Bc^{2}FL: Double-Layer Blockchain-Driven Federated Learning Framework for Agricultural IoT

Qingyang Ding, Xiaofei Yue, *Graduate Student Member, IEEE*, Qinnan Zhang, *Student Member, IEEE*, Zehui Xiong, *Senior Member, IEEE*, Jinping Chang, and Hongwei Zheng

***Abstract*—With the flourishing of the Agricultural Internet of Things (AIoT), analyzing large-volume sensor data has become a regular requirement for agricultural decision-making. Federated learning (FL), which facilitates scattered AIoT devices to train models collaboratively, has gained significant attention. However, traditional FL poses challenges in AIoT scenarios, such as wide geo-distribution, heterogeneous data distribution, and high-device risks. Existing works tend to be one-sided and remain unclear on how to tackle these issues thoroughly in AIoT. To fill the gap, we present Bc^{2}FL, a double-layer blockchain-based FL framework, which enhances both learning efficiency and security for AIoT. The double-layer blockchain, coupled with a two-stage consensus algorithm, drives the hierarchical FL process to enable efficient and reliable agricultural knowledge-sharing. In addition, Bc^{2}FL adopts an adaptive model aggregation algorithm to dynamically tune noise levels based on the model quality, further improving the learning security and model credibility. Finally, the extensive experimental results demonstrate that Bc^{2}FL not only improves the model accuracy by up to 21.17% compared with the state-of-the-art baselines, but also enhances the privacy protection within an additional error of only 2.1%.**

***Index Terms*—Agriculture IoT, blockchain, federated learning (FL), knowledge-sharing, privacy protection.**

I. Introduction

AGRICULTURE, being a fundamental pillar of economies in both developing and developed nations, has compelled governments to explore avenues for sustainable development. Combining machine learning (ML) and the Internet of Things (IoT) in pursuit of agricultural intelligence has emerged as an effective method [1], [2]. Essentially, it embeds comprehensive analytics into every aspect of decision-making in agricultural production, facilitating refined and scientific management. The Agricultural IoT (AIoT) is responsible for gathering information on crop growth, soil temperature, humidity, and other factors on the farm. By analyzing this data, ML systems provide predictions for optimal seeding times, irrigation schedules, and spraying techniques. In this way, both crop yield and quality are improved, while water and pesticide usage is reduced [3].

In general, the agricultural environment is diverse and varies by region. Each region possesses unique and valuable knowledge in specific areas, such as pest and disease management or irrigation strategies. Thus, regional ML models are not immune to mispredicting due to over-fitting. As a distributed ML paradigm without explicit data exchange, federated learning (FL) enables participants (e.g., farms) to utilize local data for training and then aggregate models from diverse sources via a centralized parameter server [4]. It provides a relatively secure solution to tackle cross-regional knowledge-sharing concerns in conventional ML-based intelligent agriculture [5], [6]. In addition, the growing computing power of edge devices also promotes the integration of FL into AIoT [7], [8]. Nonetheless, the gradient updates within FL still disclose significant information about participants, posing risks of data tampering and privacy breaches due to malicious server behavior [9], [10]. Beyond FL itself, security risks arise from adversary attacks on AIoT devices, including model attacks [11], inference attacks [12], backdoor attacks [13], and poisoning attacks [14].

Recently, blockchain, known for its reliable and decentralized nature, has emerged as a promising solution to mitigate the malicious server behavior and external attacks in FL [15]. Existing works [16], [17], [18] generally aim to reduce reliance on centralized servers via blockchain, while improving security and efficiency. Specifically, gradient update processes are encapsulated as transactions, audited and recorded by all peers in the network via a consensus process. This way creates a global, tamper-proof ledger, thus ensuring secure and reliable knowledge sharing. Other studies [19], [20], [21] also incorporate well-optimized privacy-preserving mechanisms, such as differential privacy (DP) or multiparty secure computation. Nonetheless, most of

Received 18 August 2024; revised 7 October 2024; accepted 17 October 2024. Date of publication 23 October 2024; date of current version 6 February 2025. This work was supported in part by the Research and Development Program of Beijing Municipal Education Commission under Grant KM202211417008; in part by the Beijing Social Science Foundation under Grant 23GLC037; in part by the Beijing Natural Science Foundation under Grant 9222012; in part by the Education Science Planning Project of Beijing Union University under Grant JK202212; and in part by the National Key Research and Development Program of China under Grant 2022YFC3300804. *(Corresponding authors: Xiaofei Yue; Qinnan Zhang.)*

Qingyang Ding and Jinping Chang are with the School of Management, Beijing Union University, Beijing 100101, China (e-mail: gltqingyang@buu.edu.cn; gltjinping@buu.edu.cn).

Xiaofei Yue is with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: xfyue@bit.edu.cn).

Qinnan Zhang is with the Institute of Artificial Intelligence, Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100191, China (e-mail: zhangqn@buaa.edu.cn).

Zehui Xiong is with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore 487372 (e-mail: zehui_xiong@sutd.edu.sg).

Hongwei Zheng is with the Blockchain and Privacy Computing Technology R&D Department, Beijing Academy of Blockchain and Edge Computing, Beijing 100190, China, and also with the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100191, China (e-mail: hwzheng@pku.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3485208

them cannot be used in AIoT with *wide geo-distribution*, *high-device risks*, and *heterogeneous data distribution*, due to the following basic challenges.

1) *Communication Efficiency Issues of Cross-Regional FL:* It is impractical to apply the standard FL pattern directly to large-scale IoT devices deployed across geo-distributed farms. Each participant within farms needs to transfer its local model parameters to a remote centralized server during each iteration, leading to significant communication overhead and cost. Due to the poor region-to-region network, integrating decentralized blockchain techniques would exacerbate this issue [6].
2) *Multiparty Trust Issues in Global Model Aggregation:* AIoT devices are exposed to natural environments, where they face risks of crashes or attacks. These untrusted devices may leak private information and compromise model aggregation. For instance, they may provide spurious parameters to benefit themselves, engaging in free-riding behavior [16]. Thus, farm managers often adopt defensive privacy protection policies that over-mask the local model, reducing its performance.
3) *Heterogeneous Data Distribution Issues Across Regions:* In general, natural environments and agricultural practices vary greatly from region to region, resulting in different crop needs and growing conditions. The distribution of AIoT data collected across regions is therefore distinct. This heterogeneity inherently poses challenges for each farm in accessing genuinely valuable knowledge from others [19].

In this article, we propose Bc^{2}FL, a unified FL framework built on a double-layer blockchain that facilitates efficient and secure knowledge sharing across regions for AIoT. Concretely, it enables geo-distributed farms connected by AIoT to jointly train models. To improve the efficiency and security of cross-regional parameter updates, Bc^{2}FL implements a hierarchical FL process via a well-designed double-layer blockchain. The hierarchical structure is bridged via a shared blockchain node in each farm, reducing the communication frequency across regions. To ensure transparency and consistency in parameter updates, Bc^{2}FL further adopts a two-stage consensus algorithm. Each stage operates within the corresponding blockchain layer to facilitate coordination and accounting. Besides, to improve the system's robustness, we design an adaptive global model aggregate (AGMA) algorithm, which dynamically adjusts the aggregate weight and noise levels based on the model quality, without sacrificing privacy protection. In this way, the resulting *joint global model* can effectively exploit the most valuable knowledge distilled from swarm intelligence. To sum up, the contributions of this article are summarized as follows.

1) We design Bc^{2}FL, a double-layer blockchain-driven FL framework. Coupled with the two-stage consensus algorithm, it implements an efficient hierarchical learning for AIoT, while ensuring transparency and compliance.
2) We integrate the AGMA algorithm into Bc^{2}FL, enhancing the accuracy of the noised model while reducing the risk of privacy leakage and interference from heterogeneous data distribution.
3) By extensive evaluation, we illustrate that Bc^{2}FL outperforms the baselines, achieving up to a 21.17% improvement in accuracy. Besides, it enhances privacy protection while maintaining an extra accuracy error within 2.1%.

TABLE I
COMPARISON OF Bc^{2}FL AGAINST RELATED WORK

Features	Bc^{2}FL (Ours)	PPeFL [22]	PBFL [19]	DAM-SE [16]	ChainsFL [17]	AHFL [20]	FedUC [23]
Hierarchical learning	Y	N	N	Y	Y	Y	Y
Blockchain-based	Y	N	Y	Y	Y	N	N
Privacy-enhanced	Y	Y	Y	N	N	Y	N
Aggregation-optimized	Y	N	Y	Y	N	N	Y
Stage-wise consensus	Y	N	N	N	Y	N	N

Notes: "Y" indicates that corresponding features are supported; otherwise, "N" is used.

