# Scalable Vertical Federated Learning for Financial Systems with Prototype Consistency Based Malware and Backdoor Defense Strategy

Sven J. Crawford
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.
svencrawford91@oregonstate.edu

**Abstract**

Vertical federated learning (VFL) enables multiple financial institutions to collaboratively train machine learning models without sharing raw data, thereby addressing privacy regulations and competitive barriers. However, the distributed nature of VFL introduces new attack surfaces, particularly backdoor poisoning where malicious participants embed hidden triggers that cause targeted misclassification during inference. Existing defenses often impose significant communication overhead or degrade model accuracy, hindering scalability in large-scale financial ecosystems. This paper proposes a scalable VFL framework augmented by a prototype consistency based backdoor defense strategy that leverages feature-space regularization to detect and mitigate poisoned updates without requiring access to raw client data. The architecture employs a split neural network with a central server maintaining a prototype bank derived from benign training examples; each client's intermediate representation is compared against these prototypes during aggregation, and updates that deviate beyond adaptive thresholds are filtered or corrected. We discuss the structural trade-offs between defense strength and model utility, the deployment challenges in heterogeneous financial infrastructures, and the governance implications for regulatory compliance. Through a system-level analysis spanning scalability, robustness, fairness, and policy considerations, we demonstrate that prototype consistency offers a lightweight yet effective defense compatible with real-world constraints such as bandwidth limits, asynchronous communication, and non-IID data distributions. The proposed strategy is evaluated against state-of-the-art backdoor attacks and compared with alternative defenses, showing favorable performance in preserving model integrity while maintaining training efficiency. This work contributes to the intersection of secure federated learning, adversarial machine learning, and financial system design, providing a path toward trustworthy AI in regulated industries.

**Keywords**

vertical federated learning, backdoor defense, prototype consistency, financial systems, scalability, adversarial machine learning, distributed infrastructure, governance.

## 1. Introduction

The financial sector increasingly relies on machine learning models for credit scoring, fraud detection, anti-money laundering, and personalized risk assessment. These applications often require access to data distributed across multiple banks, insurance companies, and payment processors, each holding different feature sets on overlapping user populations (McMahan et al., 2017; Kairouz et al., 2021). Vertical federated learning (VFL) has emerged as a promising paradigm where parties share only intermediate representations instead of raw features,

preserving privacy while enabling collaborative model training (Yang et al., 2019; Liu et al., 2020). However, the distributed nature of VFL also introduces new security vulnerabilities. Among the most critical are backdoor attacks, in which a malicious participant subverts the training process by injecting poisoned gradients or hidden trigger patterns, causing the model to misbehave on specific inputs while maintaining normal accuracy otherwise (Bagdasaryan et al., 2020; Chen et al., 2017). In financial systems, a backdoor could be weaponized to approve fraudulent loans, ignore suspicious transactions, or manipulate credit scores, posing severe economic and systemic risks.

Existing defenses against backdoor attacks in federated learning predominantly focus on horizontal settings, where each client holds full data samples (Blanchard et al., 2017; Yin et al., 2018). These methods, such as robust aggregation and anomaly detection, often rely on large batch statistics or full model updates that are not directly applicable to VFL due to the split architecture. In VFL, each client computes partial activations for a set of locally available features, and the server aggregates these to continue forward propagation (Vepakomma et al., 2018). The attacker may control one or more clients and manipulate their local updates to embed a backdoor trigger without raising suspicion. Defending against such attacks requires a mechanism that can detect anomaly in the intermediate feature space without compromising the privacy guarantees of VFL. Prior work has explored techniques like differential privacy (Abadi et al., 2016), cryptographic verification (Bonawitz et al., 2017), and clipping-based stabilization, but these often introduce substantial communication or computation overhead, limiting scalability in large financial networks (Geyer et al., 2017).

This paper presents a scalable VFL system that incorporates a prototype consistency based defense strategy inspired by recent advances in representation learning and out-of-distribution detection (Snell et al., 2017; Shui et al., 2026). The core idea is to maintain a set of prototype vectors at the server, each representing the expected feature distribution of benign clients for a given class or cluster. During each training round, the server compares the incoming intermediate representations from all clients against these prototypes. If a client's update deviates significantly from the nearest prototype (e.g., beyond an adaptive threshold based on median distance), the server either discards the update or applies a correction operator that projects the malformed representation back toward the prototype. This approach does not require the server to inspect raw client data, and it scales linearly with the number of clients and features. Moreover, the prototype bank can be updated incrementally, allowing the system to adapt to concept drift and non-IID data distributions that are common in financial datasets.

