate the scores of the trainer that has uploaded the new local model in the current round. The most acceptable global model is recognized as the final global model and the trainer is rewarded according to their latest scores, which indicates that this paper realizes the requirement of SDF. The requirement of CE is satisfied with the optimal block generation rate, which reduces transaction latency caused by forking issues, through a genetic algorithm [48]. Since all the local model updates and the global models are preserved in blocks, the fulfillment of SCS is low. The authors balanced the local learning time and energy cost by considering the scalability of blockchain and achieved the requirement of LTCS. Unfortunately, other requirements were not discussed by the authors.

Feng et al. [49] proposed a two-layered blockchain-assisted FL framework to manage the trust and security issues in an unfamiliar environment, where BT is used as an immutable ledger for storing model updates. A Local Model Update Chain (LMUC) is applied to record the local model updates sent by the trainers (i.e., local devices) and a Global Model Update Chain (GMUC) is utilized to record global models and the contributions of the trainers. All trainers in LMUC are associated with specific MEC nodes. Some trainers and the MEC nodes will participate in local model updates verification and reach a consensus on aggregated local model updates as miners and CNs, respectively. All MEC nodes act as miners as well as CNs in GMUC to verify the aggregated local model updates and reach a consensus on global models. The authors adopted the Byzantine Fault Tolerance (BFT) consensus mechanism in both the blockchains. Both the trainers and the miners in LMUC are settled while the miners are task-related and fixed MEC nodes in GMUC. Thus, their method can satisfy the requirements of low VCAR and low TCAR. When a task publisher issues an FL task, only those relevant MEC nodes can join this task and a task-specific chain is formed and maintained by selected MEC nodes. The throughput of the blockchain is enhanced because different task-related miners can process various task transactions simultaneously and a consensus process is performed between these task-related miners, satisfying the VE and CE requirements. Then, the trainers are elected by the associated MEC node based on their reputation, which is calculated based on historical performances and computing resources. Through trainer selection, their method cuts down the influence caused by low-quality and Non-IID data and insufficient hardware resources. Thus, their method satisfies the requirements of SRH and SDH. The trainers in LMUC would submit local model updates without privacy protection to their associated miners that would check the quality of all local model updates. Thus, their method satisfies the requirement of PAR but cannot support the requirement of IAR. Then, the task-relevant MEC nodes exchange their received local model updates and aggregate them to form global model updates, which would be distributed to the trainers subsequently. Specifically, the trainers would get paid proportional to their training data through smart contracts, which supports SDF. Note that the task publishers would cluster the trainers and dispatch different FL tasks and remunerations according to their past behaviors to maximize the task publishers' profits, which supports the requirement of Pr. However, the proposed system does not consider other requirements.

Considering the possibility that malicious trainers in FL manipulate their training data deliberately in the Industrial Internet of Things (IIoT) scenario, Zhang et al. [17] leveraged blockchain to preserve data tamper-proof and verify the integrity of training data and proposed a BT-empowered FL to detect device failure. They adopted PoW as the consensus mechanism and employed a fixed number of miners, which also work as CNs in this paper. Thus, the requirement of CE is not met. They applied a centroid distance weighted federated averaging method to reduce the impact of Non-IID data. Thus, their method satisfies the requirement of SDH. Trainers are randomly selected, they collect data from equipped sensors and create a Merkle tree to represent the data records, and only the root of the Merkle tree is stored in the block. The task publishers, which are set as verifiers here, command some trainers to join the FL tasks randomly. Thus, their system satisfies the requirements of high TCAR and low VCAR. The trainer uploads local model updates without privacy protection to the task publisher and applies Merkle trees to resolve their disputes about the training data, which will be generated based on the task publisher's validation results, thus satisfying the PAR requirement but not supporting the IAR. Notably, blockchain is not utilized to store local or global models here but to store the Merkle tree root and the centroid distance of the trainers' data as well as the size. In addition, blockchain is used to record each trainer's contribution, whereby the trainer can be paid. Therefore, the requirement of SDF is also satisfied. However, other requirements were not considered.

4.2. *Training process coordination*

Traditional FL needs a central server to coordinate the training process. However, the whole training process would be suspended if the central server is collapsed, which hinders the evolution of FL. The decentralization characteristic empowers BT to replace the central server for reliably coordinating the model training process. Usually, the coordination process is executed among miners that further manage the virtual models generated through DT. Thus, with the assistance of DT, the miners can interact with DT directly to accelerate the training process.

Qu et al. [50] devised a permissioned blockchain-enabled FL structure to enhance FL performances in fog computing and employed BT to achieve training-related data exchange between the trainers and the miners securely and stably. They adopted PoW as the consensus mechanism and added an ACK signal in block propagation to solve forking issues. Thus, their method can satisfy the requirement of FT. The trainers upload their local model updates to their associated miners without privacy protection and the miners aggregate these local model updates and reach a consensus on the aggregation results. Thus, their method cannot satisfy the requirement of IAR. The miners compare the data size of the trainers with their claimed computing time through Intel's SGX technology to calculate the contributions of the trainers. However, the authors determined the payment to the trainers is calculated to the data size-based contribution without considering the quality of their local model updates. Thus, their method satisfies our SDF requirement, but cannot support PAR. The authors designed to store both pointers to the global model and local model updates in blocks, thus satisfying the SCS requirements at low levels. In addition, they derived optimal block generation rates that reduce learning completion delays and forking probabilities, so that their system satisfies the CE requirements. Besides, the miners and the trainers are fixed in their system, which shows the fulfillment of VCAR and TCAR are both at a low level. However, other requirements were ignored.

