International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| Volume 5, Issue 2, 108-114, 2024
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V5I2P111

Original Article

Federated Learning for Privacy Preserving Fraud Detection across Financial Institutions: Architecture Protocols and Operational Governance

Rakesh Reddy Thalakanti[1], Sai Santosh Goud Bandari[2], Sai Dheeraj Sivva[3]
[1]Senior Software Engineer, Goldman Sachs, Dallas, Texas, USA.
[2]Developer TCS North Carolina, USA.
[3]Software Engineer, Independent Researcher, Charlotte, NC, USA.

Abstract - *Fraud detection is a shared problem across banks payment processors and fintech partners yet high quality models often require cross institution signals that cannot be centralized due to privacy regulation contractual constraints and competitive boundaries. Federated learning enables collaborative model training without moving raw data by sending model updates rather than records. However practical deployment in financial networks requires more than basic model averaging. It requires robust privacy protection against inference, strong security controls against malicious participants, a design that addresses non independent data distributions and a governance model that aligns with risk and compliance. This paper proposes an architecture led framework for privacy preserving federated fraud detection across financial institutions. We describe system components for horizontal and vertical federation, secure aggregation to protect individual client updates and differential privacy to bound leakage. We present model design options including tabular models representation learning and federated graph learning for transaction relationship signals. We then define an operational lifecycle that covers onboarding, auditability, drift detection, incident response and model rollback. Finally we outline an evaluation plan that measures detection lift, fairness, privacy, resilience and cost. The goal is a blueprint that allows banks and fintech partners to share learning value while preserving customer privacy and supporting regulatory defensibility.*

Keywords - *Federated Learning, Fraud Detection, Privacy Secure, Aggregation Differential, Privacy Financial, Crime Anti Money Laundering, Credit Card Fraud Governance.*

1. Introduction

Financial fraud and related crimes evolve quickly. Attackers adapt to new controls, exploit new payment rails and coordinate across accounts and institutions. Each institution observes only a partial view of the adversary, which limits detection and increases false positives. Cross institution collaboration can improve detection by learning patterns that generalize across merchant networks devices geographies and payment channels. Yet collaboration is constrained by privacy law, confidentiality commitments and competitive realities. Even when data sharing is legally possible the operational overhead of building joint data lakes, harmonizing schemas and obtaining approvals can delay action until the threat has already shifted.

Federated learning offers a path to collaborate without centralizing raw records. Clients such as banks or fintech partners train locally on their own data and share updates that are aggregated into a global model. A comprehensive monograph on federated learning highlights challenges including system heterogeneity, statistical heterogeneity, privacy risks and robustness [1]. In finance these challenges are amplified because fraud labels are sparse and delayed, class imbalance is extreme and institutions vary in product mix and customer behavior.

This paper addresses the question: what architecture enables privacy preserving federated fraud detection across banks and fintech partners while remaining practical to deploy and govern. We focus on horizontal federation where institutions share similar feature spaces and vertical federation where institutions contribute different attributes for overlapping entities. We treat privacy as a system property rather than a single mechanism. Secure aggregation protects updates in transit and at the coordinator [2]. Differential privacy bounds information leakage from model updates and model outputs [3]. We also consider robustness of the learning process under adversarial participants and operational resiliency during training and serving. The remainder of this paper is organized as follows. Section II reviews relevant concepts and prior work. Section III defines requirements. Section IV proposes a reference architecture. Section V discusses model design options. Section VI details protocol flows and security analysis. Section VII covers governance and operating model. Section VIII describes evaluation methods and a realistic case study outline. Section IX concludes.

2. Background and Related Work

2.1. Federated learning foundations

Federated learning trains a shared model from decentralized data by coordinating rounds of local training and aggregation. Modern federated learning includes methods for personalization, fairness and robustness. The work by Kairouz and coauthors consolidates methods, threat surfaces and open problems and is widely used as a foundation for system design [1]. For financial institutions the key implications are that coordination cost must be minimized, participant availability varies and data distributions differ by institution and time.