The remainder of this article is organized as follows. A brief review of related works is given in Section II. Section III describes the basic knowledge of AIoT and FL. We introduce the high-level designs of Bc^{2}FL in Section IV. The AGMA and two-stage consensus algorithms are shown in Section V and Section VI, respectively. The experimental results are presented in Section VII. Finally, this article is concluded in Section VIII.

II. RELATED WORK

Given the challenges mentioned earlier, we mainly focus on the following three subtopics of FL and present a comparison of Bc^{2}FL with the most related work in Table I.

A. Privacy-Preserving Federated Learning

Classical FL paradigm [4] proposed by Google is vulnerable to poisoning attacks, which leads to the untrustworthiness and inaccuracy of the global model. To mitigate this challenge, numerous new techniques have been proposed.

Some works [22], [24], [25] enhance and integrate DP and secure multiparty computation into FL to ensure data privacy. However, these schemes remain vulnerable to malicious gradients due to the encryption that complicates the calculation of the similarity between local gradients. To address this problem, ShieldFL [26] and PBFL [19] show a secure cosine similarity method to measure the distance between two encrypted gradients. For secure aggregation protocols, ACORN [27] enhances the computing efficiency of FL. ELSA [28], building on Prio+ [29], addresses malicious actors with an efficient and robust handling mechanism. Also, Liu et al. [30] adopted homomorphic encryption as the underlying technology and use logarithmic function-based gradient data extraction to penalize malicious clients, though this approach cannot prevent malicious server behavior. Inspired by ML technology, FELIDS [6] constructs an intrusion detection system based on the deep

learning for AIoT infrastructure. Nonetheless, the above solutions still face challenging tradeoffs between model security and accuracy, or introduce intolerable computing overhead, especially in the resource-limited IoT.

B. Hierarchical Federated Learning

Some works [31], [32] propose hierarchical FL frameworks for knowledge sharing in vehicular IoT to address distributed models and privacy concerns. However, the long-distance and high-speed movement of vehicles imposes significant demands on in-vehicle sensors, potentially resulting in increased system latency. To mitigate these problems, several efforts [16], [17], [20], [23] aim to optimize communication and aggregation efficiency. For instance, DAM-SE [16] incorporates competitive voting verification with a two-layer aggregation mechanism to decrease the communication overhead related to node security verification. Similarly, ChainsFL [17] incorporates Raft-based shard networks with a DAG-based main blockchain to enhance training efficiency and robustness. AHFL [20] strikes a balance between resource efficiency and privacy protection by offloading training to neighboring edge devices, along with a two-level DP mechanism. In this context, ShapeFL [33] further reduces communication costs and enhances learning accuracy by adjusting data distribution at the edge, while FedUC [23] improves the convergence of aggregation methods (centralized or decentralized, synchronous or asynchronous) among clusters for hierarchical FL. In practice, AIoT applications are generally not latency-sensitive [3], as crop growth conditions are relatively stable over short periods. In contrast, our objective is to implement a unified, scenario-appropriate hierarchical FL framework for AIoT applications to enhance knowledge sharing, aligning with [31] and [32]. Hence, Bc^{2}FL is designed as a cost-effective framework that maintains high accuracy and security within tolerance communication overhead. The above optimizations are orthogonal to Bc^{2}FL, but can be integrated to meet bursty demands.

C. Blockchain-Based Federated Learning

Recently, blockchain-based FL solutions have garnered considerable attention, owing to their immense potential to address security concerns in data sharing within the IoT.

Preventing the leakage of sensitive data during device fault detection is a critical challenge in industrial IoT. Wei et al. [34] utilized blockchain to empower FL participants in industrial IoT with control over shared data access. Zhang et al. [35] furthered incorporates incentive smart contracts to motivate clients to contribute their data and computing resources. In the realm of smart home IoT, Zhao et al. [21] introduced a blockchain-based privacy-preserving data-sharing framework, which integrates DP on extracted features and a novel normalization technique into FL. Similar to [36], Lu et al. [18] utilized reinforcement learning and digital twin authorization to schedule relay users, thus improving communication efficiency and enhancing privacy protection. In terms of vehicular IoT, Pokhrel and Choi [37] replaced the centralized server with blockchain and construct an analogous vehicular network model to measure the impact on system performance. In addition, Issa et al. [38] emphasized that blockchain is an excellent solution for tackling security challenges in IoT, particularly in agriculture.

Despite these benefits, the common Proof of Work (PoW) consensus algorithm within blockchain poses challenges, as it is resource-intensive and may risk privacy leakage during the verification process. Overall, further research is needed to explore how blockchain can be effectively utilized to establish a secure and trust-oriented FL framework tailored for multiheterogeneous AIoT, a crucial aspect that has been overlooked in the existing literature.

TABLE II
NOTATIONS AND THEIR DEFINITION

Notation	Definition
p_j	The participant j of federated learning process.
m_t^j	The local model of participant j after the t-th iteration.
m_t	The aggregated global model after the t-th iteration.
m_G	The convergent global model in a federated learning process.
M_G	The convergent *joint global model* of the whole system.
ϵ	The privacy budget of differential privacy.
s	The sensitivity of local differential privacy.
H_t^j	The cross-entropy of the local model of participant j after the t-th iteration.
H_t	The cross-entropy of the aggregated global model after the t-th iteration.
$\hat{m}_t^j$	The encrypted local model of participant j after the t-th iteration.
w_t^j	The model weight of participant j at the t-th iteration.
$PB_\exists$	A random private blockchain in our framework.
CB	The consortium blockchain in our framework.

III. PRELIMINARIES

In this section, we characterize the AIoT (Section III-A), and introduce the basic knowledge of FL and DP (Sections III-B and III-C). For clarity, the important notations in this article are listed in Table II.

A. Agricultural IoT

In the era of Agriculture 5.0, the deployment of IoT devices (e.g., soil sensors, weather stations, and crop monitoring tools) continues to increase, tending to form an extensive AIoT [39]. It seamlessly connects numerous scattered devices across geo-distributed farms to gather diverse data, providing a traversal infrastructure for ML-based agricultural intelligence [2].

In general, agricultural practices and settings (e.g., climate and soil types) vary from region to region, resulting in different crop demands and growth status [40]. AIoT data from individual regions often contains unique and valuable knowledge in specific areas. For instance, one region could specialize in managing specific pests and diseases, while another may have extensive expertise in optimal irrigation practices for certain crops. Thereby, regional models can easily overfit to specific agricultural settings if trained individually. This concern motivates us to engage in FL-based agricultural knowledge sharing [5], [6] *across regions*. Due to inherent disparities, however, the distribution of data from various regions is heterogeneous. This severely affects the global model convergence, making it challenging to generalize across agricultural scenarios [23].

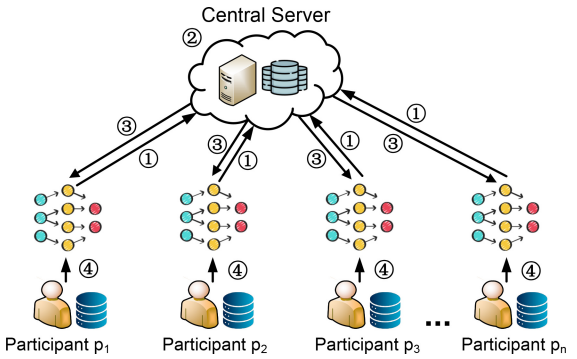


Fig. 1. Framework of FL.

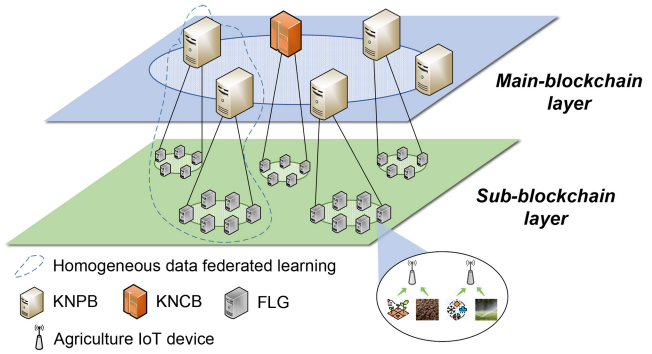


Fig. 2. System architecture of Bc^{2}FL.

On the other hand, current AIoT [41], [42], [43] protects data privacy via network authentication and device encryption, and uses blockchain to enhance traceability for efficient data management. However, due to inadequate configuration and implementation, these AIoT devices remain susceptible to external malicious attacks that compromise farm privacy. Besides, they are exposed to the natural environment for extended periods, resulting in low-quality data collection due to potential equipment damage. Significant efforts [6], [44], [45] have been made to address the above issues, demonstrating practical effectiveness. Nevertheless, there is still a lack of a unified framework to fully consider the characteristics of AIoT to achieve efficient end-to-end FL for it.

