EBFL: An Efficient Blockchain Framework for Federated Learning Services

Ze Yin, Haotian Wang, Chubo Liu, Yan Ding, *Member, IEEE*, Keqin Li, *Fellow, IEEE*, and Kenli Li, *Senior Member, IEEE*

***Abstract*—Federated Learning (FL) has emerged as a key framework to deliver AI services, recognized for its capability to construct global models while ensuring individual data. Nevertheless, FL heavily relies on a central server, which introduces significant challenges for participants to collaborate effectively and substantially limits the scalability of FL. Blockchain-based FL (BFL) offers a promising solution by replacing the central server with a decentralized blockchain system, thereby establishing a secure and trustworthy environment for FL. However, current BFL approaches face challenges in balancing high computational overhead, consistency, and security. In view of this, this paper introduces EBFL, an efficient blockchain framework for FL services. EBFL incorporates both asynchronous and synchronous advantages. A DAG-based (Directed Acyclic Graph) asynchronous computation enhances computational efficiency by mitigating delays caused by slow devices and reducing unnecessary waiting due to frequent synchronized consensus. Simultaneously, a periodic synchronized consensus mechanism is introduced during asynchronous training to ensure consistency, thereby improving security and model accuracy. Additionally, taking into account the unique characteristics of FL, we have designed a series of operations tailored for EBFL to further enhance the performance. Experimental results demonstrate that, compared to traditional synchronous BFL (TBFL) approaches, EBFL achieved a maximum speedup of up to 2.38× while retaining 92% of their accuracy. Subsequently, in-depth analytical experiments show that EBFL excels in both convergence speed and security, thereby confirming its potential to balance computational efficiency, consistency, and security.**

***Index Terms*—Blockchain framework, federated learning services, asynchronous computation, synchronous consensus.**

Received 26 March 2025; revised 27 November 2025; accepted 14 December 2025. Date of publication 18 December 2025; date of current version 5 February 2026. This work was supported in part by the National Key Research and Development Project of China under Grant 2021YFA1000600, in part by the Programs of National Natural Science Foundation of China under Grant 62441234, Grant 62502151, and Grant U25A20422, and in part by Changsha Natural Science Foundation under Grant kq2502277. *(Corresponding authors: Chubo Liu; Haotian Wang.)*

Ze Yin and Haotian Wang are with the College of Information Science and Engineering, Hunan University, Changsha 410082, China, and also with the National Supercomputing Center in Changsha, Changsha 410082, China (e-mail: zyin@hnu.edu.cn; wanghaotian@hnu.edu.cn).

Chubo Liu, Yan Ding, and Kenli Li are with the College of Information Science and Engineering, Hunan University, Changsha 410082, China, also with the National Supercomputing Center in Changsha, Changsha 410082, China, and also with the Ministry of Education Key Laboratory of "Fusion Computing of Supercomputing and Artificial Intelligence", Changsha 410082, China (e-mail: liuchubo@hnu.edu.cn; ding@hnu.edu.cn; lkl@hnu.edu.cn).

Keqin Li is with the College of Information Science and Engineering, Hunan University, Changsha 410082, China, also with the National Supercomputing Center in Changsha, Changsha 410082, China, and also with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TSC.2025.3646060

I. Introduction

In recent years, Machine Learning as a Service (MLaaS) has gained significant attention due to its ability to extract valuable and accurate knowledge from large amounts of data [1], [2], [3]. However, traditional approaches typically require data owners, such as device manufacturers and users, to upload their private data to a central server. This raises concerns about privacy and security, making data owners reluctant to share sensitive information. Consequently, FL has emerged as a key paradigm in delivering machine learning services. By allowing participants (data owners) to locally train models without exchanging private data, FL enables collaborative model training across devices while preserving data privacy and security. This approach effectively addresses the challenge of data silos and facilitates secure collaboration to improve services of global models [4].

However, several drawbacks remain [5]. FL relies on a central server, introducing significant risks such as malicious modifications to the global model, which compromise data security and model integrity. Moreover, reliance on a central server can severely undermine system reliability. For instance, a single-point failure in the central server could lead to the complete collapse of the FL system [6]. Furthermore, the inherent distrust among participants impedes their enthusiasm to actively engage in the FL.

To tackle the above drawbacks, researchers propose to combine FL and blockchain. As illustrated in Fig. 1, the central server is replaced by a blockchain system, which is a decentralized, distributed ledger based on the P2P (peer-to-peer) network [7]. Blockchain technology addresses several key issues in TFL. Its decentralized architecture eliminates the risk associated with reliance on a central server, thereby enhancing both security and system reliability. Additionally, in a distrustful environment, blockchain establishes a credible platform for participants by distributing control among all participants (or miners) [8]. This feature enables participants to securely collaborate and train a high-quality global model, even in the presence of mutual distrust.

Given this, several BFL approaches have been proposed [5], [9], [10], which are generally categorized into synchronous and asynchronous approaches [11], [12], [13]. In synchronous approaches, participants require synchronized training and must wait for the synchronization consensus. Synchronous approaches ensure that all participants equally engage in FL, thus fully utilizing the private data of participants. However, frequent synchronization introduces unnecessary time overhead,

such as frequent waiting. Additionally, disparities in computational power cause high-performance devices to wait for slower ones, leading to inefficient resource utilization. In contrast, asynchronous approaches introduce asynchronous training and consensus, improving computational efficiency. However, the asynchronous mechanism poses challenges for participants in balancing computational efficiency, consistency,[1] and security. This imbalance can lead to slower convergence speed, decreased accuracy, increased vulnerability to malicious attacks, and reduced system stability.

To address the above problems, this paper proposes an **E**fficient **B**lockchain Framework for **F**ederated **L**earning Services (EBFL), which integrates the advantages of both synchronous and asynchronous approaches. Specifically, EBFL enables participants to train their local models asynchronously based on DAG design. Simultaneously, we introduce a periodic consensus mechanism, which requires participants to engage in synchronous consensus at a fixed interval. The periodic consensus could ensure consistency and optimal model-based training. Moreover, based on the characteristic of our DAG-based structure, we present a series of operations such as Model transactions (MT) selection and reward distribution. These strategies can further ensure the accuracy and security of the EBFL framework.

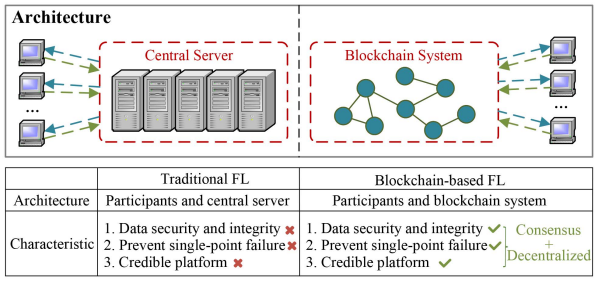
Our main contributions are as follows:

- We propose EBFL, an efficient BFL framework that integrates the advantages of both synchronous and asynchronous approaches. Asynchronous training enhances efficiency, while the periodic synchronous consensus mechanism ensures consistency.
- We introduce a set of operations to enhance the performance of EBFL. Specifically, we introduce an MT selection strategy that helps participants choose the most suitable MTs during asynchronous training, along with a periodic consensus mechanism to ensure consistency. Building on this consensus mechanism, we propose a distribution strategy that ensures fair reward allocation.
- We conduct extensive experiments, and the results indicate that, compared to TBFL approaches, EBFL achieves a maximum speedup of up to $2.38\times$ while retaining 92% of their accuracy. Furthermore, comprehensive analyses confirm that EBFL provides robust security, effective convergence, and high efficiency.

The remainder of the paper is organized as follows. Section II introduces the related work. In Section III, we present the EBFL framework in detail. We construct a series of numerical experiments and extensive analyses in Section IV. Finally, Section V concludes the proposed work.

II. Related Work

As privacy concerns gain increasing attention, BFL approaches have gradually become a key research hotspot and are being applied into various fields [14] across industrial IoT [15], [16], edge computing [17], [18], fog computing [19], IoV (internet of vehicles) [20], [21], smart grid [22], heavy haul railway [23], medicine [24], [25], and recommendation [26].



	Traditional FL	Blockchain-based FL
Architecture	Participants and central server	Participants and blockchain system
Characteristic	1. Data security and integrity ✗ 2. Prevent single-point failure ✗ 3. Credible platform ✗	1. Data security and integrity ✓ 2. Prevent single-point failure ✓ 3. Credible platform ✓ Consensus + Decentralized

Fig. 1. The comparison of traditional FL (TFL) approaches and BFL approaches.

Generally, existing BFL approaches are categorized into synchronous and asynchronous approaches as illustrated in Table I. Different from TFL approaches, synchronous approaches replace the central server with a blockchain system. Participants upload their local models to the blockchain system, and workers (or miners) collect these models to generate a new block through a consensus process. Participants then download the new block and update their models based on the transactions within the block. In contrast, asynchronous approaches adopt an asynchronous or hybrid consensus mechanism. For example, participants train their local models asynchronously on a DAG-based blockchain structure. Finally, a global model is achieved through collaboration.

It is worth mentioning that, in this paper, the model generated by aggregating local models during the training process is termed as periodic model, while the final model obtained through FL is referred to as the global model.