2.2. Secure aggregation and confidentiality of updates

Even if raw data is not shared, model updates can leak information about local records or local business patterns. Secure aggregation addresses this by enabling the coordinator to compute only the sum of client updates without seeing any individual update. Bonawitz and coauthors present a practical protocol that is communication efficient and tolerant to participant dropouts [2]. This protocol supports confidentiality between consortium members because no member can inspect another member update and the coordinator learns only aggregates.

2.3. Differential privacy for federated learning

Differential privacy provides a formal framework to bound information leakage about individual records. In federated learning it is commonly applied by clipping updates and adding noise either at the client or at the server. A practical view of this tradeoff is described in the differential privacy approach for federated learning by Tayyeh and AL Jumaili which emphasizes balancing privacy with model utility [3]. For fraud detection, strong privacy budgets can reduce detection power, so privacy design must be aligned with risk tiering and the sensitivity of feature sets.

2.4. Federated fraud detection research

Federated learning has been applied to fraud detection with increasing depth. Yang and colleagues propose FFD where banks train a fraud detection model with local behavior features and aggregate parameters to construct a shared detector without sharing datasets [5]. Abdul Salam and colleagues evaluate federated learning for credit card fraud detection with deep learning, highlighting the impact of sampling and class imbalance controls in federated settings [6]. Tang and colleagues propose a federated graph learning approach for credit card fraud detection in Expert Systems with Applications and show how graph structure can improve detection while preserving data security [4]. These works motivate a system architecture that can support multiple model families and strong governance.

2.5. Attacks and defenses

Federated systems face threats such as update poisoning, backdoor insertion, sybil participants and inference attacks. A survey in ACM Computing Surveys summarizes privacy attacks and defenses and also discusses policy and deployment considerations [7]. In a bank consortium the threat model can include compromised participant infrastructure, insider misuse, or attempts to learn competitive information through model behavior. The architecture must address both technical attacks and governance failures.

3. Problem Definition and Requirements

We consider a consortium of N financial institutions that wish to improve detection for transactions, account activity, authentication events or anti money laundering signals. Each institution has local datasets containing event features and labels. Labels may represent confirmed fraud chargebacks suspicious activity reports, or verified account takeover. The goal is to produce a shared model that improves detection at each institution without sharing raw customer records.

3.1. We define five requirement categories.

- Privacy and confidentiality: raw records must remain inside each institution. Model updates must not reveal sensitive customer information beyond agreed leakage bounds. Update confidentiality between consortium members is desirable to reduce competitive risk.
- Security and robustness: the training process must tolerate malicious or faulty participants and resist poisoning. The system must support authenticated participation, secure communication and integrity of global results.
- Accuracy and fairness: the model must maintain high recall under low false positive budgets. Performance should be stable across institutions and across subpopulations. Imbalance mitigation must be label safe.
- Operational governance: the system must support onboarding, audit trails, approvals, monitoring, incident response and rollback.
- Scalability and cost: computation and network cost must scale with participants and model size. Coordination must minimize friction so the consortium can operate on realistic cadences.

4. Reference Architecture

4.1. Roles

The architecture defines four roles: coordinator, participant institution, governance authority and optional secure compute operator. The coordinator orchestrates training rounds and hosts the global model service. Participant institutions train locally and serve the resulting model for their own transaction streams. The governance authority defines policy, privacy budgets, release criteria and risk controls. The secure compute operator is optional and may host additional cryptographic services or trusted execution environments.

4.2. Components

The system is composed of eight components.

- Local feature service transforms local events into a shared feature contract plus optional local extensions. It supports feature versioning and transformation logs.
- Local label service manages delayed labels and supports label corrections. It stores provenance so model risk teams can explain label sources.
- Local trainer performs local learning for each round and outputs a signed update.
- Privacy module performs clipping, noise addition and privacy accounting for each institution.
- Secure aggregation gateway implements secure aggregation so the coordinator can recover only aggregate updates [2].
- Global model service applies aggregated updates, maintains versioned checkpoints and publishes signed artifacts.
- Monitoring and drift service captures performance drift, data drift, privacy budget usage and security anomalies.
- Governance workflow manages approvals, exception handling and incident response, integrating with change management systems.

