Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.072081

ARTICLE

Privacy-Preserving Personnel Detection in Substations via Federated Learning with Dynamic Noise Adaptation

Yuewei Tian[1], Yang Su[2], Yujia Wang[1], Lisa Guo[1], Xuyang Wu[3,*], Lei Cao[4] and Fang Ren[3]

[1]Guiyang Power Supply Bureau, Guizhou Power Grid Co., Ltd., Guiyang, 563000, China
[2]Information Center, Guizhou Power Grid Co., Ltd., Guiyang, 563000, China
[3]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China
[4]Guizhou Power Grid Co., Ltd., Guiyang, 563000, China

*Corresponding Author: Xuyang Wu. Email: wuxuyang1217@163.com

Received: 19 August 2025; Accepted: 07 October 2025; Published: 12 January 2026

ABSTRACT: This study addresses the risk of privacy leakage during the transmission and sharing of multimodal data in smart grid substations by proposing a three-tier privacy-preserving architecture based on asynchronous federated learning. The framework integrates blockchain technology, the InterPlanetary File System (IPFS) for distributed storage, and a dynamic differential privacy mechanism to achieve collaborative security across the storage, service, and federated coordination layers. It accommodates both multimodal data classification and object detection tasks, enabling the identification and localization of key targets and abnormal behaviors in substation scenarios while ensuring privacy protection. This effectively mitigates the single-point failures and model leakage issues inherent in centralized architectures. A dynamically adjustable differential privacy mechanism is introduced to allocate privacy budgets according to client contribution levels and upload frequencies, achieving a personalized balance between model performance and privacy protection. Multi-dimensional experimental evaluations, including classification accuracy, F1-score, encryption latency, and aggregation latency, verify the security and efficiency of the proposed architecture. The improved CNN model achieves 72.34% accuracy and an F1-score of 0.72 in object detection and classification tasks on infrared surveillance imagery, effectively identifying typical risk events such as not wearing safety helmets and unauthorized intrusion, while maintaining an aggregation latency of only 1.58 s and a query latency of 80.79 ms. Compared with traditional static differential privacy and centralized approaches, the proposed method demonstrates significant advantages in accuracy, latency, and security, providing a new technical paradigm for efficient, secure data sharing, object detection, and privacy preservation in smart grid substations.

KEYWORDS: Substation; privacy preservation; asynchronous federated learning; CNN; differential privacy

1 Introduction

As critical nodes in smart grids, substations generate multimodal operational data including equipment condition monitoring streams, power load profiles, and inspection imagery [1,2]. These data types serve as determining factors for real-time dispatch and condition-based maintenance while simultaneously encompassing the confidentiality of physical infrastructure topologies and consumer electricity usage patterns [3]. Notably, spatial characteristics within the data potentially revealing equipment lay-outs through infrared thermography containing geolocation coordinates-introduce additional privacy vulnerabilities [4,5]. Any leakage during transmission could enable targeted attacks on critical power facilities, jeopardizing the secure and stable operation of power systems [6]. Consequently, establishing a multimodal data privacy preservation framework for substations holds paramount significance for ensuring power system security and advancing next-generation grid infrastructure, representing a mission-critical technology for intelligent substation implementation.

Existing substation data privacy preservation technologies predominantly rely on data anonymization and centralized federated learning [7,8]. Traditional anonymization solutions are prone to the curse of dimensionality when handling the high-dimensional spatiotemporal correlations inherent in substation multimodal data, resulting in excessive noise injection that drastically degrades data utility while remaining vulnerable to adversarial attacks exposing sensitive information [9]. The centralized architecture of conventional federated learning aggregates global model parameters on a single server, creating systemic privacy leakage risks from model inversion attacks and parameter theft [10,11]. Furthermore, its synchronous update mechanism requires plaintext gradient transmission, amplifies privacy exposure through spatiotemporal correlation inference based on substation operational characteristics. These limitations necessitate a decentralized privacy-preserving federated framework that incorporates asynchronous collaboration mechanisms and multimodal differential perturbation to address the spatiotemporal correlation challenges inherent in substation data, thereby ensuring comprehensive privacy protection. However, existing applications of AFL in smart grid scenarios still face several critical challenges. First, most AFL methods rely on static or uniform differential privacy strategies, lacking personalized privacy-utility trade-off regulation mechanisms. Second, the asynchronous nature of AFL can easily trigger consistency and robustness issues during global aggregation without centralized coordination. Furthermore, existing research rarely addresses the system-level feasibility of real-world smart grid deployments, exhibiting insufficient attention to the design and validation of trusted collaboration and scalable storage mechanisms among substations.

Current research focuses on integrating blockchain with differential privacy techniques in the power grid domain to address privacy protection and fair exchange issues in data sharing. Fotiou et al. [12] proposed a privacy-preserving statistics marketplace based on local differential privacy and blockchain. This scheme protects data providers' privacy using local differential privacy technology and utilizes blockchain smart contracts to achieve fair exchange and immutable data log recording. It is applicable to smart grid measurement data sharing scenarios, demonstrating feasibility through experimental validation for scales involving hundreds or even thousands of data providers. Zheng et al. [13] proposed a decentralized mechanism called DDP based on differential privacy for privacy-preserving computation in smart grid, which extends data sanitization from value domain to time domain by injecting Laplace noise distributively and using random permutation to shuffle measurement sequences, thus enforcing differential privacy and preventing sensitive power usage mode inference, with experiments demonstrating its outstanding performance in privacy and utility. Fan et al. [14] proposed a decentralized privacy-preserving data aggregation (DPPDA) scheme for the smart grid based on blockchain. The scheme uses the leader election algorithm to select a smart meter in the residential area as a mining node to build a block, and the node adopts the Paillier cryptosystem algorithm to aggregate user power consumption data. Meanwhile, Boneh-Lynn-Shacham short signature and SHA-256 function are applied to ensure the confidentiality and integrity of user data. Security analysis shows that the scheme meets the security and privacy requirements of smart grid data aggregation, and experimental results demonstrate that it is more efficient than existing competing schemes in terms of computation and communication overhead.

Asynchronous federated learning, with its unique decentralized architecture and efficient asynchronous parameter exchange mechanisms, can adapt to the diverse computing capabilities of substation edge devices and fluctuations in network environments, while effectively avoiding single-point failure risks inherent in centralized architectures [15,16]. This technology opens a feasible path for ensuring multimodal data privacy security in substations. In recent years, some scholars have introduced asynchronous federated learning into power systems to achieve distributed model training under data privacy protection. Li et al. [17] innovatively proposed a decentralized asynchronous adaptive federated learning algorithm that focuses on secure prediction of distributed power data, making the prediction process of distributed data more flexible, secure, and reliable. Simultaneously, Liu et al. [18] proposed a novel asynchronous decentralized federated learning framework. Under this framework, each photovoltaic power station can not only independently train local models but also actively participate in collaborative fault diagnosis through model parameter exchange. This method significantly improves the model's generalization capability while ensuring diagnostic accuracy. The global model is formed through distributed aggregation, effectively avoiding potential failures in central nodes and substantially reducing communication overhead and training time. Experimental and numerical simulation results fully verify the effectiveness and practicality of this method. Wu et al. [19] designed a privacy-preserving federated learning framework for transformer fault diagnosis that incorporates multi-level data sharing mechanisms and adaptive differential privacy protection measures. Experimental results demonstrate that this method exhibits outstanding diagnostic accuracy in identifying different types of transformer faults. Beyond power systems, recent studies have further explored federated learning in other critical infrastructures. For instance, federated learning has been applied to intrusion detection, enabling collaborative anomaly detection across distributed networks without exposing raw data, which demonstrates its strong potential in network security [20,21]. In addition, conditional generative adversarial networks have been combined with federated learning in satellite–terrestrial integrated networks, providing secure and efficient cross-domain data generation and model training [22,23]. These works highlight the broad applicability of federated learning across different domains, while our study specifically addresses the privacy-preserving challenges in smart grid substations by integrating asynchronous federated learning with blockchain and dynamic differential privacy mechanisms.

Although numerous studies have been conducted on federated learning and privacy protection, the following limitations persist: most existing research focuses on synchronous federated learning frameworks, which struggle to adapt to the asynchronous distributed characteristics of substation environments; secondly, differential privacy often employs fixed budgets, failing to account for contribution disparities among different clients, thereby compromising model utility; finally, while blockchain and storage integration primarily emphasize data integrity and traceability, they lack deep integration with differential privacy mechanisms.

To address the privacy leakage risks facing substation data and the challenge of decentralized trust establishment, this paper proposes a three-layer privacy-preserving architecture for substation data based on asynchronous federated learning. Compared to certain fully decentralized federated learning approaches, AFL maintains a degree of decentralization while achieving higher communication efficiency and stronger fault tolerance. This makes it particularly well-suited for substation environments characterized by imbalanced computational resources and inconsistent communication quality among nodes. Furthermore, through the integration of PBFT consensus mechanisms and smart contracts, AFL in this study simultaneously addresses collaborative security and decentralized training efficiency, representing a practical deployment solution under current grid system constraints. The contributions of this work are summarized as follows:

(1) A three-layer model architecture for privacy-preserving substation data is proposed, leveraging asynchronous federated learning. This architecture comprises a storage layer, service layer, and federated collaboration layer, achieving substation data privacy preservation through decentralized implementation. The framework employs blockchain to ensure data immutability, enhances off-chain storage efficiency via the IPFS, and resolves the single point failure issues inherent in traditional centralized federated learning.

