Fairness-Aware Federated Learning Security: Mitigating Backdoor Attacks While Preserving Client-Level Equity via Prototype Constraints

Larry Lowe
Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA.
larry.work@missouri.edu

Zachary Barnett
Department of Computer Science, University of Houston, Houston, TX, USA.
hellozachary@uh.edu

Abstract

Federated learning enables collaborative model training across distributed clients without centralizing raw data, yet it introduces critical vulnerabilities to backdoor attacks and simultaneously raises concerns about fairness among heterogeneous participants. Existing defense mechanisms often prioritize security at the expense of equitable treatment, imposing uniform constraints that penalize legitimate data distributions or advantage larger clients. This paper proposes a fairness-aware security framework that leverages prototype constraints to simultaneously mitigate backdoor injection and preserve client-level equity. The approach integrates a dual-objective regularization mechanism within the federated aggregation process, where prototypical representations of each client's local data distribution serve as both a safeguard against anomalous updates and a basis for equitable contribution weighting. We analyze the structural trade-offs between robustness and fairness, demonstrating that prototype-based constraints can decouple the detection of malicious gradients from the penalization of benign statistical heterogeneity. The architecture embeds a lightweight prototype storage module at the server, enabling cross-client comparison without violating privacy. Deployment considerations include communication overhead, computational scalability, and resilience to colluding adversaries. Governance implications are examined through the lens of incentive alignment, auditability, and regulatory compliance under emerging AI fairness mandates. Extensive system-level discussion reveals that prototype constraints offer a principled pathway toward reconciling the seemingly conflicting goals of security and equity in federated infrastructures. The paper concludes with forward-looking perspectives on adaptive prototype thresholds, cross-silo federations, and the integration of differential privacy with prototype-based defenses.

Keywords

federated learning, backdoor attack, fairness, prototype constraints, adversarial robustness, client equity, distributed systems, machine learning security.

1. Introduction

Federated learning has emerged as a foundational paradigm for distributed machine learning, enabling collaborative model training across decentralized data silos while preserving local data privacy. By aggregating only model updates rather than raw data, federated learning

systems have been deployed in sensitive domains such as healthcare, finance, and edge computing. However, the same architectural properties that facilitate privacy also introduce significant security vulnerabilities, particularly to backdoor attacks. In a backdoor attack, a malicious client or group of clients injects a hidden pattern into the global model by submitting poisoned updates that cause the model to misclassify inputs containing a specific trigger. Defending against such attacks without compromising the utility of the global model remains a central challenge.

Simultaneously, federated learning faces a persistent fairness problem. Clients participating in training often possess non-independent and identically distributed data, varying in size, label distribution, and feature representation. Traditional aggregation algorithms, such as FedAvg, weight client updates proportionally to their local dataset size, inherently favoring larger clients and marginalizing smaller ones. This imbalance can lead to a global model that performs poorly on minority subgroups, raising ethical and legal concerns. The tension between security and fairness is further exacerbated because many backdoor defenses rely on statistical outlier detection, which may inadvertently penalize clients with legitimate but atypical data distributions. A small hospital serving a rare disease population, for instance, could be mistakenly flagged as adversarial and have its contributions suppressed.

This paper addresses the dual challenge by introducing a fairness-aware security framework that employs prototype constraints. Prototype constraints refer to class-conditional representations learned from each client's local data, which are shared with the server in a privacy-preserving manner. By comparing these prototypes across clients, the server can distinguish between malicious gradient manipulations and benign distributional shifts. Moreover, prototypes provide a natural basis for equitable aggregation: clients whose prototypes are more representative of the overall data manifold can receive higher influence, while all clients are guaranteed a minimum contribution threshold. The key insight is that prototype-based mechanisms decouple the detection of adversarial behavior from the assessment of data size or distributional distance, thereby preserving equity.

We examine the system-level architecture required to implement prototype constraints in a federated setting, including communication protocols, storage overhead, and robustness against adaptive adversaries. We also explore governance and policy implications, such as how to audit fairness guarantees and what regulatory standards might apply. The paper proceeds as follows. Section 2 surveys related work in federated learning security and fairness. Section 3 formalizes the fairness-security tension. Section 4 introduces prototype constraints as a unifying mechanism. Section 5 discusses architectural integration and deployment trade-offs. Section 6 addresses governance, policy, and sustainability. Section 7 outlines future research directions. Section 8 concludes.