A. Synchronous Approaches

The most straightforward way to combine the blockchain and FL is the synchronous approaches [5], [19]. These approaches replace the central server with a blockchain system and design novel frameworks tailored to FL.

One of the most synchronous popular approaches is BlockFL [5], which replaces the central server by a decentralized blockchain system to enhance security and robustness. Peng et al. [32] propose VFChain, which achieves verifiable and auditable FL through a distributed committee consensus and a novel authenticated data structure.

However, certain inherent drawbacks still impede the widespread application of these approaches. All participants are required to conduct synchronous training and consensus frequently, which forces all participants to await the completion of the consensus process before commencing subsequent training round, leading to unnecessary delays that reduce the whole computational efficiency. Additionally, the mismatch of different computational power causes inefficient resource utilization, further limiting computational efficiency.

B. Asynchronous Approaches

Given the limitations of synchronous approaches, some studies propose semi-asynchronous or even (full) asynchronous

[1] In asynchronous approaches, consistency means participants share uniform information during training process. Without synchronization, they may train on different model versions, hampering convergence, lowering accuracy, and increasing vulnerability to malicious attacks.

TABLE I
THE COMPARISON BETWEEN CURRENT APPROACHES AND OUR EBFL. '/' INDICATES THAT THERE IS NO MENTION OF RELATED CONTENT OR EXISTING TECHNOLOGIES IN THE ARTICLE, AND 'SPOF' MEANS SINGLE-POINT OF FAILURE.

		Synchronous approach			**Asynchronous approach**				
		FedAvg [27]	DPFLA [28]	BlockFL [5]	BFLC [29]	DAG-FL [30]	PermiDAG [20]	S-BHAFL [31]	***Our EBFL***
Design	Category	TFL		BFL					
	Aggregation	Centralized		Decentralized					
	Incent	/		Yes			/		Yes
	Architecture	Central server		Single chain		Three-layer architecture	Hierarchical architecture		DAG-based architecture
Security	Poisoning defense	/	Yes	/	Yes		/	Yes	
	SPoF defense	/		Decentralized aggregation					
	Traceability	Not discussed		High			Limited		High
Consistency				High		Limited	High		
Efficiency				Low	Limited	High			

approaches. These approaches implement asynchronous training or consensus mechanisms, thereby accelerating the performance of the system.

We first review traditional (non-blockchain) FL approaches. Roy et al. propose BrainTorrent [33], a decentralized, server-less FL framework in which the central server is removed and clients are connected via a peer-to-peer overlay network. In each round, clients perform local training, then broadcast their current model, pull newer models, and conduct data-size–weighted aggregation followed by local fine-tuning. The entire process does not rely on server-side round control, and any client can dynamically initiate updates, realizing train-and-aggregate at the client side. Compared with centralized FL, BrainTorrent weakens synchronization constraints, avoids single-point failures and server bottlenecks, and reduces the overhead incurred by strictly synchronous aggregation. In vehicular edge computing, Mazloomi et al. propose MS-FL [34], a multi-server FL framework that deploys roadside unit (RSU) servers along the road to coordinate local training and aggregation for vehicles in their coverage areas while exchanging models over high-speed backhaul links. MS-FL further adopts performance-driven aggregation to evaluate, re-weight, and penalize outlier models across servers, and allows vehicles to continue training when moving between RSUs, thereby preserving updates that would otherwise be lost due to mobility. These traditional FL schemes, through distributed-server or server-less designs, eliminate the need for participants to trust a single central server and, to some extent, the risk of system crashes caused by a single-point of failure, while enabling more flexible training and update patterns. However, these approaches still suffer from two architectural limitations: (1) training and aggregation are not recorded in a complete, immutable manner, making auditing and traceability across rounds difficult; and (2) there is no unified global consistency guarantee, so different participants may simultaneously hold inconsistent model versions.

Then, we review the BFL approaches. Li et al. [29] propose a blockchain-based decentralized FL framework named BFLC. BFLC allows participants to asynchronously upload local models. However, during this phase, they cannot update their models based on others. FL progress still depends on the synchronous mechanism which limits efficiency. Shi et al. propose HySync [35], a hybrid FL method with effective synchronization. Unlike fully asynchronous schemes that update the global model upon every client upload, HySync introduces a time window (denoted as ω) to aggregate updates within the window using staleness-weighted averaging, performing a single global update per window. HySync reduces the number of global iterations and consequently mitigates the staleness of client updates. Sun et al. propose FedSEA [36], a semi-asynchronous FL framework for extremely heterogeneous devices. FedSEA predicts client-side completion times to dynamically adapt synchronization and training configurations. Moreover, it enables extremely slow devices to train smaller models, ensuring inclusive participation and improving efficiency under heterogeneous resource conditions. Abdmeziem et al. propose a blockchain-based FL approach [37] for IoT environments. The study addresses the node selection problem through a two-step hybrid mechanism that combines a reputation-scoring method with deep reinforcement learning (DRL). To handle device heterogeneity, it also introduces a multi-level aggregation strategy, thereby offering a novel and adaptive solution for efficient node selection in blockchain-based FL systems. Cao et al. [30] propose DAG-FL, a DAG-based FL framework that enhances efficiency through distributed asynchronous updates. DAG-FL does not adopt a synchronous consensus mechanism. Instead, each participant independently performs asynchronous voting-based consensus and training. This independence reduces consistency during training, potentially slowing convergence and increasing vulnerability to attacks. Additionally, several studies adopt a hierarchical architecture [20], [31]. Lu et al. [20] propose a blockchain-empowered asynchronous FL framework named PermiDAG for the IoV, employing a hybrid structure in which RSUs maintain the main chain, while vehicles conduct training tasks within local DAGs. Chen et al. [31] propose S-BHAFL, a hierarchical asynchronous federated learning framework that combines local training and gateway aggregation with committee-based consensus. In this architecture, the local layer is responsible for client-side training and gateway-level aggregation, while the global layer performs committee-driven validation and consensus to ensure global security. However, hierarchical data storage can cause global validation to depend on local layer data, negatively impacting traceability.

In summary, although these approaches improve training efficiency through asynchronous training and consensus, they also tend to degrade model accuracy, weaken consistency, or undermine security.

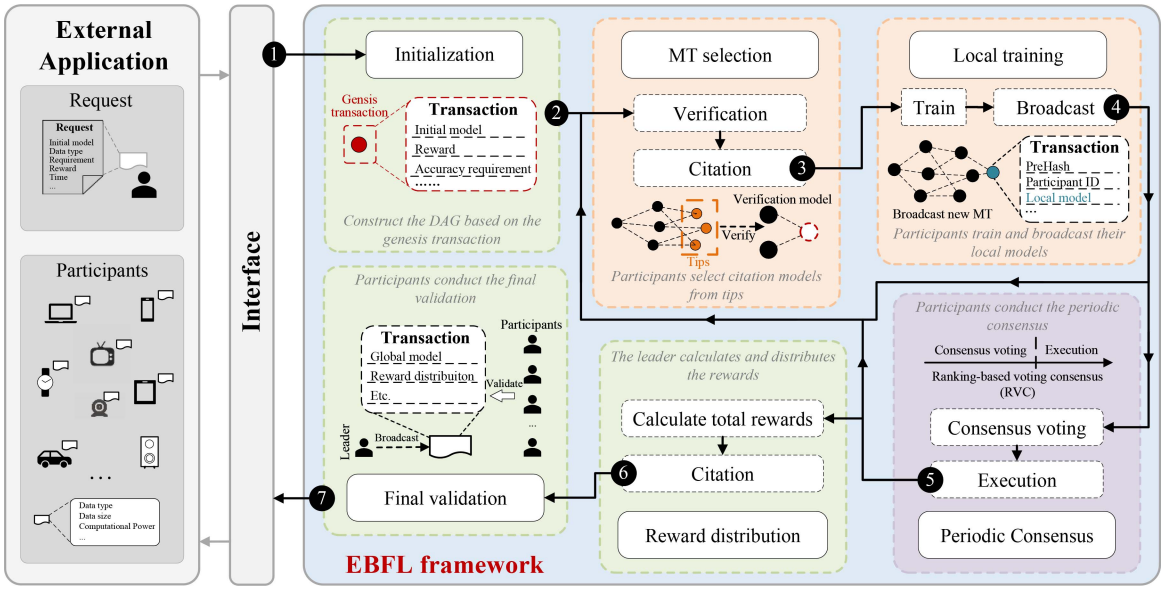


Fig. 2. The flowchart of the EBFL framework. EBFL is a synchronous-asynchronous framework where MT selection and local training (outlined with an orange dashed border) represent the asynchronous training process, while periodic consensus (outlined with a violet dashed border) represents the synchronous consensus mechanism. It primarily consists of the following steps: initialization, MT selection, local training, periodic consensus, reward distribution, and final validation.

To tackle the above challenges, we propose EBFL, which integrates the advantages of both synchronous and asynchronous approaches. It enhances computational efficiency through DAG-based asynchronous training while maintaining consistency via a synchronous periodic consensus mechanism to improve security and accuracy.

III. The Framework of EBFL

In this section, we detail our proposed EBFL framework. Section III-A presents an overview of the EBFL. Sections III-B–III-G provide details of EBFL.

