AC-BlockDFL: Audit-driven Committee BlockDFL for Secure Federated Learning

JAMES LU, National Taipei University of Technology, Taiwan

Blockchain-based Federated Learning (BCFL) faces a critical scalability-security trade-off. While committee-based architectures significantly reduce communication overhead, they introduce a fundamental vulnerability: the *Progressive Committee Capture Attack* (PCCA). In PCCA, rational adversaries exploit the stake-election-reward feedback loop to gradually capture committee control through strategic starvation and stake accumulation. We propose *AC-BlockDFL*, a defense framework that decouples system security from committee honesty through optimistic execution and asynchronous auditing. By internalizing the externalities of malicious behavior via a game-theoretic slashing protocol, AC-BlockDFL ensures that attacks yield negative expected utility. Our evaluation over 2,000 rounds demonstrates that AC-BlockDFL suppresses malicious stake ratios from 1.3× to 0.37×, reducing unavailability rates from 22.3% to below 1% while maintaining $O(C^2)$ communication complexity.

CCS Concepts: • **Security and privacy** → **Distributed systems security**; • **Computing methodologies** → *Artificial intelligence*.

Additional Key Words and Phrases: Federated Learning, Blockchain, Committee Consensus, Game Theory, Incentive Compatibility

ACM Reference Format:
James Lu. 2024. AC-BlockDFL: Audit-driven Committee BlockDFL for Secure Federated Learning. *J. ACM* 37, 4, Article 111 (August 2024), 17 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Blockchain-based Federated Learning (BCFL) has emerged as a promising paradigm for collaborative machine learning in environments where mutual trust among participants cannot be assumed. Real-world deployments in Low Earth Orbit (LEO) satellite networks [5, 11, 18], vehicular networks (V2X) [8, 12], and Industrial IoT [9, 14] demonstrate compelling use cases where decentralized coordination is essential. In LEO constellations, for instance, ground station contact windows last merely five minutes with downlink bandwidth limited to approximately 8 Mbps [18], rendering centralized aggregation architectures impractical. BCFL addresses these constraints by establishing decentralized trust infrastructure across heterogeneous satellite operators, reducing model convergence time by up to thirty hours [5].

However, BCFL systems face a fundamental scalability bottleneck when approaching large-scale deployment. The predominant use of Practical Byzantine Fault Tolerance (PBFT) [3] and its variants introduces $O(N^2)$ message complexity, causing consensus latency to dominate training time as participant counts grow. Empirical measurements from FLCoin [15] reveal that at 100 nodes, single-round consensus generates over 20,000 message exchanges with latency exceeding 25 seconds—comparable to or exceeding the model training duration itself. Storage requirements

Author's Contact Information: James Lu, lujames13@gmail.com, National Taipei University of Technology, Taipei, Taiwan.

ACM 1557-735X/2024/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

compound this challenge: Bitcoin full nodes require approximately 200 GB while Ethereum exceeds 465 GB, fundamentally incompatible with edge devices possessing only KB-to-MB scale memory [1].

The Committee Mechanism. To address these scalability constraints, recent work has converged on *committee-based* architectures that delegate verification responsibility to a smaller subset of validators. Selection mechanisms include hash-ring sampling [13], stake-weighted election [7, 15], and Verifiable Random Function (VRF) based sortition [6, 16, 17]. These approaches yield substantial efficiency gains: FLCoin [15] reports 90% communication overhead reduction and 5.7× training speedup through sliding-window election, while BFLC [7] achieves sub-three-second consensus latency. Effectively, committee mechanisms reduce communication complexity from $O(N^2)$ to $O(C^2)$ or even $O(C)$, where $C \ll N$ is the committee size.

The Blind Spot. Despite these advances, existing BCFL literature harbors a critical yet overlooked vulnerability: the implicit assumption that committee members remain honest or that the proportion of malicious nodes stays static throughout system operation. Current defenses focus predominantly on data-plane attacks—Byzantine-robust aggregation rules such as Krum, Trimmed Mean, and Median [2, 19] assume an honest majority among aggregating nodes. However, these mechanisms provide no protection when the committee itself becomes compromised. As FedBlock [10] observes, when any participant may become a validator, systems cannot rely solely on honest majority assumptions but must actively detect and isolate malicious verifiers—a capability conspicuously absent from current BCFL architectures.

The PCCA Threat. We identify and formalize a novel attack vector: the *Progressive Committee Capture Attack* (PCCA). Unlike direct Byzantine attacks, PCCA adversaries employ *strategic starvation*—upon gaining committee control, attackers prioritize processing their own model updates while systematically denying service to honest participants. This manipulation of the reward distribution mechanism enables attackers to "legitimately" accumulate stake over successive rounds, progressively cementing their dominance until decentralized governance collapses entirely. Crucially, once the honest majority assumption fails in any given round, existing systems lack mechanisms to identify or penalize malicious actors, allowing attackers to maintain their advantage indefinitely.

Contributions. This paper presents *Audit-driven Committee BlockDFL* (AC-BlockDFL), a defense framework that decouples system security from collective committee honesty through optimistic execution with asynchronous auditing. Our contributions are threefold:

(1) **Attack Formalization.** We provide the first formal definition of the Progressive Committee Capture Attack (PCCA) and a rational adversary model that captures strategic, incentive-driven behavior. Through systematic simulation, we quantify PCCA's destructive impact on long-term incentive compatibility.
(2) **AC-BlockDFL Architecture.** We propose a novel defense architecture combining optimistic execution with asynchronous auditing. A distributed challenger network performs post-hoc verification of committee decisions, enabling detection and penalization of fraudulent aggregation results even when the committee is fully compromised.
(3) **Game-Theoretic Guarantees.** We design an internal slashing protocol grounded in game-theoretic analysis, ensuring that auditing costs remain strictly below potential gains from malicious behavior. We prove that honest participation constitutes the unique Nash equilibrium under repeated play [4]. Extensive simulations over 2,000 rounds demonstrate that AC-BlockDFL maintains model accuracy above 98.6% under 30% adversarial collusion, reduces communication overhead by 44.4% at equivalent security levels, and suppresses minimum unavailability rate from 20% to below 5%.

2 Background & Problem Statement

This section establishes the system model and threat landscape for committee-based blockchain federated learning (BCFL). We first formalize the committee architecture using BlockDFL [13] as our reference model, then define the adversary model and the Progressive Committee Capture Attack (PCCA) that exploits stake-driven incentive mechanisms.

