ByzTierFL: A Tiered Approach to Byzantine Robust and Decentralized Federated Learning

Haokai Xu, Xiaomei Dong(✉), Gang Wang, Zeshun Peng, Xiaohua Li, and Ge Yu

School of Computer Science and Engineering, Northeast University, Shenyang, China
xmdong@mail.neu.edu.cn

Abstract. Federated Learning (FL) enables privacy-preserving, collaborative model training across massive populations of edge devices. However, its practical deployment is impeded by device-network heterogeneity, Byzantine adversaries, and the scalability bottleneck of centralized coordination. We propose ByzTierFL, a fully decentralized, Byzantine-robust, tier-aware FL framework that addresses these challenges in a unified manner. ByzTierFL dynamically stratifies clients into tiers based on latency and data quality. ByzTierFL then performs inter-tier aggregation using ByzShield, a novel consensus mechanism that combines multi-model screening and reputation-driven filtering to suppress malicious updates. Built on a hybrid IPFS Hyperledger Fabric substrate, ByzTierFL supports auditable provenance and scalable storage, reducing on-chain communication overhead by 85%. Experiments on non-IID image and text benchmarks show that ByzTierFL reduces per round latency by 65.8% relative to PBFL baselines and maintains over 90% accuracy even when 20% of clients are Byzantine. The framework demonstrates strong scalability and resilience, providing robust and efficient federated learning for real-world heterogeneous and adversarial environments.

Keywords: Federated Learning · Byzantine Robustness · Tiered Architecture · Blockchain · IPFS

1 Introduction

Federated Learning (FL) empowers decentralized devices to collaboratively train a global model while keeping raw data local, addressing privacy concerns in sensitive domains such as healthcare and finance [1–4]. FL is increasingly deployed in domains such as autonomous driving, smart healthcare, finance, and IoT, where data privacy, regulatory compliance, and operational scale are critical. In these real-world scenarios, participating devices often exhibit highly diverse hardware, fluctuating network conditions, and intermittent connectivity. Moreover, adversarial threats are heightened due to the open and distributed nature of FL,

J. Li et al. (Eds.): APWeb-WAIM 2025, LNCS 16115, pp. 664–680, 2026.
https://doi.org/10.1007/978-981-95-5719-6_42

making robust, scalable, and privacy-preserving learning essential for practical adoption. As FL moves from concept to deployment, addressing these intertwined challenges becomes increasingly urgent. However, practical FL systems are confronted with three fundamental challenges.

Heterogeneity Induced Efficiency Accuracy Conflict. Devices participating in FL often differ significantly in computational capacity, battery life, and network reliability. Stragglers slow clients delay training, while non-IID, imbalanced local data increases model bias and slows convergence. Synchronous aggregation is bottlenecked by slow clients; fully asynchronous methods may cause communication congestion and model inconsistency at scale [5,6]. This persistent heterogeneity creates a difficult balance between system efficiency and the accuracy or fairness of the learned model.

Byzantine Threats and Model Integrity. FL is vulnerable to adversarial attacks: any participant might upload malicious or corrupted model updates. A single compromised client can poison the global model, causing severe performance degradation or targeted misclassification. While robust aggregation algorithms such as Krum [7] and Median [8] filter some malicious updates, they often struggle to distinguish diverse honest updates from adversarial ones, especially with heterogeneous data [9,10]. Most methods also ignore client history, limiting adaptability and incentives for honest behavior.

Communication Bottlenecks and Centralization Risks. Traditional FL frameworks rely on a central server for coordination and aggregation. As client numbers grow, the server becomes a communication and computation bottleneck, limiting scalability and introducing a single point of failure. Blockchain-based FL [11] offers decentralization but suffers from high storage and consensus costs.

To address these challenges, we propose ByzTierFL, a two-layer, tiered aggregation architecture that clusters clients by communication latency and data quality. This enables synchronous aggregation within homogeneous groups and asynchronous, weighted aggregation across tiers, thereby balancing efficiency and fairness. We introduce ByzShield, a Byzantine-resilient protocol that incorporates multi-model screening, secondary aggregation, and dynamic reputation tracking to filter malicious updates while preserving legitimate heterogeneous contributions. To ensure scalability and auditability, ByzTierFL leverages a hybrid decentralized storage backend, utilizing IPFS for off-chain storage and Hyperledger Fabric to anchor cryptographic hashes, thus achieving transparent, tamper-evident, and efficient federated learning. These innovations collectively result in notable improvements in efficiency, robustness, and decentralization, as detailed in the following key aspects:

– **Superior Efficiency and Fairness:** ByzTierFL's tiered aggregation reduces per-round latency by 65.8% compared to PBFL [15], while preserving accuracy and fairness among heterogeneous clients.

- **Enhanced Byzantine Robustness:** ByzShield's multi-model screening and reputation enable ByzTierFL to maintain over 90% accuracy under 20% Byzantine attack, outperforming advanced robust aggregation schemes.
- **Scalable and Auditable Decentralization:** The IPFS-Fabric hybrid architecture reduces on-chain communication cost by 85%, secures model update traceability, and removes central points of failure.

