Blockchain-Enabled Federated Learning with Prototype Verification for Tamper-Resistant Distributed Model Training

Nils Hunt
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.
nils.work@oregonstate.edu

Nicolas Wells
Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV, USA.
nicolaswells@unr.edu

Abstract

The convergence of federated learning and blockchain technology offers a promising pathway toward tamper-resistant distributed model training, yet existing approaches often overlook the semantic integrity of learned representations. This paper presents a comprehensive framework that integrates blockchain-based immutable audit trails with prototype verification mechanisms to detect and mitigate malicious model updates in federated learning environments. Unlike conventional aggregation schemes that depend solely on statistical anomaly detection, prototype verification leverages class-level feature representations to validate the semantic consistency of contributed gradients before they are committed to the global model. The blockchain layer provides a decentralized, non-repudiable ledger of verification outcomes and model states, enabling transparent governance and post-hoc forensic analysis. We examine the structural trade-offs between verification granularity, computational overhead, and communication efficiency, and discuss how prototype anchors can be securely maintained across distributed nodes without a central authority. The system architecture is analyzed from the perspectives of scalability, adversarial robustness, and cross-silo deployment in regulated domains such as healthcare and finance. Furthermore, we explore the policy implications of embedding verifiable semantic constraints into distributed learning pipelines, including the tension between privacy preservation and auditability. This paper contributes a system-level design rationale that bridges cryptographic integrity, representation learning, and socio-technical governance, offering a blueprint for trustworthy federated learning in high-stakes applications.

Keywords

blockchain, federated learning, prototype verification, tamper resistance, distributed systems, model integrity, adversarial robustness, governance.

1. Introduction

Federated learning has emerged as a paradigm for collaborative model training across decentralized data silos without requiring raw data to leave local devices. However, the distributed nature of federated learning introduces vulnerabilities that undermine trust in the training process. Malicious participants may upload corrupted gradients to poison the global

model, while honest participants have no mechanism to verify that their contributions are faithfully aggregated. Traditional defenses, such as robust aggregation rules or differential privacy, offer partial protection but often fail against adaptive adversaries who can gradually distort model representations over many rounds [1][2]. The absence of a tamper-proof audit trail further complicates accountability: after training completes, there is no cryptographically verifiable record of which participant contributed which update at which round.

Blockchain technology, with its immutable ledger and decentralized consensus, has been proposed as a natural complement to federated learning for recording model updates and aggregation results [3][4]. Several blockchain-based federated learning systems have demonstrated that a shared ledger can prevent retrospective manipulation of training history and enable incentive mechanisms through smart contracts. Nevertheless, most existing works focus on the integrity of the update recording process rather than the semantic correctness of the updates themselves. A malicious update that passes statistical filters may still embed backdoors or cause gradual concept drift that is only detectable after deployment.

This paper argues that tamper resistance in federated learning must extend beyond transaction-level immutability to include representation-level verification. We introduce prototype verification as a mechanism to enforce that each local gradient update is consistent with a set of class prototypes that capture the expected feature distribution of the global model. By anchoring the verification criteria in a shared prototype space, the system can reject updates that deviate beyond a semantic plausibility threshold, even if the statistical properties of the gradient remain within nominal bounds. The combination of blockchain for recording and prototype verification for semantic scrutiny creates a layered defense against a wide range of adversarial behaviors.

We structure the discussion as follows. Section 2 reviews related work on federated learning security, blockchain integration, and prototype-based defenses. Section 3 describes the proposed system architecture, detailing the roles of local nodes, aggregators, verifiers, and the blockchain ledger. Section 4 analyzes tamper resistance properties and security guarantees under different threat models. Section 5 addresses governance, fairness, and policy dimensions, including the implications of centralized verification authorities and the trade-offs between auditability and privacy. Section 6 examines deployment considerations, including scalability, energy efficiency, and cross-domain adaptation. Section 7 concludes with a summary of contributions and directions for future research.

2. Background and Related Work

Federated learning faces a fundamental tension between decentralization and trust. Without a central authority to validate contributions, the system must rely on protocols that are both robust to Byzantine failures and resilient to adversarial manipulation. Robust aggregation techniques such as Krum, trimmed mean, and median-based methods have been shown to withstand a fraction of malicious updates under certain assumptions [1][6]. However, these methods are often brittle against colluding adversaries or attacks that target specific layers of the model. Furthermore, they operate purely on the vector space of gradients, ignoring the semantic meaning of the updates.