(2) A differential privacy mechanism based on dynamic noise-tuning is proposed. During model training, Laplace noise is innovatively added to intermediate results to optimize differential privacy protection effectiveness. By dynamically adjusting noise injection strategies, the mechanism balances model performance and privacy strength while protecting sensitive parameter confidentiality. Experimental verification demonstrates its significant reduction of parameter leakage risks.

(3) An asynchronous federated aggregation method integrating PBFT and smart contract collaboration is proposed. The PBFT consensus algorithm ensures parameter consistency among distributed nodes and resists malicious node attacks. By leveraging smart contracts to automatically execute model aggregation logic, this method significantly addresses the centralization dependency and data trust storage issues in traditional federated learning.

2 Related Work

2.1 Asynchronous Federated Learning

In Asynchronous Federated Learning, the server prioritizes collaboration with clients that have completed training without waiting for all clients to finish training before aggregation [24]. When a client completes its training, it immediately communicates with the server to obtain the latest global model and then initiates the next round of local training. The advantages are that no client remains idle, and the global model can promptly integrate the latest local model parameters uploaded by clients [25]. However, there are also drawbacks: the communication strategy causes a significant increase in data transmission volume since each client must communicate individually with the server; clients with limited computational capabilities have longer local training cycles, during which the global model may have undergone multiple updates through interactions with computationally powerful clients, resulting in weaker clients using outdated global models uploading these parameters may degrade global model quality [26]; frequent communication between the server and powerful clients may lead to model bias. Fig. 1 illustrates the basic process of asynchronous federated learning.

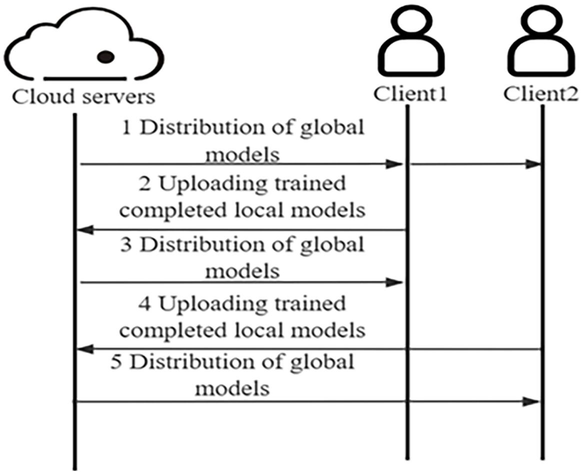

Cloud servers
Client1
Client2
1 Distribution of global models
2 Uploading trained completed local models
3 Distribution of global models
4 Uploading trained completed local models
5 Distribution of global models

Figure 1: Fundamental workflow of asynchronous federated learning

Asynchronous Federated Learning (FedAsync) [27] is a classical asynchronous federated learning algorithm. Compared with synchronous federated learning, a key distinction of asynchronous federated learning lies in its avoidance of global synchronization locks to schedule device operations. The pseudocode is as follows (Algorithm 1):

Algorithm 1: FedAsync algorithm

```
Process Server:
Initialize global parameters $w_0, a_t \leftarrow a$
Asynchronously run Scheduler() threads Updater() threads at the same time
Thread Scheduler():
Periodically triggers a subset of clients and sends them the timestamped global model
Thread Updater():
For epoch $t \in [T]$ do
Receiving information from the client $(x_{new}, r)$
Optional
Linear: $s_a(t-r) = \frac{1}{a(t-r)+1}$
Polynomial: $s_a(t-r) = (t-r+1)^{-a}$
Exponential: $s_a(t-r) = \exp(-a(t-r))$
Hinge: $s_{a,b}(t-r) = 1,$ & if $t-r \leq blat-r-b+1$
$ohterwise\ if\ Hinge \begin{cases} 1 & t-r \leq b \\ \frac{1}{a(t-r-b)+1} & otherwise \end{cases}$
$a_t \leftarrow a \times s(\cdot), s(\cdot)$ is a function of staleness
$x_t \leftarrow (1-a_t)\, x_{t-1} + a_t x_{new}$
Process Client():
For $i \in [N]$ in parallel do
If triggered by scheduler() then
Receive timestamped global models from the server $s(x_t, t)$
```

2.2 Differential Privacy

The core concept of differential privacy lies in enhancing privacy protection by adding differentially private noise to the model's raw gradients [28]. The Laplace mechanism is a canonical implementation of differential privacy, operating by injecting Laplacian-distributed random noise offsets to all data when generating target outputs [29]. The relevant definitions are as follows:

(1) Neighboring Datasets: For any two adjacent datasets D and D', given a random mechanism M algorithm operating on these adjacent datasets, when the inequality holds, S is the subset of all possible outputs generated by the M mechanism.

$$p_r[M(D)\,S] \leq e^{\varepsilon} p_r[M(D')\,S]. \tag{1}$$

Through the random mechanism M, ε−differential privacy is satisfied, where the parameter ε is the privacy protection budget. A smaller ε value indicates a higher privacy protection strength.

(2) Global Sensitivity: For a query function $f\colon \mathbb{N}^{|x|} \to \mathbb{R}^k$, the global sensitivity is defined over any two neighboring datasets D and D' as:

$$\Delta s(f, \|\bullet\|) = \max_{d(D,D')=1} \left\| f(D) - f(D') \right\|, \tag{2}$$

where $\|\bullet\|$ denotes the distance metric, typically implemented using l_1-norm or l_2-norm.

(3) Laplace Mechanism: This mechanism achieves ε−differential privacy by adding Laplace-distributed noise to query results. For any query function $f\colon \mathbb{N}^{|x|} \to \mathbb{R}^k$ with global sensitivity Δs, the Laplace

mechanism is defined as:

$$M_l\left(\left(x, f\left(\bullet\right), \epsilon\right)\right) = f\left(x\right) + \left(Y_1 + Y_2 \cdot\cdot\cdot, Y_k\right). \tag{3}$$

The Laplace mechanism satisfies the ε−differential privacy protection mechanism. Specifically, $Y_i \sim Lap\left(\frac{\Delta s}{\varepsilon}\right)$ are independently distributed random variables.

2.3 Blockchain

Blockchain technology establishes a distributed ledger system where information storage is achieved through chronologically ordered blocks [30]. Each block contains a finite set of transaction records, cryptographically linked to its predecessor via hash pointers, thereby forming an immutable chained architecture [31]. The structural configuration of blockchain is depicted in Fig. 2.

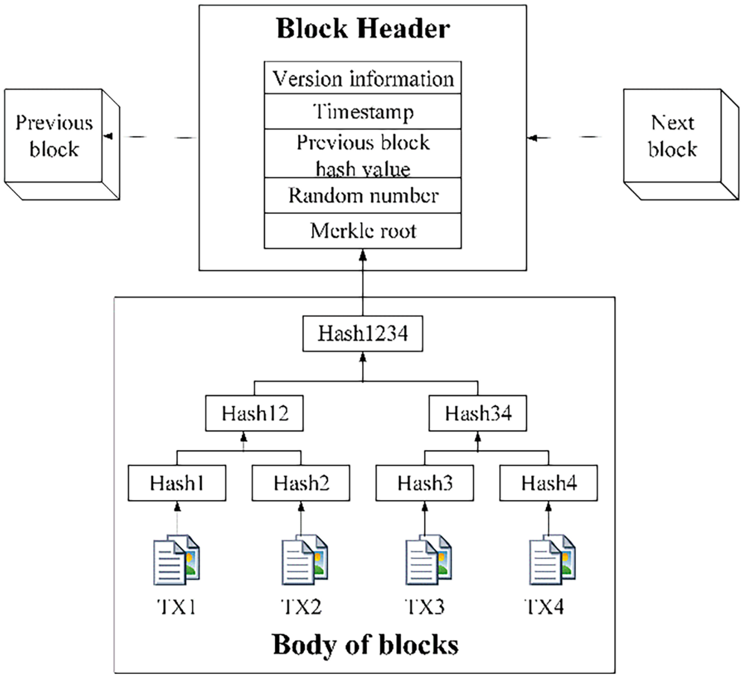


Figure 2: Blockchain architecture diagram

Within the blockchain technology framework, smart contracts serve as indispensable components that autonomously execute predefined rules and agreements upon condition fulfillment. The execution outcomes are immutably recorded on the blockchain network as tamper-evident and irreversible transaction records, requiring no external entity intervention. The flexibility of smart contracts manifests in their programmability through various programming languages to accommodate complex business logic requirements.

Consensus mechanisms constitute the core technology enabling coordinated operations among distributed nodes to maintain data consistency [32]. Given blockchain's decentralized nature, reliance on centralized authorities for transaction validation becomes infeasible. Consensus protocols thus play a pivotal role in ensuring all network nodes maintain identical ledger replicas with accurately recorded transactions [33]. As foundational elements of blockchain systems, these mechanisms effectively resolve the Byzantine Generals Problem in distributed environments.

2.4 IPFS

IPFS constitutes a decentralized peer-to-peer network for distributed file storage and sharing, designed to establish a unified file system integrating global computing resources [34]. By employing content-based addressing-where each file is uniquely identified and retrieved via cryptographic hash identifiers-IPFS enhances data reliability and accessibility through distributed storage [35].

IPFS's decentralized architecture inherently aligns with the decentralized storage requirements for substation privacy protection. Through encrypted sharding and hash verification mechanisms, this approach effectively mitigates leakage risks associated with centralized sensitive data storage while ensuring data integrity and traceability, thereby providing foundational support for high-reliability, privacy-first data sharing in smart grid systems. However, practical deployment may face challenges: IPFS could introduce model upload and indexing delays during cross-substation synchronization, while blockchain platforms may encounter throughput bottlenecks under high-frequency model submissions. Future work should integrate off-chain computation to asynchronously process model validation and aggregation, enhancing system scalability and practicality.