B. Federated Learning

Fig. 1 illustrates the standard FL paradigm that generally involves a central parameter server and n participant denoted as $\{p_1, p_2, \ldots, p_n\}$. The objective of the n participants is to cooperatively train a global model m_G without exchanging their local data. Specifically, participant p_j possesses a local dataset D_j, where $j = 1, 2, \ldots, n$. We use $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$ to denote the joint dataset. In the tth iteration, each participant p_j utilizes the dataset D_j and current global model m_{t-1} received from the parameter server to update the respective local model. The above iterative process aims to optimize the following function:

$$F(x, m, y) = \min_{m} \mathbb{E}_{(x,y)\sim\tilde{\mathcal{D}}} \mathcal{L}(x, m, y) \tag{1}$$

where x is the training data, y is the corresponding label, $\mathcal{L}(x, m, y)$ is the empirical loss function, and $\tilde{\mathcal{D}}$ is the data distribution. Next, the participant p_j sends its local model m_t^j to the central server. The central server then aggregates parameters $\{m_t^j\}_{j=1}^n$ received from n participants using the FedAvg algorithm [4], which is the current common model aggregation method and is shown in

$$m_t = \sum_{j=1}^{n} \zeta_t^j m_t^j \tag{2}$$

where $\zeta_t^j = (|D_j|/|D|)$ and $\sum_{j=1}^n \zeta_t^j = 1$.

Finally, the parameter server sends the global model m_t to all participants, and then each participant p_j continues to update the respective local model.

C. Differential Privacy

DP is a privacy-preserving model that addresses differential attacks and ensures that data can be used for research or analysis without causing data leakage.

Definition 1 (ϵ-DP [46]): Given a random function M, for any two datasets D and D' differing on at most one record, i.e., $|D \Delta D'| \leq 1$, and any subset $O \subseteq Range(M)$, the randomized algorithm B achieves ϵ-DP if

$$\Pr[B(D) \in O] \leq e^{\epsilon} \cdot \Pr\left[B\left(D'\right) \in O\right] \tag{3}$$

where $Pr[\cdot]$ returns the probability and ϵ is the privacy budget. The closer ϵ is to 0, the better privacy protection will be.

Definition 2 (Local DP (LDP) [47]): In the LDP setting, the centralized third party cannot be trusted. It transfers the work of data privatization to each user who handles and protects personal data by himself, which greatly reduces the possibility of privacy leakage. For two datasets D and D', as well as a randomized function $f : D \to \mathbb{R}^d$, the local sensitivity s is defined as

$$s = \Delta f = \max_{D'} \left\| f(D) - f\left(D'\right) \right\|_1 \tag{4}$$

$$\text{s.t. } |D \Delta D'| \leq 1 \tag{5}$$

where the local sensitivity s associated with D, reflects the biggest range of query function f in D'.

The *Laplace* mechanism is generally used to enhance ϵ-DP by adding independent zero-mean *Laplace* noise to the query results.

Definition 3 (Laplace Mechanism [48]): Given a function $f : \mathcal{D} \to \mathbb{R}^d$ over domain D, the mechanism M meets ϵ-DP after adding noise to the original output $f(D)$, which is expressed as

$$M(D) = f(D) + \left\langle \text{Lap}\left(\mu = 0, b = \frac{s}{\epsilon}\right) \right\rangle \tag{6}$$

where $Lap(\cdot)$ is the *Laplace* noise, and μ and b are location and scale parameters of *Laplace* distribution, respectively.

IV. Bc^{2}FL Design

A. Overview

The overall system architecture of Bc^{2}FL is illustrated in Fig. 2, deployed across multiple geo-distributed farms connected by AIoT. Within each farm, AIoT devices (e.g., sensors)

TABLE III
ROLE SETTINGS OF NODES IN THE DOUBLE-LAYER BLOCKCHAIN

Role	Location	Description
FLG ↓	Sub-layer	A general node in the private blockchain to perform federated learning.
KNPB ↓	Sub-layer & Main-layer	Unique within each private blockchain but shared across the consortium blockchain.
KNCB	Main-layer	Unique within the consortium blockchain to perform control functions.

Notes: "↓" indicates the node election process.

are sparse and scattered, tasked with collecting raw agricultural data (e.g., soil and weather information) for horizontal FL. The collected data, generally assumed to follow the same distribution [23], is stored in the nearest edge server. These edge servers communicate peer-to-peer through a relatively stable dedicated network (i.e., inside a farm), which is notably faster than farm-to-farm networks. To improve the security and efficiency of FL across the entire AIoT, Bc^{2}FL is developed to be decentralized by replacing the traditional centralized parameter server through a double-layer blockchain system (Section IV-B). Specifically, the layerwise blockchain nodes collaborate as the participants to carry out a hierarchical FL process and (joint) global model aggregation (Section IV-C).

B. Double-Layer Blockchain

The double-layer blockchain in Bc^{2}FL consists of a *main-blockchain layer* that deploys a consortium blockchain and a *sub-blockchain layer* deploying multiple private blockchains. These two layers are bridged through shared nodes. To ease the understanding, Table III shows the role settings of various nodes in the double-layer blockchain.

1) Sub-Blockchain Layer: The sub-blockchain layer consists of multiple private blockchains, each of which is deployed on a farm within the AIoT and responsible for implementing *local FL* on that farm.

In a private blockchain, each node (i.e., edge server), called a FL gateway (FLG), acts as a worker node, implementing the training for *local FL*. The dataset originates from a collection of nearby AIoT devices. Each FLG then transmits its learning results (i.e., local model) to other FLGs in the same private blockchain to aggregate a global model. Note that, a key FLG is required to be selected as the key node of the private blockchain (KNPB). We use a node election method based on verifiable random functions [49] to reduce the probability of malicious nodes being elected, while averting the frequent election. The responsibilities of the KNPB include collecting and packaging model training results (i.e., transactions) into candidate blocks, as well as publishing the new blocks to the private blockchain. Moreover, a classifier is embedded in the KNPB to classify the heterogeneous sensor data, enabling the realization of multiple disjoint FL processes upon homogeneous data.[1]

2) Main-Blockchain Layer: In the main-blockchain layer, the KNPBs from various farms are regarded as shared nodes to form a consortium blockchain. Each farm licensed by the consortium blockchain can build its private blockchain.

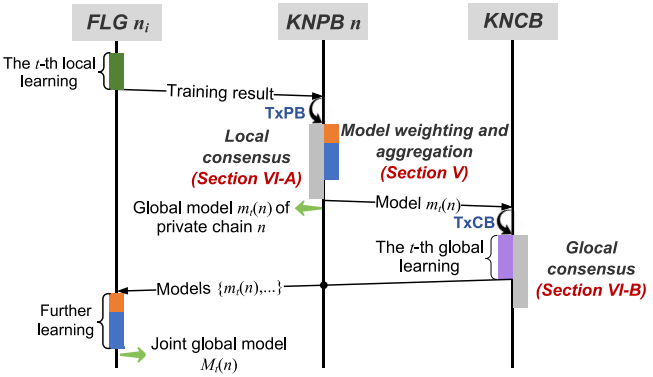


Fig. 3. Hierarchical FL process of Bc^{2}FL.

In the consortium blockchain, each node (i.e., KNPB) acts as a worker to implement the training of *global FL* over the entire AIoT. It transmits its training results of homogeneous data FL to other KNPBs to aggregate the *joint global model*,[2] and update the global model of the respective private blockchain. Similarly, a key node of the consortium blockchain (KNCB) is selected from all the KNPBs to collect and package the transactions into candidate blocks, as well as publish the new blocks to the consortium blockchain. Note that KNPBs, as blockchain nodes with special management functions, belong to both private and consortium blockchains. They bridge the double-layer blockchain through these dual identities. Moreover, delegating transaction packaging rights to the key node in both private and consortium blockchains helps mitigate potential system delays stemming from complex assignment mechanisms.

C. Hierarchical Learning Process

Coupled with the double-layer blockchain, the model training process in Bc^{2}FL is designed to be hierarchical. As shown in Fig. 3, the blockchain nodes collaborate to achieve credible and efficient FL. Specifically, it consists of the *local FL* in the sublayer and the *global FL* in the main-layer, which involves the following steps:

Step 1 (Preliminary): In the *sub-blockchain layer*, the classifier divides raw data according to its structure (e.g., image) and assigns them to the corresponding FL group. To ensure that homogeneous data are grouped together, KNPBs share respective grouping results in the *main-blockchain layer*. However, a compromised classifier may misclassify different types of data into the same group. In such cases, the system must terminate the ongoing learning process and initiate a view rotation to reposition the classifier.

Step 2 (Local FL): Within each private blockchain, FLGs collect local sensor data and execute specific FL algorithms based on the grouping results until the model converges. Following local training, each FLG shares its training result as a transaction of the private blockchain (TxPB) with others

[1]The data collected extensively by IoT devices is partitioned on demand, ensuring that each group contains homogeneous data, which is then processed by a dedicated FL model.