4.3. Feature contract and data minimization

A key system decision is the shared feature contract. The contract should include semantic definitions, allowed ranges, missing value rules and encoding guidelines. To reduce privacy risk the contract should avoid direct identifiers and should use aggregated or binned representations for sensitive values. Examples include amount bucket, merchant category, authentication method, transaction velocity counters, time of day bucket and device risk score bucket. Institutions can retain richer local features for local decisioning but the global model should rely on the shared contract for portability. Contract changes must be versioned. Each training round declares the schema version. Backwards compatibility is needed because not all institutions upgrade simultaneously. New fields should be optional and missingness should be modeled explicitly so earlier participants can still contribute.

4.4. Federation modes

Horizontal federation is used when participants share a similar feature space. This is common for payment fraud and credit card fraud. FFD is an example and shows shared model benefits without dataset sharing [5].

Vertical federation is used when participants have different feature subsets for overlapping entities, for example when a bank sees authorization details while a payment gateway sees device telemetry. Vertical federation requires entity alignment which can be supported by privacy preserving record linkage and secure computation. Because vertical federation has a higher risk profile it should start as a limited pilot with strict governance.

5. Protocol Flows

5.1. Training round sequence

A training round proceeds as follows.

- Step 1: the coordinator publishes a round configuration that includes the current global model, feature contract version, learning rate schedule, privacy parameters and minimum participation thresholds.
- Step 2: each participant selects a local training window. The window is aligned to label availability, for example training on the most recent thirty days of events with labels that have matured. Participants compute local gradients or parameter deltas using their local trainer.
- Step 3: each participant applies clipping then applies a differential privacy mechanism based on policy. The participant signs the update and submits it through the secure aggregation gateway.
- Step 4: secure aggregation runs and the coordinator recovers only an aggregated update across participants [2]. The coordinator verifies round thresholds are met and logs participation metadata.
- Step 5: the coordinator applies the aggregated update to produce a new global model checkpoint. The checkpoint is evaluated on shared validation suites where feasible.
- Step 6: the coordinator publishes the checkpoint to participants for local evaluation and potential staged rollout.

5.2. Secure aggregation design choices

Secure aggregation hides individual updates but it introduces operational decisions. A minimum K participants should contribute per round to reduce the chance that aggregated updates can be inverted or attributed. The system should prevent a round from completing when K is not met unless a documented exception is approved. Dropout tolerance is required because institutions may fail to complete a round due to operational incidents. The practical secure aggregation protocol supports such failures while still enabling aggregate recovery [2]. The coordinator should avoid aggregating by small subgroups because this reduces anonymity sets. If subgroup models are needed, they should be trained only when subgroup participant counts remain large enough and privacy analysis supports the design.

5.3. Differential privacy deployment patterns

There are two primary deployment patterns for differential privacy. Client privacy pattern: each participant adds noise locally before secure aggregation. This reduces trust requirements on the coordinator and limits leakage even if the coordinator is compromised. However it can reduce utility when participant datasets are small or when fraud labels are rare.

Server privacy pattern: participants submit clipped updates through secure aggregation then the coordinator adds noise to the aggregated update. This can provide better utility when many participants contribute. It requires that governance trusts the coordinator to apply mechanisms correctly and to retain evidence of privacy accounting. Privacy budgeting is cumulative across rounds. The system must track privacy consumption and stop training when the budget is exhausted. Practical guidance for balancing privacy and performance is discussed by Tayyeh and AL Jumaili and can inform selection of clipping norms and noise multipliers [3].