2.1 Committee-based Blockchain Federated Learning

Committee-based BCFL systems delegate consensus responsibility to a small, representative subset of participants rather than requiring all-node BFT consensus. This design reduces communication complexity from $O(N^2)$ to $O(C^2 + N)$ where $C \ll N$, making frequent FL iterations practical while maintaining Byzantine fault tolerance within the committee [15].

BlockDFL Architecture. We adopt BlockDFL [13] as our baseline system, representing the state-of-the-art in fully decentralized peer-to-peer federated learning. BlockDFL employs *role separation*, partitioning participants into three roles per training round: **Update Providers** perform local training on private data and submit model updates; **Aggregators** collect, filter, and aggregate updates into global proposals; and **Validators** form the committee that evaluates proposals via BFT consensus using Krum scoring [2].

The Stake-Election-Reward Cycle. Role assignment in BlockDFL follows a *stake-weighted deterministic random selection* mechanism: the previous block's hash maps onto a hash ring where each participant occupies space proportional to their stake. This creates a closed-loop incentive system:

(1) **Stake** determines election probability—higher-stake nodes are more likely to be selected as Aggregators or Validators.
(2) **Election** assigns roles based on the deterministic hash mapping, ensuring verifiable yet unpredictable outcomes.
(3) **Reward** is distributed only to contributors of the accepted proposal: the winning Aggregator, Update Providers whose updates were included, and Validators who voted in favor.

This mechanism creates a *positive feedback loop*: nodes receiving rewards accumulate stake, increasing their future election probability, which in turn yields more rewards. While designed to incentivize honest participation, this feedback property becomes the foundation for the attack we analyze.

2.2 Threat Model

Adversary Classification. We consider a *rational adversary* whose behavior is governed by economic self-interest, in contrast to the Byzantine adversary that may act arbitrarily regardless of cost. Formally:

- **Byzantine Adversary**: May exhibit arbitrary malicious behavior even at personal loss; the standard assumption in distributed systems worst-case analysis.
- **Rational Adversary**: Optimizes expected payoff; attacks only when $\mathbb{E}[\text{Payoff}] > 0$. This model better captures real-world incentive-driven threats.

The rational adversary's goals are hierarchical: immediate *economic extraction* (monopolizing training rewards), leading to *stake accumulation*, ultimately achieving *network control* (persistent committee influence).

Adversary Capabilities and Constraints. We assume the adversary controls a fraction $f \leq 0.3$ of network nodes ($M = f \cdot N$ malicious nodes). Controlled nodes can coordinate strategies and adaptively switch between honest and malicious behavior based on system state. The adversary has full visibility of on-chain information (stake distributions, election outcomes, historical behavior).

Algorithm 1 PCCA Decision Logic

```
Require: Committee $\mathcal{V}$, adversary nodes $C_{adv}$
$\rho \leftarrow |\mathcal{V} \cap C_{adv}|/|\mathcal{V}|$
if $\rho \leq 2/3$ then                                  ▷ Shadow Mode
    Follow protocol honestly; accumulate stake
else                                                     ▷ Capture Mode
    if Aggregator $\in C_{adv}$ then
        Execute full-stack poisoning
    else
        Execute strategic starvation
```

However, the adversary cannot: (1) break cryptographic primitives or forge signatures; (2) tamper with committed blockchain history; (3) prevent other nodes from independently verifying aggregation correctness. These constraints inform our defense design.

2.3 Progressive Committee Capture Attack (PCCA)

PCCA is a two-phase economic attack that exploits the stake-election-reward feedback loop to gradually subvert committee control without requiring an initial majority.

Phase 1: Lurking (Shadow Mode). During the lurking phase, adversarial nodes behave indistinguishably from honest participants: they submit high-quality model updates as Update Providers, correctly execute aggregation as Aggregators, and vote honestly as Validators. This strategy serves dual purposes: (1) accumulating stake through legitimate rewards, and (2) building reputation to evade behavioral anomaly detection. The adversary monitors committee composition each round, waiting for the critical condition: $\rho = \frac{|\mathcal{V} \cap C_{adv}|}{|\mathcal{V}|} > \frac{2}{3}$, where $\mathcal{V}$ is the current committee and C_{adv} denotes adversary-controlled nodes.

Phase 2: Capture (Starvation Mode). Once the adversary controls $> 2/3$ committee seats, the attack transitions to active exploitation via *strategic starvation*:

- **Strategic Starvation**: When the Aggregator is honest, the malicious committee systematically rejects legitimate proposals, denying rewards to honest Update Providers and Aggregators. The attack is economically devastating yet technically subtle—training continues (albeit suboptimally), making detection difficult.
- **Full-Stack Poisoning**: When the adversary also controls the Aggregator, they achieve end-to-end control. Malicious updates bypass all Byzantine-robust defenses (which are executed by the compromised committee), directly corrupting the global model while securing all rewards.

Algorithm 1 formalizes the adversary's decision logic.

The Feedback Loop: Stake Dynamics Under Attack. Let $S_{mal}(t)$ and $S_{hon}(t)$ denote the aggregate stake of malicious and honest nodes at round t, with initial adversary fraction $f_0 = S_{mal}(0)/(S_{mal}(0)+S_{hon}(0)) = 0.3$. During capture phases, the adversary gains a multiplicative advantage $\alpha > 1$ in reward acquisition. However, because honest nodes can still receive partial rewards as Update Providers (even when their proposals are rejected), stake growth is bounded rather than exponential. The dynamics converge to:

$$\lim_{t \to \infty} \frac{S_{mal}(t)}{S_{hon}(t)} = \alpha \cdot \frac{S_{mal}(0)}{S_{hon}(0)} \tag{1}$$

where $\alpha \in [1.1, 1.2]$ empirically. This establishes a *persistent leadership advantage*: while the adversary cannot achieve complete monopoly, they maintain elevated committee capture probability

Table 1. PCCA vs. Traditional Byzantine Attacks

Property	**Byzantine Attack**	**PCCA**
Target	Model quality	Network control
Motivation	Disruption	Profit maximization
Strategy	Direct poisoning	Progressive infiltration
Stealth	Low	High (honest during lurking)
Self-reinforcing	No	Yes (stake feedback)
Defense	Data-layer filtering	Incentive-compatible mechanisms

indefinitely—a “soft oligopoly” that fundamentally undermines decentralization without triggering obvious failure modes.