Shayan et al. [14] combined BT with FL to eliminate a central server in FL, where BT coordinates the training process among untrusted trainers. They proposed a novel consensus mechanism called Proof of Federation (PoF) based on PoS. Their proposed system contains four kinds of system entities that are trainers (i.e., peers), noiser committees, miners (i.e., verifier committees), and CNs (i.e., aggregator committees). Specifically, the noiser committees, miners, and CNs are selected based on stakes through Verifiable Random Functions (VRF) while the trainers can decide whether to join a round of FL tasks arbitrarily. Thus, the system satisfies the requirements of medium VCAR and high TCAR. The trainers would add noises generated by the noiser committees to their local model updates, which provides our satisfaction with IAR. Herein, the noises are different for different trainers. Then the trainers send the masked local model updates as well as the polynomial commitments [51] of both the original local model updates and noise to the miners. They further check the quality of received masked local model updates through multi-krum [40] and only sign the satisfied ones, which provides the system with the fulfillment of the PAR requirement. The commitments of noise vary with the trainers. Therefore, a trainer cannot plagiarize the other trainers' noise commitments. Then, the miners compare the consistency of the commitments of the masked updates with the commitments of noise and original updates. Thus, the requirement of LMF is fulfilled. Once a local model update is signed by the majority of miners, the trainers divide it into shares and transmit the shares to CNs for aggregation. Consensus on the global model update time only requires consensus among aggregators, and the number of aggregators is relatively small. Thus, their method satisfies the requirement of CE. The stakes of the trainers are proportional to their contributions to the system, thus fulfilling the requirement of SDF. However, the fulfillment of SCS is low because local model updates are stored on-chain. Unfortunately, other requirements were missed without discussion.

Pokhrel and Choi [13] utilized the public BT-based FL to minimize the end-to-end delay for the autonomous vehicle network, where BT replaces a central server and orchestrates the training process. This paper regards miners the same as CNs. The miners and trainers (i.e., autonomous vehicles) are associated randomly and anonymously. The miners are fixed during the training process while the trainers are uncertain in each round, which indicates the fulfillment of low VCAR and high TCAR. The authors adopted PoW as the consensus mechanism and enabled a miner to add an ACK signal when propagating its block for avoiding forks, which realizes the requirement of FT. The trainers calculate their local model updates and submit the local model updates with computing time to their associated miners, which consequently verify whether the computing time is correct. Thus, the requirements of IAR and PAR are not supported due to the absence of privacy-preserving techniques and verification of local model updates quality. The authors introduced an incentive mechanism to compensate the trainers according to the computing time, which supports the requirement of SDF. They also derived an optimal block arrival rate to minimize the overall system delay and decrease the forking probability. Hence, CE is satisfied. The authors considered the impact of Non-IID data and adopted the same method as in Ref. [52], which requires learning rate decay and optimal local training iterations to guarantee FL convergence rate, to make the global model converge. Therefore, the requirement of SDH is met. However, the rest requirements were not discussed in this paper.

Kim et al. [53] introduced a public blockchain into traditional FL to avoid single-point failures in future wireless systems, wherein BT is applied to verify and exchange local model updates. Similar to Ref. [13], the authors also adopted PoW as the consensus mechanism applied to the ACK signals to achieve the requirement of FT. They also derived an optimal block generation rate to minimize the average latency over the PoW process. Overcoming the forking issues also saves the resources of the miners to solve hash puzzles. Thus, the requirement of CE is also satisfied. The trainers calculate local model updates and send them to their associated miners, however, no privacy-preserving technique is applied to support the achievement of IAR. Moreover, the miners are fixed during the whole training process, thus the fulfillment of VCAR is only at a low level. The authors introduced an incentive mechanism to reward the trainers according to the amounts of them and leveraged Intel's extensions to verify their contributions (i.e., the data amounts). Hence, this paper satisfies the requirement of SDF. However, the method cannot support PAR because the miners do not verify the quality of local model updates. Since the trainers can leave or join an FL task randomly when the public blockchain is adopted, the satisfaction of TCAR is at a high level. The fulfillment of SCS is at a low level because all local model updates are preserved on-chain. However, other requirements were not discussed.

Wang et al. [18] introduced a practical federated learning structure in Unmanned Aerial Vehicles (UAV) assisted MEC network with the training orchestration of blockchain for secure model training and single-point failures resistance. They adopted PoW as the consensus mechanism and decreased the probability of forking issues by modeling a Poisson process on the mining process of PoW, however, they can hardly eliminate the influences caused by the forking issues. Therefore, the requirement of CE is satisfied while the FT requirement fails to be met. The authors utilized DP to protect the privacy of local model updates, which satisfies the requirement of IAR. They also dynamically changed the public keys of the trainers (i.e., UAVs) and miners (i.e., MEC nodes) to avoid them being detected and compromised by attackers. Thus, their method can satisfy the requirements of high VCAR and high TCAR. Miners also play the role of CNs, who are responsible for storing the entire blockchain. The trainers only store the header of each block, which satisfies the requirement of high SCS. The authors also proposed an optimal pricing strategy for trainers and task publishers by defining the Quality of Local Model updates (QoLM) as a measure of trainer contribution and determining payments to trainers based on contribution, satisfying Pr and SDF requirements. Subsequently, they used RL [42] to engage a subset of high QoLM trainers, who tend to have sufficient energy and high data quality so that their approach can meet the requirements of SDH and SRH. According to the QoLM possessed by each trainer, the task publishers calculate the global model by aggregating the trainers' local model updates with different weights, which reduces the impact of the poison attack. We can conclude that the method satisfies our PAR requirement. However, the authors did not consider other requirements.

Qu et al. [10] leveraged blockchain-enabled FL in cognitive computing to enhance the performance of industrial manufacturing, wherein blockchain is applied to coordinate the training process. They adopted a public blockchain structure, applied PoW as a consensus mechanism and solved the forking problem by adding ACK signals, thus meeting the FT requirements. In this paper, the miners, which also act as CNs, are predefined and fully anonymous since the usage of the public blockchain. However, the miners are fixed and attackers can discover their real identities through de-anonymizing techniques [22,59], which means the method only satisfies the requirement of low VCAR. The authors enabled the trainers to participate in or leave any FL task freely, thus, their method can satisfy the requirement of high TCAR. The authors further applied an RL-based Markovian decision process to model the confrontation of adversaries with poison attacks and derived the optimal aggregation strategy for satisfying the PAR requirement. The payments to the trainers are proportional to their data sizes that can be verified by the proof of elapsed time. Hence, the SDF requirement is satisfied. The authors also derived an optimal block generation rate to reduce training time for achieving the requirement of CE. IAR is not satisfied because no privacy-preserving technique is applied to protect local model updates. Although introducing off-chain and on-chain storage structures can alleviate storage pressures for the trainers, the authors did not propose any specific scheme. Moreover, their method does not concern other requirements.