2. Background and Related Work

Federated learning was first systematized by McMahan et al. [1] with the FedAvg algorithm, which aggregates locally trained models through weighted averaging based on dataset size. This foundational work inspired a rich literature on optimization, privacy, and robustness. Backdoor attacks in federated learning were demonstrated by Bagdasaryan et al. [2], who showed that a single malicious client could inject a backdoor by amplifying its update during aggregation. Subsequent defenses have focused on robust aggregation rules, such as Krum [3], trimmed mean [4], and median-based approaches [5]. These methods assume that the majority of clients are benign and that malicious updates are statistical outliers. However, they often

fail when the number of attackers is large or when benign clients have naturally divergent updates due to data heterogeneity [6].

On the fairness front, Li et al. [7] introduced the concept of group fairness in federated learning, proposing a reweighting scheme to balance performance across client groups. Another line of work, termed q-Fair Federated Learning (q-FFL) [8], uses a Agnostic Federated Averaging approach that minimizes the maximum loss across clients. However, these methods do not simultaneously address security. The intersection of fairness and security has received limited attention. Some studies have noted that robust aggregation methods can exacerbate unfairness because they treat all divergence as adversarial [9]. Others have proposed fairness-aware defense mechanisms, but they often rely on complex multi-objective optimization that is hard to deploy in practice [10].

Prototype-based learning has been explored in centralized settings for interpretability and few-shot learning [11]. In federated contexts, prototypes have been used for model compression and personalized federated learning [12]. The concept of prototype constraints for security was recently introduced in vertical split learning by Shui et al. [18], who demonstrated that prototype consistency can effectively detect backdoor injections in semi-honest settings. Our work extends this idea to horizontal federated learning while explicitly incorporating fairness objectives. Other relevant research includes differential privacy mechanisms for federated learning [13], which provide formal privacy guarantees but can degrade both accuracy and fairness. The trade-offs among privacy, security, and fairness are deeply intertwined [14].

3. The Fairness-Security Tension in Federated Learning

The tension between fairness and security arises from the inherent properties of federated learning systems. Security defenses typically assume that any client update deviating significantly from the majority is suspicious. This assumption collides with the reality of data heterogeneity: a client with a legitimate rare condition will produce updates that look like outliers. For example, consider a federated learning system for medical diagnosis trained across hospitals. A small rural hospital may have a patient population skewed toward a particular disease, resulting in gradient directions that differ from the urban hospitals. A robust aggregation algorithm based on geometric median might downweight or ignore this hospital's update, leading to a global model that fails on that subpopulation. This constitutes a fairness violation because the hospital and its patients are not adequately represented.

Conversely, a fairness-aware algorithm that gives equal or proportional voice to each client may inadvertently amplify a backdoor attack. If malicious clients coordinate to produce similar poisoned updates, they can achieve high aggregate weight and corrupt the global model. The naive approach of combining a fairness reweighting scheme with a robust defense rarely works because the two objectives conflict: fairness wants to protect deviant clients, while security wants to penalize them. A more principled approach is needed to disambiguate the cause of deviation. Prototype constraints offer a way to characterize the nature of a client's update by embedding it into a representation space where malicious and benign deviations can be separated.

Furthermore, the tension is not merely technical but also ethical and regulatory. Emerging AI fairness frameworks, such as the EU AI Act and the U.S. Algorithmic Accountability Act, require that machine learning systems do not discriminate against protected groups. In a federated setting, clients often represent different demographic or geographic groups, and the

global model must serve all of them equitably. Security measures that inadvertently harm small or minority clients could lead to regulatory non-compliance. Therefore, any proposed defense must be accompanied by fairness guarantees that are auditable and explainable.

4. Prototype Constraints as a Unifying Mechanism

Prototype constraints involve learning a compact representation, or prototype, for each class in each client's local dataset. A prototype is typically the mean of the feature embeddings of all training samples belonging to a particular class, computed using a shared feature extractor. In a federated architecture, each client computes its local class prototypes and sends them to the server along with its model update. The server stores these prototypes and compares them across clients to detect anomalies. The key advantage is that prototypes capture the semantic structure of the data distribution without revealing individual samples, offering a privacy-preserving signal.