[2]The model further aggregated from the global models provided by various private blockchains is called the *joint global model*.

in the same private blockchain. Subsequently, each FLG updates its local model using the weighted TxPB (the weighted mechanism will be introduced in Section V).

Step 3 (Publishing Blocks of Private Blockchain): In each private blockchain, the KNPB continually collects TxPBs from the corresponding private network within a predefined time interval, and packages the verified TxPBs with *local consensus* (the consensus algorithm is shown in Section VI) into a new block. After that, KNPB publishes the new block to the private blockchain for knowledge sharing on the individual farm.

Step 4 (Global FL): Similarly, each KNPB shares its model training result in the form of a transaction of consortium blockchain (TxCB) with other KNPBs in the consortium blockchain. Unlike *local FL*, each KNPB shares the received TxCBs again with all the FLGs in the corresponding private blockchain. Meanwhile, these FLGs update their local training results with the weighted TxCB. When the training results converge, the *joint global model* of the whole system can be obtained.

Step 5 (Publishing Blocks of Consortium Blockchain): The KNCB continuously collects the TxCBs from the consortium blockchain network at a predefined time interval, and packages the verified TxCBs with *global consensus* into a new block. Then, the KNCB publishes the new block into the consortium blockchain for knowledge sharing between different farms.

V. Adaptive Model Aggregation Algorithm

In this section, we propose the AGMA algorithm, which dynamically adjusts the aggregate weight and noise levels based on the model quality, without sacrificing privacy protection.

A. Algorithm Overview

Low-quality or falsified local models can severely distort the global model [15], [23]. This risk is particularly pronounced in AIoT, stemming from: 1) heterogeneous data distribution across farms, along with participants prioritizing their own interests and 2) over-masking of local models due to defensive privacy strategies. Besides, this issue is further exacerbated by hierarchical learning, even if aggregation updates are carefully recorded. Given this, Bc²FL adopts an adaptive global model aggregation (AGMA) algorithm with low-overhead and no additional communication. The process is illustrated in Fig. 4, which tackles the above problems by integrating a *dynamic weighting mechanism* and *LDP based on the adaptive Laplace mechanism*, respectively.

Intuitively, the tth AGMA iteration for any participant, as shown in Algorithm 1, adjusts the model aggregation weight based on our incentive strategy. This algorithm increases the contribution of high-quality model parameters and penalizes low-quality ones. Owing to the heterogeneous data distribution across geographically distributed farms and the risk of poisoning attacks [14], each participant is unable to directly use unvalidated model parameters for aggregating the global model [50], or even the *joint global model*. First, all received local models are verified (lines 1–4). The dynamic aggregate weights (see Section V-B) then are figured out according

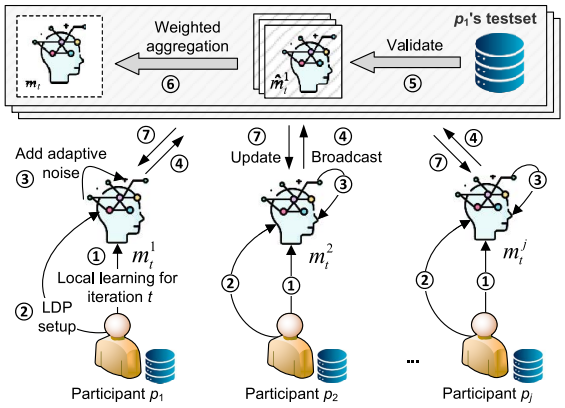


Fig. 4. Overview of the adaptive model aggregation in Bc²FL. Step ①: Local training, steps ② and ③: Noise addition via LDP based on the adaptive *Laplace* mechanism, step ④: Local parameter broadcast, steps ⑤ and ⑥: Model aggregate with dynamic weighting, and step ⑦: Global model update.

Algorithm 1: AGMA Algorithm

Input: The t-th model aggregation request, *Req*
Output: Global model, m_t or m_G

1 **for** participant $p_i \in P$ **do**
2 | Get the noised local model $\hat{m}^i_{t-1}$ of p_i;
3 | Verify the model training result H^i_t using (10);
4 **end**
5 **for** *participant* $p_i \in P$ **do**
6 | Calculate the model weight w^i_t using (11) and (12);
7 **end**
8 Aggregate the global model m_t using (7);
9 **if** $H_t > H_T$ ***and*** $t \leq T$ **then**
10 | **return** m_t;
11 **else**
12 | $m_G \leftarrow m_t$;
13 | Terminating the aggregation iteration;
14 | **return** m_G;
15 **end**

to the current and historical model quality (lines 5–7). The tth AGMA aggregation is structurally similar to the classical FedAvg, given by

$$m_t = \frac{1}{n}\sum_{j=1}^{n} w^j_t \cdot \hat{m}^j_t \tag{7}$$

where $\hat{m}^j_t$ denotes the local model of participant j with the adaptive noise (see Section V-C) and w^j_t denotes its dynamic aggregate weight, note that $\sum_j \omega^j_t = 1$. Finally, the global model m_G converges when the cross-entropy H_t drops below the threshold H_T, or when the maximum iteration limit T is reached (lines 9–15). This process can be expressed as

$$m_G \leftarrow m_t \tag{8}$$

$$\text{s.t. } H_t = H(m_t) \leq H_T \| t = T. \tag{9}$$

B. Dynamic Weighting Mechanism

At the beginning of the tth model aggregation, each participant validates the local model $\hat{m}^j_t$ received from participant

j using its own test dataset (X, Y). Note that we use cross-entropy as the validation criterion, defined as

$$H_t^j = H_t^j(f_t(X), Y) = -\sum_{x_i \in X} y_i \log f_t(x_i) \tag{10}$$

where H_t^j represents the cross-entropy of the model $\hat{m}_t^j$, y_i is the label of the test data x_i, and $f_t(x_i)$ is the output of $\hat{m}_t^j$. The smaller the value of cross-entropy, the closer the probability distribution predicted by the model is to the real result, i.e., the higher the model quality. Based on the obtained cross-entropy set $\{H_t^k\}_{k=1}^n$, we define the model quality q_t^j of m_t^j as

$$q_t^j = 1 - \frac{H_t^j(f_t(X), Y)}{\sum_{k=1}^n H_t^k(f_t(X), Y)} \tag{11}$$

where n denotes the number of participants. To avoid unnecessary overhead, we do not evaluate the model quality for each participant at each aggregation cycle. This process is carried out regularly. If the model quality of a certain participant decreases abnormally during a weight calculation round, the participant is flagged as a "*suspect*." It is then restricted until its model quality returns to normal. Based on the model quality verification history recorded as transactions, the cumulative reputation score s_t^j of participant j is derived as follows:

$$s_t^j = \frac{1}{\tau} \sum_{k=1}^{t-1} \frac{1}{1 + e^{-\log\left(q_k^j\right)}} \tag{12}$$

where τ is the number of model quality verification cycles and $q_k^j = e^{-10}$ if no record. To incentivize participants to actively verify and contribute the high-quality models, we integrate the above two metrics to adjust the aggregate weight ω_t^j of model m_t^j, i.e., $\{\omega_t^j\}_{j=1}^n = \text{Normalized}(\{q_t^j \cdot s_t^j\}_{j=1}^n)$.

C. LDP Based on the Adaptive Laplace Mechanism

DP has been extensively employed in FL to protect the data privacy of participants [51]. Among them, LDP grants users increased control and aligns well with the decentralized design of Bc^{2}FL (i.e., the centralized third party cannot be trusted). Specifically, LDP permits each participant to introduce *Laplace* noise into their local model parameters. However, this approach may lead to reduced accuracy in model transfer [21], illustrating a tradeoff between privacy protection and model performance.

Recall that AIoT devices, exposed to natural environments, are susceptible to different attacks. To bolster their security, farm managers tend to apply the defensive LDP setting (i.e., a generous privacy budget). However, this approach can lead to excessive masking of the local model due to over-provisioned noise. To mitigate the impact of LDP on model accuracy while safeguarding participant privacy, we tailor the LDP based on an adaptive *Laplace* mechanism. Let ϵ_b is the *baseline* privacy budget set by the manager, the local model $\hat{m}_t^j (t > 1)$ with adaptive *Laplace* noise is expressed as

$$\hat{m}_t^j = m_t^j + \frac{1}{1 + e^{-H_{t-1}^j}} \cdot \text{Lap}\left(\frac{s}{\epsilon_b}\right) \tag{13}$$

where s is the local sensitivity and $Lap(\cdot)$ is the raw *Laplace* noise. Consistent with [21] and [24], we find that participants with high-quality local models are less susceptible to attack and are more affected by excessive noise. The opposite holds for low-quality models. As such, we use $(1/[1 + e^{-H_{t-1}^j}]) \in [0.5, 1]$ as the adaptive noise regulation coefficient, which means that the higher the model quality, the less *Laplace* noise will be added (i.e., $\epsilon \geq \epsilon_b$, where ϵ is the actual privacy budget). Note that H_{t-1}^j represents the last cross-entropy calculation result (CECR) in the record. The coefficient also prevents the under-provisioning of noise due to the lower limit of 0.5.