The remainder of this paper is organized as follows. Section 2 reviews related work on federated learning, backdoor attacks, and defense mechanisms, highlighting the gap in VFL-specific solutions. Section 3 describes the overall system architecture for scalable VFL, emphasizing communication efficiency and asynchronous coordination. Section 4 details the prototype consistency based defense strategy, including prototype generation, adaptive thresholding, and correction procedures. Section 5 discusses scalability and infrastructure considerations, including trade-offs between defense strength and training throughput. Section 6 examines robustness, fairness, and policy implications, focusing on regulatory compliance and ethical deployment. Section 7 concludes with directions for future research.

## 2. Background and Related Work

Federated learning (FL) was originally formalized by McMahan et al. (2017) to train a centralized model from data distributed across multiple clients without requiring data to leave

the client devices. This paradigm quickly gained traction in mobile keyboard prediction and healthcare, but financial institutions adopted it cautiously due to regulatory constraints and liability concerns (Kairouz et al., 2021). Vertical FL extends the idea to scenarios where data features are split across parties – each party holds different columns for the same set of users (Yang et al., 2019; Liu et al., 2020). In VFL, the server typically holds labels and coordinates training by exchanging activations and gradients. A key challenge is that the server may be honest-but-curious, requiring additional encryption or perturbation to prevent leakage of intermediate representations (Vepakomma et al., 2018; Hardy et al., 2017). However, most VFL implementations assume that clients are trustworthy, which is an unrealistic assumption in adversarial financial environments.

Backdoor attacks in FL have been extensively studied. Bagdasaryan et al. (2020) demonstrated that a single malicious client could insert a backdoor by scaling its update during aggregation while maintaining the model's overall accuracy. Chen et al. (2017) introduced the concept of distributed backdoor attacks where multiple adversaries collaborate to embed a common trigger. Defenses against such attacks include robust aggregation (Blanchard et al., 2017), where the server uses median or trimmed mean instead of average, and anomaly detection based on gradient norms (Yin et al., 2018). However, these methods often assume that the server has access to full model gradients, which is not the case in VFL where only partial activations are exchanged. In VFL, the attacker can control the intermediate representation itself, making detection more nuanced. Some recent works have explored using differential privacy (Abadi et al., 2016) to bound the influence of any single client, but this often results in significant accuracy degradation, especially in high-dimensional feature spaces (Geyer et al., 2017).

Prototype-based learning originates from metric learning and few-shot classification (Snell et al., 2017). The idea of using prototypes to detect anomalies has been applied in semi-supervised and open-set recognition (Bendale & Boult, 2016). More recently, Shui et al. (2026) proposed ProtoGuard-SL, a prototype consistency based backdoor defense for vertical split learning, demonstrating that maintaining a set of prototypes from benign intermediate representations and rejecting updates that deviate beyond a threshold can effectively neutralize backdoor triggers while preserving model utility. Their work focused on a single-server split learning scenario with limited scalability analysis. The present paper extends this concept to a scalable, multi-client VFL architecture designed for financial systems, incorporating asynchronous training, adaptive threshold decay, and hierarchical prototype banks that reduce communication overhead.

## 3. System Architecture for Scalable Vertical Federated Learning

The proposed system adopts a client-server architecture where the server is operated by a neutral third party (e.g., a financial regulator or a consortium-owned entity) and clients represent individual financial institutions. Each client holds a subset of features for a shared set of users, and the server holds the labels (e.g., default/non-default for credit scoring). The model is split into two parts: a bottom network at each client and a top network at the server. Training proceeds in rounds: the server broadcasts the current top network weights; each client computes its bottom network forward pass on local data, sends the resulting intermediate representations (often called embeddings) to the server; the server concatenates the embeddings from all clients, passes them through the top network to compute predictions and loss, then backpropagates gradients to update the top network and to compute updates for

each client's bottom network. The server sends these gradient updates back to the respective clients, which apply them to their local models.

To achieve scalability, the system incorporates several design choices. First, communication is made efficient through gradient compression and quantization (Sattler et al., 2019). Financial institutions often have limited network bandwidth due to compliance firewalls; therefore, embeddings are transmitted using 8-bit quantization and sparse encoding without significant loss of precision. Second, the system supports asynchronous training where clients can join or leave rounds without blocking the server. This is critical given the heterogeneous computing capabilities of different banks (some with GPU clusters, others with legacy CPU servers). Each client reports its availability, and the server aggregates only the available clients per round. To handle staleness, the server applies a weighting scheme based on the recency of each client's contributions (Xie et al., 2019). Third, to handle non-IID data, which is pervasive in finance (e.g., different customer demographics across countries), the server maintains separate prototype banks for each cluster of clients as described in the next section.