5.4. Robustness controls and anomaly detection

Fraud consortia must assume a fraction of participants can be compromised. Robustness controls are implemented at three layers.

- Protocol layer: authenticated clients, signed updates and strict participation policies prevent unauthorized or repeated sybil participation.
- Aggregation layer: robust aggregation reduces the impact of outliers. Examples include trimmed mean and median based aggregators. When secure aggregation is used, robust aggregation can be implemented with additional secure computation or with statistical checks that do not require per client inspection.
- Validation layer: candidate global models are evaluated on shared challenge suites and on per participant holdout sets. Large deviations trigger investigation and potential exclusion of a participant from future rounds.

6. Model Design for Fraud Detection

6.1. Tabular models

Fraud detection often relies on tabular signals including transaction features velocity features and risk scores. Gradient boosted trees are common in central settings but can be harder to federate. Compact neural models with embeddings for categorical variables can be trained effectively in federated settings and support secure aggregation well. Abdul Salam and colleagues show that deep learning models trained under federated learning can deliver useful performance for credit card fraud detection when imbalance is handled carefully [6].

Because false positives create customer friction, calibration is essential. The system should support local calibration curves and threshold selection within governance constraints.

6.2. Label delay and semi supervised enrichment

Fraud labels can arrive weeks after events. This creates a mismatch between training data and live behavior. A practical approach is to train representations using self supervised objectives on recent unlabeled events then train the classifier head on matured labeled data. The shared representation can be federated while still keeping raw events local. A related approach is to treat fraud detection as anomaly detection with human feedback. Institutions can share learning from confirmed investigations without sharing cases by encoding outcomes as labels and training in federated rounds.

6.3. Federated graph learning

Graph features capture relational patterns such as shared devices shared merchants and repeated transaction paths. Tang and colleagues propose a federated graph learning method for credit card fraud detection that combines federated learning with graph neural networks and aims to support collaboration with data security [4]. In the proposed architecture each institution builds a local graph where nodes represent accounts merchants devices and cards while edges represent interactions. The institution computes graph embeddings locally and trains a classifier jointly through federated rounds. Cross institution graph linkage can add value but introduces risk. A staged strategy is recommended. Stage one uses local graphs and shared embedding model parameters. Stage two introduces selective linkage for a small set of hashed identifiers under strict governance. Stage three can introduce secure computation for joint neighbor aggregation if the consortium has the required cryptographic maturity.

6.4. Personalization and local policy

Institutions differ in risk tolerance and product mix. A single global model can be a strong starting point but local personalization may be needed. The architecture supports a shared backbone with local calibration and policy thresholds. This design preserves shared learning while allowing each institution to tune decisions to operational constraints.

7. Governance Operating Model

7.1. Consortium governance and model risk

A federated consortium requires governance that covers membership rules, data use restrictions, security controls, privacy budgets and model release management. A governance authority should define a model risk framework for federated models that includes documentation of feature contracts, training configurations, privacy parameters and validation evidence. Model release is treated as a controlled change. Releases require approvals based on performance lift, stability, bias assessment and privacy compliance. All artifacts are signed and stored with code hashes and configuration hashes for reproducibility.

7.2. Auditability and regulator defensibility

Regulators and internal audit teams require evidence for how decisions are made. The architecture supports auditability by keeping immutable logs for training rounds, participation metadata, privacy accounting reports and model evaluation results. Explainability is supported through feature importance summaries and reason code mappings. A privacy attack and defense survey emphasizes the importance of policy alignment and transparency for deployment decisions [7].

7.3. Incident response and rollback

If a model causes unexpected false positives or suspected poisoning the coordinator can trigger an incident response. Actions include pausing rounds, excluding a participant, rolling back to a prior checkpoint and issuing an advisory to members. Because each institution serves the model locally, institutions can also roll back independently. Incident playbooks should define required communication, decision authority and post incident analysis.