Jin et al. [54] utilized permissioned BT combined with FL to solve data sparsity problems and improve data sharing efficiency in the Internet of Medical Things (IoMT), where BT orchestrates the training process with its consensus mechanism. The authors proposed two schemes of the global model calculation with two different consensus mechanisms, which are Hasty Consensus (HstCon) and Deferred Consensus (DefCon), separately. In both HstCon and DefCon, the trainers are the IoMT devices and they send local model updates to their nearby hospitals for aggregation. The authors allowed the hospitals located in different areas to exchange their aggregated local model updates. Since only the aggregated local model update dates are transmitted between hospitals, an attacker cannot extract the local model updates to infer personal information about the trainer, and thus their approach satisfies the IAR requirements. In HstCon, The miners are identical to the trainers and the CNs. The trainers in a hospital would verify the received aggregated updates based on their local dataset and reach the consensus through Practical Byzantine Fault Tolerance (PBFT) [60] regarding whether to accept or reject the received updates, which satisfies the requirement of PAR but fails to support the requirement of CE. All miners and trainers are fixed here, which means this method can only achieve the requirements of low TCAR and VCAR. While in DefCon, the miners are elected based on the hash value of the latest block and the assets they mortgage and the tenure of a miner is k rounds. Each miner represents a hospital and they can decide whether to accept other hospital's updates or not. In DefCon, no intra-consensus exists in a hospital and a miner is responsible for making decisions on aggregation operation (i.e., consensus process), which supports the requirement of CE. Note that the trainers are still settled while the miners can be predicted through the assets they devoted. Thus, their method can meet the requirements of low TCAR and medium VCAR. Since the trainers are IoMT devices deployed in a hospital and all training data belong to the hospital, we evaluate the SDF of a hospital rather than trainers. Each hospital should be compensated differently based on their data quality, which is not referred to by the authors. The method can only achieve the requirement of low SCS since all model updates are recorded on-chain. However, other requirements were missed without any discussion.

Li et al. [55] proposed a permissioned blockchain-enabled FL framework to resist byzantine attacks (i.e., the attacks that can disturb the model accuracy) and enhance the speed of local model verification at the edge, where BT assists the global model calculation with its consensus mechanism. Trainers (i.e., edge devices) and miners (i.e., edge servers, not the verifiers we defined) are settled in their system, which only satisfies the requirement of low TCAR. The authors pre-assigned some fixed nodes (i.e., verifiers we defined) to execute local model updates validation and pre-appointed the miners (i.e., CNs we defined) who are responsible for mining blocks and reaching consensus. Thus, their method supports the requirement of low VCAR but fails to achieve the requirement of CE since all the miners are involved in the consensus process. The authors separated the verification and mining process and the miners can mine a block while the verifiers can focus on verifying local model updates with all their resources, so their approach can meet the VE requirements. The trainers should submit local model updates to the verifiers, who were responsible for determining whether to accept or discard them. Only accepted local model updates are transmitted to the miners for further aggregation. Thus, their method satisfies the requirement of PAR. The authors devised a novel consensus mechanism between the miners called Proof of Accuracy (PoA), which means the other miners consider a block as valid only the global model updates in it satisfying the accuracy deviation on their dataset and the former global model. Besides, both the local model updates and the global models are stored in IPFS and the hashs of them are stored in blocks, which supports high SCS. Task publishers would reward the trainers proportional to accuracy increment, which guarantees the requirement of SDF. However, other requirements were not considered.

4.3. Introduction of incentive to trainers

BT inherently brings unified tokens, based on which a novel incentive mechanism can be easily established. For example, we can employ the token to compensate the costs of contributors in FL, like the trainers, miners, and CNs. In addition, DT makes task publishers aware of the contributions of the trainers clearly and the task publishers can release corresponding rewards to the trainers swiftly. Furthermore, we can employ the token amount to measure the assets of the trainers/miners and select the one with the largest amount as a leader. Therefore, these system entities will be motivated to contribute.

Toyoda et al. [56] introduced a BT-based FL to compensate trainers (i.e., IoT devices in this paper) with cryptocurrencies to participate and behave as the system required. They adopted public Ethereum [45] as their platform and divided the whole training phase into multi-rounds and randomly preassigned some trainers to compute local model updates in each round, which supports the requirement of high TCAR. The local model updates computed in the previous round will be verified by the assigned trainers in the next round. The new trainers individually chose top-K high-quality local model updates based on their datasets and aggregate them as the local model update in this round. This procedure will repeat until the last round. Therefore, this paper satisfies the requirements of PAR and high VCAR. Unfortunately, the authors never specified what kind of consensus mechanism is applied and how to generate and verify, which makes it difficult to evaluate whether the requirements of FT and CE are satisfied. It is worth noting that although the local model updates are encrypted before transmission in each round, the trainers in the next round can decrypt them, so their method cannot satisfy the requirement of IAR. In their system, the authors rewarded a trainer according to how many times its local model update has been selected as the top-K in the next round. The more times the trainer chooses its local model update date, the higher the reward available to the trainer, which means satisfying the SDF requirement. Eventually, the authors used contest theory to maximize the profits of task publishers within a given budget, so their method can achieve the requirement of Pr. Since all local model updates are stored on the chain, only low SCS requirements can be achieved and no other requirements are considered.

Weng et al. [57] devised a permissioned blockchain-enabled FL to guarantee data privacy and reject malicious behaviors, in which BT is applied as an incentive mechanism to encourage trainers (i.e., participants) to join and behave honestly. The trainers whose local model updates transactions are packed in the current block would act as the miners and CNs at the same time. But their method is hard for us to evaluate the fulfillment of CE due to the uncertain amounts of CNs. The trainers are fixed while the miners are dynamic, thus their method satisfies the requirements of high VCAR and low TCAR. They adopted a novel consensus mechanism, wherein a temporary leader would be chosen randomly from the CNs to generate a block. The trainers would encrypt their local model updates before sending them to an aggregator and a secret sharing scheme is utilized to update global models, which achieves the requirement of IAR. However, the miners only check whether an update is encrypted correctly without quality verification, thus their system fails to achieve the requirement of PAR. It is worth mentioning that the authors requested all trainers to pre-frozen some deposits when entering the system, which would be forfeited and distributed to honest trainers if any malicious behavior was detected, so the requirement of IDF is supported. Furthermore, The authors also determined that the transaction fees paid by the trainers are inversely proportional to the data amount, which satisfies the requirement of SDF. Unfortunately, other requirements were not concerned.