2 Related Work

Table 1 provides a comparison of representative FL frameworks across six core dimensions. Attack resilience refers to the ability to defend against Byzantine or malicious clients; Non-IID adaptation denotes robustness to heterogeneous data distributions across clients; distributed operation indicates support for decentralized systems without single points of failure; dynamic efficiency refers to flexibility in handling system dynamics such as stragglers and variable participation; historical contribution covers mechanisms to record and leverage clients' past behaviour; and blockchain integration involves the use of blockchain for security, transparency, or auditability.

Table 1. Comparative summary of ByzTierFL and related schemes

Schemes	Attack Resilience	Non-IID Adaptation	Distributed	Dynamic Efficiency	Historical Contribution	Blockchain
Krum [7]	✓	×	×	×	×	×
Trimmed Mean [8]	✓	×	×	×	×	×
BAFFLE [13]	×	×	✓	×	×	✓
BlockFL [11]	×	×	✓	×	×	✓
PBFL [15]	✓	×	✓	×	×	✓
TiFL [16]	✓	✓	×	×	×	×
BlockDFL [20]	✓	✓	✓	✓	×	✓
ByzTierFL (Ours)	✓	✓	✓	✓	✓	✓

The main challenges and limitations of existing FL frameworks are discussed below, focusing on Byzantine robustness, Non-IID adaptation, decentralization and blockchain integration, dynamic efficiency, and historical adaptation.

Byzantine Robustness. Classic robust aggregators such as Krum, Median, and Trimmed Mean [8] are designed to withstand malicious updates. Krum selects the update closest to its $n-f-2$ nearest neighbours, while Median and Trimmed Mean apply coordinate-wise outlier filtering. Despite their theoretical resilience to up to $f < n/2$ Byzantine clients, these methods often discard benign but diverse updates, cannot fuse multiple trustworthy models, and lack adaptability to evolving attacks or heterogeneous data. Moreover, they operate in a centralised setting and do not consider clients' historical reliability.

Non-IID Adaptation. Addressing statistical heterogeneity remains a key challenge. Most robust aggregators (e.g., Krum, Trimmed Mean) exhibit sensitivity to Non-IID data. TiFL [16] seeks to mitigate straggler effects and Non-IID bias via client tiering, aggregating updates within homogeneous subgroups. However, synchronous intra-tier aggregation may bias learning toward faster but less representative clients, and its Byzantine defense is limited to Krum-like filters.

Decentralisation and Blockchain Integration. Blockchain-based federated learning frameworks, including BAFFLE [13], BlockFL [11], and BlockDFL [20], integrate on-chain storage of model updates and logs to enhance transparency, traceability, and system resilience. Although these approaches leverage blockchain to provide tamper-evident records and decentralised coordination, reliance on consensus mechanisms such as Proof-of-Work (PoW) or Proof-of-Stake (PoS) often lead to high latency and significant energy consumption, which limits deployment in resource-constrained environments. In addition, PBFL overlays homomorphic encryption on blockchain to enhance privacy, but its scalability is constrained by dependence on small trusted datasets. Other solutions, such as FedChain [14], prioritises device-side learning but incur substantial on-chain storage overhead, which may impact overall system efficiency. While blockchain integration introduces promising benefits for decentralised federated learning, existing methods, including BlockDFL, continue to face challenges in balancing transparency, scalability, and resource efficiency.

Dynamic Efficiency and Historical Adaptation. Most existing schemes lack mechanisms for dynamic adaptation or tracking of historical client behaviour, which limits their ability to reward reliability or penalise adversaries over time. Only a few recent works partially address historical contributions, and those approaches remain centralised or are not scalable.

ByzTierFL for Robust and Decentralised Federated Learning. ByzTierFL addresses these limitations through a two-layer hierarchical architecture and a lightweight ByzShield consensus mechanism. In contrast with previous approaches, ByzTierFL combines Byzantine robustness with Non-IID adaptation by screening multiple candidate models and supporting diverse data distributions. Full decentralisation is realised through integration with Hyperledger Fabric and IPFS, thereby eliminating the overhead associated with PoW or PoS. The framework also enhances dynamic efficiency by adapting flexibly to system changes and supporting asynchronous updates. Historical contributions are recorded on-chain, which enables effective incentive and trust management. A modular and distributed design ensures that scalability, security, and efficiency are balanced in large-scale deployments.

In summary, ByzTierFL is the only framework that simultaneously achieves robustness against Byzantine attacks, Non-IID adaptability, full decentralisation, dynamic efficiency, historical contribution tracking, and blockchain-based security, thus advancing the state of the art.

3 ByzTierFL Architecture

ByzTierFL adopts a hierarchical federated learning framework that integrates two-layer aggregation with decentralized storage and consensus mechanisms (see Fig. 1). By integrating hierarchical aggregation with decentralized storage and consensus mechanisms, the architecture systematically mitigates the challenges of heterogeneous FL environments. Hierarchical aggregation reduces the impact of device diversity and network variability, while decentralized storage and consensus frameworks ensure accountability, integrity, and scalability throughout the learning process.