Blockchain-based federated learning addresses the trust deficit by providing a transparent and immutable record of the training process. Works such as BlockFL [3] and DeceFL [4] propose architectures where model updates are stored on a blockchain, and smart contracts enforce reward distribution or penalize misbehavior. The consensus mechanism ensures that

once an update is committed, it cannot be altered retroactively. This property is valuable for auditing and for establishing non-repudiation in regulated industries. However, the blockchain layer does not inherently validate the quality or safety of the updates; it only provides a tamper-proof log. Malicious updates can still be stored on the ledger, and the system must rely on additional off-chain mechanisms for detection.

Prototype-based defenses have recently gained attention as a way to incorporate semantic knowledge into anomaly detection for distributed learning. Prototypes are representative feature vectors for each class, typically computed from the global model's embedding space. By comparing local updates against these prototypes, a verifier can detect whether a client's gradient is attempting to shift the decision boundary in a semantically implausible direction. The work by Shui et al. [5] introduced prototype consistency as a defense against backdoor attacks in vertical split learning, demonstrating that a deviation threshold based on cosine similarity to class prototypes can effectively filter poisoned updates without requiring access to raw data. Our framework generalizes this concept to horizontal federated learning and integrates it with a blockchain backbone to create a fully auditable verification pipeline.

Other approaches combine cryptographic techniques such as zero-knowledge proofs or secure multi-party computation to verify gradient correctness [7]. While these methods offer strong privacy guarantees, they incur substantial computational and communication overhead that limits their practicality in large-scale deployments. Prototype verification strikes a different trade-off: it requires only lightweight similarity computations and can be executed efficiently even on resource-constrained edge devices. Moreover, because prototypes are derived from the global model, they evolve naturally over the training process, reducing the need for pre-defined trusted datasets.

3. System Architecture and Prototype Verification Mechanism

The proposed system architecture consists of three primary entity types: local clients, a set of verifier nodes, and a blockchain network that maintains the global model state and verification records. Local clients perform training on their private data and compute gradient updates, which they broadcast to verifier nodes along with a proof of identity. Verifier nodes maintain a replica of the current global model and a set of class prototypes extracted from that model. Before an update is accepted into the aggregation process, it must pass a prototype consistency check.

The prototype verification mechanism operates as follows. At each communication round, the verifier node obtains the global model from the blockchain, then computes prototypes by feeding a small number of reference samples (or synthetic examples generated by the model) through the feature extraction layers. For each class label, a prototype is the mean embedding of samples belonging to that class, normalized to unit norm. When a local client submits a gradient update, the verifier applies the gradient to a temporary copy of the global model and computes the resulting change in the prototype embeddings. The client's update is considered valid if the cosine similarity between the new prototype for each class and the original prototype exceeds a predefined threshold. Intuitively, this ensures that the update does not cause a sudden, semantically unjustified shift in the representation space.

Updates that pass the prototype check are forwarded to an aggregation smart contract on the blockchain. The smart contract implements a weighted averaging of all valid updates, where the weights reflect the number of training samples contributed by each client. The aggregated global model is then written to the ledger, and a new round begins. All verification outcomes,

including timestamps, client identities, and similarity scores, are recorded immutably. This enables retrospective auditing: a regulator could later inspect the ledger to determine whether any client consistently submitted updates with borderline similarity values, indicative of potential poisoning attempts.

The choice of threshold is a critical design parameter. A strict threshold enhances security but may reject legitimate updates, slowing convergence and introducing bias toward the existing prototype distribution. A lax threshold admits more updates but reduces defense effectiveness. The system can adapt the threshold dynamically based on the variance of prototype similarities observed across rounds. For instance, if the global model is in an early training phase where representations are still shifting, a wider threshold can accommodate larger updates. As training stabilizes, the threshold can be tightened. This adaptive policy requires governance mechanisms to prevent an adversary from manipulating the threshold itself.

4. Tamper Resistance and Security Analysis

Tamper resistance in this framework operates at two levels: the semantic level enforced by prototype verification and the procedural level enforced by blockchain immutability. At the semantic level, an adversary cannot inject a backdoor that dramatically alters class-specific features without being detected, because such an update would cause prototypes to diverge beyond the acceptance threshold. However, an adversary aware of the prototype verification mechanism may attempt to craft updates that shift prototypes gradually over many rounds, staying just within the similarity boundary each time. This is a form of gradient stacking attack. The defense against such slow poisoning lies in the blockchain's audit trail: if a client consistently pushes updates in the same direction across rounds, the cumulative deviation becomes apparent when historical similarity scores are analyzed. Verifiers can implement cumulative drift detection that raises an alarm if a client's prototype similarity trend is monotonically degrading over a window of rounds.