Fan et al. [58] introduced a hybrid blockchain system consisting of the public and the permissioned blockchains to FL for selecting optimal trainers (i.e., edge nodes) that can improve the accuracy of FL models economically in edge computing, where BT brings unified rewards. Their system leverages PBFT as the consensus mechanism in the permissioned blockchain, where CNs are part of the trainers that reach a consensus on a new global model. Since the number of CNs varies in each training round, we cannot evaluate the satisfaction of CE. A task publisher issues its task requirements on the permissioned blockchain and an auction implemented by smart contracts is applied to optimize the profits of both the task publisher and the trainers, also satisfying Pr requirements. Specifically, the task publisher selects the trainers according to their bids, communication and computation capability, as well as data quality, so their method satisfies the requirements of medium TCAR, SRH, and SDH. The selected trainers would submit their local training updates to the task publisher without any privacy-preserving technique and therefore do not meet the requirements of the IAR. In each round, the task publisher will further evaluate the quality of local model updates sent by these selected trainers according to their data distribution and reject malicious and unreliable updates through Reject on Negative Influence (RONI) and FoolsGold scheme [61]. Hence, the task publisher also works as a miner in this paper. Thus, their method satisfies the requirement of low VCAR. The selected trainers are rewarded based on the quality of their local model updates, which supports the fulfillment of the SDF requirement. In addition, due to the severe scalability problems in the public blockchain, the task publishers would compensate their trainers using payment channels, which reduces the amount of broadcasted transactions and the miners can verify more local model updates within a given time. Thus, the VE is satisfied. However, how the local model updates and global models are stored is not specified, we cannot evaluate if the requirement of SCS is satisfied. Other requirements were not considered by the authors.

4.4. Trainer behavior supervision

Blockchain is essentially a distributed ledger so that BT can be utilized to record the reliability of the trainers. In a blockchain-enabled FL-backed DT, trainers can be mapped to virtual models and can be evaluated using specific criteria and the results stored on the blockchain for credibility and non-repudiation. DT also constantly and timely transfers behavior information update between the trainers and their virtual models. Hence, task publishers can quickly evaluate the behavior of trainers and reward honest trainers as well as punish dishonest trainers to build a sustainable system.

Kang et al. [16] designed a permissioned blockchain-enabled FL system that takes reputation as the evaluation results to estimate the reliability and contribution of trainers according to their behaviors and blockchain is utilized to record reputation permanently for trainer selection. They adopted PBFT as the consensus mechanism in the system. Task publishers would release their tasks with specific requirements and select the trainers based on their requirements along with reputation to improve system efficiency, which satisfies the requirements of SRH. Especially, the requirement of SDH is guaranteed because the reputation-based trainer selection scheme reduces the difference in the trainers' data distribution. For a specific task, all the trainers tend to have a high reputation, which means their training data is beneficial to the global model convergence. The selected trainers send their local model updates without encryption to the task publishers, which further verify the local model updates. Therefore, this paper realizes the requirement of PAR but fails to achieve the requirement of IAR. Then, the task publishers generate a global model based on all validated local model updates and increase the reputation of the trainers contributing to calculate global models. Subsequently, the task publishers reward the trainers according to their reputation, which means that a high-reputation trainer will obtain a high profit. Thus, their method satisfies the requirement of SDF. The miners (i.e., CNs) in this paper are preassigned to generate blocks and reach a consensus on the new block. Because all the miners need to participate in the consensus process, their system fails to support CE. The trainers and the verifiers (i.e., task publishers) are fixed. Thus, their method satisfies the requirements of low VCAR and low TCAR. Nevertheless, they devised a contract theory-based incentive mechanism to optimize the revenue of the task publishers as well as the trainers, which achieves the requirement of Pr. Unfortunately, the rest requirements were not considered.

Kang et al. [15] adopted an extremely similar infrastructure to their previous work [16] and applied smart contracts in this new structure to upload training records and reputations. Besides, a consensus process is executed among a small number of elected authorized miners, which satisfies the requirement of CE. Compared with the previous work [16], they optimized task distribution schemes among many task publishers and many trainers, which balances the revenue of both of them. Thus, in addition to the requirements they achieved in Ref. [16], the method can also satisfy the requirement of CE, but other requirements were ignored.

5. Open issues and future directions

5.1. Open issues

According to the aforementioned literature review and evaluation requirements, we figure out a few open issues.

First, model fairness is not well studied when designing the access policies of local model updates and global models. Local model updates enveloped in transactions would be plagiarized by other trainers since the transactions are public and accessible. If the local model updates are involved in virtual models construction, the stealer can build a virtual model without local training, which would demotivate the trainers to train their local model updates honestly. The authors in Ref. [14] distributed some unique noise to the trainers for preventing the local model updates from stealing. However, this method relies on the polynomial commitments of both the local model updates and noise, which incurs extra computation overheads. Moreover, task publishers not only need to compensate the trainers but also reveal their global models to the trainers, which is rather unfair to them. The authors in Ref. [9] requested all the trainers to disburse a sum of certain money for accessing global models for achieving GMF. But their method is not suitable for practical deployment. Thus, how to guarantee the model fairness and promote the sustainability of blockchain-enabled FL is still a challenge.

Second, a comprehensive incentive mechanism should not only reward cooperative behavior but also punish malicious behavior fairly, but such an incentive mechanism is still missing. Table 1 shows that most of the papers do not punish malicious trainers severely to meet the IDF requirement, except [34,57]. The authors in Ref. [34] proposed to withdraw some tokens if a trainer behaves maliciously, but they did not give a specific punishment scheme. The authors in Ref. [57] required the trainers to pre-freeze some deposits that will be forfeited when malicious behavior is detected, but the fairness of the deposit method cannot be guaranteed. Therefore, a comprehensive incentive mechanism is still absent in blockchain-enabled FL systems in order to achieve fairness for all stakeholders.