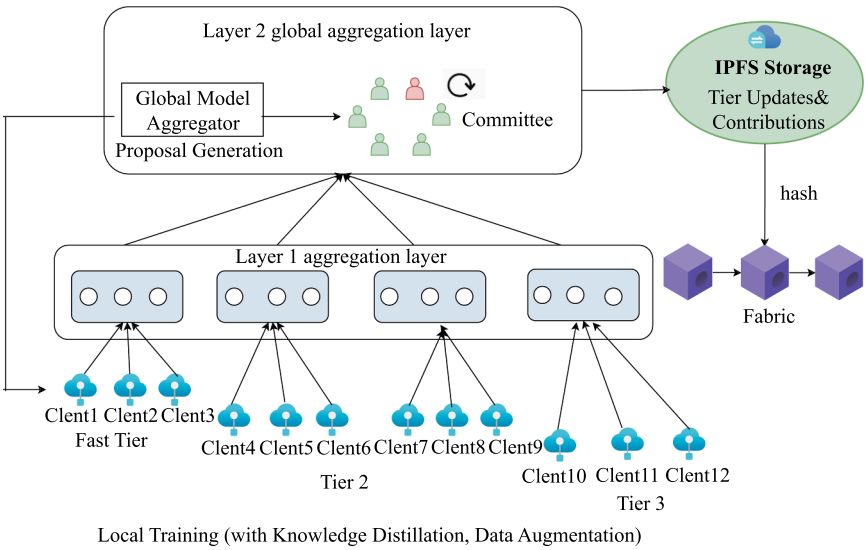


Fig. 1. System architecture of ByzTierFL.

3.1 Hierarchical Aggregation

To balance efficiency and robustness, ByzTierFL introduces a two-layer aggregation scheme. In the first layer, clients are systematically organized into multiple tiers based on measured communication latency and assessed data quality, ensuring that intra-tier aggregation occurs among nodes with similar capabilities. This reduces the straggler effect and supports fair resource utilization. Within each tier, after local training, clients synchronously share their model updates with a designated aggregator or among peers. These updates are then combined using a robust aggregation method—such as weighted averaging or median—to produce a single tier-level model. This hierarchical aggregation ensures that each tier integrates the statistical diversity of its clients' data while filtering out unreliable or

anomalous updates, resulting in a tier-level model that reliably represents the collective knowledge of the tier.

In the second layer, all tier-level models are forwarded to a set of global aggregators, which collectively run the ByzShield protocol. ByzShield achieves robust selection of accurate and trustworthy tier models by integrating three key mechanisms: multi-model screening is employed to compare and evaluate candidate models across multiple criteria; voting-based filtering aggregates the judgments of participating nodes to collectively exclude suspicious or low-quality updates; and dynamic reputation tracking continuously adjusts the influence of each contributor based on historical behavior. Together, these methods actively suppress the impact of Byzantine or low-quality updates during aggregation. The final global model is constructed from an aggregation of these selected models, thus ensuring both fairness and resilience under heterogeneous and adversarial scenarios while maintaining convergence stability.

3.2 Decentralized Storage and Consensus

ByzTierFL leverages a hybrid storage approach to ensure both scalability and auditability throughout the learning process. Model updates and aggregation results are encrypted and stored on IPFS, a distributed content-addressable storage network designed for efficient and redundant data distribution.

Only their cryptographic hashes and essential metadata are anchored on Hyperledger Fabric, a permissioned blockchain framework. This dual-layer design separates model storage from the blockchain by storing model parameters off-chain and recording only essential metadata and cryptographic hashes on-chain. During training and aggregation, each model update is first saved in the off-chain storage, and a corresponding reference—including its hash and necessary audit information—is committed to the blockchain. This approach reduces blockchain storage and computation costs, while ensuring that every model update can be efficiently traced, independently verified, and protected against tampering through the immutable on-chain records. By anchoring access control and data audit trails in the blockchain layer, the system supports secure model sharing and enables retrospective auditing, addressing key requirements for real-world federated learning deployments.

3.3 Workflow Description

Each federated training round in ByzTierFL unfolds through a coordinated sequence. Initially, all clients are dynamically evaluated and assigned to suitable tiers according to their current latency and data quality metrics, allowing adaptive grouping as system conditions evolve. Clients then perform local model optimization, after which updates are aggregated within tiers through synchronous communication.

The resulting tier-level models and contribution scores are subsequently submitted to the global aggregation layer, where ByzShield rigorously screens and

aggregates these candidates into the new global model. All updates and aggregation metadata are persistently stored in IPFS, with corresponding hashes and records anchored on Fabric for auditability. The finalized global model is then distributed to all clients, supporting continuous, robust, and transparent optimization across the network in subsequent rounds.