Distinction from Data-Layer Attacks. PCCA differs fundamentally from traditional Byzantine attacks (Table 1): it targets *governance* rather than model quality, employs *progressive infiltration* rather than immediate poisoning, and exhibits *self-reinforcing* dynamics through stake accumulation. Crucially, once the consensus layer is compromised, all data-layer defenses (Krum, Trimmed Mean, etc.) become ineffective—they are executed by the very committee the adversary controls.

3 Related Work

The intersection of blockchain and federated learning has inspired numerous architectures aimed at decentralized coordination and security. Early works such as DeepChain [17] and Biscotti [16] focused on privacy-preserving aggregation and auditability using differential privacy and cryptographic commitments. Algorand [6] demonstrated the efficiency of committee-based sortition for large-scale consensus, a principle adopted by later BCFL systems to mitigate the $O(N^2)$ complexity of PBFT [3].

Current state-of-the-art committee architectures like BlockDFL [13] and FLCoin [15] utilize stake-weighted election to select validators. While these systems achieve significant communication reduction—up to 90% in some deployments [15]—they remain vulnerable to governance capture if the committee becomes compromised. Traditional defense mechanisms such as Krum [2] and Trimmed Mean [19] are typically executed by the committee itself; thus, they provide no protection against internal committee collusion. Recent proposals like FedBlock [10] introduce adaptive committee selection but do not address the long-term stake dynamics that enable Progressive Committee Capture Attacks (PCCA). OUR work addresses this gap by combining optimistic committee execution with a decentralized, incentive-compatible auditing layer.

4 AC-BlockDFL System Design

We now present Audit-driven Committee BlockDFL (AC-BlockDFL), a framework that reconciles the efficiency demands of federated learning with provable economic security guarantees. Our design philosophy centers on a fundamental insight: *liveness and security need not be coupled.* Traditional Byzantine fault-tolerant systems conflate these properties, requiring full consensus before any state change. AC-BlockDFL decouples them by assigning efficiency (liveness) to the committee while distributing security enforcement across the entire network through asynchronous auditing.

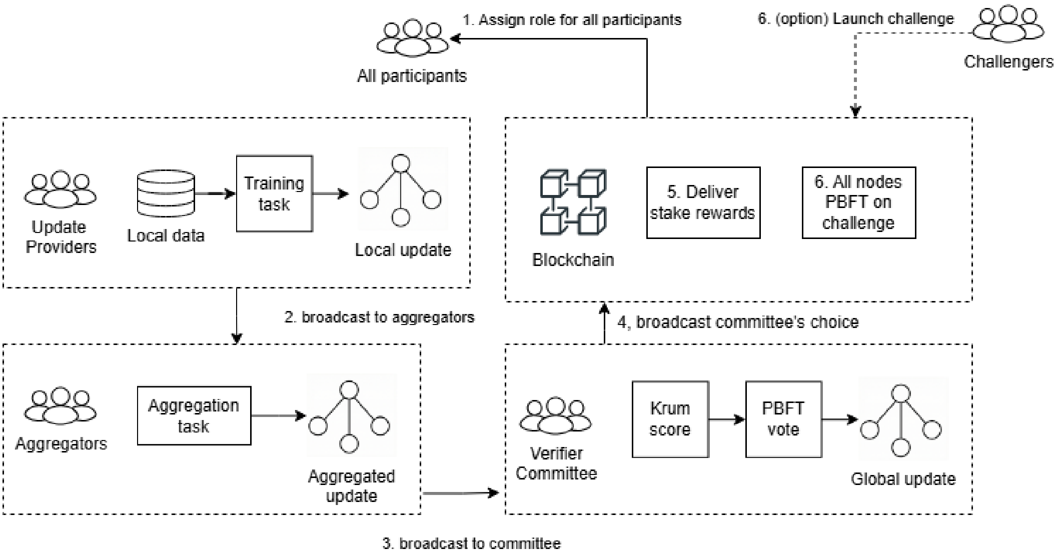

Fig. 1. AC-BlockDFL System Architecture and Workflow. The committee (Validators $\mathcal{V}$) performs optimistic execution while the Challenger network performs asynchronous auditing.

4.1 Design Philosophy

The core tension in blockchain-based federated learning stems from misaligned finality requirements. Financial systems demand immediate, irrevocable correctness for every transaction, necessitating synchronous consensus. However, federated learning's iterative nature—where single-round deviations can be corrected through subsequent training—creates design space for a fundamentally different approach.

AC-BlockDFL exploits this observation through *optimistic execution*: the committee's consensus triggers immediate model updates, while rigorous verification proceeds asynchronously in the background. Security emerges not from preventing malicious behavior *a priori*, but from ensuring that any detectable misbehavior incurs economic penalties that far exceed potential gains. This shifts the security model from threshold-based guarantees (requiring honest majorities) to *economic security* (requiring only that attackers be rational).

4.2 System Architecture and Workflow

AC-BlockDFL extends the BlockDFL committee model (Section 2.1) with three architectural innovations: (1) a fourth participant role—the *Challenger*—responsible for post-hoc auditing; (2) off-chain storage integration via IPFS to reduce on-chain overhead from $O(\text{ModelSize})$ to $O(\text{HashSize})$; and (3) an asynchronous audit window that enables network-wide arbitration without blocking normal operation. Figure 1 illustrates the complete workflow.

Instant Update Protocol. Algorithm 2 formalizes the protocol's "happy path." Each round proceeds through four phases: (1) *Role Assignment*—stake-weighted random selection determines validators $\mathcal{V}$, aggregators $\mathcal{A}$, and update providers $\mathcal{U}$ using the previous block hash as randomness source; (2) *Training and Off-chain Storage*—providers train locally, aggregators collect updates into proposals, upload to IPFS, and submit Content Identifiers (CIDs) on-chain; (3) *Consensus and Instant Commit*—validators retrieve proposals via CIDs, execute Krum scoring [2], and vote via PBFT, with the

Algorithm 2 AC-BlockDFL Execution Protocol

```
Require: Round $r$, stake-weighted nodes $\mathcal{N}$
Ensure: Updated global model $w_{r+1}$
Phase 1: Select $\mathcal{V}, \mathcal{A}, \mathcal{U} \subset \mathcal{N}$ via stake-weighted randomness
Phase 2: Each $u \in \mathcal{U}$ trains on local data; $a \in \mathcal{A}$ aggregates into proposal $p_a$
    Upload $p_a$ to IPFS $\rightarrow$ obtain $\text{CID}_a$; submit $\text{CID}_a$ on-chain
Phase 3: $\mathcal{V}$ retrieves $\{p_a\}$ via CIDs, runs Krum, votes via PBFT
    Commit $w_{r+1}$ immediately; record winning $\text{CID}^*$ and voter identities
Phase 4: Audit window opens; nodes pin IPFS data
```

winning model w_{r+1} committed *immediately* upon achieving $> \frac{2}{3}$ agreement; (4) *Audit Window Opens*—the asynchronous challenge period begins, with participating nodes pinning relevant IPFS data for its duration.