A. Overview

It is worth noting that EBFL focuses solely on FL. Like many previous studies, we assume the presence of an authoritative entity, referred to as the Interface, as shown in Fig. 2. The Interface, which could be either a central server or a blockchain system, is not the primary focus of our research. Instead, it acts as an intermediary between the EBFL system and external applications. Its authority is limited to task distribution, result collection, and participant management, with no involvement in model training, communication, or consensus processes. Specifically, when a data requester submits a request to the Interface, the Interface constructs a genesis transaction (including the initial model, total reward, accuracy requirements, etc.) and selects the most relevant participants for the required data type to collaboratively train the model. After the FL process based on EBFL, a finish transaction (including the global model, reward distribution, signatures of participants, etc.) is submitted to the Interface, which then submits the global model to the data requester and distributes rewards to the participants based on the reward distribution.

Based on the genesis transaction broadcasted by the Interface, our EBFL trains a high-quality global model through the collaborative efforts of all participants. The flowchart of EBFL is shown in Fig. 2.

In Step ❶, when the Interface receives a request, it selects the participants and broadcasts the genesis transaction to these participants. In Step ❷, based on the genesis transaction, all participants maintain a local DAG for training. In Step ❸, participants select verified MTs (model transactions), and further select a citation model as the initial model. In Step ❹, based on their selected initial model, participants train their local models and publish their new MTs. In Step ❺, all participants engage in periodic consensus to vote on a consensus ranking at fixed intervals. This periodic consensus determines the status of the FL process, such as whether the periodic model meets the required accuracy, the leader for the next round, the new periodic MT, the rewards, and other related matters. Specifically, if a majority of participants believe that the periodic model meets the required accuracy or reaches the maximum training iterations, the leader will conduct reward distribution as shown in ❻. Otherwise, EBFL will repeat the FL process (return to ❸) or re-select the leader. The details are shown in Section III-E. In Step ❻, based on the results of the previous periodic consensus, reward distribution to all participants is calculated. In Step ❼, the leader will broadcast a final transaction to the participants. The participants then validate the information in the final transaction. If the validation is successful, they broadcast their signature fields. Finally, when over $2f+1$ participants have broadcast their signature, the leader will construct a finish transaction (including the final transaction and signatures) and submit it to the Interface.

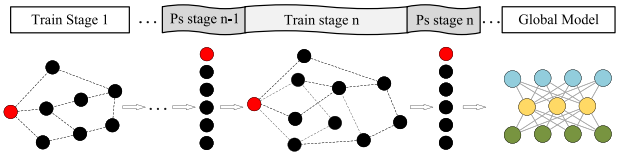


Fig. 3. The diagram of the FL process in the EBFL framework. The design integrates both synchronous and asynchronous approaches, with a periodic consensus (abbreviated as Ps in the diagram) phase occurring after each period of the asynchronous training phase.

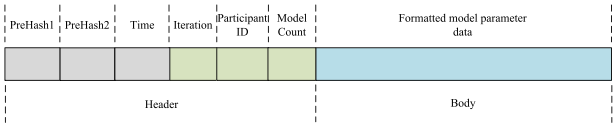


Fig. 4. The structure of an MT. An MT is composed of a header and body, where the header includes essential information, and the body includes model data.

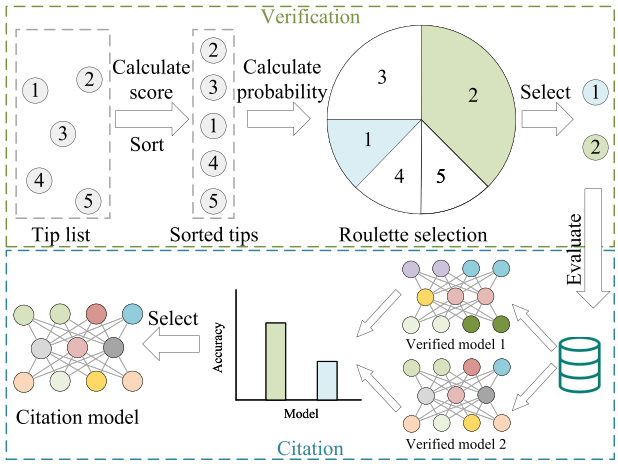


Fig. 5. The diagram of MT verification and citation. When participants need to choose an MT, they first verify two candidate MTs. After evaluating local models derived from these MTs, they select the one with the highest accuracy as the citation MT.

In the steps mentioned above, MT selection and local training (Steps ❸ and ❹) represent DAG-based asynchronous training, while periodic consensus (Step ❺) represents the periodic synchronous consensus. Fig. 3 illustrates the alternating synchronous-asynchronous training approach. This design allows EBFL to enhance computational efficiency while ensuring consistency.

B. Initialization

When a data requester needs a model, they submit a request to the Interface. The Interface then selects participants and constructs a genesis transaction, which includes essential information such as the initial model, total rewards, required accuracy, and other relevant details. This transaction provides the necessary foundation for subsequent training.

EBFL employs a DAG-based structure that incorporates both asynchronous and synchronous mechanisms. As a result, all participants are required to maintain a local DAG for training. In the first round, a leader is randomly selected, who will be responsible for managing the periodic consensus during that round.

C. MT Selection

Given the unique structure of the DAG and the characteristics of parallel asynchronous training by participants under this structure, how participants choose an appropriate MT is a crucial issue. A reasonable strategy for MT selection helps train high-quality global models and reduce malicious behavior among participants, thereby improving both security and accuracy.

Before discussing MT selection, we first present the fundamental component of the EBFL. MTs are directly stored in a DAG-based blockchain during the FL process. MTs can be categorized into verified MTs and unverified MTs (called tips). MTs contain essential information for FL training, and the structure of MTs is shown in Fig. 4. All participants are required to submit their MTs to blockchain network based on the predefined specific structure. $PreHash1$ and $PreHash2$ represent the hashes of two verified MTs, where PreHash1 is the cited one. Participants are able to abandon old tips based on the $Time$ field. $Iteration$, $ParticipantID$, and $ModelCount$ represent the consensus phase count, the identity information of participants, and the number of training rounds in a given consensus stage, respectively.

In our design, MT selection primarily comprises two components: verification and citation, as illustrated in Fig. 5. Participants need to verify several local MTs, and evaluate these verified MTs. The MT with the highest accuracy is selected as the citation MT. Then participants train their local models based on the citation model. Since the Verification phase forms the basis of the Citation phase, and the Citation phase only requires participants to evaluate verified MTs using private data, this section primarily focuses on designing a verification strategy to ensure that participants can select the most appropriate MTs.

IOTA has demonstrated that verifying two transactions before publishing a new transaction achieves an effective trade-off between security and efficiency [38]. In our verification strategy, we also follow this idea. After conducting extensive research and reflection, we believe that the critical factors in MT selection can be summarized into three principles, i.e., (1) prioritize the most recently generated tips, (2) prioritize tips on the highest recognized chain while maintaining a certain degree of randomness, and (3) prioritize tips with high accuracy. Among them, the first two principles are primarily associated with the verification phase and should be essential components of the MT selection strategy. Meanwhile, the third principle is specifically relevant to the citation phase. Given three principles, we design a novel MT selection strategy based on score metrics and roulette selection strategy. Inspired by the concept of cumulative weight in IOTA, we define the score as shown in (1)

$$s_v = w_v + \sum_{i \in ver(v)} w_i, \tag{1}$$

where v is the MT that needs to be verified, $ver(v)$ means that the MTs are verified (both directly or indirectly) by MT v. Depending on the specific task, this value can be configured accordingly to achieve different priorities. w_i represents the weight of MT i. Based on the score, participants are restricted to prioritizing the latest tips located on the highest recognized chain, which can effectively limit the lazy behavior (i.e., always selecting previously verified MTs) of participants. Then, the normalized score list $norm(s)$ of all tips is input into the roulette selection algorithm to choose two tips. The selection probability for each tip is calculated as shown in (2)

$$\Pr_v = e^{norm(s_v)} / \sum e^{norm(s)}, \tag{2}$$

where s is the score list of all tips, and s_v means the score value of the v_{th} MT. $norm(s_v)$ represents the normalized score value of the v_{th} MT. The equation is shown in (3)

$$\text{norm}\,(s_v) = L + \left(\frac{s_v - \min(s)}{\max(s) - \min(s)} \cdot (U - L) \right), \tag{3}$$

where L and U are predefined lower and upper bounds, respectively. As the length of the DAG blockchain increases, the score values of tips will continue to grow. Human intervention in the selection threshold of tips may cause significant perturbations to the stability of the blockchain, and it is also vulnerable to malicious attacks. Therefore, this paper avoids the above problem by mapping the score to a fixed interval. After MT selection, two tips are selected according to the score. Participants evaluate the accuracy of such two tips based on their private data and eventually choose the model with higher accuracy as their citation model.

EBFL guarantees participants prioritize the most recently generated tips and selection randomization based on score metric and roulette selection strategy. Then, participants can select a high-accuracy model by tip evaluation. Therefore, the proposed EBFL is able to achieve the expected goal.

