FedQFS: A Blockchain-Based Federated Learning Framework Based on Quality Auditing, Fairness Deviation, and Sybil-Resistant Similarity

Tian Fang[1], Bing Li[1, 2 (✉)], Jianglin Yu[1], Zuwei Chen[1], Baofu Han[1], and Pan Feng[1]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 210000, China
[2] School of Integrated Circuits, Southeast University, Nanjing 210000, China
bernie_seu@seu.edu.cn

Abstract. Federated Learning (FL) integrated with blockchain technology effectively mitigates the risks of single points of failure and malicious server behavior inherent in traditional federated learning architectures that rely on central servers. However, in practical implementation, the system still faces significant challenges in ensuring security and fairness. Malicious clients may upload low-quality or biased local models to disrupt global convergence, while Sybil nodes can forge multiple identities to manipulate aggregation, leading to severe performance degradation. To address these issues, this paper proposes a blockchain-based federated learning framework based on quality auditing, fairness deviation, and Sybil-resistant similarity (FedQFS). First, a quality auditing mechanism is designed to evaluate and filter local model updates through multi-dimensional metrics, effectively mitigating global accuracy degradation caused by malicious updates. Second, a Sybil-resilient identification mechanism is introduced, which leverages parameter similarity analysis to accurately detect forged identities, thereby enhancing the system's resistance against Sybil attacks. Finally, a fairness deviation quantification mechanism is incorporated to measure parameter distribution disparities and adaptively assign reasonable aggregation weights to benign clients with limited data, ensuring fairness in global model updates. Experimental results show that the proposed framework achieves over 95.5% accuracy on the MNIST dataset and maintains strong robustness under Sybil attack scenarios. When four label-flipping attackers are present, its attack success rate decreases by 18.4% compared with mainstream aggregation algorithms. These results indicate that FedQFS is effective under the evaluated MNIST setting and support the feasibility of integrating quality auditing, similarity-based defense, and fairness-aware deviation control in a blockchain-assisted federated learning pipeline.
Keywords: Federated Learning, Quality Auditing, Fairness Deviation, Sybil Attack Defense, Blockchain

1 Introduction

With the rapid advancement of big data and artificial intelligence (AI), data has become a key driver of societal intelligence [1]. Machine learning (ML), as a core tool for data analysis and knowledge discovery, is widely applied in fields such as medical diagnosis

[2], financial assessment [3], and education [4]. Despite its maturity, ML faces critical bottlenecks, notably data silos and security concerns [5]. Data sharing across organizations is often limited by commercial, privacy, and regulatory constraints, leaving high-value data fragmented and underutilized [6]. Centralized integration requires substantial computational and human resources, and existing frameworks struggle to manage massive, heterogeneous datasets efficiently, further impeding collaboration. Moreover, sharing sensitive data poses serious privacy risks, potentially causing economic, legal, or ethical consequences. Regulations such as the European Union's General Data Protection Regulation (GDPR) enhance privacy protection but can inadvertently reinforce data silos [7]. Therefore, improving data availability and sharing efficiency while preserving security and privacy remains a key challenge.

To address data silos and privacy concerns, Federated Learning (FL) has emerged as a promising distributed machine learning paradigm, enabling multiple data owners to collaboratively train a shared model without exchanging raw data [8, 9]. In FL, a central server coordinates training: each client trains a local model on private data and periodically uploads updates, which the server aggregates to generate a new global model [10]. Through iterative aggregation, the global model approaches centralized training performance while reducing data barriers and privacy risks [11]. However, FL still faces security risks: the server may become a single point of failure or compromise fairness and integrity by favoring clients or manipulating aggregation [12]. Blockchain, as a decentralized and distributed ledger technology, effectively addresses the centralization issue in FL [13]. Its inherent immutability and traceability enable a distributed, secure, and scalable training environment, allowing FL to operate in multi-party, non-trusting settings with enhanced privacy and verifiable reliability. Nevertheless, practical deployments still face security and fairness challenges [14]. Malicious clients may submit low-quality or tampered models, such as those with noise [15] or malicious gradients [16], disrupting global convergence and degrading model accuracy and generalization. Sybil attacks [17], where adversaries forge multiple identities to manipulate aggregation weights, further compromise fairness and stability. Additionally, security enforcement may conflict with data diversity: current defenses may misclassify benign updates from unique or minority clients as malicious, suppressing the model's ability to learn rare but valuable features and weakening global generalization performance.

To address the security and fairness challenges, this paper proposes a blockchain-based federated learning framework based on quality auditing, fairness deviation, and Sybil-resistant similarity (FedQFS), which effectively mitigates malicious interference in the global training process and enhances the system's resilience against attacks. First, a model quality auditing mechanism evaluates and filters local updates using multi-dimensional metrics. This mechanism reduces the adverse impact of malicious or low-quality updates on global accuracy at the source, thereby alleviating model degradation. Second, a Sybil-resilient identification mechanism is developed, which leverages model parameter similarity analysis to precisely detect and filter out forged identities, significantly improving the system's defense capability against Sybil attacks. Finally, a fairness deviation quantification mechanism is introduced to measure the parameter distribution discrepancies among clients, enabling the distinction between benign up-

dates and abnormal malicious behaviors. This mechanism adaptively assigns aggregation weights to benign clients with smaller datasets, ensuring their fair contribution to the global update. Collectively, these mechanisms form an integrated framework that jointly addresses model-quality auditing, similarity-based Sybil resistance, and fairness-aware deviation control within a blockchain-assisted federated learning setting. The main contribution of this work lies in the coordinated integration of these components into a unified training and auditing pipeline.

The structure of this paper is organized as follows: The existing blockchain-based federated learning security research in Sec. 2. Sec. 3 presents a detailed description of the proposed FedQFS framework, including the overall framework design, the core mechanisms and models of FedQFS, the client evaluation strategy based on quality auditing, the fairness deviation constraint design, and the aggregation method incorporating Sybil-resilient similarity. Section 4 shows experimental results on the MNIST dataset, demonstrating that the proposed FedQFS framework achieves high model accuracy while exhibiting strong capability against Sybil attacks. Finally, Sec. 5 concludes this paper.