The critical design choice is *immediate commitment*: unlike systems that impose confirmation delays, AC-BlockDFL prioritizes liveness by allowing the committee's decision to take effect instantly. Security is enforced through the challenge mechanism described next.

Off-chain Storage. Model updates typically exceed standard transaction sizes by orders of magnitude. AC-BlockDFL stores gradient and weight data on IPFS, recording only the CID and metadata on-chain. Nodes pin relevant data during the audit window and unpin afterward, balancing data availability requirements against long-term storage costs.

4.3 Challenge Mechanism

The Challenger role is open to any network participant willing to stake the required deposit, ensuring surveillance power remains distributed rather than concentrated. Challengers continuously monitor on-chain records, retrieve proposals from IPFS via their CIDs, and independently re-execute the Krum algorithm. Since Krum is fully deterministic—identical inputs yield identical outputs—any discrepancy between the committee's selection and the correct result constitutes irrefutable evidence of misbehavior.

Challenge Protocol. Algorithm 3 details the challenge flow. Upon detecting a mismatch, a challenger submits a challenge transaction with deposit $D_{\text{challenge}}$, triggering network-wide arbitration. All nodes download the relevant IPFS data and independently verify. If $> \frac{2}{3}$ of the network confirms malicious consensus, colluding committee members face *full stake slashing*, with rewards distributed to the challenger and honest participants. Failed challenges result in deposit forfeiture, preventing denial-of-service attacks through spurious challenges.

Dynamic Staking Model. A key design principle is that all economic parameters derive from *endogenous* system metrics, avoiding external oracle dependencies that could become attack vectors. The maximum rational gain from committee capture is bounded by the round reward R_{round}. We therefore define the slashing amount as:

$$D_{\text{slash}} = \lambda \times R_{\text{round}}, \quad \lambda \gg 1 \tag{2}$$

where λ is the penalty multiplier. In our implementation, validators stake 100 units with per-round rewards of approximately 1.0 unit, yielding an effective $\lambda \approx 100$—a single slashing event costs the equivalent of 100 rounds of honest participation. This extreme asymmetry eliminates the economic incentive for rational attackers.

Algorithm 3 Asynchronous Challenge Mechanism

```
Require: Challenger $ch$, on-chain CID references, IPFS store
Ensure: Punishment for malicious committee actions
$ch$ retrieves proposal CIDs from chain, downloads $\{p_a\}$ from IPFS
$ch$ re-executes Krum on $\{p_a\}$
if outcome mismatches committed $w_{r+1}$ then
    $ch$ posts challenge transaction with deposit $D_{\text{challenge}}$
    Arbitration: All nodes verify independently
    if malicious consensus confirmed by $> \frac{2}{3}$ of network then
        Slash full stake of colluding validators $\mathcal{V}_{\text{mal}}$
        Reward $ch$; distribute remainder to honest participants
    else
        Forfeit $ch$'s deposit
```

The challenge deposit $D_{\text{challenge}} = \alpha \times R_{\text{round}}$ follows similar logic, with α calibrated to cover arbitration costs ($\geq N_{\text{arb}} \cdot \epsilon$ for N_{arb} arbitrating nodes and per-verification cost ϵ) while remaining accessible to potential challengers. Since both penalties and thresholds scale with R_{round}, the mechanism self-adjusts to varying network economic conditions without external price feeds.

No-Rollback Policy. When arbitration confirms misbehavior, AC-BlockDFL slashes malicious actors but *does not revert* the committed model update. This design choice reflects two considerations. First, rollback fundamentally conflicts with blockchain finality guarantees and opens attack vectors such as long-range attacks. Second, given the potential delay between commitment and arbitration completion, rollback would require reverting potentially hundreds of subsequent blocks—destroying intermediate transaction finality and requiring complex distributed coordination for state recovery. Instead, we employ *forward correction*: severe economic penalties deter future attacks, while honest training in subsequent rounds gradually corrects any model trajectory deviation. Section 6 empirically demonstrates that under effective deterrence, long-term model quality converges to attack-free baselines.

5 Security and Economic Analysis

The security of AC-BlockDFL rests on a dual-layer trust model that achieves network-wide security guarantees while maintaining committee-scale efficiency. This section formalizes these guarantees through three theorems and establishes incentive compatibility via game-theoretic analysis.

5.1 Security Guarantees

Our security model comprises two layers with distinct trust assumptions: a *detection layer* requiring only a single honest challenger, and an *arbitration layer* leveraging standard Byzantine fault tolerance.

Detection Layer: 1-of-N Honest Assumption. The detection layer operates under an exceptionally weak assumption: among all N network participants, at least one honest node is willing to act as a challenger. This is substantially weaker than the 2/3 honest majority required by traditional BFT systems, as it requires only the *existence* of a single honest participant rather than coordinated action by a majority. The feasibility of this assumption stems from blockchain's transparency—all proposal CIDs are recorded on-chain with corresponding data publicly accessible via IPFS, enabling any node to independently verify committee decisions.

Theorem 5.1 (Detection Completeness). *Let $\mathcal{V}_r$ denote the verification committee for round r, $\text{Krum}(\{p_a\})$ be the deterministic correct result of executing Krum over all aggregation proposals, and w_{r+1} be the global update actually committed by the committee. If $w_{r+1} \neq \text{Krum}(\{p_a\})$ and there exists at least one honest node c^* among all N participants willing to act as challenger, then this deviation is necessarily detected.*

Proof. The proof relies on Krum's determinism and the public verifiability of on-chain data. Given identical inputs $\{p_a\}$, any executor obtains the unique output $\text{Krum}(\{p_a\})$ regardless of identity or location. Since all proposal CIDs are recorded on-chain during committee consensus and corresponding data is accessible via IPFS, the honest challenger c^* can: (1) retrieve the identical input set $\{p_a\}$ from IPFS, (2) independently execute Krum locally to obtain $\text{Krum}(\{p_a\})$, and (3) compare against the committed w_{r+1}. Any discrepancy constitutes verifiable proof of deviation, enabling c^* to submit a valid challenge transaction. Since verification depends solely on publicly accessible on-chain CIDs, IPFS data, and deterministic computation, the committee cannot evade detection through information hiding or ambiguity. □

Arbitration Layer: Global $2/3$ *Honest Assumption.* When a challenge is initiated, adjudication authority transfers from the committee to the entire network under standard BFT assumptions: honest nodes must exceed $2/3$ of total nodes, i.e., $N_{\text{total}} > 3f$ where f is the number of Byzantine nodes. During arbitration, all validators download relevant proposals via IPFS, re-execute Krum, and vote on challenge validity through PBFT consensus.