D. Local Training

After participants select the verified MTs and citation MT based on the MT selection strategy, they utilize private datasets to train their local models based on the citation model. Once training is completed, participants construct new MTs according to the predefined MT structure (as shown in Fig. 4) and broadcast them to the blockchain network. Other participants will add the MTs and their score values to their tip pool when they receive them, awaiting subsequent verification. During the training process, EBFL allows participants to enhance privacy and security by using DP, which adds noise to model parameters and defends against inversion attacks.

Since the MT selection and periodic consensus require participants to evaluate local models from others, this process incurs additional time costs. Participants must maintain an accuracy list that records the most recent accuracy of models from all participants. When a participant publishes a new MT, all participants add it to the tip pool, evaluates its accuracy, and updates their accuracy lists accordingly. The design enables the parallel execution of training and evaluation. If the MT selection phase and periodic consensus require model accuracy validation, participants first check the accuracy list for a matching entry (as shown in Fig. 4, which includes fields such as Iterations, Participant ID, and Model Count). This approach hides the latency of validation within the training process, thereby improving the efficiency of validation.

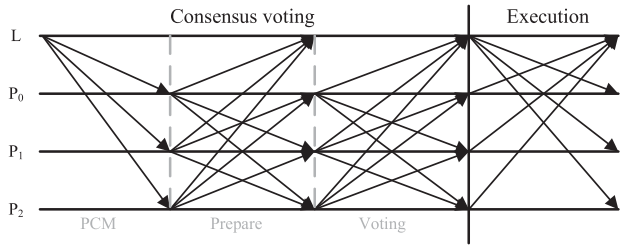


Fig. 6. The diagram of the RVC approach on EBFL. L means the leader, and P_0-P_2 stand for three participants.

E. Periodic Consensus

As previously mentioned, inconsistency among participants in asynchronous training can lead to convergence issues, reduced model accuracy, and increased vulnerability to attacks. Therefore, maintaining consistency is crucial. In EBFL, we propose a periodic consensus strategy. Specifically, all participants need to vote for a consensus ranking synchronously at fixed intervals. The consensus ranking will serve as evidence of the training results for all participants in the current round. It is used to determine whether the periodic model meets the required accuracy, to select the leader for the next round, and to distribute rewards among participants.

Unlike the traditional blockchain, DAG-based blockchain executes asynchronously and introduces significant uncertainty in the states of different participants. Therefore, in our framework, we present a novel periodic DAG-based blockchain structure, which combines the advantages of both synchronous and asynchronous mechanisms. As mentioned above, all participants engage in the periodic consensus at fixed intervals. The interval is determined by the training speed of participants, ensuring that the majority can complete a certain number of epochs within each round. The asynchronous structure is utilized to improve efficiency on the basis of ensuring a certain degree of synchronization to maintain consistency periodically.

Given that participants possess diverse datasets, it is natural for even honest participants to achieve varying accuracy levels when evaluating the same local model. As in most previous works [19], [39], [40], [41], we assume that the data distributions among honest participants are similar. Consequently, when honest participants test local models from different participants, although the accuracy rankings may vary slightly, they generally exhibit a high degree of similarity in their overall trends. In other words, honest participants tend to demonstrate consistent behavior. Therefore, for periodic consensus, this paper adopts a ranking-based voting consensus (RVC) approach. As illustrated in Fig. 6, RVC consists of two phases, consensus voting and execution, with the consensus voting phase further subdivided into three sub-phases (i.e., PCM, prepare, and voting). The process of RVC is similar to PBFT, ensuring the secure and

reliable operation of EBFL when $N \geq 3f + 1$. Here, N stands for the total number of participants. Meanwhile, f represents the maximum tolerated total number of both malicious participants and faulty honest participants, where the latter refers to honest participants who fail to vote correctly due to factors such as network fluctuations, data distribution issues, or other unforeseen circumstances. The detailed process is described as follows.

- *PCM stage:* The leader initiates the periodic consensus process at predefined intervals. First, the leader broadcasts a periodic consensus message (PCM) containing its ranking, iteration, and view number.
- *Prepare stage:* Upon receiving the PCM, participants halt their local training processes. If they verify the correctness of the PCM, they then send a prepare message to the other participants.
- *Voting stage:* When participants receive more than $2f + 1$ prepare messages, they will calculate the accuracy ranking of up-to-date MTs using their private datasets. If participants believe that the ranking of the leader is similar to their own, they will vote in favor of the ranking. Additionally, if they support the ranking, they will also vote on whether the highest-ranked model meets the accuracy requirement.
- *Execution stage:* Based on the periodic consensus described above, EBFL will perform different actions based on different voting results.

The execution phase proceeds based on voting results, which can be categorized into three cases, as detailed below.

- *Case 1:* If the majority of participants agree with the ranking and believe that the periodic model meets the accuracy requirement, the leader will package the voting results and broadcast a confirmation message. Then the leader will calculate the reward distribution for all participants and broadcast a final transaction containing essential information, such as the global model, reward distribution, and other relevant details. Participants will validate the transaction and broadcast their signatures by signing the hash value of the transaction. Once the leader collects the $2f + 1$ signatures, it broadcasts a finish transaction containing the final transaction and signatures to the Interface.
- *Case 2:* If the majority of participants agree with the ranking but believe that the periodic model does not meet the accuracy requirement, the leader will package the voting results and broadcast a confirmation message. The ranking will be determined as the consensus ranking. EBFL will then proceed with a new round of training based on this consensus ranking.
- *Case 3:* If a majority of participants vote against the ranking, the leader will be replaced, and a new voting phase will be conducted. The replacement of the leader will be conducted in descending order based on the ranking result from the previous periodic consensus. If there is no previous periodic consensus (i.e., this is the first periodic consensus), the leader will be randomly selected. This process will repeat until participants reach a consensus vote.

In case 1, EBFL will proceed to the final validation. In case 2, based on the voted consensus ranking, the participant who publishes the periodic MT will serve as the leader for the next consensus, and the periodic MT will be chosen as the genesis MT for new round training. The FL process and periodic consensus will repeat until the periodic model meets the accuracy requirement or the maximum iteration is reached. In case 3, the leader will be replaced, and a new round of RVC will be launched.

Obviously, the periodic consensus is led by the leader, and all participants work together to achieve consensus. The leader is selected randomly in the first round DAG process, and the participant with the highest average accuracy in the previous validation is the leader in the follow-up DAG-based FL process. We believe that participants with high average accuracy are more likely to be honest, so the leader is more likely to be an honest participant. In addition, participants will verify and supervise the behavior of the leaders. If a leader deliberately falsifies the accuracy ranking of local MTs or maliciously alters the parameters of the periodic model, they will be regarded as a malicious participant, and a new leader will be selected. Meanwhile, leaders also need to supervise the behavior of participants. If participants have malicious behaviors, such as frequently proposing low-quality MTs, it is necessary for leaders to initiate votes to identify and punish such malicious participants.

F. Reward

Rewards are one of the essential components of the blockchain. An equitable distribution scheme can effectively motivate participants to engage in FL. In our framework, we mainly reward honest participants. However, given that participants keep their data private, the detailed characteristics of individual datasets are unknown. Therefore, we adopt accuracy (do not consider the information of datasets) as the primary metric because we assume that, to some extent, participants with high average accuracy tend to be honest. In addition, we should punish such participants with malicious behaviors (such as lazy behavior), although they may have high average accuracy. We regard these participants as speculators, so we refuse to reward them. For example, some participants possess high-quality private data, allowing them to achieve higher accuracy with fewer training iterations during model training. However, participants who complete significantly fewer iterations than the average, whether due to lazy or malicious behaviors, will not be rewarded.

Based on the concept mentioned above, we design our reward distribution strategy. Distributing rewards fairly within the DAG is challenging due to its complex structure. However, this challenge is addressed through our unique design, whereby rewards are allocated during each periodic consensus phase. Specifically, each periodic consensus phase evenly splits the rewards, expressed as $r_t = R_{tol}/n_c$, where r_t denotes the reward allocated to each periodic consensus phase, R_{tol} is the total reward for the FL task from data requesters, and n_c indicates the maximum count of periodic consensus. The reward allocation scheme for each round is detailed in (4)

$$\begin{cases} r_{p_m} = \alpha \cdot r_t, \\ r_{p_i} = \beta \cdot r_t \cdot \text{len}\left(p_t\right)^{-1}, \text{ if } p_i \in p_t, \\ r_{p_l} = \gamma \cdot r_t, \end{cases} \tag{4}$$

where r_* denotes the reward of $*$, r_t represents the total reward of each consensus stage. The coefficients α, β and γ are the predefined proportions. It is worth mentioning that $\alpha + \beta + \gamma = 1$ and $r_{p_m} + \sum r_{pi} + r_{p_l} = r_t$. As shown in (4), our reward distribution strategy among participants is divided into three components.

- p_m represents the participant who proposes the periodic/global MT. It is commonly assumed that participant p_m is honest, and possesses a high-quality private dataset. Therefore, it is justified to reward these participants for their contributions to the FL task.
- p_i represents the set of participants who are voted to receive rewards. This implies that the participants selected through voting should be rewarded for their honest behavior and high-quality local models during the period.
- p_l represents the leader. As described in Section III-E, the periodic consensus is managed by the leader. Therefore, the honest leader should be rewarded because of their additional work.