Third, how to make the trade-off between privacy preservation and FL model practicability is a serious issue. To protect privacy, the encryption-based privacy-preserving methods in the existing literature impose an additional computational burden on the trainer, while the DP-based privacy-preserving methods affect the accuracy of the global model. The hardware-based trusted execution environment, e.g., SGX, is subject to memory restriction [62,63]. In a blockchain-enabled FL system, a miner cannot verify multiple trainers' local model updates simultaneously through SGX, which further introduces extra verification time and restraining the throughput of the whole system. Thus, the current technology cannot protect local model updates privacy without sacrificing practicability and the balance of privacy and model practicability is still an open issue.

Forth, communication latency is still the bottleneck when designing an efficient blockchain-enabled FL system, which is always neglected in existing works. Only three papers [32,34,47] satisfy the requirement of LTCS, which means that most papers do not take any measures to save communication costs. Normally, FL models contain a large number of parameters while the trainers are only equipped with constrained communication resources. As a result, high communication costs for trainers will lead to high latency in the training process, and high communication latency further limits the efficiency and usefulness of FL. The authors in Ref. [34] utilized the gradients compression method to reduce communication costs. However, their method fails to take model accuracy into consideration. The authors in Ref. [32] introduced DT to alleviate the communication latency but their method sacrifices user privacy. Therefore, how to design an efficient blockchain-enabled FL system with low communication costs and latency is still a challenging task.

Fifth, existing blockchain-enabled FL systems can not be well applied in a scenario with Non-IID data. The construction of DT relies on environment data of the physical objects, which means that if the data of a physical object obeys Non-IID, then the corresponding virtual model is also with Non-IID data. In order to achieve PAR, researchers let the miners validate the quality of received local model updates. However, the miners can scarcely distinguish Non-IID data from malicious data, which means that the local model updates generated from Non-IID data would be abandoned as malicious updates. Therefore, the generalization of the global model is adversely affected. Besides, some authors [11,14] applied the multi-krum approach to detect malicious local model updates, which assumes all data to obey IID. Hence, this detection method is not practical since the training data obey the Non-IID in practice [64]. Moreover, the existence of Non-IID data makes the global model hard to converge and introduces extra training rounds, which increases the training completion time. Therefore, how to design a feasible and efficient blockchain-enabled FL system that is compatible with Non-IID data still needs further investigation.

Sixth, the reliability of task publishers is seldom discussed. Existing literature pays a lot of attention to evaluating the reputation of the trainers and miners. However, practical task publishers could also be unreliable or even malicious. A malicious task publisher could release an FL task with high-claimed payments to compete with honest task publishers for the trainers while delaying or refusing the payments. Therefore, if there are not enough trainers, the profits of honest task publishers will be affected and trainers may not receive the claimed payments, which further affects the sustainability of the whole system. Unfortunately, an evaluation method on the task publishers is still missing and how to reduce the negative impact caused by the unreliable task publishers is still a challenge.

Seventh, a blockchain-enabled FL-backed DT is still in its infancy [1, 32]. The dynamic running state of trainers is a non-negligible factor to be considered in the deployment of FL. A task publisher can leverage the deep and timely perception of the running state to allocate resources or choose optimal trainers in order to improve FL efficiency. However, most existing works ignore the importance of trainer running states. Moreover, the virtual objects in Refs. [1,32] expose their raw training data and original local model updates, respectively, which indicates that privacy preservation is hardly considered in DT with FL. Thus, how to integrate FL with DT to improve FL efficiency without sacrificing data privacy is worth our efforts.

Last but not the least, all papers we reviewed are focused on how blockchain can make centralized federated learning better while ignoring the investigation of DFL. A prominent characteristic in DFL is decentralization and each node can share local model updates with others directly. Therefore, each trainer can perform local model aggregation based on the sharing local model updates it receives [65,66]. There exist two main issues in DFL. First, the process of model updates sharing is voluntary and all trainers in such a DFL process can obtain a better model to meet their requirements. However, the quality of the training data varies with each trainer, which results that the contributions of the trainers to the final model are different. Data-poor trainers should pay high access fees for access to the global model, while data-rich trainers should pay lower access fees or be compensated by data-poor trainers, so there is an urgent need for a fair incentive mechanism. In addition, malicious trainers may discard some of the model updates they receive and transfer inferior model updates to others, so DFL still lacks effective oversight of trainers to ensure they act honestly. Therefore, how to design a fair incentive mechanism and motivate honest behaviors in DFL is still an open issue.

5.2. Future directions

The above open issues guide the future directions. We propose several future research directions as outlined below.

Ensuring local model fairness and global model fairness is a realistic need for future blockchain enabled FL research. Local model update theft attacks enable malicious trainers to gain illegal profits by requesting others' local model updates and creating virtual objects using others' local model updates, thus undermining the fairness of local models. Therefore, the local model fairness can be achieved by detecting and resisting these attacks. Calculating the similarity of the local model updates of the trainers could be a promising technology to detect the local model update theft attack, but how to discover which one is the attacker needs further investigation. Moreover, such a calculation task introduces additional overheads to the miners. Hence, an effective and light-weight similarity detection method of local model updates is worth further exploring. Existing work [9] has applied access fees for global models to ensure global model fairness, while the fee is the same for all the global models in different training rounds, but the fee should vary with the training round. For example, the first training round should require a small number of visits, while the last training round should require a large number of visits because the global model in the later training rounds is close to the final result. In addition, the fee should be set reasonably based on the profit that the trainer can earn and the fees paid by the trainer. Thus, how to set optimal and personalized access fees is another future direction. Network watermarking [67–69] is a promising technique to guarantee the local and global model fairness. A trainer can embed a watermark into its local model and a task publisher can embed a watermark into its global model such that the trainer and the task publisher can protect their models from being plagiarized. However, this solution consumes extra computation costs [70] to inject watermarks into a model and the injected watermarks would influence the accuracy of the local and global models to some extent [71]. Thus, how to devise an efficient and practical watermarking scheme is also worth exploring.