2 Existing Research

In the field of FL security, extensive research has been conducted on poisoning attack detection, fairness preservation, and privacy protection. Singh et al [18]. proposed a micro-aggregation-based poisoning detection method to reduce false positives for minority-group updates by collaboratively clustering model updates and identifying abnormal patterns. However, this approach is highly sensitive to parameter selection and fails to adequately mitigate the computational and communication overhead inherent in distributed environments. Lyu et al [19]. developed a distributed machine learning framework that jointly considers fairness and privacy preservation. Their method filters and shares only non-sensitive data globally, while employing a Generative Adversarial Network (GAN) to produce synthetic data for global model training, thereby mitigating the accuracy degradation typically caused by differential privacy mechanisms. Nonetheless, the framework's performance heavily depends on the distributional consistency between synthetic and real data, which may adversely affect the model’s generalization capability.

DeepChain [20]. integrates Shamir secret sharing into model updates, requiring participants to encrypt local gradients and perform encrypted aggregation and iterative decryption. This significantly enhances model update security but introduces considerable computational overhead, which reduces system efficiency. Biscotti [21]. combines Multi-Krum, differential privacy, and Shamir secret sharing to enhance aggregation security, privacy protection, and defense against malicious participants. However, the deep integration of multiple security techniques substantially increases system architectural complexity, making large-scale deployment challenging. Kang et al [22]. proposed a blockchain-based node reputation evaluation method using a multi-weight logical model to manage reputation scores and quantify participant credibility. However,

the adaptability of this reputation mechanism under dynamic attack scenarios was not explicitly addressed.

Thennakoon et al [23]. proposed a decentralized defense framework with off-chain malicious model detection to reduce the accuracy degradation caused by poisoned model uploads from compromised clients. Although this framework improves robustness, its reliance on communication latency and synchronization mechanisms limits real-time performance and scalability. Furthermore, its defensive capability remains inadequate against complex attacks such as collusive poisoning and iterative (round-by-round) evolving attacks. Mehta et al [24]. investigated model poisoning in blockchain-based federated learning and proposed an adaptive weighted aggregation strategy that uses public blockchain information to compute a boosting factor and reduce the impact of single or small-scale malicious participants. However, the framework lacks explicit defense mechanisms against Sybil attacks, rendering it less effective in large-scale scenarios involving multiple forged identities.

Existing studies indicate that integrating blockchain-based federated learning with cryptographic mechanisms can enhance system security to a certain extent, it often incurs substantial computational and communication overhead, and the resulting complex encryption and decryption operations further reduce training efficiency. Differential privacy also introduces noise that inevitably degrades model accuracy, while the strong dependence of existing defense mechanisms on parameter settings increases the difficulty of real-world deployment. To address these challenges, this work proposes FedQFS, a blockchain-based decentralized FL security framework, which integrates model quality auditing, fairness deviation detection, and a Sybil-resistant similarity evaluation mechanism. By ensuring model accuracy while enhancing security and robustness, FedQFS provides a reliable and equitable solution for secure federated learning.

3 The proposed FedQFS framework

3.1 Overall Framework

The proposed FedQFS framework is shown in **Fig. 1**. At the initialization stage of the training task, the proposed framework allows each training node to autonomously declare its group affiliation, either as part of the majority or minority group. It should be noted that self-declared group affiliation is introduced as a simplified design assumption for consortium-style federated learning, where participants are enrolled through a partially trusted registration process. In stronger adversarial environments, malicious nodes may misreport group identities to obtain additional deviation tolerance. Therefore, the self-declared group label in the current version serves as an initialization mechanism rather than a fully trust-free fairness guarantee. More robust designs based on verifiable group attributes, reputation-assisted registration, or blockchain-backed identity certification will be considered in future work. Based on this declaration, each node is required to maintain its model deviation within the normal fluctuation range of its respective group, thereby accommodating heterogeneous data distributions while preserving aggregation stability. Meanwhile, the framework incorporates intra-group

model quality auditing and Sybil attack detection mechanisms to dynamically adjust the aggregation weights of individual nodes, ensuring both fairness and robustness in the global update process. After each aggregation round, the leader node uploads the updated global model parameters to the blockchain network. In the current implementation, the Leader serves as an auditable coordination node rather than a fully trusted centralized server. It collects local updates, performs aggregation-related computation, and publishes aggregation outcomes to the blockchain for traceable record keeping. Therefore, the blockchain layer mainly provides immutability, traceability, and auditability for the training process, rather than completely eliminating coordination logic. Since this design still retains a certain degree of coordination centrality, a more decentralized aggregation mechanism will be investigated in future work. Participating nodes then retrain their local models based on the latest global model, iteratively repeating this process until the system meets the predefined convergence condition. Accordingly, strengthening the framework against more adversarial coordination scenarios remains an important direction for future work.

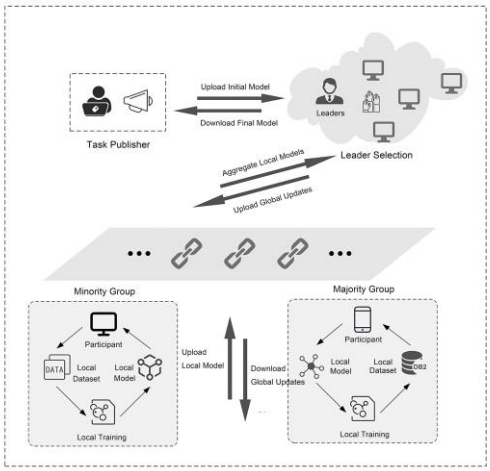


Fig. 1. FedQFS overall framework.