At the procedural level, the blockchain prevents any entity from altering the recorded updates, verification results, or global model states after the fact. This property is essential for accountability in domains such as clinical decision support, where model provenance must be demonstrable to regulatory bodies. Even if a malicious update bypasses prototype verification, the fact that it was recorded with a timestamp and client identity allows for post-hoc damage assessment and model rollback. The blockchain consensus mechanism also makes it difficult for a single entity to censor valid updates, as long as the verifier nodes are sufficiently decentralized.

The integration of prototype verification introduces a new attack surface: an adversary could try to corrupt the prototype computation itself. If an attacker controls a verifier node, they could compute prototypes using a poisoned reference set, thereby establishing a flawed baseline that passes malicious updates. To mitigate this, the prototype computation should be replicated across multiple verifier nodes using a distributed consensus protocol. Each verifier independently computes prototypes from the same global model, and the final prototype used for verification is the median or trimmed mean of the individual computed prototypes. This approach is analogous to robust aggregation at the representation level.

Another security consideration is the trade-off between verification latency and batch size. In synchronous federated learning, verifier nodes must process all incoming updates before proceeding to the next round. If the number of clients is large, the verification step may become a bottleneck. The system can adopt asynchronous verification, where updates are

validated and committed to the blockchain as they arrive, and the global model is updated only after a quorum of valid updates has been collected. Asynchronous operation increases throughput but complicates convergence guarantees. The blockchain can help by recording the order and timing of each verified update, enabling later reconstruction of the model state sequence.

5. Governance, Fairness, and Policy Implications

The deployment of a blockchain-enabled federated learning system with prototype verification raises important governance questions. Who sets the threshold for prototype similarity? Who determines the reference samples used to compute prototypes? In a fully decentralized setting, these parameters could be established through on-chain voting mechanisms among verifier nodes or clients. However, voting introduces its own vulnerabilities, such as Sybil attacks or plutocratic bias where clients with more data have disproportionate influence. A more practical approach for regulated environments is to assign a governance board of domain experts that approves the initial prototype computation procedure and updates the threshold adaptation rules. The blockchain then immutably records all governance decisions, creating a transparent policy log.

Fairness implications arise because prototype verification relies on the global model's representations, which may encode historical biases present in the training data. If the global model has poor performance for certain minority classes, the corresponding prototypes may be distorted, causing legitimate updates from clients with minority data to be rejected more frequently. This could exacerbate representational harms. To address this, the prototype verification threshold should be class-specific, perhaps set inversely proportional to the class sample size, or periodically recalibrated using held-out fairness metrics recorded on the blockchain. Additionally, clients from underrepresented groups should have a mechanism to appeal verification rejections, with the appeal process also recorded on-chain for transparency.

Policy implications extend to privacy regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). While federated learning reduces the need to share raw data, the blockchain ledger containing model updates and verification metadata may still constitute personal data if the updates can be linked back to individuals. Differential privacy can be applied to local updates before verification, but this may degrade prototype similarity calculations. A compromise is to have verifier nodes operate in a trusted execution environment (TEE) that processes updates without exposing them to other parties. The blockchain then stores only the verification result and a hash of the update, not the update itself. This design preserves privacy while maintaining auditability, though it introduces hardware trust assumptions.

The tension between auditability and privacy is inherent. A fully auditable system requires that all verification outcomes be publicly visible, which could enable adversaries to infer the training data distribution by analyzing which updates are rejected. Conversely, if verification outcomes are kept private, then malicious updates cannot be tracked retrospectively. One resolution is to use zero-knowledge proofs to show that an update passed verification without revealing the update content. However, this adds computational overhead. The choice between transparency and privacy must be made based on the threat model and regulatory requirements of the deployment domain.

6. Deployment and Sustainability Considerations

Deploying a blockchain-enabled federated learning system at scale requires careful attention to computational and energy costs. The blockchain consensus mechanism, whether Proof of Work, Proof of Stake, or a Byzantine fault-tolerant protocol, imposes overhead that can be substantial for large training rounds. Proof of Stake mechanisms are generally more sustainable and are preferable for long-running learning tasks. Additionally, the verification process itself adds a layer of computation: each verifier must compute prototypes and evaluate similarity for every client update. For deep neural networks with many classes, computing prototypes may require forward passes through the feature extractor for each reference sample. This cost can be amortized by using a small set of synthetic reference samples generated from the global model's generative capabilities, or by caching prototype computations across rounds when the model changes only incrementally.

Communication efficiency is another factor. In a system where updates are broadcast to multiple verifiers, the network bandwidth may become a bottleneck. One solution is to use a committee-based verification approach where a subset of verifier nodes is randomly selected each round to perform the prototype check. The blockchain records which committee members were selected and their verification decisions, allowing the broader network to verify the committee's honesty through cryptographic commitments. This reduces communication overhead while maintaining distributed trust.