A comprehensive and fair incentive mechanism is highly expected in the practical deployment. Penalties are studied and incorporated into existing reward-based incentives to achieve comprehensive and fair incentives. The process of constructing virtual models can be intentionally or unintentionally misguided, so the design of punishment is critical. In addition, punishment is an easily accessible and effective way to deter malicious behavior and thwart trainers from doing evil. Weng et al. [57] have demanded the task publishers should require the trainers to pay the same deposits before allowing the trainers to join FL tasks. However, the amounts of deposits should vary with different trainers that are endowed with different data qualities or data volumes. For instance, those trainers equipped with large data volume or high data quality should be required to pay fewer deposits, in order to attract them to join. Thus, an incentive mechanism including rational punishments is worth exploring.

Balancing the significance between privacy and practicability is worth our efforts. The system preference should be considered when determining the priority of privacy or practicability in a blockchain-enabled FL system. Different systems have different preferences in privacy and practicability, based on which we can perform corresponding measures. For instance, in a privacy-favored blockchain-enabled FL system with a global model accuracy requirement, we propose to adopt effective but complex encryption methods to protect privacy without sacrificing global model accuracy. Similarly, DP can be applied when this system raises efficiency requirements. In an extreme practicability-oriented situation, privacy preservation is far less significant than practicability and we can even abandon privacy preservation. However, how to derive the system preference needs further exploration. Besides, devising a more effective privacy preservation method within the minimum impact on model practicability is another direction.

Lowering the communication costs of the trainers in blockchain-enabled FL is a meaningful and interesting topic. Model compression technique would be an effective method. In some cases that the majority of trainers are in unstable internet connections, the model compression method is relatively effective to enhance the efficiency of the communication among the miners and the trainers. Structured updates and sketched updates [37] could be promising model compression methods but they sacrifice model accuracy. Therefore, how to balance the trade-off between model compression and model accuracy and apply suitable model compression methods can be a further research direction. Furthermore, an effective model compression without model accuracy loss is highly expected. DT is another powerful technique to lower the communication costs because the virtual models of the trainers can be created and deployed at the miner directly, which avoids long-distance local model updates transmission. However, the creation is a time-consuming and tough task. Thus, the efficient creation of DT is an interesting research direction.

A blockchain-enabled FL system that is compatible with Non-IID data is worth exploring. DT cannot change the nature of Non-IID data existence. Non-IID data are essential in achieving model generalization; therefore, an effective Non-IID data detection method to discover the local model updates that are trained by Non-IID data but are detected as malicious by poison attack detection methods is of urgent need. In order to further improve the convergence of training the global model with

Non-IID data, trainer selection [38] may be a suitable solution. However, the trainer selection method will constrain the generalization of global models since the trainers are selected according to model preferences [35]. Therefore, this method can only be applied when the model convergence rate and accuracy are significantly important. Zhao et al. [46] have demonstrated that sharing a small subset of training data on-chain can alleviate the storage pressure caused by Non-IID data. However, this method cannot be applied when the shared data contain reveal sensitive information. Hence, designing a blockchain-enabled FL system with model generalization, high accuracy, and high model convergence rate needs further exploration.

An evaluation method on task publishers should be studied to eliminate the negative impact caused by unreliable task publishers on honest task publishers. Trust management [72] on task publishers is a promising technology. Trust acts as a crucial role in a pervasive network [73,74] and trust management will be a powerful tool to build up trust between the trainers and the task publishers. When trust is introduced, the trainers can select the FL tasks according to the trust value of the task publishers. The FL tasks published by high-trust task publishers are likely to be selected since the trainers could have a high possibility to obtain the payment. Furthermore, the profits of the honest task publishers are also guaranteed with enough trainers. Thus, applying trust management to evaluate the reliability of task publishers is very meaningful and worth exploring.

Integrating FL with DT is a practical requirement in the industrial environment and investigating various methods for different purposes regarding virtual objects creation and representation is an appealing direction. For instance, the creation of virtual objects usually contain sensitive information regarding training data or local model updates. DP can be leveraged to create virtual objects for protecting data privacy. However, if the virtual objects are designed for the purpose of resource allocation, both the training data and local model updates are allowed not to be contained in the virtual objects. Thus, exploring different methods of virtual objects creation is essential. Notably, modeling virtual objects and completing FL tasks are both time-consuming. How to balance the time allocation to maximize the profits of the task publishers is another future direction.

Combining DT and BT with DFL is an alluring direction for making DFL better. Each system node can design a virtual model for other nodes by leveraging its received information. In this way, all nodes gain clear and intuitive insight into the entire system and make real-time decisions accordingly. BT can monitor the behavior of each node and compensate them fairly, which could attract more trainers and encourage the trainers in DFL to perform honestly. BT brings an incentive mechanism as well as reliable local model updates records. Each node in DFL will be treated impartially and all nodes' operations are executed under supervision. Thus, DT and BT generate positive impacts on DFL and DT supported by blockchain-enabled DFL is worth exploring.

6. Conclusion

The problems faced by DT with traditional AI and the limitations of FL arouse our speculation about the prospects of DT supported by blockchain-enabled FL. In this paper, we surveyed blockchain-enabled FL systems and explore their success in the application of DT. We summarized the general structure of DT that is supported by blockchain-enabled FL and proposed a series of requirements to evaluate the effectiveness of the existing blockchain-enabled FL systems. We classified the existing literature into four categories based on the functionality of BT in the FL and further evaluated them with our proposed requirements. We discovered that the research on blockchain-enabled FL faces a number of open issues and the study of DT supported by blockchain-enabled FL is still in its infancy. Based on these open issues, we proposed some interesting research directions for future investigation.

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.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62072351; in part by the Academy of Finland under Grant 308087, Grant 335262, Grant 345072, and Grant 350464; in part by the Open Project of Zhejiang Lab under Grant 2021PD0AB01; and in part by the 111 Project under Grant B16037.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.dcan.2022.08.001.

References

[1] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Communication-efficient federated learning and permissioned blockchain for digital twin edge networks, IEEE Internet Things J. 8 (4) (2021) 2276–2288.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[3] 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. 8 (16) (2021) 12806–12825.

[4] M. Ali, H. Karimipour, M. Tariq, Integration of blockchain and federated learning for Internet of Things: recent advances and future challenges, Comput. Secur. 108 (2021) 102355.