3.2 Framework Concept and Model

The core concept of the proposed FedQFS framework is to achieve a comprehensive audit and quantitative assessment of local model quality through an inter-participant evaluation mechanism after each participant uploads its local model updates. Specifically, each participant uses its local dataset to conduct loss testing on the models submitted by other participants in the previous round. The resulting errors are fed back to the Leader as model quality metrics for determining aggregation weights in the subsequent global update. Unlike conventional FedAvg, which determines aggregation weights mainly based on local dataset size and individual training loss, FedQFS integrates three factors: cross-participant error evaluations, cosine similarity between each local model and the previous global model, and local dataset scale. By combining these

factors, FedQFS provides a more comprehensive measure of model quality. Furthermore, Before formal weight allocation, the Leader further calculates the deviation degree of each local model to distinguish benign gradients from malicious ones, thereby reducing the impact of abnormal updates on global aggregation and improving attack resistance. For clarity, the notation used in this section is unified as follows. The local model trained by participant P_i at round t is denoted by $M_i^{(t)}$, while the global model aggregated by the Leader at round t is denoted by $GlobalModel_t$. Accordingly, the previous-round global model is denoted by $GlobalModel_{t-1}$. In addition, T denotes the total number of communication rounds, whereas τ denotes the deviation threshold used to filter invalid local updates.

Algorithm 1 FedQFS

Input: local dataset $D = \{D_1, D_2, \dots, D_n\}$, training node set $P = \{P_1, P_2, \dots, P_n\}$, influence coefficient α, β, δ, minority group determination threshold θ_{fair}, compensation coefficient γ, deviation threshold τ

Output: global aggregation model $GlobalModel_T$

Initialize $GlobalModel_0$

each training node P_i declares group type G_i

for $t = 1$ to T **do**

  training node P_i trains a local model $M_i^{(t)}$ based on its local dataset D_i and $GlobalModel_{t-1}$

  $M^{(t)} = \{M_1^{(t)}, M_2^{(t)}, \dots, M_n^{(t)}\}$

  $\{\mathrm{Q}\left(M_i^{(t)}\right)\} \leftarrow \mathrm{Auditing}(D_i, M_t)$

  $\Gamma_i^{(t)} = \sum_{s=1}^{t}\left(M_i^{(s)} - GlobalModel_{s-1}\right)$

  $S_i = \sum_{j \neq i} Sim\left(\Gamma_i^{(t)}, \Gamma_j^{(t)}\right)$

  $BaseQM_i^{(t)}$ is computed according to Eq. (8)

  $QM_i^{(t)} = BaseQM_i^{(t)} - \gamma \cdot max\left(0, \theta_{fair} - \frac{|G_i|}{n}\right)$

  $P_{valid} = \left\{P_i \,\middle|\, QM_i^{(t)} \leq \tau\right\}$

  for each $P_i \in P_{valid}$ **do**

    $W_i \leftarrow Norm\left(\alpha \cdot \hat{Q}_i + \beta \cdot \hat{S}_i + \delta \hat{N}_i\right)$

  end for

  $GlobalModel_t = \sum W_i \cdot M_i^{(t)}$

end for

return $GlobalModel_T$

The FedQFS framework consists of two core types of participating entities: the leader node (Leader) and the participant nodes (Participants). The leader node is primarily responsible for maintaining blockchain consensus and managing the global model, while the participant nodes perform local model training using their respective datasets. Assume that the system comprises a training alliance $P = \{P_1, P_2, \dots, P_n\}$ consisting of n participant nodes. The datasets held by these nodes conform to the characteristics of

horizontal federated learning, where all nodes share an identical label space but exhibit heterogeneous data distributions across samples.

The FedQFS framework comprises two core model entities: local models and the global model. The local models are independently trained by each participant node using its private dataset. Specifically, at communication round t, participant P_i trains a local model, denoted by $M_i^{(t)}$, based on its local dataset. After local training, the participant submits $M_i^{(t)}$ to the Leader for subsequent evaluation and global aggregation.

The global model is generated by the Leader in each round through weighted aggregation of the collected valid local models. The aggregated result at round t, denoted by $GlobalModel_t$, serves as the initialization model for the next round of local training. The overall model training and aggregation process is summarized in Algorithm 1.

3.3 Model Quality Auditing

The quality auditing mechanism conducts a multi-dimensional evaluation of local models through a distributed consensus process. Its core architecture is built upon blockchain smart contracts, by leveraging the tamper-proof transaction records on the blockchain to fully document the auditing process, thereby ensuring the transparency and credibility of the evaluation results. The model quality auditing function $Auditing(D_k, M_t)$ takes two core parameters as input, where D_k representing the local dataset of training node P_k, and $M = \{M_1^{(t)}, M_2^{(t)}, \dots, M_n^{(t)}\}$ representing the set of n local models submitted by training nodes in the current round. The execution process of the model quality auditing function $Auditing(\cdot)$ can be systematically delineated into four key steps:

1. For the local dataset $D_k = \{(x_i, y_i)\}_{i=1}^N$ of training node P_k, which contains N samples, let $y_i \in \{1,2, \dots, \mathrm{C}\}$ denote the true label corresponding to each data item x_i. During the model quality auditing process, node P_k iterates through the set of local models M_t submitted in the current round and sequentially invokes each model to perform inference testing on all data items x_i within its local dataset D_k. The calculation equation is as follows:

$$\hat{y}_i^j = Inference(M_j^{(t)}, x_i) \tag{1}$$

   where $Inference(\cdot)$ denotes the model inference function applied to the test data, and $\hat{y}_i^j$ represents the predicted probability distribution of sample x_i produced by the local model $M_j^{(t)}$. This full-dataset inference evaluates the generalization capability of each local model and reduces evaluation bias caused by sample selection.

2. Training node P_k computes the loss value $L_k^{(j)}$ of each local model using a multi-class cross-entropy loss function, based on the ground-truth labels y_i and the predicted results $\hat{y}_i^j$ obtained in the previous step. This loss function effectively quantifies the deviation between the model's predicted outputs and the actual labels, the calculation is expressed as follows:

$$L_k^{(j)} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} I\,(y_i = c) log(\hat{y}_i^j, c) \tag{2}$$

where C denotes the total number of label categories for y_i, and $I(\cdot)$ represents the indicator function, which equals 1 when the condition inside the parentheses is satisfied and 0 otherwise. $\left(\hat{y}_i^j, c\right)$ refers to the c-th component of the predicted probability distribution. This loss computation effectively transforms the performance of the local model on its local dataset into a quantifiable metric, thereby providing fundamental data support for subsequent model quality evaluation.