3 Method

3.1 AFL Substation Data Privacy Protection System Framework

The proposed asynchronous federated learning model for substation data privacy consists of three layers, as shown in Fig. 3. The storage layer employs IPFS for secure off-chain data storage, complementing blockchain to enhance decentralization; the service layer handles operational workflows with local model training, blockchain-audited processes, and certificate-based authentication; while the federated collaboration layer orchestrates cross-substation model training using blockchain and smart contracts, which aggregate local models to update global parameters-achieving privacy-preserving optimization through decentralized synchronization.

The model assumes there are n substations, with each substation S_i $(i = 1, 2, \cdots, n)$ possessing its dataset D_i. Each substation trains a local model m_i on its local dataset D_i, then uploads the model. Through smart contracts in the federated collaboration layer, global model M is aggregated. Each substation subsequently retrains a new local model m_i' based on the global model M, uploads it again, and generates an updated global model M', iterating this process until model convergence.

This framework employs the PBFT consensus algorithm for model aggregation, leveraging its high fault tolerance to rigorously validate client parameters before aggregation-mitigating malicious nodes while ensuring stringent data consistency beyond lightweight alternatives. For decentralized security, substations register on-chain, encrypt data via ECC for dual on/off-chain storage, and utilize smart contracts for global model aggregation. This dual mechanism eliminates central server reliance, prevents tampering, and secures both data privacy and training processes.

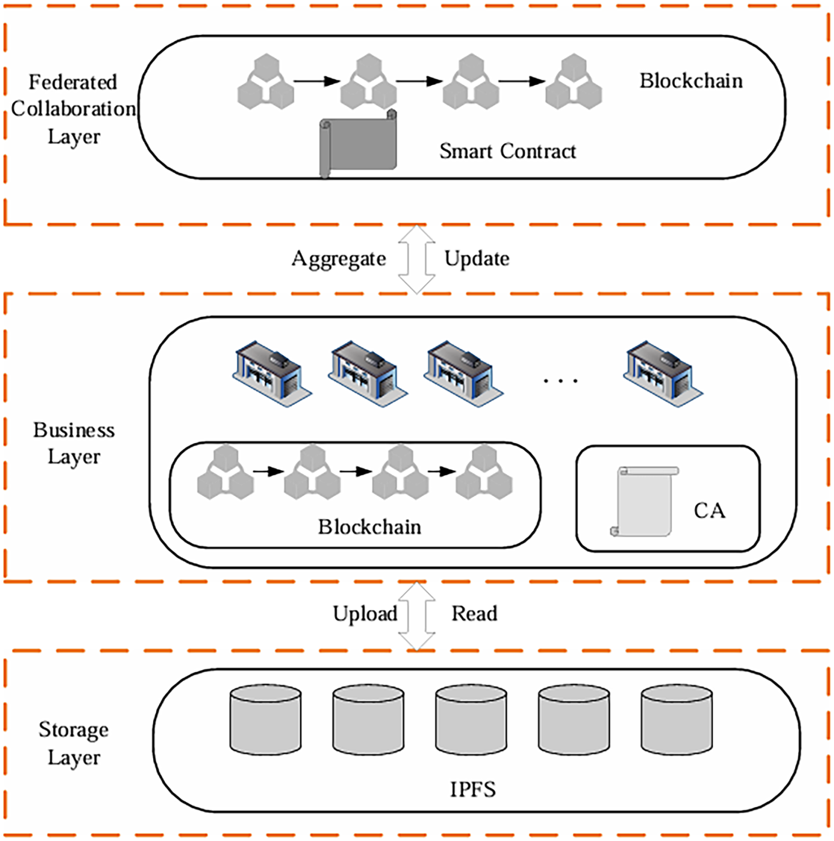


Figure 3: Asynchronous federated learning-based system model for substation data privacy preservation

3.2 AFL Processes for Substations

Within the substation data privacy framework, AFL enables decentralized model training as shown in Fig. 4. Authenticated substations sign/store data on IPFS, receiving unique hashes which are digitally signed and uploaded to the blockchain; following on-chain verification and broadcast, each substation retrieves initial parameters from the federation layer, trains local models using its dataset, uploads models to global nodes, and invokes smart contracts for parameter aggregation, generating an updated global model whose parameters are stored on-chain and synchronized across all substations; this process iterates from local retraining until model convergence, with IPFS handling distributed storage, blockchain recording metadata, and smart contracts executing aggregation.

To address client training inconsistencies in AFL, the system implements an asynchronous aggregation trigger and integrates directional consistency verification within PBFT consensus-applying penalizing weighting to clients whose parameters persistently deviate from the global optimization direction, thereby enhancing model stability.

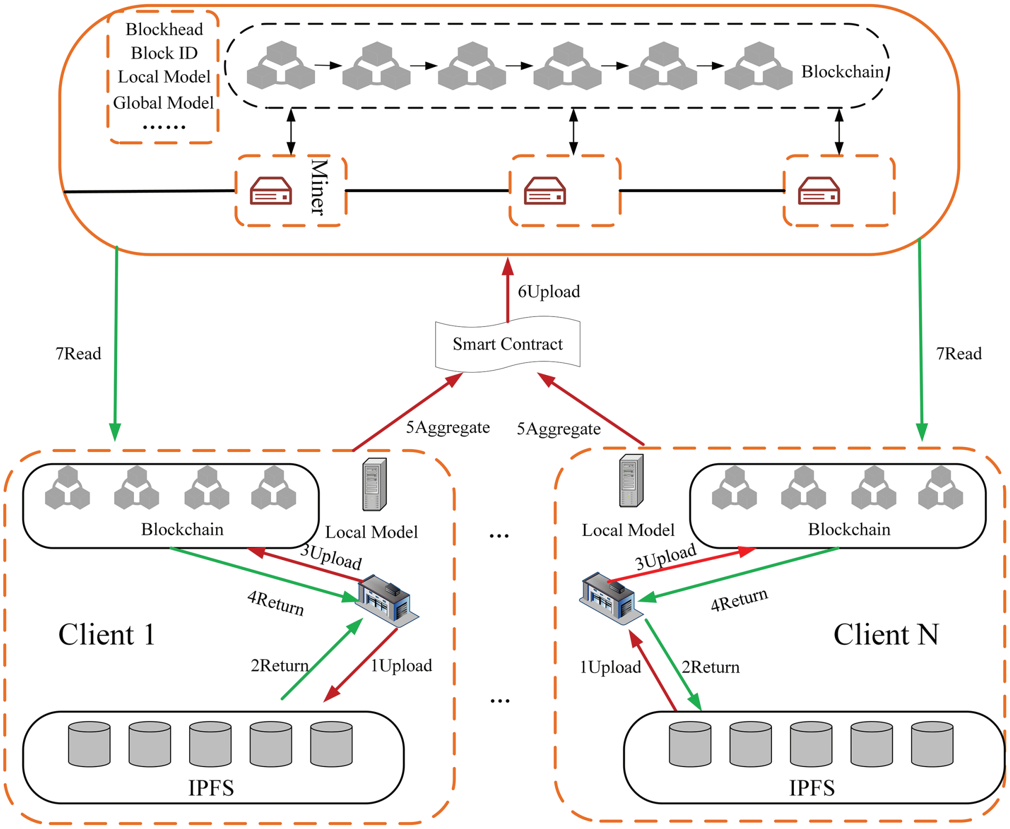


Figure 4: Asynchronous federated learning workflow for substations

3.3 Dynamically Tuned Differential Privacy Mechanism

To resolve temporal inconsistencies in asynchronous federated learning that challenge uniform DP, this paper proposes a dynamic DP mechanism. This approach jointly optimizes privacy granularity through: (1) A server-side DP budget allocation system using contribution factors and upload frequency factors to dynamically assign client-specific budgets; and (2) Client-level adaptive perturbation intensity adjustment collectively enabling balanced privacy protection and model performance despite update heterogeneity.