8. Evaluation Methodology and Case Study Outline

8.1. Metrics

Evaluation metrics should include AUC, area under precision recall curve, recall at fixed false positive rate and expected loss reduction under institution specific cost models. Per institution lift over a local baseline is essential because benefits may vary by data volume and fraud mix.

Privacy metrics include differential privacy parameters and empirical tests such as membership inference against the released model. Security metrics include robustness under simulated poisoning and backdoor attempts. Operational metrics include training round success rate, time per round and network cost.

8.2. Case study outline for a consortium pilot

Consider a pilot with five institutions consisting of two banks, one payment processor and two fintech partners. Each participant contributes a shared feature contract of one hundred features and trains a compact neural classifier. The pilot uses secure aggregation with a minimum participation of five so that each round aggregates across all members [2]. Differential privacy is applied using a moderate noise multiplier guided by the privacy performance balance described in [3].

Baseline models are local only models trained independently at each institution. The pilot trains a shared model for eight rounds per month. Each institution evaluates the shared model on a local holdout set and reports metrics. If the shared model improves recall at a fixed false positive rate, institutions then test a staged rollout on a small fraction of live traffic.

Graph features are introduced in month two. Each institution builds a local graph of accounts merchants and devices. Graph embeddings are learned locally and the embedding model parameters are updated through federated rounds based on the federated graph learning approach [4]. Institutions compare lift from graph enriched features to the tabular only baseline.

8.3. Expected benefits and limits

Prior federated fraud detection work suggests that shared learning can improve detection for participants with smaller datasets or narrow fraud coverage. FFD presents a foundational example of such benefits [5]. Deep learning based federated fraud studies show that careful imbalance management can further improve performance [6]. Graph based federation can improve detection for coordinated fraud patterns that are poorly captured by independent features [4].

Limits include label inconsistency across institutions, differences in fraud definitions and varied customer friction policies. These limits motivate a governance process that harmonizes label definitions and allows local thresholding while keeping the shared model stable.

9. Conclusion

Federated learning provides a practical way for banks and fintech partners to collaborate on fraud detection while keeping customer data local. To be viable at enterprise scale it must be supported by secure aggregation, differential privacy, robustness controls and an operating model that is auditable and regulator ready. This paper proposed a reference architecture and detailed protocol flows, model options and governance practices. The framework aims to enable measurable detection lift with bounded privacy risk and manageable operational overhead across a multi institution network.

References

[1] P. Kairouz et al., "Advances and Open Problems in Federated Learning," Foundations and Trends in Machine Learning, vol. 14, no. 1 2, pp. 1 210, 2021, doi: 10.1561/2200000083.

[2] K. Bonawitz et al., "Practical Secure Aggregation for Privacy Preserving Machine Learning," in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017, doi: 10.1145/3133956.3133982.

[3] Wang, L., Jia, R., & Song, D. (2020). D2P-Fed: Differentially private federated learning with efficient communication. arXiv. https://doi.org/10.48550/arXiv.2006.13039

[4] Gunda, S. K. G. (2023). The Future of Software Development and the Expanding Role of ML Models. International Journal of Emerging Research in Engineering and Technology, 4(2), 126-129. https://doi.org/10.63282/3050-922X.IJERET-V4I2P113

[5] W. Yang et al., "FFD: A Federated Learning Based Method for Credit Card Fraud Detection," in Advances in Knowledge Discovery and Data Mining, 2019, doi: 10.1007/978-3-030-23551-2_2.

[6] Zheng, W., Yan, L., Gou, C., & Wang, F.-Y. (2020). Federated meta-learning for fraudulent credit card detection. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI) (pp. 4654–4660). https://doi.org/10.24963/ijcai.2020/642

[7] Pan, Z., Wang, G., Li, Z., Chen, L., Bian, Y., & Lai, Z. (2023). 2SFGL: A simple and robust protocol for federated graph-based fraud detection. arXiv. https://doi.org/10.48550/arXiv.2310.08335

[8] N. F. Aurna et al., "Federated Learning Based Credit Card Fraud Detection: Performance Analysis with Sampling Methods and Deep Learning Algorithms," in 2023 IEEE International Conference on Cyber Security and Resilience, 2023, doi: 10.1109/CSR57506.2023.10224978.