3. After computing the loss vector $L_k = [L_k^1, L_k^2, \ldots, L_k^n]$ for all local models, the training node P_k encapsulates this loss vector into a blockchain transaction T_k, appends a digital signature, and broadcasts it to the network. The corresponding computation is given as follows:

$$T_k = Sign_k(\langle AUDIT, t, k, L_k \rangle) \tag{3}$$

where $Sign_k(\cdot)$ denotes the digital signature function of node P_k, and t represents the unique identifier of the current training round, used to prevent cross-round data confusion. After the transaction is generated, the on-chain smart contract automatically verifies its signature, and only transactions that pass signature verification are recorded on the blockchain, thereby preventing malicious nodes from tampering with the model quality auditing results.

4. Upon collecting the audit transactions $\{T_1, T_2, \ldots, T_n\}$ submitted by the training nodes, the leader node parses the transaction data to extract the loss vectors of each local model. Subsequently, the leader node computes the comprehensive quality evaluation for each local model according to equation (4).

$$Q\left(M_j^{(t)}\right) = \frac{1}{n}\sum_{k=1}^{n} L_k^j \tag{4}$$

The comprehensive evaluation result not only reflects the predictive accuracy of each local model but also integrates multidimensional factors such as data representativeness. It serves as the basis for weight allocation during subsequent global model aggregation and provides key threshold references for detecting anomalous node deviations, thereby effectively ensuring the convergence and stability of the federated learning global model.

3.4 Sybil Node Identification Scheme Based on Model Parameter Similarity

The model quality evaluation mechanism can effectively mitigate the interference of malicious models in the training process. When a training node submits a malicious local update, it typically exhibits significantly elevated aggregate error during the model quality audit, resulting in a substantial reduction of its weight during global aggregation. However, this mechanism has notable limitations in defending against Sybil attacks: an adversary can submit coordinated erroneous updates through multiple fabricated node identities, making it difficult for a quality-assessment-based approach alone to detect such malicious behavior, as each fabricated node's local model may still present a seemingly ‘normal’ quality evaluation.

To address this issue, this paper proposes a Sybil node detection scheme based on model parameter similarity. In order to achieve their attack objectives, Sybil nodes inevitably submit local models with highly consistent parameters across the fabricated identities, whereas honest nodes, due to the inherent heterogeneity of their local data distributions, naturally produce diverse model updates. Therefore, by quantifying the parameter similarity among local models, an effective framework for detecting Sybil attacks can be established. The specific implementation process is as follows:

1. In each iteration of federated learning, the leader node first collects the model update gradients submitted by all training nodes. Let the local model of training node P_i in the t-th round be $M_i^{(t)}$, and let the global model from the previous round be $GlobalModel_{t-1}$. The cumulative historical gradient $\Gamma_i^{(t)}$ is obtained by summing the update gradients over rounds, as shown in equation (5).

$$\Gamma_i^{(t)} = \sum_{s=1}^{t}\left(M_i^{(s)} - GlobalModel_{s-1}\right) \tag{5}$$

2. After obtaining the historical gradients $\Gamma_i^{(t)}$, a similarity matrix $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ can be constructed based on these gradients, as shown in equation (6):

$$S_{ij} = Sim\left(\Gamma_i^{(t)}, \Gamma_j^{(t)}\right) = \begin{cases} \dfrac{\Gamma_i^{(t)} \cdot \Gamma_j^{(t)}}{\left\|\Gamma_i^{(t)}\right\|_2 \cdot \left\|\Gamma_j^{(t)}\right\|_2}, & i \neq j \\ 0, & i = j \end{cases} \tag{6}$$

where $\Gamma_i^{(t)}$ and $\Gamma_j^{(t)}$denote the historical gradients of training nodes P_i and P_j, respectively. Each matrix element S_{ij} is defined as the cosine similarity between the historical gradients of nodes P_i and P_j, with a value range of $[-1,1]$. A value of S_{ij} close to 1 indicates that the gradient directions of the two nodes are highly aligned, whereas a value close to -1 indicates opposing gradient directions. To eliminate the influence of a node on its own similarity calculation, the diagonal elements of the similarity matrix S_{ij} are set to zero when $i = j$.

3. Based on the constructed similarity matrix, the total similarity S_i for each node can be further computed. Due to the inherent heterogeneity of local data distributions, honest nodes typically exhibit a relatively uniform distribution of S_i values within a reasonable range. In contrast, Sybil nodes, which submit models highly similar to multiple forged identities, tend to have significantly higher S_i values compared to honest nodes. According to this criterion, any node whose S exceeds a predefined threshold R is flagged as a suspicious node. During subsequent global model aggregation, suspicious nodes are assigned lower aggregation weights, thereby effectively mitigating their potential influence on the global model.

3.5 Deviation Quantification Mechanism and Model Aggregation

In federated learning, honest nodes should not be excluded due to the inherent characteristics of their local data, and all legitimate model updates should have an equitable opportunity to contribute. However, there is an inherent conflict between the inclusion

of data diversity and resilience against poisoning attacks. If aggregation weights rely solely on model quality auditing results, genuine variations from non-IID or minority-group clients may be misidentified as anomalous updates or even malicious behavior. Such misjudgment may discriminate against minority participants and induce decision bias in the global model by excluding critical features, thereby adversely affecting the overall performance and fairness of the model.

To resolve this conflict, this paper proposes a fairness-aware deviation quantification mechanism, which constructs a comprehensive deviation metric based on model quality auditing results and group-specific feature adjustments.