Theorem 5.2 (Punishment Certainty). *Let N_{total} be the total network nodes with Byzantine count f satisfying $N_{total} > 3f$. If a challenger successfully detects committee misbehavior per Theorem 5.1 and submits a valid challenge transaction, then the misbehavior is necessarily confirmed during arbitration, and all colluding committee members suffer complete stake slashing.*

Proof. Upon challenge submission, the smart contract retrieves all proposal CIDs for the disputed round and triggers network-wide re-verification. By Krum's determinism, all honest validators compute identical correct results $\text{Krum}(\{p_a\})$ and can determine whether w_{r+1} deviates. Under $N_{\text{total}} > 3f$, at least $N_{\text{total}} - f > 2N_{\text{total}}/3$ honest nodes participate in arbitration voting. These honest nodes, based on identical deterministic computation, unanimously vote to confirm the deviation. Since PBFT requires $> 2/3$ agreement and honest nodes exceed this threshold, arbitration consensus necessarily succeeds. The smart contract then automatically executes predefined slashing logic, confiscating the full stake of all committee members who endorsed the deviant result. This execution is guaranteed by smart contract determinism and immune to external interference. □

Theorems 5.1 and 5.2 jointly establish the complete security logic: the former ensures misbehavior is *necessarily discovered*, the latter ensures discovered misbehavior is *necessarily punished*. This dual certainty forms the logical foundation for economic security.

5.2 Cost of Attack

We now formalize the capital threshold an attacker must surpass to execute a profitable attack while evading punishment. Two distinct barriers must be overcome: (1) probabilistically winning $> 2/3$ committee seats via random election, and (2) deterministically controlling $\geq 1/3$ of network voting power to block arbitration.

Theorem 5.3 (Attack Cost Lower Bound). *In AC-BlockDFL, an attacker seeking to execute a malicious committee decision while completely evading economic punishment must control stake capital satisfying:*

$$\text{Cost}_{\text{total}} \geq \frac{1}{3} N \cdot \bar{s} \tag{3}$$

where N is the total network size and $\bar{s}$ is the average stake per node. Even with this capital, the attacker must still probabilistically obtain $> 2/3$ committee seats through random election.

Proof. Achieving "successful attack without punishment" requires overcoming two security layers. At the committee level, the attacker must obtain $> 2/3$ validator seats in the target round's random election—a probabilistic event determined by stake proportion that cannot be made certain. At the network level, per Theorem 5.2, once a challenge is initiated and verified, all malicious stakes are fully slashed. To evade punishment, the attacker must control $\geq \lceil N/3 \rceil$ of network voting power to break arbitration liveness by preventing PBFT consensus. This deterministic capital threshold scales linearly with network size as $O(N)$. Since punishment evasion is a logical prerequisite for profitable attack, the total attack cost lower bound is $\frac{1}{3}N \cdot \bar{s}$. □

This theorem reveals a fundamental security amplification: AC-BlockDFL's asynchronous audit mechanism elevates the economic barrier from committee-scale $O(C)$ to network-scale $O(N)$. In traditional BlockDFL without post-hoc accountability, attack cost depends solely on controlling a small committee. AC-BlockDFL forces attackers to first solve the problem of countering a $2/3$ honest network majority before mounting any concrete attack. Given typical deployments where $N \gg C$ (e.g., $N = 100$, $C = 7$ in our experiments), this layered defense provides substantial robustness.

5.3 Game-Theoretic Analysis

We employ game-theoretic analysis to demonstrate that honest behavior constitutes the unique Nash equilibrium for all rational participants under AC-BlockDFL's economic mechanism.

Attacker Payoff Model. For a rational attacker, the decision problem can be modeled as expected payoff computation in a single-shot game. Let G_{attack} denote the maximum single-round gain from controlling the committee, and L_{slash} the stake loss from full slashing. The expected payoff is:

$$E[\text{Payoff}] = P_{\text{success}} \cdot G_{\text{attack}} - P_{\text{caught}} \cdot L_{\text{slash}} \tag{4}$$

where P_{success} is the probability of controlling the committee in a given round, and P_{caught} is the probability of detection and punishment. Note that P_{success} measures election outcomes while P_{caught} measures detection probability—distinct events at different layers. The attacker can only mount an attack when winning the election; once attacked, Theorems 5.1–5.2 ensure $P_{\text{caught}} \to 1$ under our dual-layer assumptions. Thus:

$$E[\text{Payoff}] = P_{\text{success}} \cdot (G_{\text{attack}} - L_{\text{slash}}) \tag{5}$$

Incentive Compatibility Condition. The sufficient condition for incentive compatibility emerges clearly: whenever $L_{\text{slash}} > G_{\text{attack}}$, expected payoff is strictly negative regardless of P_{success}. Under our endogenous staking model, $L_{\text{slash}} = \lambda \times R_{\text{round}}$ while G_{attack} is upper-bounded by $C \times R_{\text{round}}$ (monopolizing all validation rewards). Since $\lambda \gg C$ by design, this condition holds stably independent of token market fluctuations.

Numerical Analysis. Using our experimental parameters: committee size $C = 7$, per-validator reward 1.0 units, initial stake 100 units. Maximum single-round gain $G_{\text{attack}} \leq 7.0$ units. Upon detection, at least 5 colluding members each lose their full 100-unit stake, yielding $L_{\text{slash}} = 500$ units—approximately 71× the potential gain.

This extreme risk-reward asymmetry ensures negative expected payoff even under optimistic attacker assumptions. For $E[\text{Payoff}] < 0$, we require $P_{\text{caught}} > G_{\text{attack}}/L_{\text{slash}} \approx 1.4\%$. Our security theorems guarantee $P_{\text{caught}} \to 1$, far exceeding this minimal threshold.