[5] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system. https://www.debr.io/section/2183-whitepaper, 2008.

[6] P. Lin, Q. Song, D. Wang, F.R. Yu, L. Guo, V.C.M. Leung, Resource management for pervasive-edge-computing-assisted wireless VR streaming in Industrial Internet of Things, IEEE Trans. Ind. Inf. 17 (11) (2021) 7607–7617.

[7] P. Lin, Q. Song, F.R. Yu, D. Wang, L. Guo, Task offloading for wireless VR-enabled medical treatment with blockchain security using collective reinforcement learning, IEEE Internet Things J. 8 (21) (2021) 15749–15761.

[8] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Blockchain and federated learning for 5G beyond, IEEE Netw 35 (1) (2021) 219–225.

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

[10] Y. Qu, S.R. Pokhrel, S. Garg, L. Gao, Y. Xiang, A blockchained federated learning framework for cognitive computing in Industry 4.0 networks, IEEE Trans. Ind. Inf. 17 (4) (2021) 2964–2973.

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

[12] 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. Inf. 16 (6) (2020) 4177–4186.

[13] S.R. Pokhrel, J. Choi, Federated learning with blockchain for autonomous vehicles: analysis and design challenges, IEEE Trans. Commun. 68 (8) (2020) 4734–4746.

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

[15] J. Kang, Z. Xiong, X. Li, Y. Zhang, D. Niyato, C. Leung, C. Miao, Optimizing task assignment for reliable blockchain-empowered federated edge learning, IEEE Trans. Veh. Technol. 70 (2) (2021) 1910–1923.

[16] J. Kang, Z. Xiong, D. Niyato, S. Xie, J. Zhang, Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory, IEEE Internet Things J. 6 (6) (2019) 10700–10714.

[17] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S.K. Lo, S. Chen, X. Xu, L. Zhu, Blockchain-based federated learning for device failure detection in Industrial IoT, IEEE Internet Things J. 8 (7) (2021) 5926–5937.

[18] Y. Wang, Z. Su, N. Zhang, A. Benslimane, Learning in the air: secure federated learning for UAV-assisted crowdsensing, IEEE Trans. Netw. Sci. Eng. 8 (2) (2021) 1055–1069.

[19] 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.

[20] L. Peng, W. Feng, Z. Yan, Y. Li, X. Zhou, S. Shimizu, Privacy preservation in permissionless blockchain: a survey, Digit. Commun. Netw. 7 (3) (2021) 295–307.

[21] W. Feng, Z. Yan, Mcs-chain: decentralized and trustworthy mobile crowdsourcing based on blockchain, Future Generat. Comput. Syst. 95 (2019) 649–666.

[22] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: architecture, consensus, and future trends, in: 2017 IEEE International Congress on Big Data (BigData Congress), IEEE, 2017, pp. 557–564.

[23] B. Putz, G. Pernul, Detecting blockchain security threats, in: 2020 IEEE International Conference on Blockchain (Blockchain), IEEE, 2020, pp. 313–320.

[24] J. Xie, F.R. Yu, T. Huang, R. Xie, J. Liu, Y. Liu, A survey on the scalability of blockchain systems, IEEE Netw 33 (5) (2019) 166–173.

[25] R. Yang, F.R. Yu, P. Si, Z. Yang, Y. Zhang, Integrated blockchain and edge computing systems: a survey, some research issues and challenges, IEEE Commun. Surv. Tutor. 21 (2) (2019) 1508–1532.

[26] Y. Liu, F.R. Yu, X. Li, H. Ji, V.C.M. Leung, Blockchain and machine learning for communications and networking systems, IEEE Commun. Surv. Tutor. 22 (2) (2020) 1392–1431.

[27] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, H.V. Poor, Federated learning for Internet of Things: a comprehensive survey, IEEE Commun. Surv. Tutor. 23 (3) (2021) 1622–1658.

[28] F. Mostafa, L. Tao, W. Yu, An effective architecture of digital twin system to support human decision making and AI-driven autonomy. https://onlinelibrary.wiley.com/doi/epdf/10.1002/cpe.6111, 2021.

[29] M.M. Rathore, S.A. Shah, D. Shukla, E. Bentafat, S. Bakiras, The role of AI, machine learning, and big data in digital twinning: a systematic literature review, challenges, and opportunities, IEEE Access 9 (2021) 32030–32052.

[30] W. Sun, N. Xu, L. Wang, H. Zhang, Y. Zhang, Dynamic digital twin and federated learning with incentives for air-ground networks, IEEE Trans. Netw. Sci. Eng. 9 (1) (2020) 321–333.

[31] Q. Song, S. Lei, W. Sun, Y. Zhang, Adaptive federated learning for digital twin driven Industrial Internet of Things, in: 2021 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2021, pp. 1–6.

[32] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks, IEEE Trans. Ind. Inf. 17 (7) (2021) 5098–5107.

[33] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, Y. Zhang, Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing, IEEE Trans. Veh. Technol. 70 (6) (2021) 6073–6084.

[34] L. Cui, X. Su, Z. Ming, Z. Chen, S. Yang, Y. Zhou, W. Xiao, Creat: blockchain-assisted compression algorithm of federated learning for content caching in edge computing, IEEE Internet Things J. 9 (16) (2020) 14151–14161.

[35] Y. Liu, J. Peng, J. Kang, A.M. Iliyasu, D. Niyato, A.A.A. El-Latif, A secure federated learning framework for 5G networks, IEEE Wireless Commun. 27 (4) (2020) 24–31.

[36] L. Melis, C. Song, E. De Cristofaro, V. Shmatikov, Exploiting unintended feature leakage in collaborative learning, in: 2019 IEEE Symposium on Security and Privacy (SP), IEEE, 2019, pp. 691–706.

[37] J. Konečný, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: strategies for improving communication efficiency. https://arxiv.org/abs/1610.05492, 2017.

[38] H. Wu, P. Wang, Node selection toward faster convergence for federated learning on non-iid data, IEEE Trans. Netw. Sci. Eng. 9 (5) (2022) 3099–3111.

[39] S. King, S. Nadal, PPcoin: peer-to-peer crypto-currency with proof-of-stake. https://archive.org/details/PPCoinPaper, 2012.