Owing to the composability property of LDP [47], it is recognized that the adaptive *Laplace* mechanism consumes a relatively high-overall privacy budget. However, we emphasize its effectiveness in addressing the majority of defensive privacy-preserving strategies.

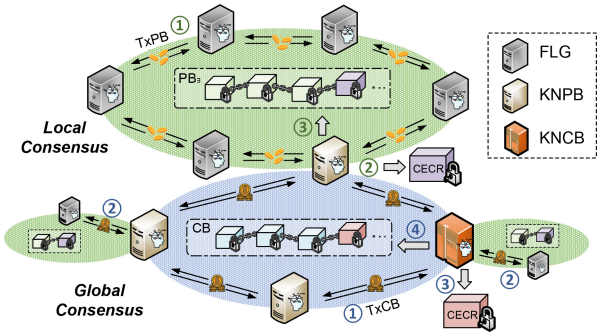


Fig. 5. Overview of the two-stage consensus algorithm, in which the local consensus processes are shown in green and the global consensus in blue.

VI. Two-Stage Consensus Algorithm

"Proof" consensus algorithms are mainly applied in public blockchains. They require elaborate incentive mechanisms or substantial computing power to solve cryptographic puzzles, potentially degrading blockchain performance. Given that both the consortium blockchain and private blockchains within Bc^{2}FL are permissioned blockchains, we thus propose a "nonproof" consensus algorithm. To accommodate the double-layer blockchain while preserving its global consistency, the consensus algorithm is designed to be two-stage, which consists of a *local consensus* stage in the sub-blockchain layer (Section VI-A) and a *global consensus* stage in the main-blockchain layer (Section VI-B). For clarity, their overview and step-by-step execution are illustrated in Fig. 5.

A. Local Consensus in the Sub-Blockchain Layer

Bc^{2}FL adopts a uniform local consensus algorithm for each private blockchain in the sublayer, as illustrated in Algorithm 2. Specifically, the detailed consensus process for a random private blockchain $PB_\exists$ is outlined as follows:

Step 1 (TxPB Transmission and Global Model Aggregation): During *local FL*, each FLG continuously collects learning results (i.e., TxPBs) sent by others, and aggregates the global model of $PB_\exists$ using the AGMA algorithm. Subsequently, each FLG calculates the (CECR, i.e., training loss) using local test data and broadcasts it within $PB_\exists$ (lines 1–3).

Algorithm 2: Local Consensus Algorithm

Input: Global model of $PB_{\exists}$, m_t
Output: New block of $PB_{\exists}$, B_n
1 **for** participant p_i *in* $PB_{\exists}$ **do**
2 | Calculate and broadcast $CECR_i$ of model m_t;
3 **end**
4 $\tau, k \leftarrow 1$;
5 Initialize the candidate block B_c;
6 Initialize the new block B_n;
7 **while** $\tau \leq T_{PB_{\exists}}$ *or* $k \leq 2N_{\exists}/3$ **do**
8 | KNPB collects *CECRs* and *TxPBs* into the candidate block B_c of $PB_{\exists}$;
9 | **for** $CECR_i$ *in CECRs* **do**
10 | | **if** $CECR_i \leq H_{PB_{\exists}}$ **then**
11 | | | $k \leftarrow k + 1$;
12 | | **end**
13 | **end**
14 | $\tau \leftarrow \tau + 1$;
15 **end**
16 $B_n \leftarrow B_c$;
17 Broadcast and append the new block B_n to $PB_{\exists}$;
18 **return** block B_n;

Algorithm 3: Global Consensus Algorithm

Input: *Joint global model* of the whole system, M_t
Output: New block of CB, B_N
1 **for** participant p_i *in* $PB_{\exists}$ **do**
2 | Calculate and broadcast $CECR_i$ of model M_t;
3 **end**
4 **for** $KNPB_i$ *in* CB **do**
5 | Collect *CECRs* in $PB_{\exists}$ where $KNPB_i$ belongs;
6 | Broadcast *CECRs* in CB;
7 **end**
8 $\tau, k \leftarrow 1$;
9 Initialize the candidate block B_C;
10 Initialize the new block B_N;
11 **while** $\tau \leq T_{CB}$ *or* $k \leq 2N_B/3$ **do**
12 | KNCB collects *CECRs* and *TxCBs* into the candidate block B_C of CB;
13 | **for** $CECR_i$ *in CECRs* **do**
14 | | **if** $CECR_i \leq H_{CB}$ **then**
15 | | | $k \leftarrow k + 1$;
16 | | **end**
17 | **end**
18 | $\tau \leftarrow \tau + 1$;
19 **end**
20 $B_N \leftarrow B_C$;
21 Broadcast and append the new block B_N to CB;
22 **return** block B_N ;

Step 2 (CECR Collection and Block Generation): The KNPB, as the key node of $PB_{\exists}$, constantly collects the CECRs within a predefined time interval and packages these TxPBs and CECRs into a candidate block (line 8). Although each participant being capable of monitoring TxPBs and CECRs, the candidate blocks are still packaged by the KNPB. This ensures that participants do not have to compete for billing rights in the permissioned blockchain, and it also reduces the computing power consumption of other nodes.

Step 3 (CECR Validation and Block Publication): The KNPB continuously tracks the CECRs. Let $N_{\exists}$ be the number of nodes in $PB_{\exists}$. The KNPB will publish the candidate block into $PB_{\exists}$ when the vast majority (i.e., more than $2N_{\exists}/3$) of CECRs are below the threshold $H_{PB_{\exists}}$ or upon reaching the timeout $T_{PB_{\exists}}$ (lines 7–15). The newly published block will be linked with the preceding block, and both parameters and CECRs will be permanently recorded on the blockchain.

B. Global Consensus in the Main-Blockchain Layer

The global consensus in the *main-blockchain layer* contains both the consensus process done by the KNPBs in the consortium blockchain and the FLGs in each private blockchain during the *joint global model* aggregation phase, as shown in Algorithm 3. The detailed consensus process of the consortium blockchain *CB* is described as follows.

Step 1 (TxCB Cross-Layer Transmission): During *global FL*, each KNPB broadcasts the parameters of the global model (i.e., TxCB), which has been consented to the $PB_{\exists}$ to which it belongs. Afterward, each KNPB continues to broadcast the received TxCBs within its $PB_{\exists}$, facilitating the sharing of training results across different private blockchains.

Step 2 (Further Learning and Joint Global Model Aggregation): The FLGs that receive TxCBs in $PB_{\exists}$ implement further learning by aggregating the *joint global model* with the AGMA algorithm. Especially, the model weights reflect the training quality of the global model in $PB_{\exists}$. After that, each FLG will calculate and broadcast the CECR of the *joint global model* to others in $PB_{\exists}$ (lines 1–3).

Step 3 (CECR Collection and Block Generation): The KNPB continuously collects the CECRs sent by the FLGs in $PB_{\exists}$ within a predefined time interval and broadcasts them in *CB* (lines 4–7). Then, the KNCB packages the CECRs and TxCBs into a candidate block (line 12).

Step 4 (CECR Validation and Block Publication): The KNCB continuously tracks the CECRs. Let N_B be the number of nodes in the double-layer blockchain. The KNCB will publish the candidate block to *CB* when the vast majority (i.e., more than $2N_B/3$) of CECRs are below the threshold H_{CB} or upon reaching the timeout T_{CB} (lines 11–19). The newly published block will be linked with the preceding block, and both parameters and CECRs will be permanently recorded on the blockchain.

VII. Experimental Evaluation

In this section, we first present the experimental setup and datasets (Section VII-A) and evaluate the overall performance of Bc^{2}FL (Section VII-B). Next, we analyze the effectiveness of its components (Section VII-C). Finally, we demonstrate the system overhead of Bc^{2}FL (Section VII-D).