Based on the reward distribution strategy, EBFL allocates rewards to participants who demonstrate honest behavior and contribute high-quality local models. Since the reward distribution relies on periodic consensus, rewards are not distributed to all participants after every periodic consensus. When the majority of participants believe that the model meets the accuracy requirements or the FL process reaches the maximum training iterations, the leader will calculate the total rewards for each participant based on the consensus rankings from all periodic consensus rounds.

G. Final Validation

To ensure security, participants must perform a final validation before the leader submits the finish transaction to the Interface. The final validation aims to ensure that the leader cannot maliciously tamper with the global model, reward distribution, or other related data when submitting the finish transaction. This is achieved by relying on the signature fields of all participants.

As described in Section III-F, the leader will calculate rewards for each participant once a majority of participants agree that the periodic model meets the requirement. Then the leader will compose and broadcast a final transaction containing the global model and essential information, such as participant reward distribution. Participants will then validate this transaction and broadcast their signature fields. Once the majority of participants have broadcast their signature fields, the leader will broadcast the finish transaction, which contains both the final transaction and the participant signatures, to the Interface. The lifecycle of EBFL concludes. All participants are still required to retain the training data for a certain period for verification purposes. Once this period expires, the participants delete the data to free up storage space. In this context, it is implied that participants only need to store the training data temporarily, without requiring long-term data storage. Even if participants engage in multiple tasks, this approach will not cause excessive storage pressure. Consequently, this design is particularly advantageous for devices with limited storage capacity, such as IoT devices and mobile devices.

TABLE II
THE DESCRIPTION OF DATASETS

Datasets	Model	Train/Test Examples	Class
Flowers	MobileNetV3	2939/731	5
CIFAR-10	MobileNetV3	50000/10000	10

Upon receiving the finish transaction, the Interface validates its correctness based on signatures. It then submits the task to the data requester and distributes rewards to participants according to the allocation specified in the finish transaction.

IV. EXPERIMENT

In this section, we evaluate the performance of our EBFL framework. Specifically, in Section IV-A, we introduce the experimental settings, including the datasets and models used in experiments. In Section IV-B, we compare the accuracy of models obtained from our EBFL approach and baseline approaches. Finally, we analyze security, convergence, efficiency, parameters, and MT selection in Sections IV-C to IV-G.

A. Experimental Settings

Similar to previous studies [41], [42], we perform extensive simulation experiments to verify the effectiveness of EBFL. To assess the performance of EBFL, we select FedAvg[2] [27], BFLC [29], Non-synchronous Consensus DAG (NDAG), and S-BHAFL [31] as baseline approaches[3]. NDAG is a DAG-based blockchain approach that does not incorporate the periodic consensus process. FedAvg is a classic FL method, while TBFL is a synchronous scheme that replaces the central server with a blockchain system [5]. In our setting, we assume that each block contains all up-to-date local MTs, so the aggregation results produced by FedAvg and TBFL are very similar, and we refer to this baseline collectively as FedAvg (TBFL). BFLC is a novel BFL approach with a design that differs from our EBFL framework. S-BHAFL is a blockchain-based hierarchical asynchronous scheme. Additionally, to flexibly and conveniently construct data partitions, we focus on an image classification task and use several image datasets. A brief summary of these datasets is provided in Table II.

As mentioned above, the ultimate goal of FL is to train a high-quality global model without private data sharing, where initial models are provided by the requester. In our experiments, we choose MobileNetV3, a lightweight model, as the target model [43]. The code of MobileNetV3 can be found in https://github.com/Fafa-DL/Awesome-Backbones.

In this paper, we choose 3 classical metrics to evaluate the FL performance, i.e., precision, recall, and F1 score. To make the

[2]https://github.com/CedricShang/federated-learning-master/tree/main

[3]NDAG is a comparison approach specifically designed for this research. The implementations of BFLC and S-BHAFL in our experiments are based on the algorithmic descriptions provided in the original papers. For a fair comparison, we adapt their code to use the same MobileNetV3 model and datasets (Flowers and CIFAR-10) as our EBFL framework. However, some implementation details and parameter settings are not fully specified in the original papers, so our reimplementations may exhibit minor deviations from the original approaches described in those works.

TABLE III
THE PARAMETER SETTINGS OF EBFL AND ALL BASELINE APPROACHES USED IN OUR EXPERIMENTS

Method	Parameters	Values
EBFL	Time for one training slot	T
	Local epochs per training slot (T)	5
	Periodic consensus interval	$4T$
	Total local epochs per asynchronous period	20
	Maximum number of consensus rounds	5
	Normalization upper bound (U)	4
	Normalization lower bound (L)	1
	Average total local epochs per participant	100
NDAG	Local training epochs between model selection	5
	Average total local epochs per participant	100
FedAvg (TBFL)	Local training epochs per aggregation	5
	Number of aggregations	20
BFLC	Maximum updated blocks per aggregation	40
	Maximum number of aggregations	25
	Average total local epochs per participant	100
S-BHAFL	Local epochs per aggregation	5
	Number of aggregations	20

results intuitive and easy to understand, we present these metrics as percentages by multiplying their values by 100.

We shuffle and divide the dataset into 10 parts as the private dataset for the 10 participants. In the experiments, the parameters of all approaches, such as the learning rate, are set to be the same. Table III details the parameter settings of EBFL and baseline approaches. For EBFL, each training round consists of 5 epochs, with T denoting the time duration of a training stage (as shown in Fig. 3). All participants engage in periodic consensus at intervals of $4T$, and this process is repeated 5 times, amounting to a total of 100 training epochs for each participant. Due to the challenge of precisely regulating the training frequency for each participant within the DAG structure, we guarantee that the average total epochs for all participants in both EBFL and the NDAG algorithm remain approximately 100. In addition, we configure that the FedAvg (TBFL) conducts model aggregation every 5 epochs, totaling 20 model aggregations. For BFLC, aggregation is triggered when participants propose 40 update blocks, resulting in the creation of a model block with the updated global model. With a total of 25 aggregations, each participant typically undergoes an average of 100 training epochs.

B. Experimental Results

In this section, we conduct experiments to prove the performance of our EBFL framework.

Table IV shows that the EBFL framework achieves commendable accuracy compared to the FedAvg (TBFL), reaching 92% on both the Flowers and CIFAR-10 datasets. Specifically, in the Flowers dataset, the three metrics decreased by 3.63, 5.32, and 5.45, respectively. Similarly, in the CIFAR-10 dataset, the three metrics decreased by 5.25, 5.75, and 5.47. The reason is that participants are required to verify two MTs by utilizing the MT selection strategy, and eventually choose one as the citation MT by comparing the accuracy of two verified MTs based on their private dataset. Due to the verification strategy, EBFL may not take full advantage of the private data from all participants. Specifically, participants are required to choose MTs from tips. Some MTs with lower accuracy may not be selected first, resulting in such parts of the datasets being underutilized.

TABLE IV
THE NUMERICAL RESULTS

Dataset	Method	Prec.	Recall	F1 score
Flowers	**EBFL**	72.43	70.13	69.65
	NDAG	61.11	56.62	55.11
	FedAvg (TBFL)	76.06	75.45	75.10
	BFLC	68.64	67.54	67.58
	S-BHAFL	71.05	70.15	69.96
CIFAR-10	**EBFL**	69.28	68.78	68.89
	NDAG	65.31	64.00	63.88
	FedAvg (TBFL)	74.53	74.53	74.36
	BFLC	59.94	60.02	59.90
	S-BHAFL	54.17	54.39	53.84

However, in the real world, these datasets are often an important addition. Additionally, asynchronous approaches (such as DAG-based methods) suffer from difficulties in maintaining consistency among participants, leading to a decrease in accuracy. This is also a common challenge faced by existing asynchronous FL frameworks. Despite this, the MT selection strategy also offers significant advantages. Through the strategy, honest participants are able to avoid pollution from low-quality local models and attacks from malicious participants. We conduct related experiments as shown in Section IV-C. Furthermore, compared with NDAG, EBFL has shown significant performance enhancements, with increases of 14.54 and 5.01 in F1 score on the Flowers and CIFAR-10 datasets, respectively. This is because EBFL introduces the periodic consensus mechanism, which ensures consistency among participants. Meanwhile, the periodically voted optimal model allows all participants to train based on the current optimal model, further improving accuracy.

Finally, we compare EBFL with hybrid approaches that combine synchronous and asynchronous training rather than relying on a single training mode. On the Flowers dataset, EBFL achieves accuracy comparable to S-BHAFL and slightly outperforms BFLC. On the CIFAR-10 dataset, EBFL achieves noticeably higher accuracy than both BFLC and S-BHAFL.

C. Security Analysis

As mentioned above, to improve the efficiency of BFL approaches, asynchronous mechanisms are introduced. However, asynchronous approaches face challenges in balancing computational efficiency, consistency and security. This paper proposes EBFL to tackle the aforementioned problems. Therefore, comprehensive security analysis is essential for demonstrating the robustness and reliability of the proposed approach.

In our EBFL framework, the periodic consensus is a non-negligible part to malicious participants. As described in (4), malicious participants realize that they may obtain more rewards when they get a higher ranking from periodic consensus. Therefore, they may attempt to maliciously tamper with the MT ranking of other participants during the consensus phase.