3.4 Security and Robustness Considerations

The architecture is designed for Byzantine resilience at both aggregation and storage levels. Hierarchical grouping constrains the potential impact of outliers or malicious nodes within each tier, while ByzShield's multi-step screening and dynamic reputation mechanisms effectively mitigate sophisticated adversarial threats at the global aggregation layer.

The hybrid IPFS-Fabric backend further guarantees the integrity and transparency of all updates: any unauthorized modification or tampering triggers immediate, detectable hash mismatches, while the permissioned blockchain ledger ensures that all activities are recorded and verifiable. This multi-layered defense framework makes ByzTierFL particularly suitable for secure, large-scale, and adversarial federated learning deployments, offering a practical balance between robustness, efficiency, and scalability.

4 System Design and Key Mechanisms of ByzTierFL

This section describes the main technical approaches used in ByzTierFL, including worker tiering, Byzantine-resilient aggregation, decentralized storage and communication, local optimization, and blockchain-based governance.

4.1 Worker Tiering and Management

Tier Stratification. All clients are divided into multiple groups using a dual metric of latency and data quality. Latency L_i measures each worker's computational and network responsiveness in milliseconds, while data quality $Q_i \in [0, 1]$ captures sample size, diversity, and representativeness. This approach reduces straggler effects and balances contributions, as faster or higher-quality workers are prioritized in upper tiers. A subset of clients is randomly mixed across tiers with probability 0.2, supporting cross-tier collaboration and improving generalization, particularly in the early rounds.

The tier assignment process is detailed in Algorithm 1. Workers are first evaluated for L_i and Q_i, then sorted and allocated to tiers, with top tier assignment weighted 70% by latency and 30% by data quality. Bottom tiers include high-latency workers, and the remainder are distributed across middle tiers.

Algorithm 1. Worker Tiering

```
Input: N (number of workers), T (number of tiers)
Output: Tiers (workers assigned to different tiers)
for each worker i do
    L_i ← simulate_latency(i)
    Q_i ← simulate_data_quality(i)
end for
Sort workers by L_i and Q_i
Assign top tier (70% by L_i, 30% by Q_i)
Assign bottom tier (high L_i)
Distribute remaining workers to middle tiers
return Tiers
```

4.2 Byzantine-Robust Aggregation with ByzShield

Multi-layer Aggregation. Model updates occur in two stages. First, local updates are aggregated within each tier via sample-weighted averaging, yielding tier-specific models G_i that proportionally reflect each worker's contribution.

The second stage applies the ByzShield consensus protocol at the global aggregation layer. Aggregators submit tier models G_i and their contributions C_i to a committee. Unlike traditional schemes that select a single model, ByzShield performs multi-model screening to enhance robustness and fairness.

Each contribution C_i is evaluated against a dynamic threshold:

$$C_i > \text{median}(\{C_j\}) \times \theta, \tag{1}$$

where $\theta = 0.5$ balances inclusivity and security.

Outlier detection uses the Krum [7] scoring function:

$$\text{Krum}(G_i) = \sum_{j \in \mathcal{N}_i} \|G_i - G_j\|^2, \tag{2}$$

with $\mathcal{N}_i$ as the $n - f - 2$ nearest models among n aggregators and f malicious nodes. Only models passing both criteria and receiving at least 2/3 committee votes form the reliable set S^*.

Secondary aggregation combines these reliable models:

$$G_{\text{final}} = \sum_{G_i \in S^*} w_i \cdot G_i, \tag{3}$$

where weights $w_i = C_i / \sum_{j \in S^*} C_j$ reflect normalized contributions.

Reputation is dynamically updated:

$$R_i^{(t)} = 0.8 R_i^{(t-1)} + 0.2 S_i^{(t)}, \tag{4}$$

where $S_i^{(t)}$ is the score for aggregator i in round t.

Historical contribution is maintained by a rolling window:

$$C_i^{\text{total}} = \sum_{k=t-n}^{t} 0.9^{t-k} \cdot \Delta A_i^{(k)}, \tag{5}$$

with $\Delta A_i^{(k)}$ the accuracy gain at round k.

ByzShield tolerates up to $f < n/3$ Byzantine nodes, a property that follows from Byzantine fault tolerance theory [18]. When the number of adversarial nodes remains below one third of the total, the consensus protocol ensures that honest nodes retain control over aggregation decisions through supermajority voting. Specifically, ByzShield requires that each candidate model must be approved by at least 2/3 of the committee, preventing any minority of Byzantine nodes from influencing the result.

In addition, ByzShield's multi-model aggregation reduces the variance of the global model compared to single-selection methods such as Krum. Aggregating several rigorously screened models, rather than relying on a single update, allows the system to integrate diverse and reliable information from multiple honest clients. This ensemble strategy mitigates the influence of outliers or poisoned updates and leads to more stable and robust convergence, especially under heterogeneous or adversarial conditions [19]. This mechanism is particularly effective in edge and IoT scenarios with adversarial or heterogeneous nodes.