A. Experimental Setup

Experimental Setting: We implement the Bc^{2}FL framework based on Python v3.6.10 and PyTorch v0.4.1. The double-layer

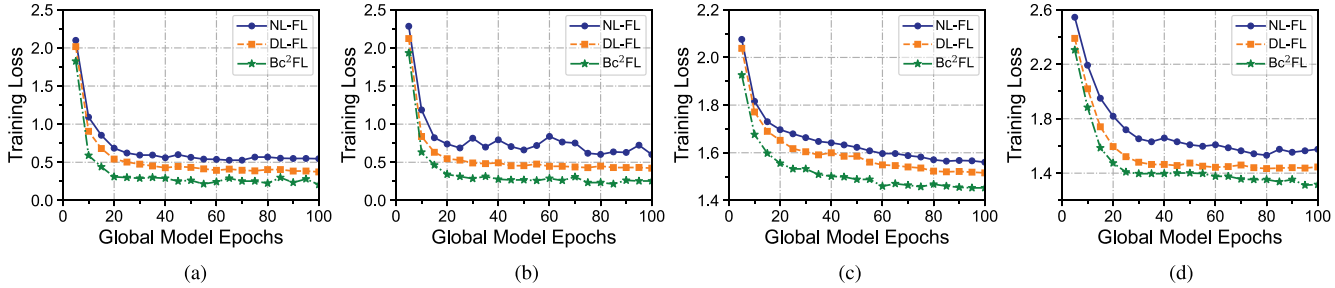


Fig. 6. Training loss of various methods. (a) MLP and MNIST dataset. (b) CNN and MNIST dataset. (c) MLP and CIFAR-10 dataset. (d) CNN and CIFAR-10 dataset.

TABLE IV
SIMULATION PARAMETERS

Parameter	Value
Learning rate	0.01
SGD momentum	0.5
Laplace location	0
Batch size	64
Local model epochs	5
Activation function	ReLu

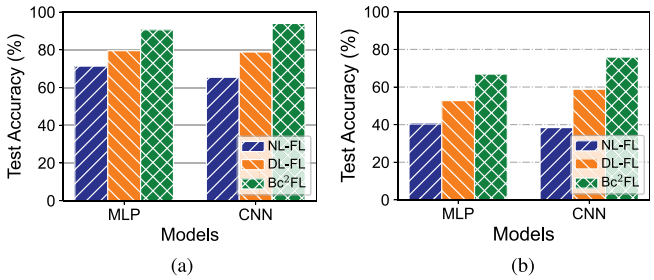


Fig. 7. Learning accuracy of various methods. (a) MNIST dataset. (b) CIFAR-10 dataset.

blockchain system is established using `Python Flask` to provide fundamental services (e.g., transaction creation and block publishing). Besides, our proposed AGMA algorithm is deployed in the form of smart contracts.

Dataset: We select two widely used real datasets, MNIST [52] and CIFAR-10 [53], for model training and evaluation. These datasets are evenly divided according to the number of worker nodes in the *sub-blockchain layer* to simulate the horizontal federation learning.

1) The MNIST dataset [52] is a grayscale image dataset of hand-written digits, comparing 60 000 training samples and 10 000 test samples.
2) The CIFAR-10 dataset [53] is a 10-class color image dataset, with 50 000 training and 10 000 test samples.

Classifiers: We use a multilayer perceptron (MLP) and a convolutional neural network (CNN) as the basic neural classifiers to be learned. They can classify images in both of the datasets above. More specifically, we utilize the stochastic gradient descent (SGD) algorithm to iteratively update the local gradient. The MLP model consists of two fully connected hidden layers with a total of 64 neurons, and the CNN model has a convolutional kernel size of 5×5 and a step size of 1. The other parameter settings are shown in Table IV.

B. Overall Performance

We discuss the overall performance improvement of Bc^{2}FL against two baselines: 1) a nonlayered traditional FL framework (NL-FL) that employs the AGMA algorithm for global model aggregation and 2) the start-of-the-art double-layered FL framework (DL-FL) [23] with average update. In our experiments, we generate *Laplace* noise using a random noise-added function (`random.Laplace`), with a privacy budget ϵ set to 0.3. Each farm is equipped with 30 edge servers.

We evaluate the overall performance of Bc^{2}FL in terms of training loss and learning accuracy. Fig. 6 shows that Bc^{2}FL achieves the lowest loss compared to the baselines for both the CNN and MLP models. Although the AGMA algorithm is applied in NL-FL, its loss remains higher than that of DL-FL, demonstrating that the double-layered learning algorithm outperforms nonlayered FL. Fig. 7 illustrates the best learning outcomes among the frameworks during training. Compared to the second-best DL-FL, Bc^{2}FL improves test accuracy by an average of 12.41% and 21.17% for the MLP and CNN models, respectively. This indicates that the AGMA algorithm effectively improves training quality via the dynamic tuning of local model weights. The models perform better on MNIST compared to CIFAR-10. The reason is that MNIST consists of grayscale images with only one channel, whereas CIFAR-10 has RGB images with three channels. The additional channels imply more complex feature extraction during training, which accordingly results in lower learning accuracy.

In a nutshell, our hierarchical learning framework leverages the *main-blockchain layer* to aggregate underlying knowledge from the sublayer, thereby enhancing the relevance of all data. This process exemplifies the concept of swarm intelligence, resulting in improved final learning accuracy.

C. Deep Dive of Bc^{2}FL

We conduct a deep dive into Bc^{2}FL to evaluate the impact of the node number and privacy budget on the performance of Bc^{2}FL, and its overall security.

The Impact of Privacy Budget on Model Performance: We examine the performance of the MLP model under different

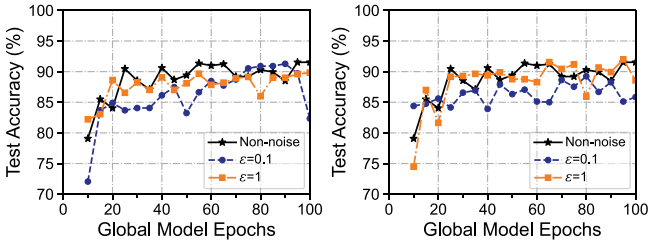


Fig. 8. Impact of privacy budget ϵ on model learning performance. (a) Fixed noise. (b) Adaptive noise.

TABLE V
MODEL ACCURACY ERROR WITH VARIOUS NOISE-ADDING SCHEMES

Privacy budget ϵ	Fixed noise (%)	Adaptive noise (%)
0.1	2.713	2.074
0.3	2.284	1.861
1	1.058	0.956
5	0.468	0.374

privacy budgets, with ϵ set to 0.1 and 1. Recall that the smaller the value of ϵ, the larger the added *Laplace* noise in LDP.

Fig. 8 illustrates a comparison of model accuracy between the traditional fixed noise addition and the proposed adaptive noise scheme in the AGMA algorithm. As the privacy budget increases, both schemes exhibit gradual improvements in model accuracy and convergence speed. Nonetheless, the results in Table V highlight that the adaptive noise scheme, which adjusts according to model quality, significantly reduces accuracy errors compared to the fixed approach. With privacy budgets set at 0.1, 0.3, 1 and 5, the average model accuracy error decreases by 0.639%, 0.423%, 0.102%, and 0.094%, respectively. In summary, Bc^{2}FL can achieve a tradeoff between privacy protection and dynamic reconfigurability of learning performance using the AGMA algorithm, especially when the privacy budget is small.

The Impact of Node Number on Model Performance: We compare the training loss and test accuracy of the MLP and CNN models in Bc^{2}FL across varying numbers of participating nodes. Specifically, we maintain a consistent configuration for the number of nodes per private blockchain in the sublayer, within the range of $[10, 30, 60, 100]$. Each private blockchain independently and randomly partitions the dataset into 100 segments, ensuring each node receives 600 MNIST samples and 500 CIFAR-10 samples.

As depicted in Fig. 9, both MLP and CNN models training on the MNIST dataset experience a decline in training loss and an improvement in training accuracy, as the number of joint model aggregations increases. Concretely, a higher number of nodes correlates with a more significant reduction in loss, and the variability in loss gradually diminishes. When the number of nodes exceeds 30, the training loss consistently converges to below 0.5, and the accuracy stabilizes above 85%. As shown in Fig. 10, the learning performance of MLP and CNN models on CIFAR-10 mirrors that of MNIST. Consistent with earlier findings, the model loss stabilizes below 1.6, and the accuracy of the model trained with 100 nodes is 42.55% lower than that on MNIST.

Security Evaluation: To assess the security of Bc^{2}FL, we compare it with vanilla federated learning (Vanilla-FL) [4] and a blockchain-based (Vanilla-BL) framework [15]. Specifically, Vanilla-FL assumes that all participants are trustworthy, and Vanilla-BL employs a nonlayered consortium blockchain that implements a fundamental consensus process. Following [54], we simulate poisoning attacks by altering the labels of local datasets at participating nodes, where the attack intensity $\rho_1 \in [0, 1]$ denotes the proportion of modified data labels. The attack intensity of each malicious node is randomly distributed within the interval $[0, 0.5]$ to simulate real-world conditions.