[40] P. Blanchard, E.M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: byzantine tolerant gradient descent, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, ACM, 2017, pp. 118–128.

[41] J. Benet, IPFS - content addressed, versioned, P2P file system. https://arxiv.org/abs/1407.3561, 2014.

[42] R.S. Sutton, A.G. Barto, Reinforcement Learning: an Introduction, MIT Press, Cambridge, Massachusetts, 2018.

[43] S.J. Alsunaidi, F.A. Alhaidari, A survey of consensus algorithms for blockchain technology, in: 2019 International Conference on Computer and Information Sciences (ICCIS), IEEE, 2019, pp. 1–6.

[44] H. Chai, S. Leng, Y. Chen, K. Zhang, A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of Vehicles, IEEE Trans. Intell. Transport. Syst. 22 (7) (2021) 3975–3986.

[45] G. Wood, et al., Ethereum: a secure decentralised generalised transaction ledger, Ethereum project yellow paper 151 (2014) 1–32.

[46] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-iid data. https://arxiv.org/abs/1806.00582, 2018.

[47] L. Feng, Y. Zhao, S. Guo, X. Qiu, W. Li, P. Yu, Blockchain-based asynchronous federated learning for Internet of Things, IEEE Trans. Comput. (2021), https://doi.org/10.1109/TC.2021.3072033.

[48] L. Deng, P. Yang, W. Liu, An improved genetic algorithm, in: 2019 IEEE 5th International Conference on Computer and Communications (ICCC), IEEE, 2019, pp. 47–51.

[49] L. Feng, Z. Yang, S. Guo, X. Qiu, W. Li, P. Yu, Two-layered blockchain architecture for federated learning over mobile edge network, IEEE Netw 36 (1) (2021) 1–14.

[50] 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.

[51] A. Kate, G.M. Zaverucha, I. Goldberg, Constant-size commitments to polynomials and their applications, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2010, pp. 177–194.

[52] X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, On the convergence of FedAvg on non-iid data. https://arxiv.org/abs/1907.02189, 2020.

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

[54] H. Jin, X. Dai, J. Xiao, B. Li, H. Li, Y. Zhang, Cross-cluster federated learning and blockchain for internet of medical Things, IEEE Internet Things J. 8 (21) (2021) 15776–15784.

[55] Z. Li, H. Yu, T. Zhou, L. Luo, M. Fan, Z. Xu, G. Sun, Byzantine resistant secure blockchained federated learning at the edge, IEEE Netw 35 (4) (2021) 295–301.

[56] K. Toyoda, J. Zhao, A.N.S. Zhang, P.T. Mathiopoulos, Blockchain-enabled federated learning with mechanism design, IEEE Access 8 (2020) 219744–219756.

[57] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, W. Luo, Deepchain: auditable and privacy-preserving deep learning with blockchain-based incentive, IEEE Trans. Dependable Secure Comput. 18 (5) (2021) 2438–2455.

[58] S. Fan, H. Zhang, Y. Zeng, W. Cai, Hybrid blockchain-based resource trading system for federated learning in edge computing, IEEE Inter. Things J. 8 (4) (2021) 2252–2264.

[59] J. Zayuelas Muñoz, Detection of bitcoin miners from network measurements. https://upcommons.upc.edu/bitstream/handle/2117/133503/137903.pdf, 2019.

[60] M. Castro, B. Liskov, Practical byzantine fault tolerance, in: USENIX Symposium on Operating Systems Design and Implementation, USENIX, 1999, pp. 173–186.

[61] C. Fung, C.J.M. Yoon, I. Beschastnikh, Mitigating sybils in federated learning poisoning. https://arxiv.org/abs/1808.04866, 2020.

[62] S. Fei, Z. Yan, W. Ding, H. Xie, Security vulnerabilities of SGX and countermeasures: a survey, ACM Comput. Surv. 54 (6) (2021) 1–36.

[63] G. Liu, Z. Yan, W. Feng, X. Jing, Y. Chen, M. Atiquzzaman, Sedid: an SGX-enabled decentralized intrusion detection framework for network trust evaluation, Inf. Fusion 70 (6) (2021) 100–114.

[64] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R.G.L. D'Oliveira, H. Eichner, S.E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P.B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S.U. Stich, Z. Sun, A.T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F.X. Yu, H. Yu, S. Zhao, Advances and open problems in federated learning. https://arxiv.org/abs/1912.04977, 2021.

[65] A. Lalitha, O.C. Kilinc, T. Javidi, F. Koushanfar, Peer-to-peer federated learning on graphs. https://arxiv.org/abs/1901.11173, 2019.

[66] A.G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, C. Wachinger, Braintorrent: a peer-to-peer environment for decentralized federated learning. https://arxiv.org/abs/1905.06731, 2019.

[67] F. Boenisch, A survey on model watermarking neural network. https://arxiv.org/abs/2009.12153, 2020.

[68] L. Fan, K.W. Ng, C.S. Chan, Rethinking deep neural network ownership verification: embedding passports to defeat ambiguity attacks, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, ACM, 2019, pp. 4714–4723.

[69] R. Namba, J. Sakuma, Robust watermarking of neural network with exponential weighting, in: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, ACM, 2019, pp. 228–240.

[70] B.G.A. Tekgul, Y. Xia, S. Marchal, N. Asokan, WAFFLE: watermarking in federated learning, in: 2021 40th International Symposium on Reliable Distributed Systems (SRDS), IEEE, 2021, pp. 310–320.

[71] H. Li, E. Wenger, S. Shan, B.Y. Zhao, H. Zheng, Piracy resistant watermarks for deep neural networks. https://arxiv.org/abs/1910.01226, 2020.

[72] Z. Yan, P. Zhang, A.V. Vasilakos, A survey on trust management for Internet of Things, J. Comput. Syst. Sci. 42 (2014) 120–134.

[73] Z. Yan, Y. Chen, Y. Shen, A practical reputation system for pervasive social chatting, J. Comput. Syst. Sci. 79 (5) (2013) 556–572.

[74] Z. Yan, P. Wang, W. Feng, A novel scheme of anonymous authentication on trust in pervasive social networking, Inf. Sci. 445 (2018) 79–96.