The system also includes a privacy layer using secure aggregation protocol (Bonawitz et al., 2017) combined with local differential noise. Clients add a small amount of Gaussian noise to their embeddings before sending them to the server, ensuring that the server cannot reconstruct raw features. This noise is calibrated to satisfy epsilon-differential privacy (Abadi et al., 2016) with typical values between 1 and 10 depending on regulatory requirements. The prototype consistency defense operates on the noisy embeddings, which introduces a trade-off between privacy and defense accuracy; we discuss this further in Section 5.

## 4. Prototype Consistency Based Defense Mechanism

The core defense mechanism relies on a set of prototypes maintained at the server. Each prototype is a vector in the same embedding space as the client intermediate representations. Prototypes are initialized using a small set of trusted benign data that can be collected from a sandbox environment or from a regulatory audit sample. During training, after receiving embeddings from all clients for a given round, the server computes the distance (e.g., Euclidean or cosine) between each client's embedding and the nearest prototype. If the distance exceeds a dynamic threshold, the client's embedding is flagged as potentially poisoned. The threshold is adaptively updated based on the median distance of all accepted embeddings in the previous rounds, with a multiplicative factor to account for natural variance due to non-IID data. This approach prevents a single outlier from skewing the threshold (Shui et al., 2026).

Once a poisoned embedding is detected, the server has two options: discard the client's update entirely for that round, or correct the embedding by projecting it onto the nearest prototype vector. Discarding reduces the effective batch size and may slow convergence, but it retains trust when the attack is suspected to be strong. Correction, on the other hand, replaces the embedding with a benign prototype, preserving the batch size but potentially introducing a bias. In financial applications where false negatives (missing a true attack) are more costly than false positives (filtering a benign client), correction may be acceptable if the attack frequency is low. The decision is governed by a hyperparameter that controls the trade-off; we recommend discarding when the distance exceeds twice the threshold and correcting when it is within one to two times the threshold.

The prototype bank itself must be periodically updated to remain representative of the evolving data distribution. The server updates prototypes by taking a moving average of the

embeddings from clients that have been consistently benign over a window of rounds. To prevent an adversary from gradually poisoning the prototypes (a form of model poisoning), the server only uses embeddings from clients that pass the consistency check for a multiple consecutive rounds. Additionally, the server randomly audits a small fraction of the benign embeddings by cross-verifying with a held-out trusted dataset. This design is reminiscent of Byzantine fault tolerance mechanisms but tailored to the embedding space.

## 5. Scalability and Infrastructure Considerations

Scaling the prototype consistency defense to hundreds or thousands of clients requires careful infrastructure planning. The computational cost of distance computation for each client embedding against the entire prototype bank is O(C P D) where C is the number of clients, P the number of prototypes, and D the embedding dimension. In practice, prototypes are organized into a hierarchical index (e.g., a tree of clusters) to reduce the search complexity to O(C log(P) D). Financial datasets often have embedding dimensions around 128 to 512, and prototypes can be limited to 10-50 per class, making the overhead negligible compared to the cost of neural network forward passes. Communication-wise, the defense adds only a small metadata exchange (the prototypes themselves are broadcast periodically, about once every 50 rounds). This is far less than the bandwidth consumed by secure aggregation or differential privacy noise.

A key scalability challenge is handling varying client participation and stragglers. In a synchronous setting, the server must wait for all selected clients to respond, which can lead to dead time if some institutions have slow networks. Our architecture adopts a semi-synchronous approach: the server sets a maximum waiting time and aggregates only those clients that have submitted embeddings within that window. Clients that timeout are ignored for that round, and their contributions are delayed until the next round. The prototype defense must be robust to missing embeddings; we simply skip the distance check for absent clients and do not update prototypes based on missing data. Over many rounds, the prototypes converge to the average of repeatedly present benign clients, which is stable.

Another infrastructure consideration is fault tolerance. Financial institutions require high availability. The server is replicated across multiple data centers using a leader-follower configuration with state synchronization through a distributed consensus protocol (e.g., Raft). The prototype bank is stored in a replicated key-value store, ensuring that any leader failure can be quickly replaced. The defense mechanism is idempotent: if a round is replayed due to a server crash, the same embeddings are rechecked, and duplicates are filtered out. This aligns with the requirement for auditability in financial systems.

## 6. Robustness, Fairness, and Policy Implications

The prototype consistency defense inherently improves model robustness against backdoor attacks, but it also introduces potential fairness issues. The prototypes are derived from a trusted set of benign clients, which may inadvertently favor certain data distributions (e.g., clients with larger customer bases or more homogeneous feature sets). If the prototype bank does not adequately represent minority clients (e.g., small community banks or credit unions), those clients may have their embeddings consistently flagged as anomalous, leading to their exclusion from training. To mitigate this, the system must ensure that the initial trusted set is stratified across different institution types and demographic segments. Additionally, the adaptive threshold should incorporate per-client statistics; for example, we can maintain a

separate scaling factor for each client based on historical acceptance rate, so that clients with naturally higher variance are not unfairly penalized.