We assess the impact of poisoning attacks on model accuracy across different proportions of malicious nodes, denoted as ρ_2. We set ρ_2 uniformly for each farm. As shown in Fig. 11, Vanilla-FL assumes all data is trustworthy, resulting in a significant increase in test error as the number of malicious nodes increases. When ρ_2 is less than 0.3, Vanilla-BL shows greater robustness than others by using consensus to audit the data. However, when a majority of nodes are maliciously attacked (i.e., a 51% attack [55]), its test error rises significantly. In contrast, Bc^{2}FL employs the AGMA algorithm to dynamically adjust aggregate weights for low-quality models from malicious nodes in time. Thus, it demonstrates enhanced resilience against poisoning attacks. Furthermore, the layered framework of Bc^{2}FL enables the creation of multiple ledgers, each managed by the private blockchain responsible for block publication, further strengthening its security.

D. System Overhead

We utilize the stress testing tool *Siege v4.0.4* and the request access tool *Postman* to measure the external performance impact of the double-layer blockchain during model aggregation.

Fig. 12 illustrates the average computing delay and storage overhead required by each node. As depicted in Fig. 12(a), as the aggregate transaction volume increases, the communication overhead in the system increases significantly, and each node requires more space to record remote model parameters, so the storage overhead basically shows linear growth. Similarly, Fig. 12(b) shows that the storage overhead increases linearly as the block length increases. Besides, affected by the randomness of mining difficulty, the average computing delay exhibits a fluctuating growth trend. As a result, Bc^{2}FL employs the double-layer blockchain to facilitate model aggregation, enabling performance comparable to that of independent FL without significantly impacting model training.

VIII. CONCLUSION

In this article, we introduce Bc^{2}FL, a unified double-layer blockchain-based FL framework for AIoT. The hierarchical architecture and learning process of Bc^{2}FL are well-suited to large-scale, scattered, and privacy-preserving AIoT scenarios, facilitating the integration of knowledge from farms across various regions to enhance model generalization. Bc^{2}FL employs

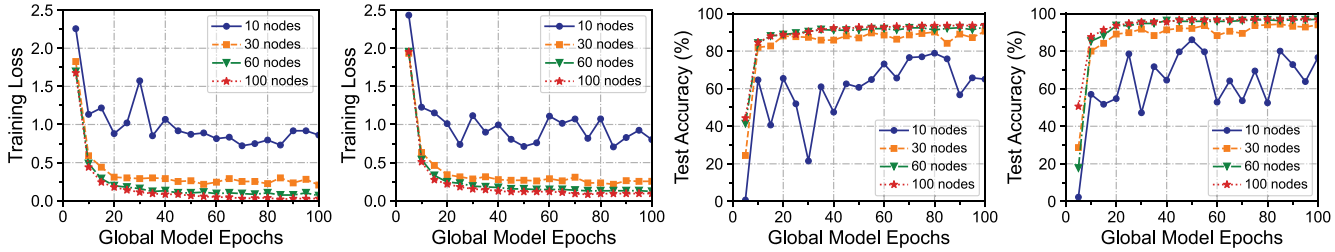


Fig. 9. Impact of node number on model learning performance under MNIST dataset. (a) Training loss of MLP. (b) Training loss of CNN. (c) Test accuracy of MLP. (d) Test accuracy of CNN.

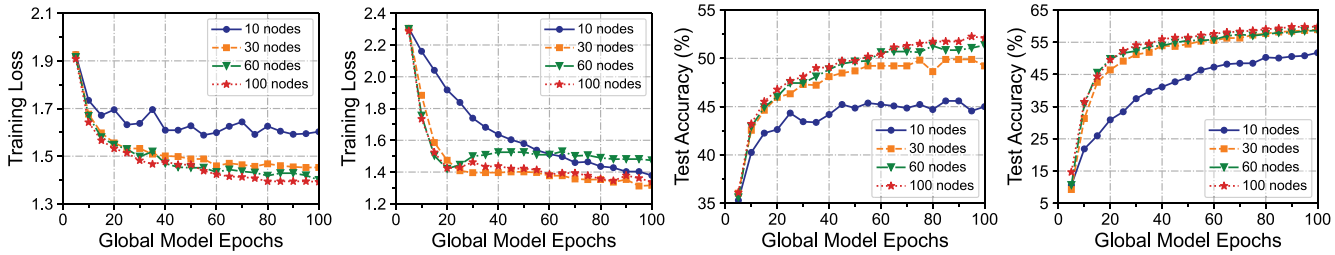


Fig. 10. Impact of node number on model learning performance under CIFAR-10 dataset. (a) Training loss of MLP. (b) Training loss of CNN. (c) Test accuracy of MLP. (d) Test accuracy of CNN.

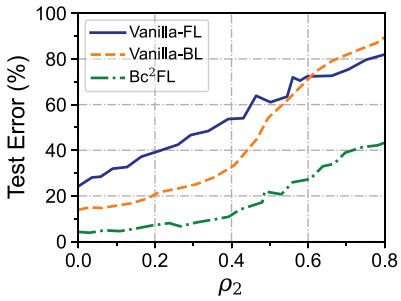


Fig. 11. Impact of malicious node share ρ_2 on test error.

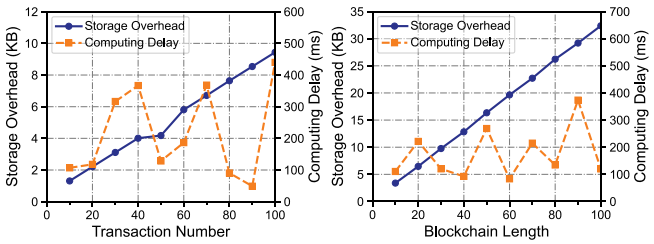


Fig. 12. Overhead of double-blockchain system on Bc^{2}FL. (a) Impact of transaction number. (b) Impact of blockchain length.

a two-stage consensus algorithm to secure and streamline the hierarchical FL process, thus ensuring transparency and regulatory compliance. Moreover, the AGMA algorithm dynamically adjusts noise levels based on model quality, providing the adaptive privacy protection for intermediate model parameters. The experimental results validate the effectiveness and security of Bc^{2}FL, showcasing that it outperforms the state-of-the-art baseline solutions in model accuracy.

In summary, Bc^{2}FL provides an effective and secure FL solution with considerable potential for deployment in AIoT settings. Bc^{2}FL utilizes swarm intelligence to extract the most valuable knowledge efficiently. Our future research will aim to refine the algorithm and architecture (e.g., enhancing communication efficiency) to better manage more complex application scenarios and achieve higher performance requirements.

REFERENCES

[1] I. Attri, L. K. Awasthi, T. P. Sharma, and P. Rathee, "A review of deep learning techniques used in agriculture," *Ecol. Informat.*, vol. 77, Nov. 2023, Art. no. 102217.

[2] R. J. Patil, I. Mulage, and N. Patil, "Smart agriculture using IoT and machine learning," *J. Sci. Res. Technol.*, vol. 1, no. 3, pp. 47–59, 2023.

[3] A. Vij, S. Vijendra, A. Jain, S. Bajaj, A. Bassi, and A. Sharma, "IoT and machine learning approaches for automation of farm irrigation system," *Procedia Comput. Sci.*, vol. 167, pp. 1250–1257, 2020.

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

[5] H. K. Kondaveeti, G. B. Sai, S. A. Athar, V. K. Vatsavayi, A. Mitra, and P. Ananthachari, "Federated learning for smart agriculture: Challenges and opportunities," in *Proc. IEEE Int. Conf. Distrib. Comput. Electr. Circuits Electron.*, 2024, pp. 1–7.

[6] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K.-K. R. Choo, and M. Nafaa, "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things," *J. Parallel Distrib. Comput.*, vol. 165, pp. 17–31, Jul. 2022.

[7] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020.

[8] A. Mitra, S. P. Mohanty, and E. Kougianos, "ToEFL: A novel approach for training on edge in smart agriculture," in *Proc. ACM Great Lakes Symp.*, 2024, pp. 657–662.

[9] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Security Privacy*, 2019, pp. 691–706.

[10] J. Zhang, L. Zhao, K. Yu, G. Min, A. Y. Al-Dubai, and A. Y. Zomaya, "A novel federated learning scheme for generative adversarial networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 3633–3649, May 2023.

[11] N.-B. Nguyen, K. Chandrasegaran, M. Abdollahzadeh, and N.-M. Cheung, "Re-thinking model inversion attacks against deep neural networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16384–16393.

[12] M. Bertran, S. Tang, A. Roth, M. Kearns, J. H. Morgenstern, and S. Z. Wu, "Scalable membership inference attacks via quantile regression," *Proc. 37th Int. Conf. Adv. Neural Inf. Proces. Syst.*, vol. 36, 2024, pp. 314–330.

[13] T. D. Nguyen, T. A. Nguyen, A. Tran, K. D. Doan, and K.-S. Wong, "IBA: Towards irreversible backdoor attacks in federated learning," *Proc. 37th Conf. Adv. Neural Inf. Proces. Syst.*, vol. 36, 2024, pp. 1–13.