After the server receives the perturbed gradient $\tilde{g}_i^{(t)}$ uploaded by client i, it calculates the cosine similarity between the gradient and the global model update direction as the current round's contribution factor:

$$v_i^{(t)} = 1 + \cos\left(\tilde{g}_i^{(t)}, \Delta w^{(t)}\right) \tag{4}$$

where the model parameter change $\Delta w^{(t)}$ is defined as:

$$\Delta w^{(t)} = w^{(t)} - w^{(t-1)} \tag{5}$$

The training round index is denoted by t, and $\Delta w^{(t)}$ represents the model parameter changes in the t-th round. The expectation of $v_i^{(t)}$ follows the von Mises distribution approximation: $\mathbb{E}\left[v_i^{(t)}\right] \approx 1 + e^{-\sigma^2 t}$, where σ^2 quantifies data distribution divergence among clients. This indicates higher contribution weights ($v_i^{(t)} \to 2$) in early training stages and convergence to equilibrium ($v_i^{(t)} \to 1$) as t increases. To mitigate the impact of anomalous fluctuations in individual rounds, a moving average method is employed to calculate the historical average contribution factor:

$$\overline{v}_i \leftarrow \beta \cdot \overline{v}_i + (1-\beta) \cdot v_i^{(t)} \tag{6}$$

where $\overline{v}_i$ indicates the historical average contribution factor for client i, and $\beta \in (0,1)$ denotes the smoothing coefficient. Such a contribution-based adaptive allocation mechanism is, in essence, a form of adaptive filtering strategy. This approach aligns closely with the defensive effects and improvement directions revealed in existing studies on adaptive aggregation against model poisoning attacks [36]. Simultaneously, an upload frequency factor ϕ_i is introduced to quantify the client's model update frequency within a specific timeframe. The privacy budget for the next training round is dynamically configured as:

$$\varepsilon_i^{(t+1)} = \varepsilon_{base} \cdot \frac{1-\overline{v}_i}{1+\phi_i} \tag{7}$$

where ε_{base} represents the foundational privacy budget. When $\overline{v}_i$ is large, the client's updates have a more significant impact on the global model, and the privacy risk is higher. Therefore, the budget should be reduced to enhance noise protection. When ϕ_i is high, the privacy loss accumulates rapidly due to the composition effect. Thus, the per-round budget should also be reduced to maintain the overall constraint. This mechanism aligns with the sensitivity control principle and the composition theorem of differential privacy. When a client exhibits substantial training contributions, heightened privacy protection becomes imperative; consequently, its allocated budget should be reduced. Conversely, frequent model uploads by a client amplify its influence over the global model. This dynamic differential privacy allocation inherently introduces a randomization effect, which aligns with existing research demonstrating that randomized aggregation helps defend against model poisoning attacks [37]. To ensure global $(\varepsilon, 0) - DP$, the base budget ε_{base} is constrained by:

$$\varepsilon_{base} = \frac{\varepsilon}{n \cdot T_{\max} \cdot k'}, k = \mathbb{E} = \left[\max_j \left(\frac{1-\overline{v}_i}{1+\phi_i}\right)\right] \tag{8}$$

This mechanism achieves personalized privacy protection and rational resource allocation by dynamically adjusting budget distribution according to client behavioral characteristics. The server-side asynchronous differential privacy budget control algorithm is described as follows (Algorithm 2).

Algorithm 2: Server-side budget control process

```
Input: Total privacy budget $\varepsilon$, Client set, Moving average coefficient $\beta$, Max rounds $T_{\max}$
Output: Dynamic privacy budget $\varepsilon_i$ for each client
1: Initialize model parameters $w_0$, set base budget $\varepsilon_{base} = \varepsilon / (n \cdot T_{\max} \cdot k)$
2: Initialize for each client $i{:}\overline{v}_i = 1$, upload frequency count $\phi_i = 0$
3: After client uploads perturbed gradient $\tilde{g}_i^{(t)}$
    (1) Calculate model variation: $\Delta w^{(t)} = w^{(t)} - w^{(t-1)}$
    (2) Compute contribution factor: $v_i^{(t)} = 1 + \cos\left(\tilde{g}_i^{(t)}, \Delta w^{(t)}\right)$
    (3) Update historical contribution: $\overline{v}_i \leftarrow \beta \cdot \overline{v}_i + (1-\beta) \cdot v_i^{(t)}$
    (4) Update upload frequency: $\phi_i \leftarrow \phi_i + 1$
    (5) Calculate next-round budget: $\varepsilon_i^{(t+1)} = \varepsilon_{base} \cdot \frac{1-\overline{v}_i}{1+\phi_i}$
    (6) Aggregate perturbed gradients and update model: $w^{(t)} \leftarrow w^{(t)} \cdot \eta \cdot \tilde{g}_i^{(t)}$
```

Differential privacy aggregation logic processes verified client gradients via clipping, noise injection (base budget $\varepsilon = 0.1$), and weighted averaging. Using smoothing coefficient $\beta = 0.8$ updates clients' historical contribution factors, mitigating single-round anomalies while preserving trend responsiveness. Privacy budget initialization ($\overline{v}_i = 1$) ensures fairness; dynamic adjustments based on historical contributions and update frequency subsequently occur, with contribution factors initialized as $\phi_i = 0$ and iteratively refined via client-submitted model updates to govern ongoing allocations. Clients receive global parameters $w^{(t)}$ and current budget $\varepsilon_i^{(t)}$ for local training—yielding raw gradients $\nabla w_i^{(t)}$, while implementing adaptive clipping thresholds based on precision feedback to match local convergence speed.

The client first records the current gradient norm:

$$G_t = \left\| \nabla w_i^{(t)} \right\|_2 \tag{9}$$

A historical norm collection $\mathcal{G} = \{C_0, G_1, \cdots, G_t\}$ is maintained, which records historical gradient norms and initial clipping thresholds. The gradient clipping threshold C_t is dynamically updated via a $Quantile(\cdot)$ function, utilizing the median of the historical gradient norms $\boldsymbol{\mathcal{G}}$ as the quantization metric. Concurrently, the validation accuracy q_t plays a pivotal role in this update process. The updated cutting threshold as:

$$C_t = Quantile(\boldsymbol{\mathcal{G}}, Median) \times (1 + \lambda(1 - q_t)) \tag{10}$$

where $\lambda = \frac{\log T_{\max}}{\sqrt{T_{\max}}}$ is a convergence-optimal decay factor ensuring $\lim_{t \to T_{\max}} C_t = Median(\boldsymbol{\mathcal{G}})$. The local validation accuracy $q_t \in [0, 1]$ serves as utility feedback: when $q_t \to 1$, C_t relaxes to retain more gradient information; when $q_t \to 0$, C_t tightens to control sensitivity. The clipping operation is executed as:

$$g_i^{(t)} = \frac{\nabla w_i^{(t)}}{\max\left(1, \frac{\left\| \nabla w_i^{(t)} \right\|_2}{C_t}\right)} \tag{11}$$

This step bounds the L_2 sensitivity: $\Delta S = \sup_{D,D'} \left\| g_i(D) - g_i(D') \right\|_2 \leq 2C_t$ for adjacent datasets D, D'.

Adding Laplace Noise:

$$\tilde{g}_i^{(t)} = g_i^{(t)} + Laplace\left(0, \frac{2C_t}{\varepsilon_i^{(t)}}\right) \tag{12}$$

Finally, the client uploads the perturbed gradients to the server for participation in global model updating. The client-side gradient clipping and noise perturbation algorithm are described as follows (Algorithm 3).

Algorithm 3: Client-side gradient clipping and noise perturbation

```
Input: Model parameters $w^{(t)}$, privacy budget $\varepsilon_i^{(t)}$, initial threshold $C_0$
Output: Perturbed gradient $g_i^{(t)}$
Initialize gradient norm collection: $\mathcal{G} \leftarrow \{C_0\}$
Train local model to obtain gradient $\nabla w_i^{(t)}$
Compute norm: $G_t = \left\| \nabla w_i^{(t)} \right\|_2$
validation accuracy $q_t$, update clipping value: $C_t = Quantile\left(\mathcal{G}, Median\right) \times \left(1 + \lambda\left(1 - q_t\right)\right)$
Clip gradient: $g_i^{(t)} = \frac{\nabla w_i^{(t)}}{\max\left(1, \frac{\left\|\nabla w_i^{(t)}\right\|_2}{C_t}\right)}$
Add Laplace noise: $\tilde{g}_i^{(t)} = g_i^{(t)} + Laplace\left(0, \frac{2C_t}{\varepsilon_i^{(t)}}\right)$
Upload $\tilde{g}_i^{(t)}$ to the server for global model updating
```

On the client side (Algorithm 3), both local clipping and noise injection operations have a complexity of $O\left(d\right)$, and the communication complexity for uploading the perturbed gradient is also $O\left(d\right)$. On the server side (Algorithm 2), each round requires aggregating gradients uploaded by n clients, resulting in a computational complexity of $O\left(nd\right)$, along with one global noise injection step of complexity $O\left(d\right)$. In terms of communication, the server needs to distribute and receive n vectors, leading to a communication complexity of $O\left(nd\right)$. Therefore, the per-round computational complexity is $O\left(nd\right)$, and the per-round communication complexity is $O\left(nd\right)$. If the global training runs for T rounds, the overall overhead is $O\left(T \cdot nd\right)$.

3.4 PBFT and Smart Contract-Coordinated Asynchronous Federated Aggregation

3.4.1 PBFT Consensus Algorithm Construction

The PBFT consensus algorithm demonstrates superior scalability, enabling sustained high-efficiency operation amid expanding substation node quantities and complex environments—delivering robust security for reliable substation data interactions. The consensus process of the algorithm is detailed in Fig. 5.

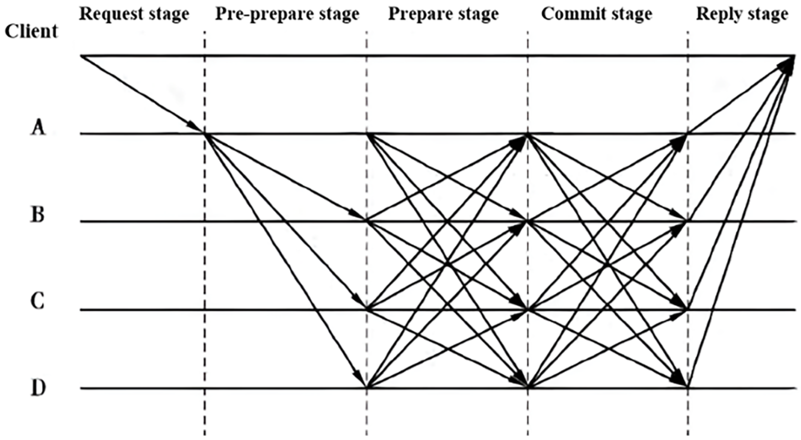


Figure 5: PBFT algorithm

Within the PBFT consensus framework, the primary node initiates aggregation proposals while replicas validate requests. If the primary consecutively fails to broadcast valid proposals, the system automatically triggers rotation—promoting the next designated replica. Nodes transmitting falsified digests or demonstrating persistent unresponsiveness are excluded to prevent Byzantine attacks from compromising global aggregation. The contract layer enforces response timeouts and rotation logic, maintaining robustness with a tolerance limit of $f < \frac{N-1}{3}$ faulty substations.