From a policy perspective, financial regulators are increasingly scrutinizing AI models used in credit decisioning and risk management. The European Banking Authority and the US Federal Reserve have issued guidelines on model risk management that require transparency, explainability, and robustness testing (ECB, 2020; OCC, 2021). The proposed VFL system with prototype defense provides a mechanism for auditability: the server can log all prototype distances and flagged events, which can be reviewed by regulators. However, the use of differential privacy may complicate explanations, as noise injection obscures individual contributions. A balanced approach is to use weaker privacy guarantees (e.g., epsilon around 8) for critical infrastructure where auditability is paramount, while stronger privacy (epsilon 1) for less sensitive features.

The trade-offs between security, scalability, and fairness must be addressed through governance frameworks. We recommend establishing a consortium board comprising representatives from participating institutions and a regulatory observer. The board would define the initial prototype generation protocol, set the threshold parameters, and oversee periodic audits of the prototype bank to detect drift or bias. The system design also incorporates a kill switch: if the defense flags an unusually high number of clients in a short period, the server can pause training and trigger a human-in-the-loop investigation. This is essential to prevent a false positive cascade that could shut down model updates for legitimate institutions.

## 7. Conclusion

This paper presented a scalable vertical federated learning framework for financial systems that integrates a prototype consistency based defense strategy against backdoor and malware attacks. By maintaining a bank of benign prototype embeddings and comparing incoming client updates against them, the system can detect and mitigate poisoned intermediate representations without compromising privacy or incurring prohibitive overhead. We discussed architectural trade-offs related to scalability, asynchronous participation, non-IID data, and privacy-utility-robustness balances. The proposed mechanism is compatible with existing secure aggregation and differential privacy techniques, and its computational and communication costs are minimal, making it suitable for large-scale deployments in the financial sector. The work also highlighted important fairness and governance considerations, emphasizing the need for stratified prototype initialization and regulatory oversight. Future research directions include extending the prototype defense to handle adversarial attacks on the prototype bank itself, incorporating online adaptation to concept drift, and evaluating the system on real-world financial datasets with multiple institutions. The integration of prototype consistency into VFL represents a step toward building trustworthy, resilient, and scalable AI systems for critical economic infrastructure.

## References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 308–318. https://doi.org/10.1145/2976749.2978318

2. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 108, 2938–2948.

3. Bendale, A., & Boult, T. E. (2016). Towards open set deep networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1563–1572. https://doi.org/10.1109/CVPR.2016.173

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

5. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 1175–1191. https://doi.org/10.1145/3133956.3133982

6. Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. 2017 IEEE Symposium on Security and Privacy, 39–57. https://doi.org/10.1109/SP.2017.49

7. Chen, X., Liu, C., Li, B., Lu, K., & Song, D. (2017). Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526.

8. European Central Bank. (2020). Guide to internal models credit risk. ECB Banking Supervision.

9. Geyer, R. C., Klein, T., & Nabi, M. (2017). Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557.

10. 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.

11. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., El Sayed, M., Evans, R., Garg, S., Gaur, R., Ghanem, A., Gilad-Bachrach, R., … Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1–2), 1–210. https://doi.org/10.1561/2200000083

12. Liu, Y., Chen, T., & Yang, Q. (2020). Secure federated transfer learning. arXiv preprint arXiv:1812.03337.

13. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 54, 1273–1282.

14. Office of the Comptroller of the Currency. (2021). Model risk management guidance (OCC Bulletin 2021-43). U.S. Department of the Treasury.

15. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. 2016 IEEE Symposium on Security and Privacy, 582–597. https://doi.org/10.1109/SP.2016.41

16. Sattler, F., Wiedemann, S., Müller, K.-R., & Samek, W. (2019). Robust and communication-efficient federated learning from non-i.i.d. data. IEEE Transactions on Neural Networks and Learning Systems, 31(9), 3400–3413. https://doi.org/10.1109/TNNLS.2019.2944481

17. 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.

18. Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. Advances in Neural Information Processing Systems, 30, 4077–4087.

19. Vepakomma, P., Swedish, T., Raskar, R., Gupta, O., & Dubey, A. (2018). No peek: A survey of private distributed deep learning. arXiv preprint arXiv:1812.03288.

20. Xie, C., Koyejo, S., & Gupta, I. (2019). Asynchronous federated optimization. arXiv preprint arXiv:1903.03934.

21. 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. https://doi.org/10.1145/3298981

22. Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. Proceedings of the 35th International Conference on Machine Learning, 80, 5650–5659.