Cross-domain deployment presents challenges because the prototype verification mechanism is domain-specific. A model trained on medical images will have fundamentally different representation spaces than a model trained on financial transaction sequences. The prototype-based defense must be tailored to the structure of the embedding space in each domain. For example, in natural language processing, prototypes may be derived from sentence embeddings, and similarity metrics may need to account for anisotropy in the embedding space. Our framework is designed to be modular: the update verification logic can be swapped out without altering the blockchain infrastructure, enabling specialization for different application domains.

Sustainability also involves the continued evolution of the prototype verification criteria as the global model improves. If the model changes significantly between rounds, the prototypes become stale and may reject updates that are actually beneficial. A common practice is to re-compute prototypes from the latest global model after each aggregation step. In our architecture, this re-computation is triggered automatically by smart contracts that monitor the blockchain for new global model entries. The latency between model update and prototype recomputation must be minimized to avoid a window of vulnerability during which stale prototypes are used.

7. Conclusion

This paper has presented a comprehensive framework for tamper-resistant distributed model training that integrates blockchain-based immutable record-keeping with prototype verification for semantic consistency checking. By requiring that each local gradient update be consistent with class-level prototypes derived from the global model, the system provides a defense against backdoor attacks and gradual poisoning that goes beyond statistical anomaly detection. The blockchain layer ensures that all verification outcomes and model states are recorded transparently and cannot be retroactively altered, enabling accountability and post-hoc auditing. We have discussed key structural trade-offs, including the selection of verification thresholds, the resilience against adaptive adversaries, and the governance mechanisms needed to maintain fairness and adaptiveness. Policy implications regarding

privacy, regulatory compliance, and bias mitigation have been analyzed, and deployment considerations such as scalability, energy efficiency, and cross-domain adaptability have been addressed. The proposed architecture offers a blueprint for building trustworthy federated learning systems in high-stakes environments where semantic integrity and procedural transparency are paramount. Future work should explore empirical evaluations of the prototype verification overhead and the interplay with differential privacy, as well as formal security proofs for the combined blockchain-verification protocol.

References

1. Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In Advances in Neural Information Processing Systems 30 (pp. 119–129).

2. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. In Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (pp. 2938–2948). PMLR.

3. Kim, H., Park, J., Bennis, M., & Kim, S. L. (2020). Blockchained on-device federated learning. IEEE Communications Letters, 24(6), 1279–1283.

4. Yuan, Y., & Wang, F. Y. (2020). Blockchain-based federated learning for safe driving. IEEE Transactions on Intelligent Transportation Systems, 21(12), 5199–5213.

5. Shui, Y., Jin, R., Dou, Z., & Gao, Z. (2026). ProtoGuard-SL: Prototype Consistency Based Backdoor Defense for Vertical Split Learning. arXiv preprint arXiv:2604.03595.

6. Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In Proceedings of the 35th International Conference on Machine Learning (pp. 5650–5659). PMLR.

7. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1175–1191).

8. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (pp. 1273–1282). PMLR.

9. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), 50–60.

10. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1–2), 1–210.

11. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Unpublished manuscript.

12. Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation (pp. 173–186). USENIX.

13. Zhang, C., Li, S., Xia, H., Wang, Y., & Du, X. (2021). Blockchain-based privacy-preserving federated learning scheme. IEEE Transactions on Network and Service Management, 18(3), 3856–3869.

14. Fung, C., Yoon, C. J., & Beschastnikh, I. (2018). Mitigating Sybils in federated learning using reputation-based defense. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (pp. 6478–6488).

15. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-IID data. arXiv preprint arXiv:1806.00582.

16. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security (pp. 1–11).

17. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 308–318).

18. Xu, J., Wang, H., & Chen, L. (2022). Federated learning over blockchain: A survey. IEEE Internet of Things Journal, 9(21), 20995–21013.

19. Cao, X., Jia, J., & Gong, N. Z. (2022). Prototype-based adversarial training for robust federated learning. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (pp. 427–440).

20. Li, D., & Wang, J. (2019). FedMD: Heterogenous federated learning via model distillation. arXiv preprint arXiv:1910.03581.

21. Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4), 211–407.

22. Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G., Smith, G., & Thorne, B. (2017). Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. arXiv preprint arXiv:1711.10677.

23. Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 10(2), Article 12.

24. Zhan, Y., Zhang, J., Li, P., & Li, K. (2022). A survey of federated learning for edge computing: Architectures, challenges, and applications. IEEE Communications Surveys & Tutorials, 24(3), 1728–1758.