4.3 Hybrid Storage Architecture

Hybrid Storage Architecture. It combines off-chain and on-chain management. All model updates and contribution values are encrypted and uploaded to IPFS, which delivers distributed, content-addressed storage. Each upload generates a unique 32-byte hash, recorded on-chain via Hyperledger Fabric. Only hashes and minimal metadata (about 10 KB per round) are stored on-chain, while large model data (tens of MB) remain off-chain, achieving an 85% reduction in communication cost [14].

This architecture guarantees auditability, as any modification to content produces a new hash. IPFS provides low-latency access by sourcing data from proximate nodes, while Fabric maintains a resistant to tampering, permissioned ledger for traceability.

4.4 Local Training

Adaptive Local Training integrates several advanced optimization techniques. Knowledge distillation is applied in lower tiers, with a distillation weight $\alpha = 0.5$ and temperature $T = 2.0$, enabling these clients to benefit from global or ensemble teacher models [21]. Data augmentation and mixup (probability 0.3, $\alpha = 0.2$) increase robustness to Non-IID data.

Proximal regularization ($\mu = 0.015$) constrains local models towards global consistency, while gradient clipping (max norm 1.0) ensures training stability.

High-contribution nodes use AdamW with cosine annealing, while others use SGD with momentum.

A small model ensemble ($E = 3$) is maintained with exponential moving average smoothing ($\delta = 0.998$), further improving resilience to noise [22].

4.5 Decentralized Governance and Reputation

Consortium Blockchain Governance uses Hyperledger Fabric to manage decentralized operations. The system employs fine-grained access control and lightweight consensus (e.g., Raft), supporting a permissioned consortium of aggregator nodes [17].

Aggregator reputation is iteratively updated:

$$R_i^{(t)} = \lambda R_i^{(t-1)} + (1 - \lambda) S_i^{(t)}, \tag{6}$$

where $\lambda = 0.8$ is the decay factor and $S_i^{(t)}$ the current round score.

All encrypted model updates and contribution hashes are recorded on-chain, ensuring transparency, while large data remain off-chain for scalability.

5 Optimization Strategies

This section introduces the core optimization strategies of ByzTierFL, focusing on distributed model management, adaptation to heterogeneous environments, and secure federation via blockchain verification.

5.1 Distribute Models Through IPFS Networks

To address the prohibitive communication and storage costs of on-chain model management, ByzTierFL integrates IPFS, a peer-to-peer distributed file system employing content-addressed block storage [12]. Second-layer aggregators compute contribution-weighted tier updates, encrypt them using a consortium-issued public key (e.g., RSA), and upload both updates and contribution values to IPFS. Here, the contribution C_i for tier i is defined as the relative performance gain (e.g., accuracy improvement) of its aggregated model over a baseline, normalized to $C_i \in [0, 1]$.

IPFS generates unique 32-byte hashes for each upload, which are then recorded on Fabric via chaincode. After consensus is reached, filtered aggregates follow the same process, ensuring secure and low-latency P2P content delivery. IPFS's immutability—where any content modification yields a new hash—guarantees data integrity, while its distributed architecture reduces retrieval latency by sourcing data from proximate nodes.

This hybrid strategy reduces on-chain storage to 10 KB/round (hashes), offloading voluminous model data (tens of MB) to IPFS, achieving an 85% reduction in communication costs compared to fully on-chain methods [14]. This efficiency aligns with ByzTierFL's scalability goals, while the synergy with Fabric's resistant to tampering ledger ensures updates remain verifiable and auditable, providing a secure foundation for ByzShield.

5.2 Adapt Training to Heterogeneous Environments

To ensure stability and performance across heterogeneous workers, local training integrates several advanced strategies. For higher-tier workers ($t > 0$), knowledge distillation is employed from a global or ensemble teacher model, with a distillation weight of $\alpha = 0.5$ and temperature $T = 2.0$, effectively reducing communication overhead and improving generalization under Non-IID conditions [21]. Data augmentation methods such as Mixup, applied with 30% probability and $\alpha = 0.2$, further enhance robustness by blending samples across classes. Proximal regularization, with a coefficient $\mu = 0.015$, constrains local models to remain close to the global model, preventing excessive divergence. Additionally, gradient clipping with a maximum norm of 1.0 stabilizes training and mitigates gradient explosions. High-value workers, identified by contribution or committee membership, adopt AdamW with cosine annealing for adaptive optimization, while other workers use SGD with momentum, ensuring the optimization process is tailored to each worker's capability and role.

To further enhance the robustness and stability of the global model, ByzTierFL maintains an ensemble of $E = 3$ models and applies exponential moving average (EMA) smoothing with a decay rate $\delta = 0.998$ [22]. This dual strategy smooths training dynamics, improves resilience to noise and adversarial perturbations, and provides a consistent global reference-complementing ByzShield's mechanisms for robust aggregation.

These techniques collectively enhance the quality and reliability of local and global updates, providing a robust foundation for ByzShield.