In the fairness-aware deviation quantification mechanism, the first step is to quantify the deviation of a local model $M_i^{(t)}$, which is represented by the difference between the model quality auditing result $Q\left(M_i^{(t)}\right)$ and the average quality assessment score. The computation of the average quality assessment score is shown in equation (7).

$$AvgQ = \frac{1}{n}\sum_{i=1}^{n} Q\left(M_i^{(t)}\right) \tag{7}$$

After obtaining the difference value, the basic deviation $BaseQM_i^{(t)}$ of the local model $M_i^{(t)}$ is quantified based on this difference. During the computation of the basic deviation, the differences corresponding to minority groups are assigned lower weights to ensure fairness. The corresponding calculation equation is shown as follows:

$$BaseQM_i^{(t)} = \begin{cases} \omega \cdot \left|Q\left(M_i^{(t)}\right) - AvgQ\right|, & if\ G_i\ is\ minority \\ \left|Q\left(M_i^{(t)}\right) - AvgQ\right|, & if\ G_i\ is\ majority \end{cases} \tag{8}$$

where $0 < \omega < 1$, and $AvgQ$ denotes the average quality assessment score. When the local model $M_i^{(t)}$ originates from a minority group, its loss deviation contribution to the overall deviation is appropriately reduced.

To further safeguard the fairness of minority groups, the final deviation score $QM_i^{(t)}$ of the local model $M_i^{(t)}$ is computed by incorporating a group-specific correction term into the baseline deviation. The corresponding equation is shown as follows:

$$QM_i^{(t)} = BaseQM_i^{(t)} - \gamma \cdot max\left(0, \theta_{fair} - \frac{|G_i|}{n}\right) \tag{9}$$

where θ_{fair} denotes the threshold for identifying minority groups, $max(\cdot)$ represents the maximum operator, and γ is the fairness compensation coefficient. When the proportion of nodes belonging to a given group, denoted as $\frac{|G_i|}{n}$, falls below the threshold θ_{fair}, the fairness compensation term becomes effective and provides additional deviation tolerance for minority-group participants.

After computing the final deviation, any local model whose deviation exceeds the threshold τ is identified as an invalid update submitted by a malicious node and subsequently removed. The remaining local models proceed to the global aggregation stage. In the proposed framework, the aggregation algorithm jointly incorporates the comprehensive quality evaluation score $Q\left(M_i^{(t)}\right)$, the cosine similarity sum S_i, and the sample

size N_i of each training node. Based on these factors, the final aggregation weight assigned to each local model is determined according to the following equation:

$$W_i = \frac{\alpha \cdot \hat{Q}_i + \beta \cdot \hat{S}_i + \delta \hat{N}_i}{\sum_{i=1}^{m} (\alpha \cdot \hat{Q}_i + \beta \cdot \hat{S}_i + \delta \hat{N}_i)} \tag{10}$$

where $\alpha, \beta, \delta > 0$ denotes the multiplicative influence coefficients, with $\alpha + \beta + \delta = 1$ to ensure balanced weight distribution. m represents the total number of valid nodes participating in the aggregation process.

As shown in equations (11) and (12), $\hat{Q}_i$ and $\hat{N}_i$ represent the normalized values of the comprehensive quality assessment result $Q\left(M_i^{(t)}\right)$ and the training node's sample size N_i, respectively. $\hat{Q}_i$ is obtained through the inverse normalization function $Norm_{rev}(\cdot)$, while $\hat{N}_i$ is obtained through the forward normalization function $Norm(\cdot)$. Furthermore, $\hat{S}_i$ is defined by a piecewise function so that suspicious nodes receive lower effective aggregation weights. Here, R_{max} denotes the upper bound used in the similarity normalization.

$$\begin{cases} \hat{Q}_i = Norm_{rev}[Q\left(M_i^{(t)}\right); Q_{min}, Q_{max}] \\ \hat{N}_i = Norm(N_i; N_{min}, N_{max}) \end{cases} \tag{11}$$

$$\hat{S}_i = \begin{cases} \frac{S_i - S_{min}}{R_{max} - S_{min}}, & if\ node\ P_i\ is\ normal \\ \frac{2R_{max} - S_i}{R_{max} - S_{min}}, & if\ node\ \ P_i\ is\ suspicious \end{cases} \tag{12}$$

4 Experimental Results

4.1 Experimental Environment

All simulation experiments in this work were conducted under a uniform hardware configuration and software environment. The experimental testbed was built on Ubuntu 20.04 LTS. The blockchain module was implemented using Hyperledger Fabric, with smart contracts developed in Go. Hyperledger Fabric v1.4.1 was selected mainly for its compatibility with the existing development environment and the stability of the smart-contract toolchain used in our testbed. Therefore, this study focuses on framework design and robustness verification under the current prototype setting, rather than a full system-level benchmark covering cross-version portability, communication cost, on-chain latency, and scalability overhead. In the current implementation, the blockchain layer records audit-related results and aggregation outcomes for traceability, while a more comprehensive deployment-oriented evaluation is left for future work. The federated learning module was implemented in Python.

4.2 Experimental Dataset

This work employs the MNIST dataset to evaluate and validate the model training process. As a classical handwritten digit benchmark in machine learning, MNIST contains

60000 training samples and 10000 test samples. Each sample is a 28×28 grayscale image with a digit label ranging from 0 to 9. Owing to its reliability and wide adoption, MNIST is commonly used for assessing algorithmic performance.

As shown in **Fig. 2**, the CNN used for MNIST classification consists of two 5×5 convolutional layers and two fully connected layers, followed by Softmax logistic regression for multi-class classification.

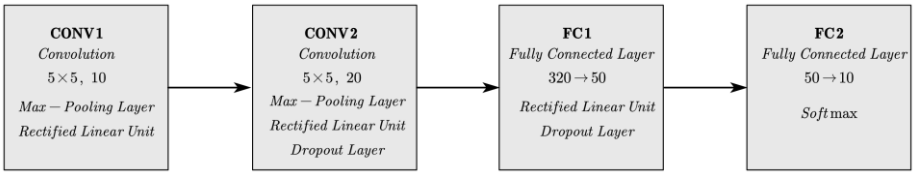


Fig. 2. Illustration of the CNN Architecture on the MNIST Dataset.

The dataset is split into 70% for training and 30% for testing. In addition, a series of federated learning experiments are conducted on the FATE platform. The experiments assume that the data follow an independent and identically distributed (IID) setting. The detailed experimental configuration is as follows: the number of participating nodes is 10, the global learning rate is set to 0.01, the batch size is 32, the number of local training epochs is 10, and the total number of global communication rounds is 200.