3.4.2 Smart Contract Design

Solidity-implemented smart contracts encode PBFT consensus logic to interface with substation devices via Go, facilitating bidirectional blockchain interactions: devices upload immutable operational data to the chain and autonomously adjust states based on PBFT-validated instructions. Upon meeting response thresholds or time limits, contracts trigger on-chain aggregation—executing parameter averaging using stored digests, then broadcast updated global models through event mechanisms. This entirely decentralized workflow ensures tamper-resistant ordering and autonomous contract execution for asynchronous federated learning. The asynchronous federated learning aggregation algorithm is detailed as follows (Algorithm 4).

Algorithm 4: Asynchronous federated learning aggregation

```
Input: Number of substations $n$, Local models $\mathrm{m}'_i$, Waiting time $T$, Threshold waiting time $T'$
Output: New global model $\mathrm{M}'$
    1: Each substation uploads its local model $\mathrm{M}'$
    2: When the number of substations $n$ reaches the required threshold or the waiting time $T \geq T'$, the local models $\mathrm{m}'_i$ re-averaged to obtain the new global model $\mathrm{M}'$
    3: The new global model $\mathrm{M}'$ is uploaded to the blockchain through consensus
    4: End
```

4 Experiments

This study employs a power grid-provided dataset comprising 5000 infrared surveillance units (4000 static images, 100 video clips of 5–30 s duration) collected from 5 distinct substations (1000 units per substation). Due to the limited number of infrared image samples, this study employed random data augmentation techniques (flipping, rotation, scaling, etc.) during training to expand the effective sample size; meanwhile, the risks of overfitting were mitigated through multi-client federated aggregation and the regularization effect of differential privacy noise. Video key frames are extracted as image sequences to capture spatiotemporal behavior patterns. CNN is selected as the foundational model for its superior spatial feature extraction capability in identifying local structural variations within infrared imagery, aligning with technical objectives while enabling industrial-grade deployment.

All samples undergo preprocessing/normalization with multi-category annotation: 3200 normal behavior instances, 800 unauthorized entry, 600 illegal operation, and 400 not wearing a safety helmet, demonstrating a controlled class imbalance that aligns with real-world substation operational profiles.

The PBFT consensus was deployed on a testbed consisting of 5 substation nodes. Each node operated on industrial-grade hardware. The blockchain platform utilized Hyperledger Fabric 2.4 with a block size of 1 MB and transaction throughput capped at 350 TPS. Network latency was emulated using Linux Traffic Control to simulate WAN conditions. The CNN architecture comprises two convolutional layers (30 channels for low-level feature extraction, 80 channels for high-level spatial feature capture) followed by five fully connected layers. Optimized for substation data characteristics and task requirements, pooling layers enhance robustness and computational efficiency through dimensionality reduction, while fully connected layers integrate convolutional features to refine classification performance. This network efficiently processes substation multi-modal data, achieving an optimal balance between computational efficiency and classification accuracy, as shown in Fig. 6. Model performance and privacy preservation were evaluated using accuracy, loss, and F1-score metrics.

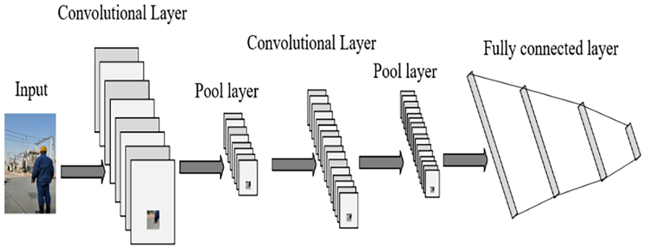
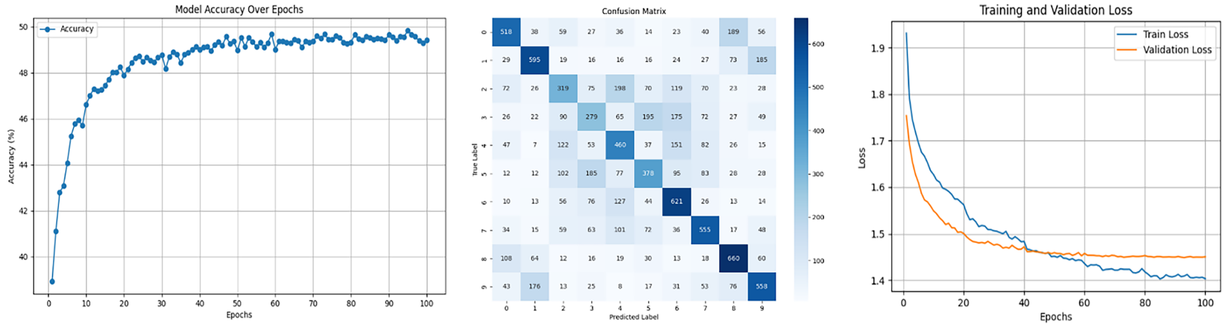


Figure 6: Improved CNN architecture schematic

4.1 Analysis of Key Performance Indicators of the Model

The proposed asynchronous federated learning model initially employs a baseline CNN architecture for training, achieving an accuracy of 49.03%, an F1-score of 0.486, and a loss of 1.49. As shown in Fig. 7, the experimental results demonstrate suboptimal model performance prior to architectural optimization.



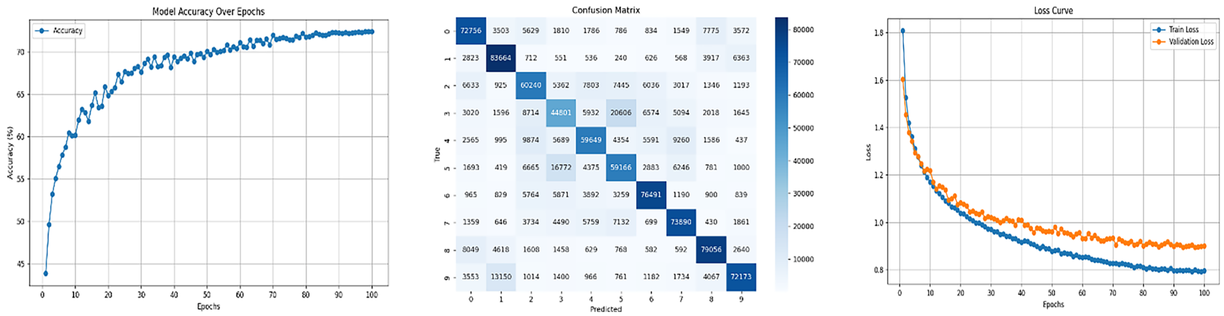
	0	1	2	3	4	5	6	7	8	9
0	518	38	59	27	36	14	23	40	189	56
1	29	595	19	16	16	16	24	27	73	185
2	72	26	319	75	198	70	119	70	23	28
3	26	22	90	279	65	195	175	72	27	49
4	47	7	122	53	460	37	151	82	26	15
5	12	12	102	185	77	378	95	83	28	28
6	10	13	56	76	127	44	621	26	13	14
7	34	15	59	63	101	72	36	555	17	48
8	108	64	12	16	19	30	13	18	660	60
9	43	176	13	25	8	17	31	53	76	558

(a) (b) (c)

Figure 7: Performance metrics of the baseline CNN architecture. (**a**) Model accuracy; (**b**) Confusion matrix; (**c**) Model loss

Building upon the original CNN architecture, an enhanced CNN model was developed with the application of data augmentation techniques to improve the diversity of the training dataset. For learning rate scheduling, the Cosine Annealing Learning Rate Scheduler was employed to dynamically adjust learning rates, enabling smoother model convergence. Additionally, early stopping and regularization methods were incorporated during the training process. To evaluate the enhanced CNN architecture, a comparative analysis was conducted against a baseline model. This baseline model was trained on an identical dataset encompassing the multi-modal information from substations utilized for the enhanced model. Furthermore, an equivalent number of training epochs was applied to both models during the training process. The experimental results are presented in Fig. 8.

The experimental results demonstrate that the enhanced model achieves an accuracy of 72.34%, an F1-score of 0.72, and a loss of 0.89, representing significant improvements over the baseline CNN model. Comparative metrics of the original and enhanced models are summarized in Table 1.



	0	1	2	3	4	5	6	7	8	9
0	72756	3503	5629	1810	1786	786	834	1549	7775	3572
1	2823	83664	712	551	536	240	626	568	3917	6363
2	6633	925	60240	5362	7803	7445	6036	3017	1346	1193
3	3020	1596	8714	44801	5932	20606	6574	5094	2018	1645
4	2565	995	9874	5689	59649	4354	5591	9260	1586	437
5	1693	419	6665	16772	4375	59166	2883	6246	781	1000
6	965	829	5764	5871	3892	3259	76491	1190	900	839
7	1359	646	3734	4490	5759	7132	699	73890	430	1861
8	8049	4618	1608	1458	629	768	582	592	79056	2640
9	3553	13150	1014	1400	966	761	1182	1734	4067	72173

(a) (b) (c)

Figure 8: Performance metrics of the improved CNN architecture. (**a**) Model accuracy; (**b**) Confusion matrix; (**c**) Model loss

Table 1: Model comparison

Evaluation indicators	Accuracy	Loss	F1
Original model	48.97%	1.49	0.486
Improved	72.34%	0.89	0.72

4.2 Privacy-Preserving Data Encryption and Model Aggregation Performance Analysis

To validate the efficiency of data encryption and successful decryption post-encryption, datasets ranging from 100 to 500 samples were encrypted using the SM2 algorithm. The encryption time was measured for each dataset size, with test results shown in Fig. 9. Empirical results demonstrate that encryption time increases proportionally with data volume, exhibiting a linear growth trend as the dataset size escalates.