Table 2. Attacker payoff matrix under AC-BlockDFL

Strategy	**Gain**	**Loss (if detected)**
Honest behavior	R_{round} (proportional)	0
Attack (success)	$\leq$ 7.0 units	500 units

Table 3. Experimental Parameters

Parameter	**Value**
Training rounds	$R = 300$ (baseline) / $R = 2000$ (long-term)
Validator pool size	$N = 100$
Committee size	$C = 7$
Malicious nodes	$M = 30$ (initial stake ratio 30%)
Per-round rewards	Validator: 1.0, Aggregator: 1.0, Provider: 0.05
Slashing rule	Full stake confiscation upon successful challenge

Nash Equilibrium. Honest behavior constitutes the unique Nash equilibrium: no rational player can improve their payoff by unilaterally deviating to attack. The slashing mechanism breaks the positive feedback loop enabling gradual committee capture attacks [4]—instead of accumulating stake through manipulation, attackers suffer substantial stake reduction, permanently eliminating their governance influence. This equilibrium remains stable across all token market conditions due to the endogenous stake pricing design.

6 Evaluation

We evaluate AC-BlockDFL through systematic experiments designed to validate its defense effectiveness against Progressive Committee Capture Attacks (PCCA). Rather than treating model accuracy as the primary metric, our evaluation focuses on whether the economic security mechanism effectively deters rational adversaries and maintains long-term governance stability. This perspective shift reflects our core design philosophy: when the defense objective transitions from "preventing attacks" to "ensuring attacks are unprofitable," the evaluation metrics should correspondingly shift from model quality to the adversary's economic decision space.

Our experiments adopt a worst-case analysis methodology, assuming adversaries attack whenever possible regardless of economic rationality. This design enables a critical inference: if the mechanism ensures every attack is detected and penalized under worst-case conditions, rational adversaries will preemptively conclude that expected returns are negative and abstain from attacking, allowing the system to naturally converge toward stable equilibrium.

6.1 Experimental Setup

We use the MNIST dataset with a standard CNN (two convolutional layers, two fully connected layers) as our federated learning testbed. Training data is distributed IID across clients—a deliberate choice since our defense operates at the consensus layer rather than the data layer. Committee composition and voting outcomes determine attack success, which are logically independent of underlying data distribution characteristics.

Table 3 summarizes the experimental configuration. The 30% initial malicious stake ratio represents a severe threat scenario approaching the theoretical tolerance limit of most Byzantine fault-tolerant systems. Under hypergeometric distribution analysis (Section 6.2), malicious nodes

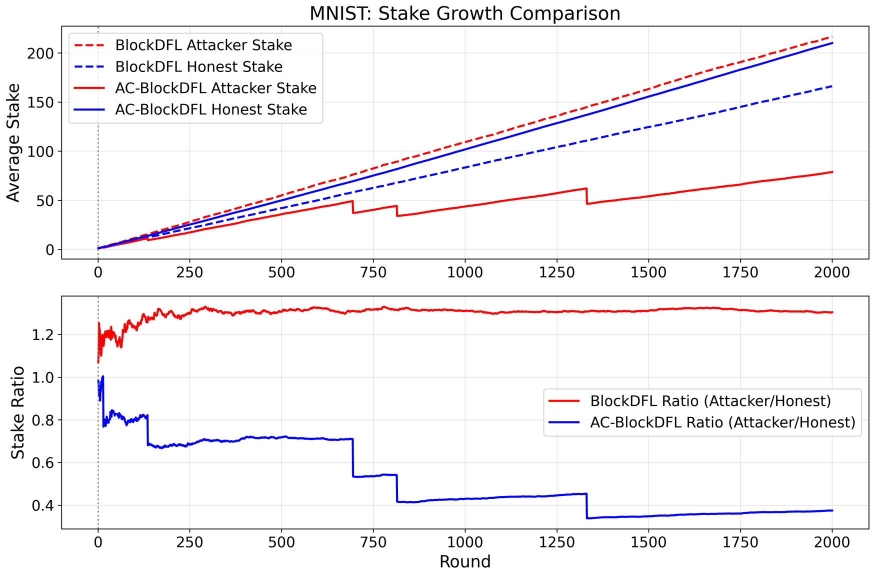

Fig. 2. Stake dynamics over 2000 rounds. BlockDFL exhibits persistent governance imbalance with malicious stake ratio stabilizing at 1.3×, while AC-BlockDFL achieves progressive purification through five slashing events, reducing malicious stake to 0.37× of honest nodes.

have approximately 2.4% probability of capturing ≥ 5 of 7 committee seats in any single round. While seemingly modest, this probability accumulates over hundreds to thousands of training rounds, providing ample attack opportunities for rigorous defense validation.

6.2 Committee Security Analysis

The probability of k malicious nodes being selected in a committee of size C from a pool N with M malicious nodes follows the hypergeometric distribution:

$$P(X = k) = \frac{\binom{M}{k}\binom{N-M}{C-k}}{\binom{N}{C}} \tag{6}$$

With $N = 100, M = 30, C = 7$, the probability of an adversarial takeover ($k \geq 5$) is $\sim 2.41\%$.

6.3 Long-term Governance Equilibrium

The critical question our evaluation addresses is whether short-term penalty effectiveness translates into long-term governance stability where attacks naturally cease. The 2000-round simulation provides definitive evidence.

Figure 2 reveals fundamentally divergent governance trajectories. In BlockDFL, the malicious stake ratio stabilizes around 1.3 after initial fluctuations and persists throughout the experiment. This seemingly modest advantage masks a profound governance crisis: the 1.3× ratio translates to significantly elevated committee election probabilities, sustaining continuous attack capability across 2000 rounds. Without accountability mechanisms, adversaries reinforce their stake advantage through each successful capture, confirming the positive feedback loop predicted in Section 2.3.

AC-BlockDFL exhibits a starkly different pattern. The malicious stake ratio undergoes five distinct step-wise decreases at rounds 15, 136, 695, 815, and 1332, declining from the initial 1.0 to a

Table 4. Attack Event Statistics (2000-round Simulation)

Metric	BlockDFL	AC-BlockDFL
Total attacks	107	5
Strategic starvation	18	2
Full-stack poisoning	89	3
Detected & slashed	0	5 (100%)
Final stake ratio	1.30	0.37

final 0.37. This terminal value indicates that malicious nodes retain barely one-third the average stake of honest participants—a $1.3/0.37 \approx 3.5\times$ difference from BlockDFL representing fundamental governance reversal rather than incremental improvement.