4.3 Experimental Attack Settings

To evaluate the robustness of the proposed framework under various attack scenarios, the experimental attack settings are summarized in **Table 1**. In the FL attack paradigm, data poisoning is one of the most prevalent attack vectors. Such attacks can be specifically categorized into targeted attacks (TA) and untargeted attacks (UA), both of which fundamentally rely on the label flipping (LF) operation as their core mechanism.

Table 1. Attack Scenario Settings

Attack Scenario	Scenario Description
TA^n	n malicious nodes perform the same label-replacement attack.
$TA_\alpha \times TA_\beta$	Malicious nodes perform heterogeneous label-replacement attacks.
UA^n	n malicious nodes perform a untargeted attack.
NA	No attack is performed.

In this work, targeted attacks, also referred to as backdoor attacks, are constructed as follows: malicious nodes uniformly flip the labels of samples originally labeled as 3 to label 8 in their local datasets and then participate in federated learning training using the modified data. The attack scenario TA^n denotes the presence of n malicious nodes in the cluster performing this label-flipping operation. To further evaluate the proposed framework's robustness against combined attacks, a composite attack scenario

$TA_\alpha \times TA_\beta$ is designed, in which two malicious nodes independently perform label-flipping operations: one node flips all samples labeled 3 to 8, while the other flips all samples labeled 1 to 4, while both nodes simultaneously participate in the training process.

Untargeted attacks represent a conventional data poisoning paradigm. In this attack, malicious nodes randomly replace the labels of all samples in their local datasets with other class labels, and then use the manipulated data to participate in the federated learning process. The attack scenario UA^n denotes the presence of n malicious nodes in the cluster executing such untargeted attacks.

4.4 Experimental Evaluation Metrics

In this experiment, the core evaluation metrics selected are the test dataset Accuracy, Loss, and Attack Rate. The definitions and calculation methods for each metric are detailed below.

1. **Accuracy**: Accuracy is used to measure the overall classification performance of the model on the test set. Since the MNIST dataset involves a multi-class classification task, the accuracy is computed as:

$$Accuracy = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} 1(\hat{y} = y_i) \tag{13}$$

where N_{test} denotes the number of test samples, y_i denotes the true label of sample i, $\hat{y}_i$ denotes the predicted label, and $\mathbf{1}(\cdot)$ is the indicator function that equals 1 when the prediction is correct, and 0 otherwise.

2. **Loss**: Loss quantifies the discrepancy between the model's predictions and the ground-truth labels. In this experiment, the training process employs the categorical cross-entropy loss function, which is computed as follows:

$$Loss = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} I(y_i = c) \log\left(\hat{y}_i^j, c\right) \tag{14}$$

where $Loss$ denotes the loss value, N is the number of samples, and C represents the total number of classes. $I(\cdot)$ is an indicator function that equals 1 when the true category of sample i matches c, and 0 otherwise. $(\hat{y}_i^j, c)$ denotes the c-th component of the predicted probability distribution for sample i. A smaller loss indicates better model fitting.

3. **Attack Rate**: Attack Rate measures the effectiveness of adversarial behaviors and serves as a key indicator for evaluating the model's security under malicious interference. In this experiment, the Attack Rate is computed as follows:

$$AttackRate = \frac{1}{M} \sum_{i=1}^{m} \frac{count(\hat{y}_i = c_\beta)}{count(y_i = c_\alpha)} \tag{15}$$

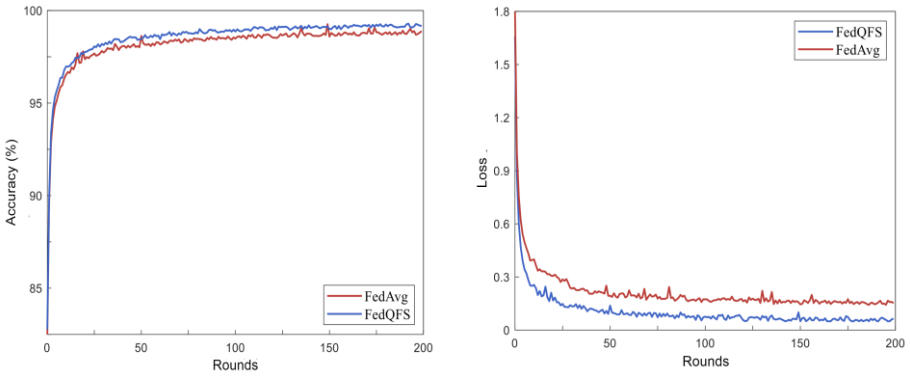
where M denotes the total number of samples subjected to label flipping, and $count(\cdot)$ is a counting function that increments by one whenever a sample i with true class c_α is misclassified by the model as the flipped label c_β. The attack rate

quantifies the extent to which malicious manipulation interferes with the model's prediction outcomes.

4.5 Experimental Results and Analysis

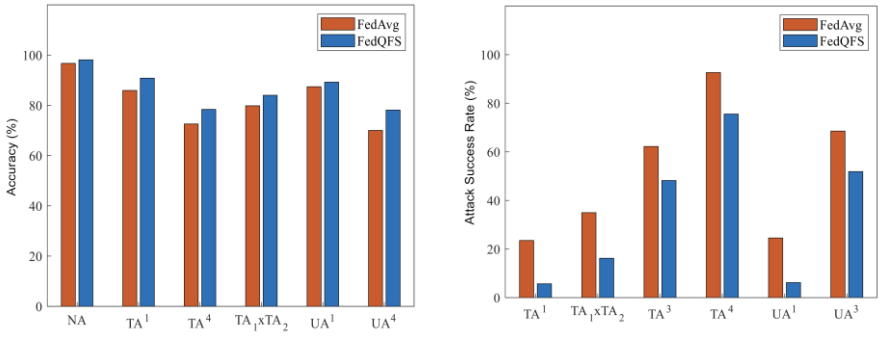
The experiments encompass comparative analyses across four dimensions, with the specific settings and corresponding results summarized as follows.