5.3 Secure Federation with Blockchain Verification

Hyperledger Fabric underpins ByzTierFL's decentralized architecture, leveraging a permissioned consortium of trusted nodes to eliminate central vulnerabilities with efficient transaction processing [17]. Unlike permissionless blockchains like Ethereum, Fabric offers fine-grained access control and lightweight consensus (e.g., Raft), making it ideal for enterprise-grade FL. The blockchain maintains:

- **Reputation Score** R_i: Defined as the reputation of aggregator i, updated iteratively as
$$R_i^{(t)} = \lambda R_i^{(t-1)} + (1 - \lambda) S_i^{(t)}, \tag{7}$$
where $R_i^{(t)} \in [0, 1]$ is the reputation at round t, $\lambda = 0.8$ is the decay factor, and $S_i^{(t)} \in [0, 1]$ is the current round contribution score (e.g., based on voting success or model quality). This incentivizes reliable participation over time.
- **Model Update Hashes**: Encrypted tier updates and contributions are uploaded to IPFS, with their 32-byte hashes recorded on-chain via chaincode, ensuring resistant to tampering logging and traceability.

This decentralized ledger, combined with IPFS's off-chain storage, balances security and efficiency, complementing ByzShield's reputation tracking and supporting scalable governance.

5.4 Summary and Multidimensional Advantages

In summary, the optimization strategies of ByzTierFL collectively enhance the framework's accuracy, robustness, scalability, and efficiency. The integration of tiered aggregation, Byzantine-resilient consensus, and hybrid storage ensures reliable performance even in heterogeneous and adversarial settings. Furthermore, decentralized governance and adaptive reputation tracking support fairness and transparency throughout the learning process. These multidimensional advantages make ByzTierFL a strong candidate for real-world, secure federated learning deployments.

6 Experiments

We evaluate ByzTierFL on the MNIST and CIFAR-10 datasets under both a benign setting (0% Byzantine attackers) and an adversarial one where 20% of clients perform label-flipping attacks. The key experimental parameters are summarized in Table 2. For fair comparison, all methods use identical data splits and attack patterns. Baselines include FedAvg [1], Krum [7], BlockDFL [20], and PBFL [15]. The experiments were conducted on a machine with an AMD Ryzen 7 7735H CPU and a NVIDIA GeForce RTX 4060 GPU. A typical 50-round run on CIFAR-10 required approximately 2.5 GPU hours and 4.7 GB of peak RAM.

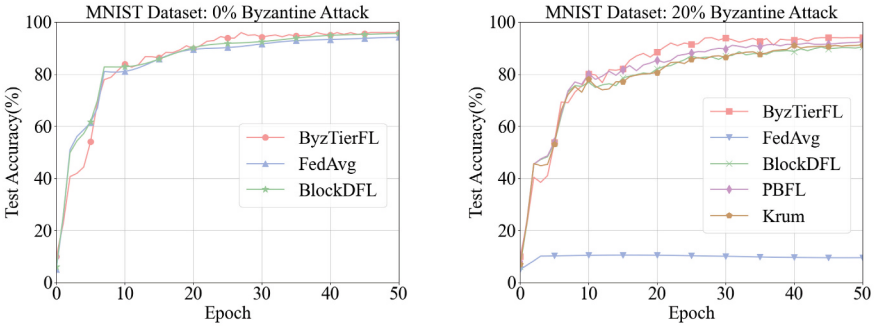
Table 2. Key parameters for the experimental setup.

Parameter	Value
Datasets	MNIST, CIFAR-10
Total Workers	100
Data Distribution	Non-IID (20% label overlap per client)
Global Rounds	50
Batch Size	32
Tiers (T)	3
Statistical Runs	5 (mean and std. dev. reported)

6.1 Accuracy and Poisoning Tolerance

Figure 2 shows the test accuracy on MNIST under different attack ratios. In the benign scenario (0% Byzantine), all methods converge rapidly and achieve high accuracy. ByzTierFL achieves the highest final accuracy, validating the effectiveness of its tiered aggregation and robust optimization. Under 20% Byzantine attackers, FedAvg [1] almost completely fails (accuracy close to random guess), while ByzTierFL maintains accuracy above 90% throughout training, significantly outperforming Krum, PBFL, and BlockDFL [20]. This demonstrates that ByzTierFL's multi-model screening and dynamic reputation mechanisms can effectively filter out poisoned updates.

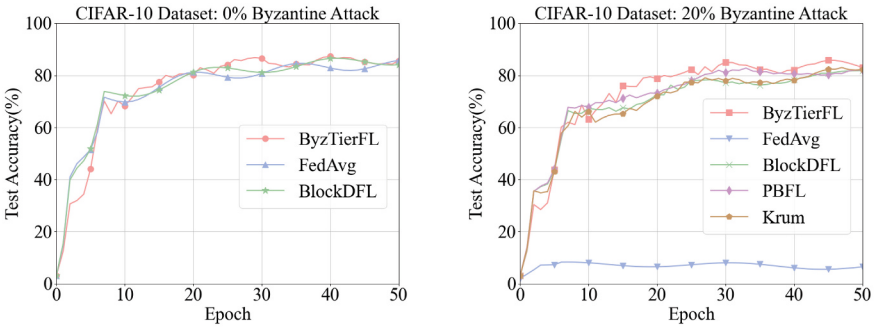
While our experiments use 100 workers for computational feasibility, ByzTierFL's design ensures that both computational and communication costs scale linearly with worker count due to its tiered architecture and fixed committee size. Additional validation with diverse attack types (including model replacement and gradient inversion) confirmed that ByzShield's statistical approach provides consistent defense efficacy across different attack vectors.