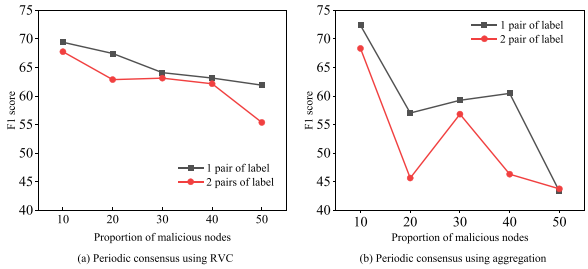


Fig. 7. The analysis of poisoning attacks based on label flipping. (a) shows the results when EBFL adopts the RVC approach in periodic consensus, while (b) presents the results when model aggregation is applied in periodic consensus. The experiment is conducted on the Flowers dataset, which comprises five classes. The black line represents the results of flipping one pair of labels, whereas the red lines indicate the results of flipping two pairs of labels.

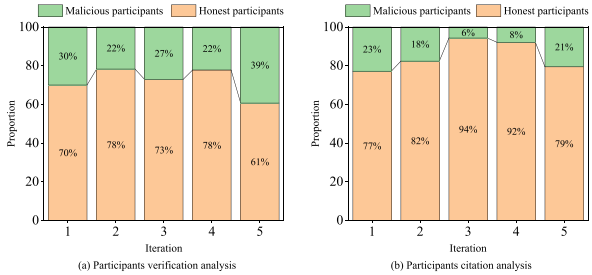


Fig. 8. The verification and citation analysis experiments for malicious and honest participants. (a) and (b) represent the verification and citation experiments, respectively. Among them, the orange segments represent the proportions for honest participants, while the green segments represent those for malicious participants.

However, as previously assumed, honest participants obtain similar results when evaluating the same local model. The voting process is able to protect the security of the EBFL system against attacks from malicious participants.

In the proposed EBFL framework, participants select two MTs and evaluate the accuracy of selected MTs. Then they train their local models based on the citation model so that the impact of poisoning attacks can be avoided to a certain extent. In addition, with the majority of participants being honest, the periodic consensus can ensure that participants always train their local models based on the high-accuracy periodic models of honest participants, thus effectively defending against poisoning attacks of malicious participants.

We also conduct poisoning attack experiments, and the results are shown in Fig. 7. In this experiment, we assume that some participants might attempt to destroy the system by adopting label flipping. To isolate the effect of the poisoning attacks, malicious participants are configured to exclusively conduct attacks during the training phase, while the remaining phases proceed unaffected. The results demonstrate that as the number of malicious participants increases, the accuracy of the global model gradually decreases. Fig. 7(a) shows that when there are 50% malicious participants in the system, the accuracy decreases by 7.5 and 12.39, respectively, compared to scenarios with only 10% malicious participants. However, we find that when there are only 30% malicious participants, the accuracy remains satisfactory (i.e., decreasing by 5.32 and 4.63, respectively) compared to 10% malicious participants. Since the experiments are conducted under situations where malicious participants perform attacks exclusively during the training phase, the observed limited accuracy degradation reflects the effectiveness of our EBFL framework in defending against poisoning attacks. As a comparison, we replace the RVC with the aggregation approach in the periodic consensus phase. The aggregation means the leader will aggregate all up to date models of all participants as a new periodic model. The results (as shown in Fig. 7(b)) indicate that when only 10% of participants are malicious, the accuracy of the aggregation-based approach is higher than that of the RVC-based approach, since the aggregation approach can fully exploit the data contributed by all participants. However, when the proportion of malicious participants exceeds 20%, the accuracy drops significantly. This degradation occurs because the aggregation approach indiscriminately aggregates all the latest local models, including malicious ones, which leads to a decline in the accuracy of the periodic model, gradient oscillations, and even convergence failure. These results demonstrate that, in complex adversarial environments, the proposed EBFL framework can provide stronger security guarantees than the aggregation-based baseline.

To further demonstrate the effectiveness of our EBFL framework, we analyze the citation and verification of MTs during the poisoning attack experiment, where the proportion of malicious participants is 50%. As shown in Fig. 8, the validation proportion of malicious participants is 28%, and the citation proportion is 15.2%. Although the experimental results are obtained under the assumption that malicious participants only conduct attacks during the training phase, we demonstrate that our MT selection and periodic consensus design can defend against malicious behavior. Additionally, the proposed EBFL framework is compatible with privacy protection strategies, such as differential privacy (DP). For instance, our EBFL could integrate the DP strategy by allowing participants to add generated noisy data to further protect the privacy and security of participants. It is noteworthy that the paper exclusively focuses on the framework design, thus we do not take the extra privacy protection strategy into consideration. In other words, our implementation and experiments do not include privacy-preserving policies such as DP.

D. Convergence Analysis

Convergence analysis is a crucial criterion in machine learning. A well-converged model generally performs more stably. Additionally, observing convergence helps determine the optimal point to stop training, thereby avoiding unnecessary computational resource usage and identifying potential overfitting. In FL, convergence analysis can reflect the effectiveness of collaborative training. A well-converged global model can effectively demonstrate the ability of the proposed framework in model selection, training, and validation.

In this section, we conduct a convergence analysis to assess the performance of the EBFL framework. As illustrated in Fig. 9(a) and (c), the accuracy exhibits a rapid increase during the first 5

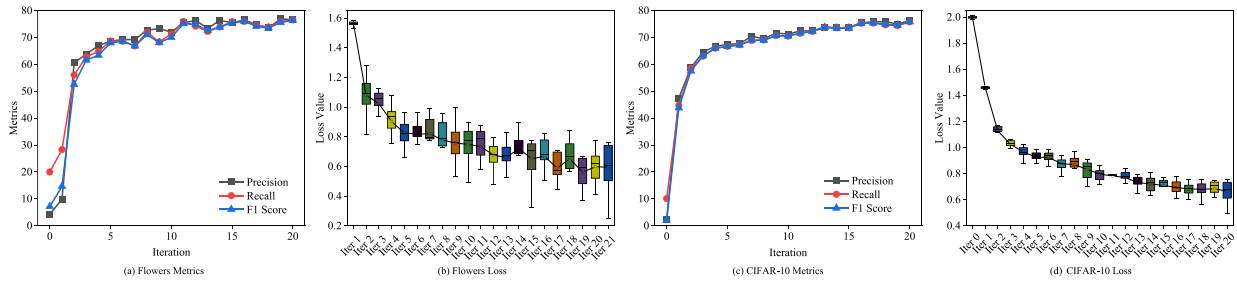


Fig. 9. The convergence analysis on Flowers and CIFAR-10. (a) and (b) illustrate the convergence of metrics and loss on the Flowers dataset, while (c) and (d) present the metrics and loss on the CIFAR-10, with all experiments conducted using 20 periodic consensus.

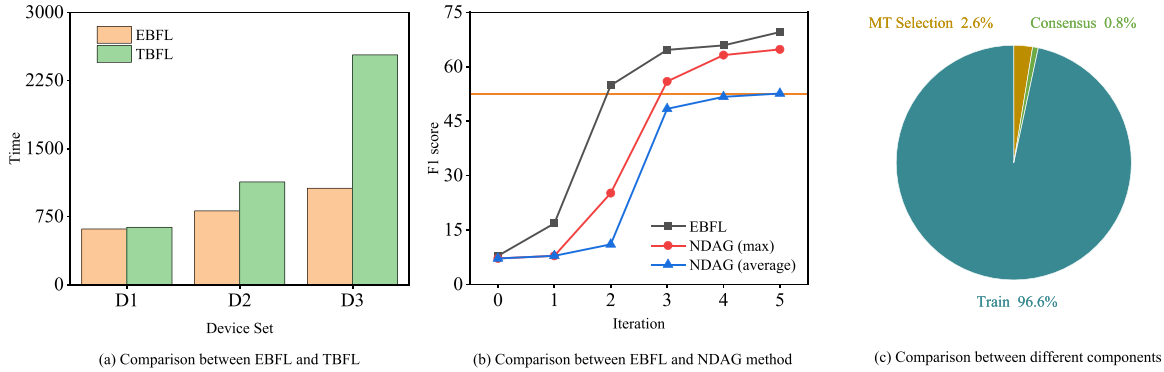


Fig. 10. The efficiency comparison. (a) compares the computational efficiency of EBFL with TBFL, (b) compares EBFL with the NDAG approach, and (c) presents the time-overhead proportions across different EBFL components.

periodic consensus phases. In the subsequent training process, the accuracy rises slowly and converges gradually. The highest values of precision, recall, and F1 score are 76.91, 76.53, and 76.35 on the Flowers dataset, and 76.28, 75.78, and 75.86 on the CIFAR-10 dataset, respectively. Fig. 9(b) and (d) illustrate the variation trends in loss values on the Flowers dataset and the CIFAR-10 dataset, respectively. The box plots show the distribution and variability of loss values for 10 participants using the periodic model. The connecting lines represent the mean loss values, which decrease rapidly during the initial stages and gradually stabilize, indicating that the model converges over time. The results demonstrate that the EBFL framework exhibits excellent convergence in the FL process, further validating the effectiveness of the proposed synchronous-asynchronous framework and MT selection strategies.

E. Efficiency Analysis

We propose EBFL to address the challenges of current BFL approaches. The EBFL framework adopts a synchronous-asynchronous structure that effectively balances computational efficiency, security, and consistency. In this section, we design experiments to evaluate its computational efficiency in comparison with TBFL and NDAG.