[14] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2020, pp. 480–501.

[15] 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.

[16] S. Xuan, M. Jin, X. Li, Z. Yao, W. Yang, and D. Man, "DAM-SE: A blockchain-based optimized solution for the counterattacks in the Internet of Federated Learning systems," *Secur. Commun. Netw.*, vol. 2021, pp. 1–14, Jun. 2021.

[17] S. Yuan, B. Cao, M. Peng, and Y. Sun, "ChainsFL: Blockchain-driven federated learning from design to realization," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–6.

[18] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2276–2288, Feb. 2021.

[19] Y. Miao, Z. Liu, H. Li, K.-K. R. Choo, and R. H. Deng, "Privacy-preserving Byzantine-robust federated learning via blockchain systems," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2848–2861, 2022.

[20] Y. Guo, F. Liu, T. Zhou, Z. Cai, and N. Xiao, "Privacy vs. efficiency: Achieving both through adaptive hierarchical federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1331–1342, Apr. 2023.

[21] Y. Zhao et al., "Privacy-preserving blockchain-based federated learning for IoT devices," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1817–1829, Feb. 2021.

[22] B. Wang, Y. Chen, H. Jiang, and Z. Zhao, "PPeFL: Privacy-preserving edge federated learning with local differential privacy," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15488–15500, Sep. 2023.

[23] Q. Ma, Y. Xu, H. Xu, J. Liu, and L. Huang, "FedUC: A unified clustering approach for hierarchical federated learning," *IEEE Trans. Mob. Comput.*, vol. 23, no. 10, pp. 9737–9756, Oct. 2024.

[24] S. Truex et al., "A hybrid approach to privacy-preserving federated learning," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2019, pp. 1–11.

[25] L. Chen, D. Xiao, Z. Yu, and M. Zhang, "Secure and efficient federated learning via novel multi-party computation and compressed sensing," *Inf. Sci.*, vol. 667, May 2024, Art. no. 120481.

[26] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "ShieldFL: Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1639–1654, 2022.

[27] J. Bell et al., "ACORN: Input validation for secure aggregation," in *Proc. USENIX Secur. Symp.*, 2023, pp. 4805–4822.

[28] M. Rathee, C. Shen, S. Wagh, and R. A. Popa, "ELSA: Secure aggregation for federated learning with malicious actors," in *Proc. IEEE Symp. Secur. Privacy*, 2023, pp. 1961–1979.

[29] S. Addanki, K. Garbe, E. Jaffe, R. Ostrovsky, and A. Polychroniadou, "Prio+: Privacy preserving aggregate statistics via boolean shares," in *Proc. Int. Conf. Secur. Cryptogr. Netw.*, 2022, pp. 516–539.

[30] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4574–4588, 2021.

[31] 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.

[32] H. Zhou, Y. Zheng, H. Huang, J. Shu, and X. Jia, "Toward robust hierarchical federated learning in Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5600–5614, May 2023.

[33] Y. Deng et al., "A communication-efficient hierarchical federated learning framework via shaping data distribution at edge," *IEEE/ACM Trans. Netw.*, vol. 32, no. 3, pp. 2600–2615, Jun. 2024.

[34] J. Wei et al., "A redactable blockchain framework for secure federated learning in industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17901–17911, Sep. 2022.

[35] 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.

[36] Q. He et al., "A blockchain-based scheme for secure data offloading in healthcare with deep reinforcement learning," *IEEE/ACM Trans. Intell. Syst. Netw.*, vol. 32, no. 1, pp. 65–80, Feb. 2024.

[37] 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.

[38] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-based federated learning for securing Internet of Things: A comprehensive survey," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–43, 2023.

[39] S. Bera, T. Dey, A. Mukherjee, and D. De, "FLAG: Federated learning for sustainable irrigation in agriculture 5.0," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2303–2310, Feb. 2024.

[40] J. Nie, J. Jiang, Y. Li, J. Li, Y. Qiao, and S. Ercisli, "UAVEC-FLchain: Distributed multi-regional jujube orchard joint yield estimation for secure agricultural-IoT applications," *IEEE Internet Things J.*, vol. 25, Apr. 2024, Art. no. 101143.

[41] A. Vangala, A. K. Das, A. Mitra, S. K. Das, and Y. Park, "Blockchain-enabled authenticated key agreement scheme for mobile vehicles-assisted precision agricultural IoT networks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 904–919, 2022.

[42] H.-T. Wu and C.-W. Tsai, "An intelligent agriculture network security system based on private blockchains," *J. Commun. Netw.*, vol. 21, no. 5, pp. 503–508, 2019.

[43] A. Vangala, A. K. Das, and J.-H. Lee, "Provably secure signature-based anonymous user authentication protocol in an Internet of Things-enabled intelligent precision agricultural environment," *Concurr. Comput. Pract. Exp.*, vol. 35, no. 16, 2023, Art. no. e6187

[44] M. A. Ferrag, L. Shu, H. Djallel, and K.-K. R. Choo, "Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0," *Electronics*, vol. 10, no. 11, p. 1257, 2021.

[45] V. Puri, V. K. Solanki, and G. J. R. Aponte, "Blockchain and federated learning based integrated approach for agricultural Internet of Things," in *Proc. Int. Conf. Intell. Syst. Netw.*, 2023, pp. 240–246.

[46] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Languag., Program.*, 2006, pp. 1–12.

[47] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, "Generating synthetic decentralized social graphs with local differential privacy," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2017, pp. 425–438.

[48] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.*, 2006, pp. 265–284.

[49] N. Jiang, F. Bai, L. Huang, Z. An, and T. Shen, "Reputation-driven dynamic node consensus and reliability sharding model in IoT blockchain," *Algorithms*, vol. 15, no. 2, p. 28, 2022.

[50] S. Shen, T. Zhu, D. Wu, W. Wang, and W. Zhou, "From distributed machine learning to federated learning: In the view of data privacy and security," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 16, 2022, Art. no. e6002.

[51] 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. Informat.*, vol. 16, no. 6, pp. 4177–4186, Jun. 2020.

[52] Y. LeCun. "The MNIST database of handwritten digits." 1998. [Online]. Available: http://yann. lecun. com/exdb/mnist/

[53] A. Krizhevsky,"Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, USA, Rep. TR-2009, 2009.

[54] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[55] N. Schweitzer, A. Stulman, R. D. Margalit, and A. Shabtai, "Contradiction based gray-hole attack minimization for ad-hoc networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2174–2183, Aug. 2017.

Qingyang Ding received the Ph.D. degree from the Central University of Finance and Economics, Beijing, China, in 2020.

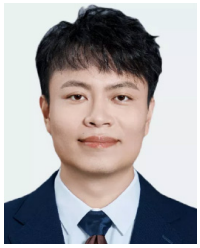
He is currently a Lecturer with the School of Management, Beijing Union University, Beijing, China. His research interests include federated learning, blockchain, and digital transformation.

Xiaofei Yue (Graduate Student Member, IEEE) received the M.E. degree from Northeastern University, Shenyang, China, in 2022. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China.

His current research interests include distributed systems, cloud/serverless computing, and data analytics.

Qinnan Zhang (Student Member, IEEE) received the Ph.D. degree from the Central University of Finance and Economics, Beijing, China, in 2023.

She is currently a Postdoctoral Fellow with the Institute of Artificial Intelligence, Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing. Her current research interests include blockchain, federated learning, incentive mechanism, game theory, and edge intelligence.

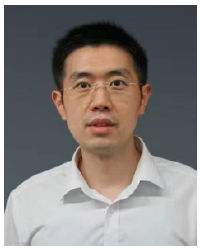


Zehui Xiong (Senior Member, IEEE) received the B.Eng. degree with (Highest Hons.) in telecommunications engineering from Huazhong University of Science and Technology, Wuhan, China, in 2016, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2019.

He is an Assistant Professor with Singapore University of Technology and Design, Singapore, and also an Honorary Adjunct Senior Research Scientist with Alibaba-NTU Singapore Joint Research Institute, Singapore. He was a Visiting Scholar with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA, and a Visiting Scholar with Broadband Communications Research Lab, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His current research interests include wireless communications, Internet of Things, blockchain, edge intelligence, and AIGC.

Jinping Chang received the Ph.D. degree in management from Beijing University of Technology, Beijing, China, in 2013.

He is currently an Associate Professor with the School of Management, Beijing Union University, Beijing. His research interests include digital transformation, project management, and big data analysis and application.



Hongwei Zheng received the Doctoral degree in quantitative economics from The Ohio State University, Columbus, OH, USA, in 2010.

He is a Senior Researcher with Beijing Academy of Blockchain and Edge Computing, Beijing, China. His research interests include complex networks, blockchain, and artificial intelligence.