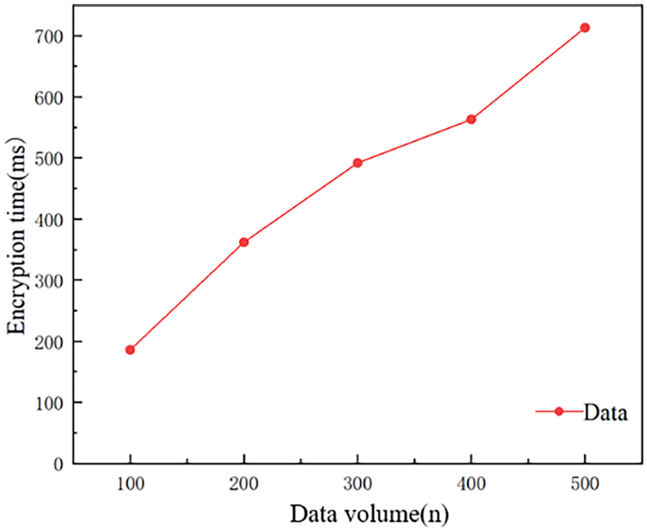
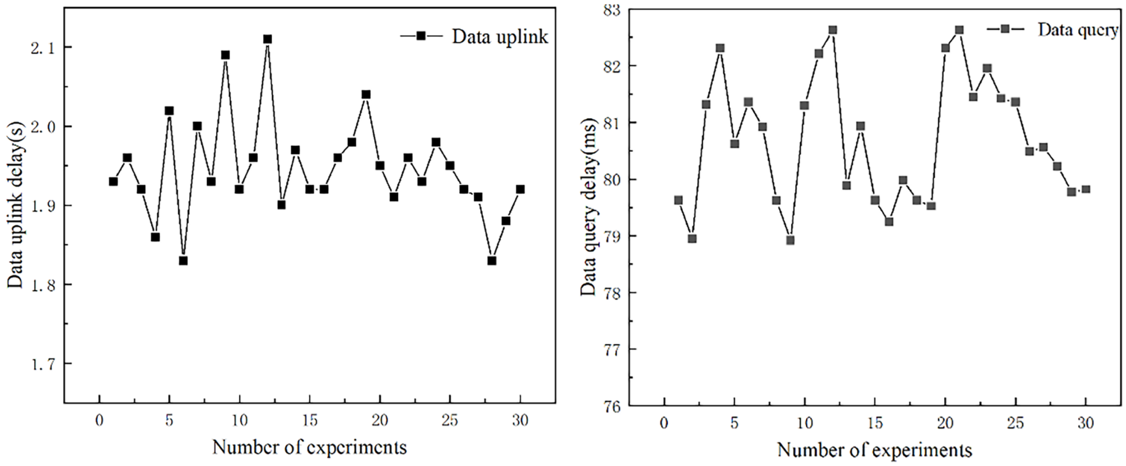


Figure 9: Data encryption delay

Subsequently, the aggregation-to-chain latency and query latency were tested to evaluate the effectiveness of the asynchronous federated learning model aggregation. Each test round comprised 30 trials. The aggregation-to-chain latency results are depicted in Fig. 10a. Experimental results indicate an average aggregation-to-chain latency of 1.58 s. Furthermore, query latency tests were conducted on the aggregated on-chain data, as shown in Fig. 10b. The results demonstrate an average query latency of 80.79 ms, confirming the model's capability to meet substation requirements for data aggregation and query operations.



(a)

(b)

Figure 10: Model aggregation performance benchmarking. (**a**) Aggregation-to-Chain latency; (**b**) Query latency

4.3 Numerical Results of Dynamic Noise Tuned Differential Privacy Mechanism

To ensure a rigorously fair comparison between the proposed dynamic differential privacy mechanism and the static uniform noise-injection approach, both schemes were allocated an identical privacy budget ($\varepsilon = 0.1$), thereby ensuring equivalent privacy protection levels. The experimental results are illustrated in Fig. 11, the dynamic mechanism demonstrates consistent superiority across multiple performance dimensions. It not only attained a higher final accuracy of 72.34% compared to approximately 68.1% for the static method, but also achieved a lower loss value of 0.89 and a superior F1-score of 0.72. These improvements are attributed to the mechanism's ability to adaptively modulate noise injection based on each client's contribution level and update frequency, effectively preserving informative gradients from reliable clients while suppressing potentially disruptive updates from unreliable or malicious participants. In contrast, the static uniform noise scheme introduces unmodulated interference across all updates, impairing the learning process and resulting in slower convergence and reduced stability. Furthermore, the dynamic approach promotes more consistent convergence behavior and enhances robustness against potential poisoning attacks, underscoring its practical utility in real-world federated learning scenarios such as substation monitoring.

Fig. 12 comprehensively quantifies the impact of privacy budget ε on federated learning systems. The privacy budget ε in this study is set within the range of [0.01, 1.0] not to recommend the direct use of $\varepsilon = 1.0$ in practical deployment, but to systematically characterize the complete spectrum of the privacy-utility trade-off. This range quantitatively reveals the impact trajectory of different privacy budgets on model performance: as ε increases, the noise amplitude decreases, model accuracy gradually improves, loss decreases, and convergence accelerates. Notably, experimental results indicate that $\varepsilon = 1.0$ represents a critical balancing point in this task: while maintaining strict privacy constraints, the model still achieves an accuracy of 68.4%, a loss of 0.98, and converges in 98 rounds, balancing both privacy and utility. In more sensitive practical applications, it is common practice to select $\varepsilon \leq 1.0$ or even smaller budget values to obtain stronger privacy protection. In the future, we will extend the experiments to scenarios with $\varepsilon < 1.0$ to validate the applicability and robustness of the proposed method under extreme privacy conditions.

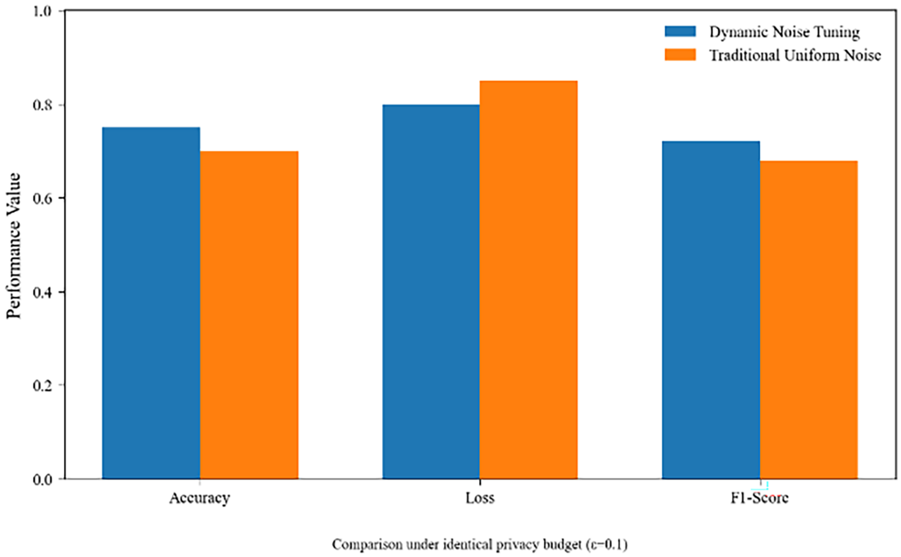
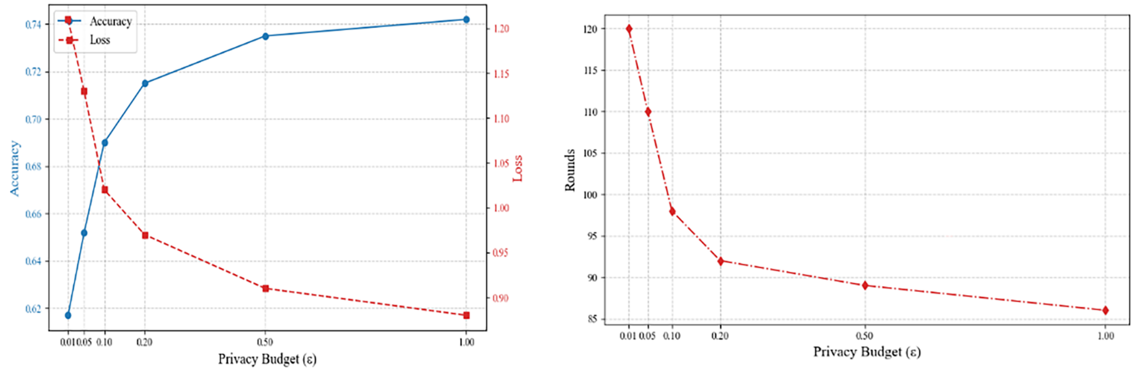


Figure 11: Feasibility analysis of differential privacy mechanisms



(a) (b)

Figure 12: Model performance with privacy budget. (**a**) Privacy budget and model accuracy/loss relationship; (**b**) Relationship between privacy budget and model convergence

Fig. 13 experimental results demonstrate the convergence behaviors under different β values. When $\beta = 0.1$, the model converges faster in the initial stage but stagnates at a lower accuracy in later phases. $\beta = 0.3$ achieves an optimal balance between efficiency and stability, reaching 81% accuracy by the mid-phase, making it the best choice for substation scenarios. While $\beta = 0.5$ attains the highest final accuracy, it exhibits the slowest convergence speed. Further increasing β to 0.7 results in a smoother convergence curve and similar final accuracy to $\beta = 0.5$, but with even slower convergence. At $\beta = 0.9$, excessive smoothing leads to delayed update responses and limited model improvement. Overall, a moderate value (e.g., $\beta = 0.3$) provides the best trade-off between convergence efficiency and final performance, whereas excessively large values may weaken the model's learning capability.