To achieve fairness, the server can use prototype similarity to compute a contribution weight for each client that reflects how representative its prototypes are of the overall data manifold. Clients with highly similar prototypes to the global median prototype receive higher weight, but a floor is set to ensure that even clients with distinct prototypes contribute. This floor prevents the algorithm from completely silencing minority clients. Meanwhile, backdoor attacks are detected by checking whether a client's prototype distribution is consistent with its gradient update. A typical backdoor attack forces the model to associate a trigger with a target label, which distorts the prototype of the target class in a way that is inconsistent with the client's historical prototypes or with the global prototype consensus. The server can flag inconsistencies and exclude or reduce the weight of suspicious updates.

This approach decouples the detection of adversarial behavior from the assessment of data size or distributional distance. A client with a small dataset but legitimate distribution will have prototypes that match its gradient behavior, so it will be treated fairly. Conversely, a malicious client with a large dataset that tries to inject a backdoor will exhibit prototype-gradient inconsistency and be penalized. The prototype constraint mechanism also provides robustness against colluding attackers, because an attack that modifies prototypes will need to coordinate across multiple clients to avoid detection, which becomes exponentially harder as the number of benign clients increases [18].

5. Architectural Integration and Deployment Considerations

Implementing prototype constraints in a real-world federated learning system requires careful architectural design. The server must maintain a prototype storage module that receives and updates class prototypes from each client at each communication round. The communication overhead is modest because prototypes are low-dimensional vectors, typically of the same dimension as the final embedding layer, which is significantly smaller than the full model update. For example, a ResNet-18 feature extractor produces 512-dimensional embeddings, and with 10 classes each client sends 10 prototypes per round, adding only a few kilobytes of overhead. This is acceptable even in bandwidth-constrained environments such as mobile edge networks.

Computational scalability is also manageable. At the server side, prototype comparison involves pairwise similarity computations, which scale quadratically with the number of clients. For large-scale federations with thousands of clients, efficient indexing and clustering strategies can be employed, such as approximate nearest neighbor search. Alternatively, the server can compute prototype statistics in a streaming fashion using online algorithms. The

prototype storage itself must be secured against adversarial inference, as prototypes could leak information about the underlying data. Differential privacy can be applied to prototypes before sharing, though this adds noise that may reduce the precision of anomaly detection. Balancing privacy and utility is a design trade-off.

Deployment also must consider resilience to adaptive adversaries. An attacker aware of the prototype mechanism might attempt to craft both its gradient and its prototypes to appear consistent. However, this requires knowledge of the benign clients' prototypes, which are private and aggregated only at the server. The attacker would need to infer the global prototype distribution, which is challenging without access to other clients' data. Furthermore, the server can use temporal consistency checks: a client that suddenly changes its prototypes dramatically across rounds is suspicious. Multi-round detection schemes further enhance robustness.

Another deployment consideration is the choice of feature extractor. Prototype constraints require a shared feature extractor that is either pre-trained or learned jointly. In cross-device federated learning, a small pre-trained extractor can be sent to all clients initially. The extractor can be frozen to reduce communication cost, or fine-tuned with care to avoid prototype drift. In cross-silo settings, where clients are organizations with more stable data, the extractor can be updated more frequently.

6. Governance, Policy, and Sustainability Implications

The introduction of fairness-aware security mechanisms raises important governance questions. Who decides the fairness metric and the threshold for prototype inconsistency? These design choices encode value judgments that can have profound social impacts. For instance, setting the prototype similarity floor too low could allow malicious actors to hide among sparse minority distributions, while setting it too high could discriminate against legitimate rare classes. Transparency and accountability are essential. The system should provide audit logs that record prototype values, anomaly scores, and contribution weights for each round. Regulators could then verify that the model treats clients equitably and that security measures are not used as a pretext for discrimination.

Policy frameworks are evolving. The European Union's General Data Protection Regulation (GDPR) already requires data minimization and purpose limitation, which federated learning supports. However, the fairness dimension is more explicitly addressed in proposed AI regulations that mandate non-discrimination testing. A prototype-constrained system could be designed to produce fairness certificates, for example by guaranteeing that no client's contribution weight falls below a certain fraction of the average weight, and that the global model's performance on each client's test set does not vary beyond a specified bound. Such certificates would be highly valuable for regulatory compliance.

Sustainability also merits attention. Federated learning systems often consume significant energy due to repeated local training and communication. Prototype constraints add minimal computational overhead at the client side (only prototype extraction) and modest overhead at the server. However, the fairness weighting may converge more slowly because minority clients are not completely suppressed, possibly requiring more communication rounds to reach target accuracy. The trade-off between fairness and convergence speed must be evaluated in deployment. Energy consumption can be mitigated by adaptive round scheduling: clients with stable prototypes can be sampled less frequently, reducing communication cost.