Based on theoretical analysis, the EBFL has higher training efficiency over the TBFL approach, attributed to its unique design. More specifically, in TBFL, participants must wait to reach a consensus after they perform a certain number of training epochs to obtain the global model. The need for frequent waiting among participants leads to a decrease in computational efficiency. Furthermore, different devices possess varying computational power, which may result in situations where faster devices wait for slower ones. This waiting, in turn, further affects the computational efficiency. The EBFL framework effectively addresses the limitations of TBFL by integrating synchronous and asynchronous architectures. Based on the unique design, the EBFL ensures consistency and security while enabling asynchronous training among participants, thereby minimizing the waiting time for consensus and counteracting the performance degradation due to varied device computing capabilities.

To validate our theoretical analysis, we conduct an experiment, with the results shown in Fig. 10(a). D1-D3 represent three groups, each consisting of 10 devices. In D1, every device operates at the same speed; in D2, the speed difference among devices is small (the slowest device runs at nearly 54% of the speed of the fastest); in D3, the speed difference is significant (the slowest device operates at 24% of the speed of the fastest). Since our periodic consensus involves multiple communication rounds, similar to Practical Byzantine Fault Tolerance (PBFT) [44], we assume that the BFT approach employs PBFT, with consensus time matching that of our periodic consensus. As the results show, when using the D1 group, EBFL and TBFL exhibit similar training speeds, with minor differences due to the number of consensus rounds. With the D2 group, EBFL achieves 1.39x

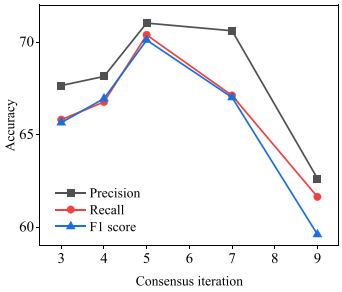


Fig. 11. The parameter analysis on the Flowers dataset. The x-axis represents the maximum number of consensus phases, and y-axis indicates accuracy.

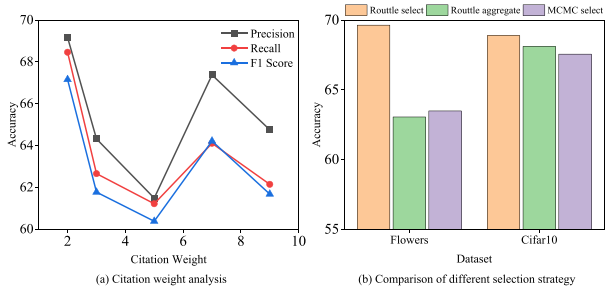


Fig. 12. The analysis of MT selection. (a) illustrates the impact of assigning different weights to the citation models on accuracy, while (b) compares the accuracy of three selection strategies on the Flowers and CIFAR-10 datasets.

speedup. In the D3 group, the significant differences in device speeds make the wait time in TBFL more pronounced, resulting in 2.38x speedup. In real-world scenarios, where significant differences exist between devices, EBFL demonstrates a performance advantage.

Compared to the NDAG approach, our EBFL framework introduces periodic consensus, allowing all participants to continue their training process based on the global optimal model. This enhancement leads to greater training efficiency in EBFL. Fig. 10(b) shows the comparison results between EBFL and the NDAG on the Flowers dataset, incorporating five periodic consensus phases. Since the NDAG lacks synchronous consensus, we calculate the maximum (red line) and average (blue line) F1 score values at intervals of every $4T$ within the training process. Based on the slope variation of the three lines, we can find that the training efficiency of EBFL is higher than that of NDAG. Specifically, we observe that the training rounds required by EBFL, NDAG (max), and NDAG (average) to reach the same model accuracy (as indicated by the orange line in the figure) occur around the second, third, and fifth rounds, respectively. Results show that EBFL achieves almost a 20% improvement in efficiency compared to NDAG (max), as it converges one round faster to achieve a similar F1 score, and almost a 60% improvement compared to NDAG (average), converging three rounds faster. The experimental results demonstrate that our unique design effectively enhances the accuracy, computational efficiency, and convergence speed of the FL process.

Finally, since EBFL introduces additional components such as MT selection and periodic consensus, we further analyze the time overhead of different components for a single participant within one asynchronous training round, as shown in Fig. 10(c). The results show that the training phase accounts for 96.6% of the total time, whereas the remaining components contribute about 3.4%, which is negligible compared with the training time. Therefore, the extra overhead is not the dominant factor affecting the overall efficiency.

F. Parameter Analysis

Finally, we conduct a parameter analysis visualization (as shown in Fig. 11) on the Flowers dataset to evaluate the impact of varying the maximum number of consensus phases.

We find that as the maximum number of consensus increases, the accuracy rises correspondingly. This trend is attributed to the increased frequency of participants training their local models using the global optimal models. However, when the maximum number of consensus phases exceeds 5, the improvement in accuracy tends to flatten out, and the accuracy declines when the number of consensus phases reaches 7. This is because we set the total training epochs to 100. When the number of consensus phases is set too high, participants lack adequate time to select and train their local models based on the models of other participants, which hampers the full utilization of the dataset. Based on our experimental results, we can demonstrate that the setting of the consensus phases significantly influences accuracy. Therefore, in our experiments, unless otherwise specified, the maximum number of consensus phases is set to 5.

G. MT Selection Analysis

The proposed EBFL is a hybrid design that combines the advantages of asynchronous training and synchronous periodic consensus. In the asynchronous phase, the selection of high-quality models on the DAG is crucial for ensuring both the security and the accuracy of the framework. In this section, we conduct extensive experiments to validate the effectiveness of the proposed MT selection strategy.

Fig. 12(a) shows the parameter analysis results. Among all the weight settings in this experiment, accuracy reaches its peak when the weight is set to 2, at which point the selection process achieves a balance between randomness and the preference for high-weight models. As the weight continues to increase, the accuracy initially declines. When the weight becomes very large, the randomness in MT selection diminishes and the process stabilizes, consistently favoring high-weight models. Although the accuracy partially recovers at this stage, the loss of randomness limits data utilization, leading to a lower upper bound on performance. Fig. 12(b) presents the accuracy comparison among different selection strategies. The results show that the proposed MT selection scheme achieves higher accuracy than both the MCMC-based and aggregation-based schemes, indicating that our strategy of evaluating candidate citation models using private validation data from participants is effective and reasonable.

In summary, Fig. 12(a) and (b) demonstrate the effectiveness of the proposed MT selection strategy. The weight setting allows participants to apply different preferences when selecting models, providing greater flexibility. Although participants may

occasionally make inaccurate judgments during the citation process due to the limited quantity or diversity of data in their private datasets, the periodic consensus ensures that the periodically optimal model is selected as the foundation for the next round. Consequently, such occasional misjudgments have only a minor impact on the overall accuracy.

V. Conclusion

This paper proposes the EBFL framework, offering a novel paradigm for efficient and secure BFL. To achieve the optimal trade-off between computational efficiency, consistency, and security, we consider the advantages of both synchronous and asynchronous executions. Based on this analysis, we propose a DAG-based blockchain structure that addresses the drawbacks of existing approaches. Specifically, the EBFL framework improves training efficiency by allowing participants to train asynchronously based on the DAG structure, while maintaining consistency through synchronous periodic consensus, thus ensuring security and accuracy. Furthermore, we design a series of operations tailored to our EBFL such as the MT selection strategy and reward distribution strategy, which enhance security and accuracy. Finally, extensive experiments demonstrate that our EBFL framework achieves satisfactory accuracy. Detailed analyses of security, convergence, efficiency, and parameter settings conclude that the EBFL framework offers excellent security and computational efficiency, making it highly applicable to existing machine learning algorithms.

References

[1] M. Ribeiro, K. Grolinger, and M. A. Capretz, "MLaaS: Machine learning as a service," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl.*, 2015, pp. 896–902.

[2] N. Kourtellis, K. Katevas, and D. Perino, "Flaas: Federated learning as a service," in *Proc. 1st Workshop Distrib. Mach. Learn.*, 2020, pp. 7–13.

[3] J. Zhu, C. Gao, Z. Yin, X. Li, and J. Kurths, "Propagation structure-aware graph transformer for robust and interpretable fake news detection," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2024, pp. 4652–4663.

[4] W. Li, P. Yu, Y. Cheng, J. Yan, and Z. Zhang, "Efficient and privacy-enhanced federated learning based on parameter degradation," *IEEE Trans. Serv. Comput.*, vol. 17, no. 5, pp. 2304–2319, Sep./Oct. 2024.

[5] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

[6] H. Zhang, J. Bosch, and H. H. Olsson, "Federated learning systems: Architecture alternatives," in *Proc. 27th Asia-Pacific Softw. Eng. Conf.*, 2020, pp. 385–394.

[7] W. Liang, Y. Liu, C. Yang, S. Xie, K. Li, and W. Susilo, "On identity, transaction, and smart contract privacy on permissioned and permissionless blockchain: A comprehensive survey," *ACM Comput. Surv.*, vol. 56, no. 12, pp. 1–35, 2024.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Bitcoin*, vol. 4, no. 2, 2008, Art. no. 15.