Fig. 14 quantitatively demonstrates the performance superiority of dynamic clipping over static methods: dynamic clipping achieves significantly higher initial accuracy and maintains persistent performance advantages throughout the training trajectory. Although both methods converge to comparable final accuracy, dynamic clipping stabilizes at target precision 20 epochs earlier, empirically validating that adaptive gradient clipping mechanisms substantially accelerate convergence efficiency while enhancing training stability.

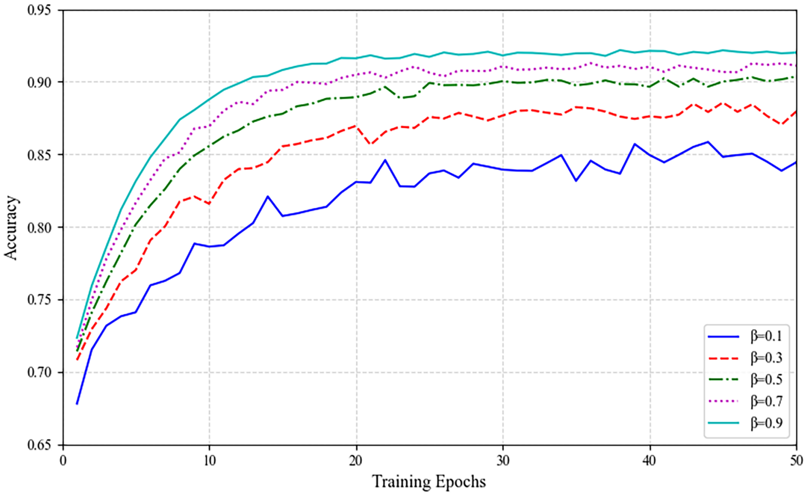


Figure 13: Convergence comparison under different β values

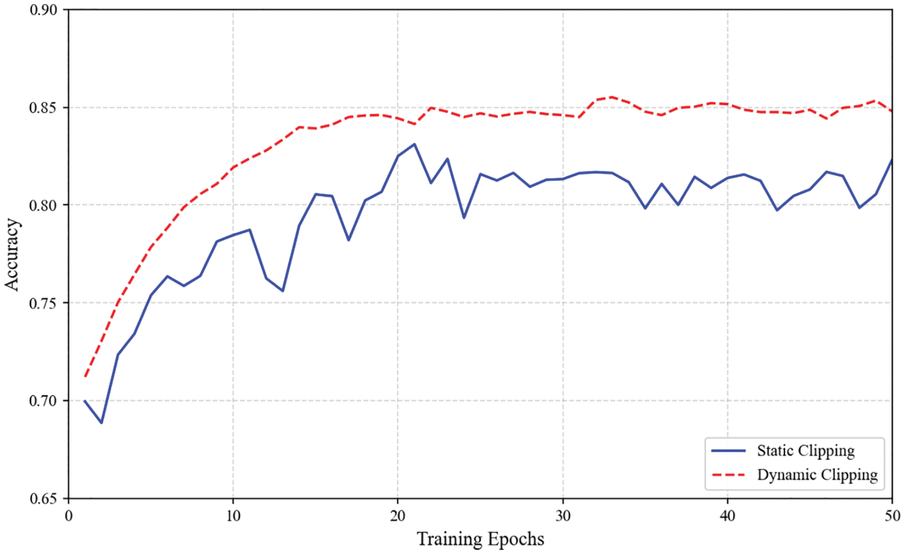


Figure 14: Performance advantage of dynamic clipping and static clipping

Fig. 15 presents a comparative analysis between our proposed method and several baseline approaches, including DDP, DP-FedAvg, FedAsync with static DP, and a Byzantine-resilient federated learning method. The results demonstrate that while DDP achieves the highest performance in a centralized environment, it provides no privacy protection. DP-FedAvg introduces strong noise to ensure privacy but at the cost of reduced model accuracy. FedAsync with DP shows certain efficiency advantages in asynchronous training, yet its improvements in the privacy-utility trade-off remain limited. The Byzantine-resilient method exhibits robustness against malicious updates; however, its overall accuracy is still lower than our approach due to additional aggregation constraints. In contrast, our proposed method outperforms all baselines, achieving the highest accuracy and F1-score while maintaining the lowest loss value. These results highlight the comprehensive advantages of our dynamic privacy budget allocation mechanism in terms of utility, convergence stability, and robustness.

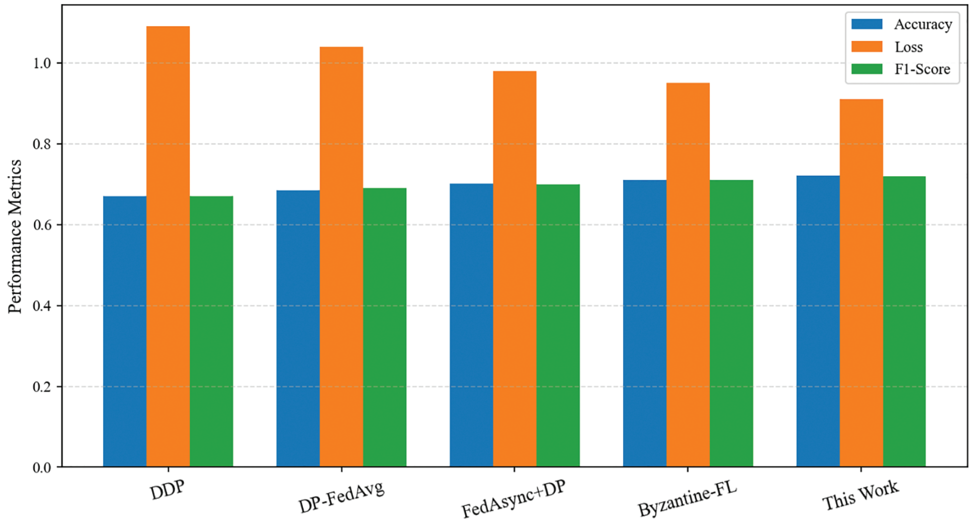


Figure 15: Comparison of performance between the proposed method and traditional federated learning

5 Conclusion

To address the challenges of centralized single-point failures and synchronous gradient leakage in substation data privacy protection, this paper proposes a three-tiered architecture for substation data privacy preservation based on asynchronous federated learning. The architecture achieves privacy protection through a dynamic noise-tuning mechanism while enhancing decentralized aggregation efficiency via the integration of PBFT and smart contract-coordinated consensus. Experimental results demonstrate that the proposed architecture significantly improves model performance while ensuring substation data privacy. Specifically, the enhanced CNN model exhibits a 23.37% improvement in accuracy and a 0.234 increase in F1-score compared to the baseline model. Additionally, the data aggregation-to-chain latency (1.58 s) and query latency (80.79 ms) meet substation operational requirements, providing a viable solution for smart grid data privacy protection.

Future research may be directed toward several important extensions of the present work. First, algorithmic optimization remains a crucial avenue, particularly through the development of more efficient strategies for dynamic privacy budget allocation and lightweight aggregation mechanisms, with the aim of reducing computational and communication overhead in resource-constrained environments. Second, the applicability and robustness of privacy-preserving federated learning frameworks merit further validation in complex and large-scale grid environments, including cross-regional and multi-node scenarios, where heterogeneous data and dynamic conditions pose additional challenges. Finally, the design of cross-regional blockchain interoperability protocols constitutes a promising research direction, as such protocols are essential for enabling secure, scalable, and collaborative data exchange across heterogeneous blockchain infrastructures in power system applications.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by the National Natural Science Foundation of China, grant number 61605004, the Fundamental Research Funds for the Central Universities, grant number FRF-TP-19-016A2, and Guizhou

Power Grid Co., Ltd. 2024 first batch of services (2024–2026 technology R&D services for science and technology projects (in addition to national and SGCC key projects)), grant number 060100KC23100012.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Yuewei Tian, Xuyang Wu and Fang Ren; methodology, Yuewei Tian, Xuyang Wu and Fang Ren; software, Fang Ren; validation, Fang Ren; formal analysis, Xuyang Wu; investigation, Yang Su; resources, Xuyang Wu; data curation, Xuyang Wu; writing—original draft preparation, Xuyang Wu and Fang Ren; writing—review and editing, Yuewei Tian and Fang Ren; visualization, Yujia Wang and Lisa Guo; supervision, Fang Ren; project administration, Lei Cao; funding acquisition, Fang Ren. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the Corresponding Author, [Xuyang Wu], upon reasonable request.