[9] Gunda SK, Yettapu SDR, Bodakunti S, Bikki SB. Decision Intelligence Methodology for AI-Driven Agile Software Lifecycle Governance and Architecture-Centered Project Management, 2023 Mar. 30;4(1):102-8. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I1P112

[10] H. Brendan McMahan et al., "Learning Differentially Private Recurrent Language Models," in International Conference on Learning Representations, 2018, doi: 10.48550/arXiv.1710.06963.

[11] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in 2017 IEEE Symposium on Security and Privacy, 2017, doi: 10.1109/SP.2017.41.

[12] L. Song, R. Shokri and P. Mittal, "Privacy Risks of Securing Machine Learning Models Against Adversarial Examples," in Proceedings of ACM CCS, 2017, doi: 10.1145/3133956.3134066.

[13] J. Konečný, H. B. McMahan, D. Ramage and P. Richtárik, "Federated Optimization: Distributed Machine Learning for On Device Intelligence," 2016, doi: 10.48550/arXiv.1610.02527.

[14] Alsaeedi, A., & Khan, M. (2019). Software defect prediction using supervised machine learning and ensemble techniques: A comparative study. Journal of Software Engineering and Applications, 12, 85–100. https://doi.org/10.4236/jsea.2019.125007

[15] M. Jagielski et al., "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning," in 2018 IEEE Symposium on Security and Privacy, 2018, doi: 10.1109/SP.2018.00021.

[16] P. Blanchard, E. M. El Mhamdi, R. Guerraoui and J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," in NeurIPS, 2017, doi: 10.48550/arXiv.1703.02757.

[17] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin and V. Shmatikov, "How To Backdoor Federated Learning," in Proceedings of AISTATS, 2020, doi: 10.48550/arXiv.1807.00459.

[18] Shah, H., & Sahatiya, P. (2022). A comparative analysis study of software defect prediction using machine learning algorithms on NASA datasets. Journal of Harbin Institute of Technology, 54(9), 67–76. Retrieved from http://hebgydxxb.periodicales.com/index.php/JHIT/article/view/1301

[19] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated Learning for Emoji Prediction in a Mobile Keyboard," *arXiv preprint*, 2019, doi: 10.48550/arXiv.1906.04329.

[20] H. Li, K. Ota, and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018, doi: 10.1109/MNET.2018.1700202.

[21] Al-Nawaiseh, O., & Hammad, M. (2022). Software defect prediction using stacking generalization of optimized tree-based ensembles. Applied Sciences, 12(9), 4577. https://doi.org/10.3390/app12094577

[22] Z. Ma, H. Hu, Y. Li, and J. Liu, "Privacy-Enhancing Technologies in Federated Learning: A Survey," *Information Fusion*, vol. 92, 2023, doi: 10.1016/j.inffus.2022.08.016.

[23] L. Xu, W. Wang, and Z. Qin, "FedSafe: A Federated Learning Framework for Secure and Fair Model Training," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4123–4135, 2023, doi: 10.1109/TIFS.2023.3284442.

[24] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint*, 2017, doi: 10.48550/arXiv.1704.04861.

[25] S. Wang, T. Tuor, T. Salonidis, K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019, doi: 10.1109/JSAC.2019.2904348.

[26] M. Chen, Z. Yang, W. Saad, C. Yin, and M. Shikh-Bahaei, "A Joint Learning and Communications Framework for Federated Learning over Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2021, doi: 10.1109/TWC.2020.3024620.

[27] Shah, H., & Sahatiya, P. (2022). A comparative analysis study of software defect prediction using machine learning algorithms on NASA dataset. Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology, 54(9), 67–76. Retrieved from http://hebgydxxb.periodicales.com/index.php/JHIT/article/view/1301