Performance Benchmarking in the Absence of Malicious Nodes: As shown in **Fig. 3**, using FedAvg as the baseline, both the proposed FedQFS framework and FedAvg achieve classification accuracies exceeding 95% in the federated learning setting without malicious node participation, demonstrating comparable performance. It is noteworthy that throughout the entire training process, the loss value of FedQFS consistently remains lower than that of FedAvg. This observation indicates that under normal training conditions, FedQFS attains superior model performance and enhanced training stability.



(a) Accuracy (b) Loss

Fig. 3. Performance Benchmarking in the Absence of Malicious Nodes.



(a) Accuracy (b) Attack Success Rate

Fig. 4. Robustness Evaluation under Multiple Attack Scenarios.

Robustness Evaluation under Multiple Attack Scenarios: As shown in **Fig. 4**, the experimental results indicate that in the label-flipping attack scenarios TA^1, $TA_1 \times TA_2$, and TA^4, the adversary misleads the model by altering the labels of training samples, forcing it to learn incorrect decision patterns. As the attack intensity increases, both FedQFS and FedAvg exhibit a consistent decline in classification accuracy, while the attack rate rises correspondingly. In particular, under attack scenario TA^4, FedAvg achieves only 72.58% accuracy, whereas its attack rate surges to 92.64%, demonstrating that the low-quality model updates originating from malicious clients exert a severely detrimental impact on the training process.

Comparative Evaluation of Anti-Collusion Performance under Sybil Attack Scenarios: To evaluate the attack resilience of different schemes under Sybil attack scenarios, in attack scenario TA^1, malicious nodes replicate their own attack datasets and participate in the federated learning process.

To enhance the comparative analysis, the similarity-based FoolsGold [25] scheme is also included in this experiment. The experimental results shown in **Fig. 5**, indicate that when the number of malicious nodes reaches two, the attack success rate against FedAvg approaches 100%, demonstrating that the malicious nodes have fully achieved their attack objectives. FoolsGold, which relies on similarity-based detection, exhibits significantly improved defense effectiveness when two or more identical malicious nodes are present. Notably, when only a single malicious node exists, the similarity among honest nodes is higher than that between the malicious node and others. Consequently, the honest nodes receive lower aggregation weights, while the weight of the malicious node is higher because it is not similar to other nodes. In contrast, the proposed FedQFS framework achieves a balanced performance under both scenarios: when only one malicious node is present, the model quality auditing mechanism prevents its gradient updates from being fully recognized by other nodes, resulting in a lower effective aggregation weight and a correspondingly lower attack rate compared to FoolsGold. When the number of malicious nodes exceeds two, the Sybil node identification mechanism activates, drastically reducing the aggregation weight of suspicious nodes to near zero, thereby enabling the system to defend against attacks with an effectiveness of approximately 99%.

Normal Gradient Compatibility Test: To better evaluate the compatibility of the deviation detection mechanism with minor updates to the normal gradient, this experiment designates a subset of training nodes on the MNIST dataset as minority groups. These nodes are assigned a core dataset dominated by samples labeled as 3, comprising 90% of the data, with the remaining 10% consisting of digits 8 and 9, which share similar features with 3, to ensure that gradient updates primarily focus on optimizing features of digit 3. Additionally, the number of local training iterations for these minority nodes is limited to simulate minor updates to the normal gradient. In this experiment, the group of minority nodes is compared with the cluster of nodes under attack scenario TA^n, where the number of minority nodes is consistently set to n. The variation in the model's accuracy on digit 3 is then observed as n increases. As shown in **Fig. 6**, the accuracy of the model trained with normally updated minority nodes is consistently higher than that obtained under attack scenario TA^n. This demonstrates that the pro-

posed FedQFS framework can effectively distinguish between malicious gradient updates and normal gradients with minor updates, thereby confirming its superior compatibility with such normal gradients.

Based on the above experimental results, the proposed FedQFS demonstrates a strong capability to withstand interference from malicious nodes and effectively reduce the negative impact of low-quality model parameters on global model aggregation. In addition, FedQFS can accurately distinguish Sybil attacks from normal nodes' minor gradient updates, thereby significantly improving the overall quality of model parameters in federated learning.

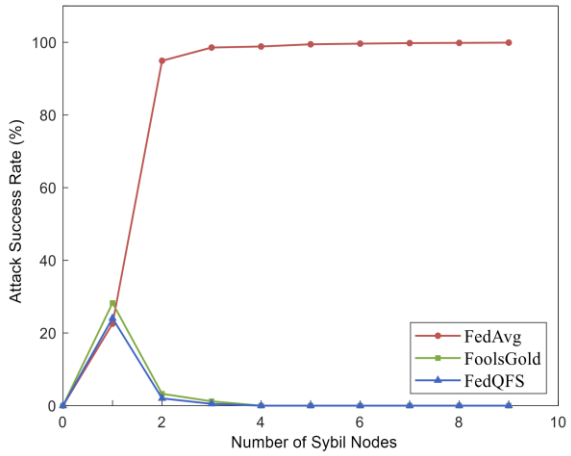


Fig. 5. Attack Rates of Different Schemes under Sybil Attack Scenarios.

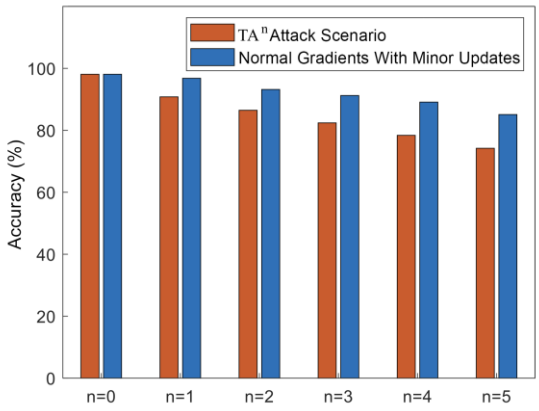


Fig. 6. Normal Gradient Compatibility Test.