Ethics Approval: Not applicable.

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

References

1. Wu B, Hu Y. Analysis of substation joint safety control system and model based on multi-source heterogeneous data fusion. IEEE Access. 2023;11:35281–97. doi:10.1109/ACCESS.2023.3264707.
2. Zhou F, Wen G, Ma Y, Geng H, Huang R, Pei L, et al. A comprehensive survey for deep-learning-based abnormality detection in smart grids with multimodal image data. Appl Sci. 2022;12(11):5336. doi:10.3390/app12115336.
3. Hasan MK, Habib AA, Shukur Z, Ibrahim F, Islam S, Razzaque MA. Review on cyber-physical and cyber-security system in smart grid: standards, protocols, constraints, and recommendations. J Netw Comput Appl. 2023;209:103540. doi:10.1016/j.jnca.2022.103540.
4. Livada Č, Glavaš H, Baumgartner A, Jukić D. The dangers of analyzing thermographic radiometric data as images. J Imaging. 2023;9(7):143. doi:10.3390/jimaging9070143.
5. Sutherland N, Marsh S, Priestnall G, Bryan P, Mills J. InfraRed thermography and 3D-data fusion for architectural heritage: a scoping review. Remote Sens. 2023;15(9):2422. doi:10.3390/rs15092422.
6. Abdelkader S, Amissah J, Kinga S, Mugerwa G, Emmanuel E, Mansour DA, et al. Securing modern power systems: implementing comprehensive strategies to enhance resilience and reliability against cyber-attacks. Results Eng. 2024;23:102647. doi:10.1016/j.rineng.2024.102647.
7. Luan Z, Lai Y, Xu Z, Gao Y, Wang Q. Federated learning-based insulator fault detection for data privacy preserving. Sensors. 2023;23(12):5624. doi:10.3390/s23125624.
8. Fernández JD, Menci SP, Lee CM, Rieger A, Fridgen G. Privacy-preserving federated learning for residential short-term load forecasting. Appl Energy. 2022;326(1):119915. doi:10.1016/j.apenergy.2022.119915.
9. Putrama IM, Martinek P. Enhancing protection in high-dimensional data: distributed differential privacy with feature selection. Inf Process Manag. 2024;61(6):103870. doi:10.1016/j.ipm.2024.103870.
10. Yazdinejad A, Dehghantanha A, Karimipour H, Srivastava G, Parizi RM. A robust privacy-preserving federated learning model against model poisoning attacks. IEEE Trans Inf Forensics Secur. 2024;19:6693–708. doi:10.1109/TIFS.2024.3420126.
11. Sharma A, Marchang N. A review on client-server attacks and defenses in federated learning. Comput Secur. 2024;140:103801. doi:10.1016/j.cose.2024.103801.
12. Fotiou N, Pittaras I, Siris VA, Polyzos GC, Anton P. A privacy-preserving statistics marketplace using local differential privacy and blockchain: an application to smart-grid measurements sharing. Blockchain Res Appl. 2021;2(1):100022. doi:10.1016/j.bcra.2021.100022.
13. Zheng Z, Wang T, Bashir AK, Alazab M, Mumtaz S, Wang X. A decentralized mechanism based on differential privacy for privacy-preserving computation in smart grid. IEEE Trans Comput. 2022;71(11):2915–26. doi:10.1109/TC.2021.3130402.

14. Fan H, Liu Y, Zeng Z. Decentralized privacy-preserving data aggregation scheme for smart grid based on blockchain. Sensors. 2020;20(18):5282. doi:10.3390/s20185282.
15. Alsharif MH, Kannadasan R, Wei W, Nisar KS, Abdel-Aty AH. A contemporary survey of recent advances in federated learning: taxonomies, applications, and challenges. Internet Things. 2024;27:101251. doi:10.1016/j.iot.2024.101251.
16. Shabbir A, Manzoor HU, Manzoor MN, Hussain S, Zoha A. Robustness against data integrity attacks in decentralized federated load forecasting. Electronics. 2024;13(23):4803. doi:10.3390/electronics13234803.
17. Li Q, Liu D, Cao H, Liao X, Lai X, Cui W. Decentralized asynchronous adaptive federated learning algorithm for securely prediction of distributed power data. Front Energy Res. 2024;11:1340639. doi:10.3389/fenrg.2023.1340639.
18. Liu Q, Yang B, Wang Z, Zhu D, Wang X, Ma K, et al. Asynchronous decentralized federated learning for collaborative fault diagnosis of PV stations. IEEE Trans Netw Sci Eng. 2022;9(3):1680–96. doi:10.1109/TNSE.2022.3150182.
19. Wu Q, Dong C, Guo F, Wang L, Wu X, Wen C. Privacy-preserving federated learning for power transformer fault diagnosis with unbalanced data. IEEE Trans Ind Inform. 2024;20(4):5383–94. doi:10.1109/TII.2023.3333914.
20. Khraisat A, Alazab A, Singh S, Jan T, Gomez A Jr. Survey on federated learning for intrusion detection system: concept, architectures, aggregation strategies, challenges, and future directions. ACM Comput Surv. 2024;57(1):1–38. doi:10.1145/3687124.
21. Li J, Tong X, Liu J, Cheng L. An efficient federated learning system for network intrusion detection. IEEE Syst J. 2023;17(2):2455–64. doi:10.1109/JSYST.2023.3236995.
22. Jiang W, Han H, Zhang Y, Mu J, Shankar A. Intrusion detection with federated learning and conditional generative adversarial network in satellite-terrestrial integrated networks. Mob Netw Appl. 2024;2024(4):1–14. doi:10.1007/s11036-024-02435-4.
23. Zhao L, Geng S, Tang X, Hawbani A, Sun Y, Xu L, et al. ALANINE: a novel decentralized personalized federated learning for heterogeneous LEO satellite constellation. IEEE Trans Mob Comput. 2025;24(8):6945–60. doi:10.1109/TMC.2025.3545429.
24. Sun S, Zhang Z, Pan Q, Liu M, Wang Y, He T, et al. Staleness-controlled asynchronous federated learning: accuracy and efficiency tradeoff. IEEE Trans Mob Comput. 2024;23(12):12621–34. doi:10.1109/TMC.2024.3416216.
25. Ji Y, Chen L. FedQNN: a computation–communication-efficient federated learning framework for IoT with low-bitwidth neural network quantization. IEEE Internet Things J. 2023;10(3):2494–507. doi:10.1109/JIOT.2022.3213650.
26. Wan Y, Qu Y, Ni W, Xiang Y, Gao L, Hossain E. Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: a comprehensive survey. IEEE Commun Surv Tutor. 2024;26(3):1861–97. doi:10.1109/COMST.2024.3361451.
27. Li J, Jiang M, Qin Y, Zhang R, Ling SH. Intelligent depression detection with asynchronous federated optimization. Complex Intell Systems. 2023;9(1):115–31. doi:10.1007/s40747-022-00729-2.
28. Gao Q, Sun H, Wang Z. DP-EPSO: differential privacy protection algorithm based on differential evolution and particle swarm optimization. Opt Laser Technol. 2024;173(1):110541. doi:10.1016/j.optlastec.2023.110541.
29. Sathish Kumar G, Premalatha K, Uma Maheshwari G, Rajesh Kanna P, Vijaya G, Nivaashini M. Differential privacy scheme using Laplace mechanism and statistical method computation in deep neural network for privacy preservation. Eng Appl Artif Intell. 2024;128(2):107399. doi:10.1016/j.engappai.2023.107399.
30. Parida NK, Jatoth C, Reddy VD, Hussain MM, Faizi J. Post-quantum distributed ledger technology: a systematic survey. Sci Rep. 2023;13(1):20729. doi:10.1038/s41598-023-47331-1.
31. Feng Q, He D, Zeadally S, Khan MK, Kumar N. A survey on privacy protection in blockchain system. J Netw Comput Appl. 2019;126:45–58. doi:10.1016/j.jnca.2018.10.020.
32. Lashkari B, Musilek P. A comprehensive review of blockchain consensus mechanisms. IEEE Access. 2021;9:43620–52. doi:10.1109/ACCESS.2021.3065880.
33. Yue K, Zhang Y, Chen Y, Li Y, Zhao L, Rong C, et al. A survey of decentralizing applications via blockchain: the 5G and beyond perspective. IEEE Commun Surv Tutor. 2021;23(4):2191–217. doi:10.1109/COMST.2021.3115797.

34. Balduf L, Henningsen S, Florian M, Rust S, Scheuermann B. Monitoring data requests in decentralized data storage systems: a case study of IPFS. In: 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS); 2022 Jul 10–13; Bologna, Italy. p. 658–68. doi:10.1109/ICDCS54860.2022.00069.
35. Shi R, Cheng R, Han B, Cheng Y, Chen S. A closer look into IPFS: accessibility, content, and performance. Proc ACM Meas Anal Comput Syst. 2024;8(2):1–31. doi:10.1145/3656015.
36. Nabavirazavi S, Taheri R, Shojafar M, Iyengar SS. Impact of aggregation function randomization against model poisoning in federated learning. In: 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom); 2023 Nov 1–3; Exeter, UK. p. 165–72. doi:10.1109/TrustCom60117.2023.00043.
37. Nabavirazavi S, Taheri R, Ghahremani M, Iyengar SS. Model poisoning attack against federated learning with adaptive aggregation. In: Adversarial multimedia forensics. Cham, Switzerland: Springer Nature; 2023. doi:10.1007/978-3-031-49803-9_1.