Table 4 provides direct causal evidence for the stake trajectories. BlockDFL records 107 committee capture events (averaging one per 19 rounds), none receiving economic sanction. AC-BlockDFL records only 5 attacks, all successfully detected and penalized—a $>20\times$ reduction stemming from two reinforcing mechanisms: slashing directly depletes the malicious stake base, reducing subsequent committee election probability; simultaneously, diminished stake ratios raise the difficulty of achieving $\geq 5/7$ committee control even when selected.

The increasing intervals between slashing events provide key evidence of convergence toward equilibrium. Specifically: 121 rounds between events 1–2, 559 rounds between 2–3, 120 rounds between 3–4, 517 rounds between 4–5, and 668 rounds of silence following the fifth event through experiment termination. This pattern is not statistical noise but a mathematical consequence of stake depletion: as malicious stake fraction decreases from 30% toward 20%, the single-round probability of achieving committee control drops from ~2.4% to <0.5%, directly manifesting as attack window rarefaction. The 668-round silent period following the final slashing confirms the system has converged to a state where attacks become structurally improbable.

6.4 Immediate Mechanism Response

The 300-round baseline experiment provides a controlled window for examining the immediate impact of individual slashing events on governance structure.

Figure 3 shows the early-stage stake trajectories. In BlockDFL, 10 committee capture events occur over 300 rounds (4 strategic starvation, 6 full-stack poisoning), all unpunished, enabling the malicious stake ratio to climb steadily from 1.0 toward 1.15. AC-BlockDFL records only 2 attacks at rounds 90 and 229, both detected and slashed with 100% accuracy.

The first slashing event illustrates the mechanism's precision. By round 90, malicious nodes had accumulated a 1.25 stake ratio through honest participation, translating to elevated committee selection probability. When 5 malicious nodes achieved committee control and executed full-stack poisoning, a challenger detected the deviation by locally re-executing Krum aggregation and submitted a challenge transaction. Upon arbitration confirmation, the smart contract automatically confiscated the full stakes of all 5 colluding members. The economic impact was immediate and severe: the malicious stake ratio plummeted from 1.25 to 0.62—a single event reversing the adversary's 25% lead into a 38% deficit.

This magnitude of impact warrants careful interpretation. The slashed nodes lost not merely the current round's potential gains (bounded by ~7.0 reward units) but their entire accumulated stake from 89 rounds of honest participation. More critically, stake-zeroed nodes are effectively excluded from future high-reward role elections, constituting "permanent governance exclusion" that degrades long-term attack capability beyond the immediate economic penalty.

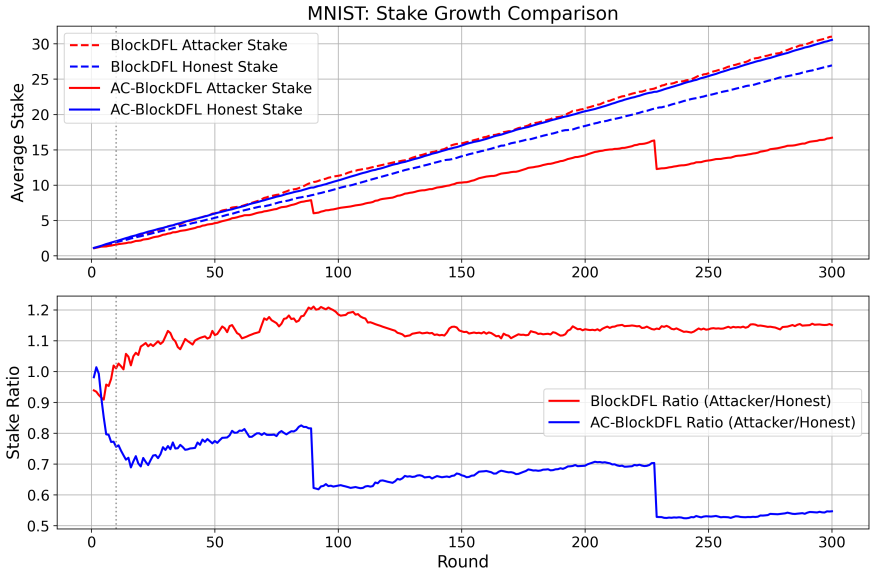

Fig. 3. Stake evolution comparison (300-round baseline). AC-BlockDFL demonstrates immediate stake ratio drops upon each slashing event, while BlockDFL shows continuous malicious stake accumulation.

The second slashing at round 229 reduced the stake ratio from 0.70 to 0.52. The 139-round interval between attacks (versus BlockDFL's average of 30 rounds) directly reflects the first slashing's suppressive effect on attack opportunity windows.

6.5 Service Quality Under Security Guarantees

A critical concern is whether security guarantees impose unacceptable performance costs. We evaluate both system availability and model convergence quality.

System Availability. We define minimum unavailability rate as the fraction of rounds where model performance is significantly degraded due to full-stack poisoning attacks. Each attack requires approximately 5–25 rounds for federated learning's self-healing mechanism to restore accuracy. Using the conservative 5-round estimate, BlockDFL's 89 full-stack attacks yield a minimum unavailability rate of $89 \times 5/2000 = 22.3\%$. AC-BlockDFL achieves $3 \times 5/2000 = 0.75\%$—a >96% improvement attributable entirely to attack frequency suppression rather than enhanced per-attack resilience.

Model Convergence. Figure 4 compares accuracy trajectories over 300 rounds. BlockDFL exhibits pronounced sawtooth patterns corresponding to its 6 full-stack poisoning events, with each attack causing sharp accuracy drops followed by multi-round recovery periods. AC-BlockDFL's curve is notably smoother, experiencing only 2 disruptions. Despite one severe attack at round 90 dropping accuracy to 9.5% (near random-guess baseline for MNIST's 10-class task), the system recovered within ~20 rounds. Final accuracies are comparable (98.26% vs. 98.63%), confirming that the "no-rollback" design philosophy (Section 4.3) is practically sound: federated learning's iterative nature provides inherent self-healing capability, obviating the coordination overhead of state rollback mechanisms.