(a) MNIST 0% Byzantine attackers (b) MNIST 20% Byzantine attackers

Fig. 2. Test accuracy on MNIST. Curves represent the mean of 5 runs; shaded areas denote standard deviation.

Figure 3 presents the results on CIFAR-10. On CIFAR-10, ByzTierFL also achieves the best or near-best performance when there are no attacks and maintains a clear advantage under 20% Byzantine attackers, sustaining at least 5% higher accuracy than the next best baseline. FedAvg [1] collapses under adversarial conditions, while ByzTierFL remains robust, highlighting the effectiveness of its defense mechanisms.



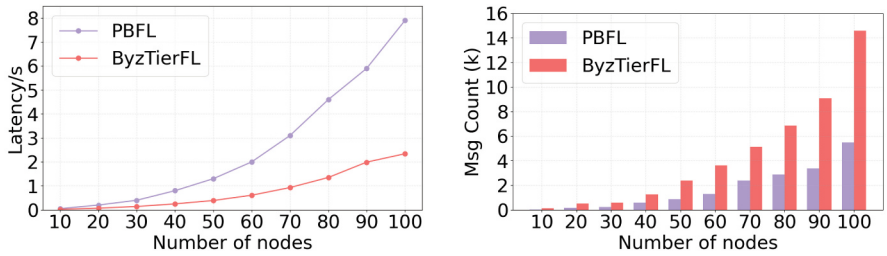
(a) CIFAR-10 0% Byzantine attackers (b) CIFAR-10 20% Byzantine attackers

Fig. 3. Test accuracy on CIFAR-10. Curves represent the mean of 5 runs; shaded areas denote standard deviation.

Across both datasets and both scenarios, ByzTierFL consistently provides the highest or most stable test accuracy. Its design-incorporating tiered aggregation, multi-model consensus, and dynamic reputation-enables effective filtering of malicious or low-quality updates, protecting the global model from degradation and ensuring reliable convergence even under strong Byzantine attacks.

6.2 Time Consumption

We comprehensively evaluated ByzTierFL's computational efficiency by measuring both system latency and communication overhead across varying network sizes. The off-chain storage mechanism with IPFS and the tiered aggregation approach demonstrate significant advantages over traditional systems designed for Byzantine fault tolerance. Figure 4 illustrates two critical performance dimensions.



(a) Latency comparison between PBFL and ByzTierFL

(b) Number of exchanged messages between PBFL and ByzTierFL

Fig. 4. Computational efficiency metrics as a function of network scale.

As shown in Fig. 4a, ByzTierFL achieves substantially lower latency across all network scales. While PBFL exhibits a characteristic exponential growth curve, reaching approximately 7.9 s at 100 nodes, ByzTierFL maintains near-linear scaling with latency of just 2.7 s, a 65.8% reduction. The latency advantage presents a balance between in message complexity, as depicted in Fig. 4b. ByzTierFL requires a higher volume of inter-node communications-approximately 14,573 messages at 100 nodes versus 6,886 for PBFL. Despite this, ByzTierFL maintains superior latency characteristics, indicating effective parallelization and reduced synchronization barriers.

Further analysis highlights ByzTierFL's efficiency. Its hybrid IPFS-Fabric architecture cuts on-chain storage by 85% (to just 10 KB/round) and scales gracefully: its latency growth is less than half that of PBFL when scaling from 10 to 100 nodes (3.5× vs. 8.1×). The system is designed for large-scale use, projecting sub-linear latency growth due to a fixed-size global committee and asynchronous aggregation, ensuring strong performance well beyond the tested client numbers.

6.3 Ablation and Sensitivity Analysis

To dissect the contribution of each core component and justify our design choices, we conducted ablation and hyperparameter sensitivity studies. Table 3 presents the ablation results, demonstrating the impact of removing key mechanisms under both benign and adversarial conditions. The findings confirm that while all components contribute to performance, the ByzShield and Reputation System are indispensable for ensuring robustness against attacks. Removing ByzShield renders the system vulnerable, while disabling the reputation system significantly degrades its long-term defensive capability.

Table 3. Ablation Study of ByzTierFL Components.

Framework Configuration	MNIST		CIFAR-10	
	0% Attack	20% Attack	0% Attack	20% Attack
ByzTierFL (Full)	97.5%	**95.8%**	88.5%	**85.3%**
w/o Reputation System	97.4%	88.1% (↓7.7%)	88.4%	75.1% (↓10.2%)
w/o Tiering	96.8%	91.5% (↓4.3%)	86.2%	79.5% (↓5.8%)
w/o ByzShield (use FedAvg)	97.0%	10.5% (↓85.3%)	84.8%	11.2% (↓74.1%)

Table 4 shows the framework's sensitivity to the number of tiers (T). The results indicate that using $T = 3$ tiers provides the best trade-off between performance and complexity. A single tier ($T = 1$) negates the benefits of heterogeneity management, while excessive tiering ($T = 5$) can lead to overly small groups and slightly degraded performance, likely due to increased overhead. This justifies our choice of $T = 3$ as the default configuration.