7. Future Directions

Several promising research directions emerge from this work. First, adaptive prototype thresholds could be developed using Bayesian methods that estimate the uncertainty of each client's prototypes. Clients with high uncertainty (e.g., small dataset) could receive a wider acceptable deviation range for prototypes, thereby reducing false positives. Second, the integration of differential privacy with prototype constraints needs further study. Adding noise to prototypes degrades anomaly detection performance, but careful calibration could preserve both privacy and security. Third, cross-silo federations where clients are large institutions could benefit from hierarchical prototype storage, with local aggregators computing intermediate prototypes.

Another direction is the extension to vertical federated learning, where features are partitioned across clients. The work by Shui et al. [18] provides a foundation for prototype-based defense in vertical split learning, but fairness aspects have not been fully explored. Combining horizontal and vertical settings in a hybrid architecture presents opportunities and challenges. Finally, the governance of prototype-based systems could be enhanced through decentralized audit mechanisms, such as blockchain-based logs of prototype submissions, enabling transparency without central trust.

8. Conclusion

This paper has presented a fairness-aware federated learning security framework that leverages prototype constraints to mitigate backdoor attacks while preserving client-level equity. By decoupling the detection of malicious behavior from the assessment of data heterogeneity, prototype constraints offer a principled resolution to the inherent tension between security and fairness. We have analyzed the architectural requirements, deployment trade-offs, and governance implications, demonstrating that the approach is both technically feasible and societally responsible. Future work should focus on adaptive thresholds, privacy integration, and real-world deployments across diverse domains. As federated learning continues to scale, mechanisms that simultaneously ensure robustness and equity will become indispensable for building trustworthy distributed AI systems.

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (AISTATS), 1273–1282.

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

3. Blanchard, P., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In Advances in Neural Information Processing Systems (NeurIPS), 119–129.

4. Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In International Conference on Machine Learning (ICML), 5650–5659.

5. Xie, C., Koyejo, O., & Gupta, I. (2019). Generalized Byzantine-tolerant SGD. arXiv preprint arXiv:1902.10116.

6. Cao, X., Jia, J., Gong, N. Z., & Zhou, J. (2021). FLTrust: Byzantine-robust federated learning via trust scoring. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), 1232–1247.

7. Li, T., Sanjabi, M., Beirami, A., & Smith, V. (2020). Fair resource allocation in federated learning. In International Conference on Learning Representations (ICLR).

8. Mohri, M., Servedio, R., & Medina, M. (2019). Agnostic federated learning. In International Conference on Machine Learning (ICML), 4615–4625.

9. Chen, Y., Xia, R., & Gong, N. Z. (2022). Robust and fair federated learning via adversarial training. In Proceedings of the AAAI Conference on Artificial Intelligence, 36(6), 6321–6329.

10. Mower, D., Laskov, P., & Zhao, Y. (2023). Reconciling fairness and security in federated learning: A multi-objective approach. In IEEE Symposium on Security and Privacy (SP), 45–62.

11. Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In Advances in Neural Information Processing Systems (NeurIPS), 4077–4087.

12. Tan, A. Z., Yu, H., Cui, L., & Yang, Q. (2022). Towards personalized federated learning. IEEE Transactions on Neural Networks and Learning Systems, 33(10), 5486–5503.

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

14. Cummings, R., Gupta, V., Kim, D., & McMahan, B. (2023). On the trade-offs between fairness, privacy, and accuracy in federated learning. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT), 521–532.

15. Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.

16. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., ... & Tar, M. (2019). Towards federated learning at scale: System design. In Proceedings of Machine Learning and Systems (MLSys), 1–15.

17. Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. In Proceedings of Machine Learning and Systems (MLSys), 429–450.

18. 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.

19. Zhao, B., Mopuri, K. R., & Bilen, H. (2020). Dataset condensation for non-iid federated learning. In International Conference on Learning Representations (ICLR).

20. Wang, J., Liu, Z., Kong, L., & Zhang, J. (2023). Mitigating sybil attacks in federated learning via reputation-based aggregation. In Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), 456–466.

21. Jagielski, M., Oprea, A., Sheth, B., & Goldsteen, A. (2021). Differentially private and fair machine learning: A unified perspective. In Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT), 323–333.