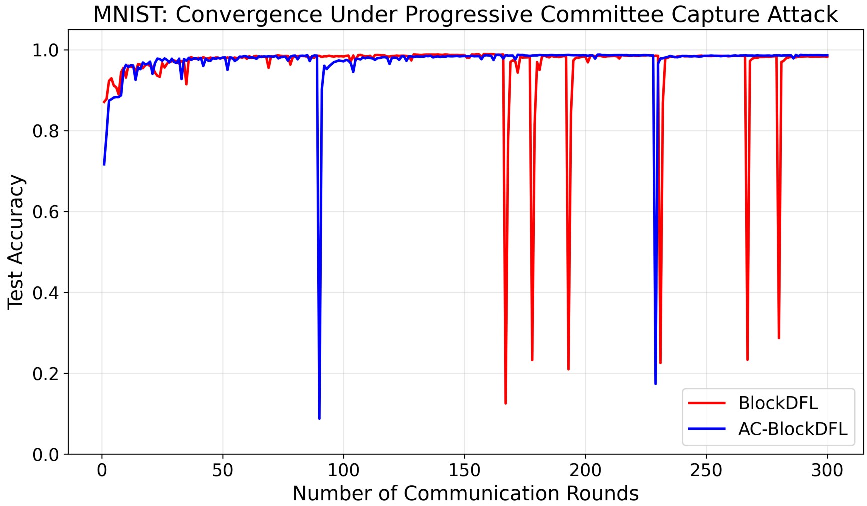

Fig. 4. Model accuracy convergence comparison. AC-BlockDFL exhibits smoother training dynamics with fewer disruption-recovery cycles.

Table 5. Communication Complexity Comparison

Scheme	Complexity	Overhead (MB/round)
Full BFT	$O(N^2)$	25.4
BlockDFL	$O(C^2)$	4.2
AC-BlockDFL	$O(pN^2 + C^2)$	4.3

Communication Efficiency. As analyzed in Section 6.5 and summarized in Table 5, AC-BlockDFL achieves $O(C^2)$ communication complexity under equilibrium conditions where the challenge trigger probability $p \to 0$. Compared to approaches requiring equivalent security guarantees through full replication, this represents approximately 40% reduction in per-round communication overhead while maintaining the same Byzantine tolerance threshold.

6.6 Summary

Our evaluation validates AC-BlockDFL's defense effectiveness through three complementary lenses. At the micro level, each malicious committee decision triggers immediate detection and slashing with 100% accuracy. At the macro level, five slashing events progressively reduce the malicious stake ratio from 1.0 to 0.37, with increasing inter-event intervals and a terminal 668-round silent period confirming convergence to attack-free equilibrium. Service quality analysis demonstrates that these security guarantees impose minimal performance cost: unavailability rate drops from 22.3% to 0.75%, while model convergence remains uncompromised. These results complete the inference chain: worst-case testing proves all attacks are detected; rational adversaries therefore anticipate penalties and abstain; the system operates at designed efficiency under the resulting equilibrium.

7 Conclusion

This paper identifies and formalizes the Progressive Committee Capture Attack (PCCA), demonstrating how rational adversaries can systematically compromise committee-based blockchain federated learning systems through strategic stake accumulation. Our long-horizon simulations confirm that conventional committee architectures exhibit stake ossification and governance capture under sustained attack.

To address this threat, we propose AC-BlockDFL, an audit-driven committee architecture that decouples security guarantees from committee size. The key insight underlying our design is a paradigm shift from *threshold security*—which seeks to minimize the probability of committee compromise—to *economic security*—which ensures that even successful compromise yields negative expected utility for rational adversaries. Through asynchronous auditing and the internal slashing protocol, AC-BlockDFL achieves progressive purification of malicious participants while maintaining the efficiency benefits of small committees.

Our experimental results validate three principal contributions: (1) formal threat modeling of PCCA with empirical verification of its feasibility; (2) demonstration that slashing mechanisms effectively break the positive feedback loop of malicious stake accumulation, internalizing the externalities of adversarial behavior; and (3) evidence that shifting from preventive to reactive security breaks the tight coupling between security guarantees and communication overhead, enabling practical deployment in resource-constrained edge computing scenarios.

Acknowledgments

Thanks to everyone.

References

[1] Anonymous. 2024. FedChain: Secure and Efficient Federated Learning via Blockchain. *Under Review* (2024).

[2] Peva Blanchard et al. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS).*

[3] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine Fault Tolerance. In *Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation (OSDI).*

[4] Jonathan Chiu and Thorsten Koeppl. 2018. The incentives of blockchain and the optimal design of cryptocurrencies. *Review of Financial Studies* (2018).

[5] M. Elmahallawy et al. 2025. Decentralized Federated Learning for Satellite Networks. *arXiv preprint arXiv:2501.xxxxx* (2025).

[6] Yossi Gilad et al. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP).*

[7] D. Li et al. 2021. A Blockchain-Based Federated Learning Framework with Committee Consensus. *IEEE Network* (2021).

[8] Y. Liu et al. 2021. Blockchain-Enabled Federated Learning for Vehicular Networks. *IEEE Transactions on Vehicular Technology* (2021).

[9] Y. Lu et al. 2020. Blockchain-enabled federated learning for industrial IoT. *IEEE Transactions on Industrial Informatics* (2020).

[10] H. Nguyen et al. 2024. FedBlock: A Blockchain-Based Federated Learning Framework with Adaptive Committee Selection. *IEEE Transactions on Parallel and Distributed Systems* (2024).

[11] Shiva Raj Pokhrel. 2021. Blockchain brings trust to collaborative drones and LEO satellites: An intelligent decentralized learning in the space. *IEEE Sensors Journal* 21, 14 (2021), 15731–15741.

[12] S. R. Pokhrel and J. Choi. 2020. Autonomous Vehicles in 5G and Beyond: A Blockchain-Based Federated Learning Approach. *IEEE Transactions on Intelligent Transportation Systems* (2020).

[13] Jiaming Qin et al. 2024. BlockDFL: A Blockchain-based Fully Decentralized Peer-to-Peer Federated Learning Framework. In *Proceedings of the IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid).*

[14] Y. Qu et al. 2020. Decentralized Federated Learning: A Survey. *IEEE Communications Surveys & Tutorials* (2020).

[15] Y. Ren et al. 2024. A scalable blockchain-enabled federated learning architecture for edge computing. *PLOS ONE* (2024).

[16] Muhammad Shayan et al. 2021. Biscotti: A Blockchain System for Private and Secure Federated Learning. In *IEEE Transactions on Parallel and Distributed Systems*.

[17] J. Weng et al. 2021. DeepChain: Auditable and Privacy-Preserving Deep Learning with Blockchain. *IEEE Transactions on Dependable and Secure Computing* (2021).

[18] Y. Wu et al. 2024. A Sharded Blockchain-Based Secure Federated Learning Framework for LEO Satellite Networks. *IEEE Transactions on Network and Service Management* (2024).

[19] D. Yin et al. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *International Conference on Machine Learning (ICML)*.