5 Conclusion

This paper presents FedQFS, a blockchain-assisted federated learning framework that integrates quality auditing, fairness-aware deviation quantification, and similarity-based Sybil resistance. Rather than introducing a completely new standalone aggregation rule, the proposed framework coordinates these three mechanisms within a unified training and auditing process to improve robustness and fairness under malicious-update scenarios. Experimental results on the MNIST dataset show that FedQFS is effective under the evaluated setting in reducing the influence of malicious participants and distinguishing suspicious updates from benign minority-like deviations. Nevertheless, the current study is still limited to a relatively simplified benchmark setting, and broader validation under more challenging datasets, non-IID distributions, and larger-scale deployments remains part of our future work.

References

1. Jagatheesaperumal, S.K., Rahouti, M., Ahmad, K., Al-Fuqaha, A., Guizani, M.: The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions. IEEE Internet of Things Journal **9**(15), 12861–12885 (2022).
2. Ozsahin, D.U., Taiwo Mustapha, M., Mubarak, A.S., Said Ameen, Z., Uzun, B.: Impact of feature scaling on machine learning models for the diagnosis of diabetes. In: 2022 International Conference on Artificial Intelligence in Everything (AIE). pp. 87–94 (2022).
3. Ahmed, S., Alshater, M.M., El Ammari, A., Hammami, H.: Artificial intelligence and machine learning in finance: A bibliometric review. Research in International Business and Finance 61, 101646 (2022).
4. S M, D., N, R., K, S., Pareek, P.K.: Machine learning based education system with sentiment analysis for students. In: 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon). pp. 1–6 (2022).
5. Grosse, K., Bieringer, L., Besold, T.R., Biggio, B., Krombholz, K.: Machine learning security in industry: A quantitative survey. IEEE Transactions on Information Forensics and Security 18, 1749–1762 (2023).
6. Cui, L., Qu, Y., Xie, G., Zeng, D., Li, R., Shen, S., Yu, S.: Security and privacy-enhanced federated learning for anomaly detection in iot infrastructures. IEEE Transactions on Industrial Informatics 18(5), 3492–3500 (2022).
7. Goddard, M.: The EU general data protection regulation (GDPR): European regulation that has a global impact. International Journal of Market Research 59(6), 703–705 (2017).
8. Huang, W., Ye, M., Shi, Z., Li, H., Du, B.: Rethinking federated learning with domain shift: A prototype view. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16312–16322 (2023).
9. Zhang, T., Gao, L., He, C., Zhang, M., Krishnamachari, B., Avestimehr, A.S.: Federated learning for the internet of things: Applications, challenges, and opportunities. IEEE Internet of Things Magazine 5(1), 24–29 (2022).
10. Yuan, L., Wang, Z., Sun, L., Yu, P.S., Brinton, C.G.: Decentralized federated learning: A survey and perspective. IEEE Internet of Things Journal 11(21), 34617–34638 (2024).
11. Liu, Y., Kang, Y., Zou, T., Pu, Y., He, Y., Ye, X., Ouyang, Y., Zhang, Y.Q., Yang, Q.: Vertical federated learning: Concepts, advances, and challenges. IEEE Transactions on Knowledge and Data Engineering 36(7), 3615–3634 (2024).

12. Kumar, K.N., Mohan, C.K., Cenkeramaddi, L.R.: The impact of adversarial attacks on federated learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 46(5), 2672–2691 (2024).
13. Miao, Y., Liu, Z., Li, H., Choo, K.K.R., Deng, R.H.: Privacy-preserving byzantine-robust federated learning via blockchain systems. IEEE Transactions on Information Forensics and Security 17, 2848–2861 (2022).
14. Huang, W., Ye, M., Shi, Z., Wan, G., Li, H., Du, B., Yang, Q.: Federated learning for generalization, robustness, fairness: A survey and benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 46(12), 9387–9406 (2024).
15. Mao, Z., HuiLi, Huang, Z., Tian, Y., Zhao, X., Zhang, H.: Ghostwriting-federal learning key technology research for big data privacy protection. In: 2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI). pp. 387–390 (2022).
16. Zhang, R., Deng, X.: A byzantine-robust method for decentralized federated learning through gradient filtering and weight transformation. In: 2024 IEEE International Conference on Unmanned Systems (ICUS). pp. 1375–1380 (2024).
17. Wang, Z., Hu, Q., Zou, X., Hu, P., Cheng, X.: Can we trust the similarity measurement in federated learning? IEEE Transactions on Information Forensics and Security 20, 3758–3771 (2025).
18. Singh, A.K., Blanco-Justicia, A., Domingo-Ferrer, J.: Fair detection of poisoning attacks in federated learning on non-i.i.d. data. Data Mining and Knowledge Discovery 37(5), 1998–2023 (2023).
19. Lyu, L., Yu, J., Nandakumar, K., Li, Y., Ma, X., Jin, J., Yu, H., Ng, K.S.: Towards fair and privacy-preserving federated deep models. IEEE Transactions on Parallel and Distributed Systems 31(11), 2524–2541 (2020).
20. Weng, J., Weng, J., Zhang, J., Li, M., Zhang, Y., Luo, W.: Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Transactions on Dependable and Secure Computing 18(5), 2438–2455 (2021).
21. Shayan, M., Fung, C., Yoon, C.J.M., Beschastnikh, I.: Biscotti: A blockchain system for private and secure federated learning. IEEE Transactions on Parallel and Distributed Systems 32(7), 1513–1525 (2021).
22. Kang, J., Xiong, Z., Niyato, D., Xie, S., Zhang, J.: Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. IEEE Internet of Things Journal 6(6), 10700–10714 (2019).
23. Thennakoon, R., Wanigasundara, A., Weerasinghe, S., Seneviratne, C., Siriwardhana, Y., Liyanage, M.: Decentralized defense: Leveraging blockchain against poisoning attacks in federated learning systems. In: 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC). pp. 950–955 (2024).
24. Mehta, S., Saini, R.: A blockchain framework for federated learning: Privacy, security, and scalability. In: 2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI). vol. 3, pp. 1–5 (2025).
25. Fung, C., Yoon, C.J.M., Beschastnikh, I.: The limitations of federated learning in sybil settings. In: Proc. 23rd Int. Symp. Research in Attacks, Intrusions and Defenses (RAID). pp. 301–316 (2020).