[9] J. Sun, Y. Wu, S. Wang, Y. Fu, and X. Chang, "Permissioned blockchain frame for secure federated learning," *IEEE Commun. Lett.*, vol. 26, no. 1, pp. 13–17, Jan. 2022.

[10] M. B. Singh, H. Singh, and A. Pratap, "Energy-efficient and privacy-preserving blockchain based federated learning for smart healthcare system," *IEEE Trans. Serv. Comput.*, vol. 17, no. 5, pp. 2392–2403, Sep./Oct. 2024.

[11] X. Fu, H. Wang, and P. Shi, "A survey of blockchain consensus algorithms: Mechanism, design and applications," *Sci. China Inf. Sci.*, vol. 64, pp. 1–15, 2021.

[12] A. S. Bano et al., "SoK: Consensus in the age of blockchains," in *Proc. 1st ACM Conf. Adv. Financial Technol*, 2019, pp. 183–198.

[13] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron.*, 2018, pp. 1545–1550.

[14] D. Hou, J. Zhang, K. L. Man, J. Ma, and Z. Peng, "A systematic literature review of blockchain-based federated learning: Architectures, applications and issues," in *Proc. 2nd Inf. Commun. Technol. Conf.*, 2021, pp. 302–307.

[15] W. Zhang et al., "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2021.

[16] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[17] L. Cui et al., "CREAT: Blockchain-assisted compression algorithm of federated learning for content caching in edge computing," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14151–14161, Aug. 2022.

[18] Z. Wang, Q. Hu, Z. Xiong, Y. Liu, and D. Niyato, "Resource optimization for blockchain-based federated learning in mobile edge computing," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15166–15178, May 2024.

[19] Y. Qu et al., "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5171–5183, Jun. 2020.

[20] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4298–4311, Apr. 2020.

[21] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3975–3986, Jul. 2021.

[22] Z. Wang, M. Ogbodo, H. Huang, C. Qiu, M. Hisada, and A. B. Abdallah, "AEBIS: AI-enabled blockchain-based electric vehicle integration system for power management in smart grid platform," *IEEE Access*, vol. 8, pp. 226409–226421, 2020.

[23] G. Hua, L. Zhu, J. Wu, C. Shen, L. Zhou, and Q. Lin, "Blockchain-based federated learning for intelligent control in heavy haul railway," *IEEE Access*, vol. 8, pp. 176830–176839, 2020.

[24] M. A. Rahman, M. S. Hossain, M. S. Islam, N. A. Alrajeh, and G. Muhammad, "Secure and provenance enhanced Internet of Health Things framework: A blockchain managed federated learning approach," *IEEE Access*, vol. 8, pp. 205071–205087, 2020.

[25] R. Kumar et al., "Blockchain-federated-learning and deep learning models for COVID-19 detection using CT imaging," *IEEE Sensors J.*, vol. 21, no. 14, pp. 16301–16314, Jul. 2021.

[26] Y. Wang, Y. Tian, X. Yin, and X. Hei, "A trusted recommendation scheme for privacy protection based on federated learning," *CCF Trans. Netw.*, vol. 3, pp. 218–228, 2020.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[28] X. Feng, W. Cheng, C. Cao, L. Wang, and V. S. Sheng, "DPFLA: Defending private federated learning against poisoning attacks," *IEEE Trans. Serv. Comput.*, vol. 17, no. 4, pp. 1480–1491, Jul./Aug. 2024.

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

[30] M. Cao, L. Zhang, and B. Cao, "Toward on-device federated learning: A direct acyclic graph-based blockchain approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2028–2042, Apr. 2023.

[31] Y. Chen, L. Yan, and D. Ai, "An robust secure blockchain-based hierarchical asynchronous federated learning scheme for Internet of Things," *IEEE Access*, vol. 12, pp. 165280–165297, 2024.

[32] Z. Peng et al., "VFchain: Enabling verifiable and auditable federated learning via blockchain systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 173–186, Jan./Feb. 2022.

[33] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," 2019, *arXiv:1905.06731*.

[34] F. Mazloomi, S. Shah Heydari, and K. El-Khatib, "A novel multi-server federated learning framework in vehicular edge computing," *Future Internet*, vol. 17, no. 7, 2025, Art. no. 315.

[35] G. Shi, L. Li, J. Wang, W. Chen, K. Ye, and C. Xu, "HySync: Hybrid federated learning with effective synchronization," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst.*, 2020, pp. 628–633.

[36] J. Sun et al., "Fedsea: A semi-asynchronous federated learning framework for extremely heterogeneous devices," in *Proc. 20th ACM Conf. Embedded Networked Sensor Syst.*, 2022, pp. 106–119.
[37] M. R. Abdmeziem, H. Akli, R. Zourane, and A. A. Nacer, "Towards a distributed nodes selection mechanism for federated learning applied to blockchain-based IoT," *Internet Things*, vol. 27, 2024, Art. no. 101276.
[38] S. Popov, "The Tangle," *White Paper*, vol. 1, no. 3, 2018, Art. no. 30.
[39] Y. Qu, S. R. Pokhrel, S. Garg, L. Gao, and Y. Xiang, "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2964–2973, Apr. 2021.
[40] M. Shayan, C. Fung, C. J. Yoon, and I. Beschastnikh, "Biscotti: A blockchain system for private and secure federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1513–1525, Jul. 2021.
[41] L. Feng, Y. Zhao, S. Guo, X. Qiu, W. Li, and P. Yu, "BAFL: A blockchain-based asynchronous federated learning framework," *IEEE Trans. Comput.*, vol. 71, no. 5, pp. 1092–1103, May 2022.
[42] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734–4746, Aug. 2020.
[43] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.
[44] M. Castro et al., "Practical byzantine fault tolerance," in *Proc. 3rd Symp. Operat. Syst. Des. Implementation*, 1999, vol. 99, no. 1999, pp. 173–186.

Ze Yin received the MS degree in computer science and technology from Southwest University, China, in 2022. He is currently working toward the PhD degree in computer science and technology with Hunan University, China. His research interests include blockchain, distributed computing, high-performance computing, and artificial intelligence.

Haotian Wang received the PhD degree in computer science from Hunan University, China, in 2023, and the BS degree from the School of Information Engineering, Nanchang University, China, in 2018. He previously completed a one-year joint PhD program with Nanyang Technological University. He is currently a postdoctoral fellow with Hunan University, China. He has authored or coauthored more than ten papers in journals and conferences, such as IEEE Transactions on Parallel and Distributed Systems. His research interests include parallel computing, tensor compilation, and artificial intelligence. He is an ACM member.

Chubo Liu received the BS and PhD degrees in computer science and technology from Hunan University, China, in 2011 and 2016, respectively. He is currently a full professor of computer science and technology with Hunan University. He has authored or coauthored more than 40 papers in journals and conferences such as *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Industrial Informatics*, IEEE Internet of Things Journal, *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, *Theoretical Computer Science*, ISCA, DAC, and NPC. His research interests include parallel and distributed computing, computer architecture, artificial intelligence, game theory, and approximation and randomized algorithms. He won the IEEE TCSC Early Career Researcher Award in 2019. He is an ACM member.

Yan Ding (Member, IEEE) received the PhD degree in computer science from Hunan University, China, in 2021. He is currently an assistant professor with Hunan University. He has authored or coauthored eight papers in journals and conferences, such as Design Automation Conference, *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Industrial Informatics*, *Journal of Parallel and Distributed Computing*, *Computers & Security*, and the 17th IEEE International Symposium on Parallel and Distributed Processing with Applications (IEEE ISPA 2019). His research interests include parallel computing, mobile edge computing, Big Data, artificial intelligence, and architecture. He was the recipient of the Outstanding Paper Award in the 17th IEEE ISPA.

Keqin Li (Fellow, IEEE) received the BS degree in computer science from Tsinghua University, in 1985, and the PhD degree in computer science from the University of Houston, in 1990. He is currently a SUNY distinguished professor with the State University of New York and national distinguished professor of Hunan University, China. He has authored or coauthored more than 1110 journal articles, book chapters, and refereed conference papers. He holds nearly 75 patents announced or authorized by the Chinese National Intellectual Property Administration. He is listed in ScholarGPS Highly Ranked Scholars (2022–2024) and among the top 0.002% of more than 30 million scholars worldwide based on composite scores for research productivity, impact, and quality in the recent five years. He is an AAAS Fellow, AAIA Fellow, and ACIS Founding Fellow. He is a member of Academia Europaea.

Kenli Li (Senior Member, IEEE) received the PhD degree in computer science from the Huazhong University of Science and Technology, China, in 2003. From 2004 to 2005, he was a visiting scholar with the University of Illinois with Urbana-Champaign. He is currently a cheung kong professor of computer science and technology with Hunan University (HNU), vice-president of HNU, dean of the College of Computer Science and Electronic Engineering, HNU, and director of the National Supercomputing Center, Changsha, China. He has authored or coauthored more than 350 research papers in international conferences/journals. He is a fellow of the CCF. He is serving or has served as an associate editor for *IEEE Transactions on Computers*, *IEEE Transactions on Industrial Informatics*, and *IEEE Transactions on Sustainable Computing*. His research interests include parallel and distributed processing, high-performance computing, and Big Data management.