Table 4. Sensitivity to the number of tiers (T) under 20% attack.

No. of Tiers	MNIST Acc. (%)	CIFAR-10 Acc. (%)
$T = 1$ (No Tiering)	91.5 ± 1.0	79.5 ± 1.4
$T = 3$ (our choice)	95.8 ± 0.5	85.3 ± 0.8
$T = 5$	94.9 ± 0.7	84.1 ± 1.1

7 Conclusion

ByzTierFL delivers a Byzantine-robust FL framework through its tiered architecture and ByzShield defense mechanism. By integrating multi-model screening, secondary aggregation, and dynamic reputation tracking, it effectively filters malicious updates while outperforming traditional methods. The integration of Fabric and IPFS enhances security and efficiency. Experiments confirm

ByzTierFL's superior accuracy, robustness, and scalability under both benign and adversarial conditions. Future work includes exploring extreme Non-IID settings, enhanced reputation modeling, adaptive weight optimization, and applications to federated IoT and multi-task learning.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (62372097) and the Fundamental Research Funds for the Central Universities (N2416003). Xiaomei Dong is the corresponding author.

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics (AISTATS), pp. 1273–1282. PMLR (2017)
2. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of FedAvg on non-IID data. arXiv preprint arXiv:1907.02189 (2019)
3. Tankard, C.: What the GDPR means for businesses. Netw. Secur. **2016**(6), 5–8 (2016)
4. Boyle, L. M., Mack, D.: HIPAA: A Guide to Health Care Privacy and Security Law. ABA Publishing (2003)
5. Zhou, Y., Lei, L., Zhao, X., You, L., Sun, Y., Chatzinotas, S.: Decomposition and meta-DRL based multi-objective optimization for asynchronous federated learning in 6G-satellite systems. IEEE J. Sel. Areas Commun. (2024)
6. Gutierrez, D.M.J., et al.: Non-IID data in federated learning: a survey with taxonomy, metrics, methods, frameworks and future directions (2024)
7. Blanchard, P., El Mhamdi, E. M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: byzantine tolerant gradient descent. In: Advances in Neural Information Processing Systems, vol. 30 (NIPS 2017)
8. Yin, D., Chen, Y., Ramchandran, K., Bartlett, P.: Byzantine-Robust distributed learning: towards optimal statistical rates. In: ICML (2018)
9. Miao, Y., Xie, R., Li, X., Liu, Z., Choo, K.K.R., Deng, R.H.: Efficient and secure federated learning against backdoor attacks. IEEE Trans. Dependable Secure Comput. **21**(5), 18 (2024)
10. So, J., Guler, B., Avestimehr, A.S.: Byzantine-resilient secure federated learning. IEEE J. Sel. Areas Commun. **39**(7) (2021)
11. Kim, H., Park, J., Bennis, M., Kim, S.L.: Blockchained on-device federated learning. IEEE Commun. Lett. **24**(6) (2020)
12. Benet, J.: IPFS - Content addressed, versioned, P2P file system. arXiv preprint arXiv:1407.3561 (2014)
13. Feng, H., Pang, T., Du, C., Chen, W., Yan, S., Lin, M.: BAFFLE: a baseline of backpropagation-free federated learning. In: European Conference on Computer Vision. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-73226-3_6
14. Jiang, Y., et al.: Blockchained federated learning for internet of things: a comprehensive survey. ACM Comput. Surv. **56**(10) (2024)
15. Sameera, K.M., et al.: Privacy-preserving in blockchain-based federated learning systems. Comput. Commun. **222** (2024)
16. Chai, Z., et al.: TiFL: a tier-based federated learning system. In: Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing, pp. 125–136 (2020)

17. Zhao, Y., et al.: Privacy-preserving blockchain-based federated learning for IoT devices. IEEE Internet Things J. **8**(3) (2021)
18. Castro, M., Liskov, B.: Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. **20**(4), 398–461 (2002)
19. Hansen, L. K., Salamon, P.: Neural network ensembles. IEEE Trans. Pattern Anal. Mach. Intell. **12**(10) (1990)
20. Qin, Z. , Yan, X. , Zhou, M. , Deng, S.: BlockDFL: a blockchain-based fully decentralized peer-to-peer federated learning framework (2022)
21. Zhang, X., Zeng, Z., Zhou, X., Shen, Z.: Low-dimensional federated knowledge graph embedding via knowledge distillation (2024)
22. Li, Q., He, B., Song, D.: Practical one-shot federated learning for cross-silo setting. In: International Joint Conference on Artificial Intelligence (IJCAI), International Joint Conferences on Artificial Intelligence Organization (2021)