

Integrating Blockchain and AI for Secure and Scalable Financial Transaction Monitoring

Ozgur Gurbuz and Mohammad Salam

Department of Computer Science, Chicago State University

ogurbuz@csu.edu

msalam@csu.edu

<https://doi.org/10.68414/LRCJ3332>

Abstract

The global financial system processes trillions of dollars in transactions daily, creating an urgent need for monitoring systems that are secure, scalable, privacy-preserving, and capable of detecting sophisticated fraud in real time. Despite a decade of development, existing approaches are fundamentally fragmented: rule-based anti-money laundering (AML) systems generate excessive false positives, centralized machine learning models introduce data privacy risks, and legacy database architectures create single points of failure. This paper proposes and evaluates an integrated framework that combines permissioned blockchain infrastructure with federated machine learning to address these limitations holistically. The proposed architecture anchors transaction records on a Hyperledger Fabric blockchain, reducing reliance on centralized points of failure and ensuring tamper-evident auditability. Concurrently, federated gradient-boosted and graph neural network (GNN) models detect anomalous behavior without requiring raw transaction data to leave institutional boundaries. Experimental results on the PaySim and IEEE-CIS datasets show that the federated GNN component achieves an AUC-ROC of 0.961 and a recall of 93.4% on fraudulent transactions, outperforming isolated institutional models and slightly exceeding the centralized LightGBM baseline. Blockchain benchmarking shows 340 ms average commit latency at over 2,000 TPS, and smart contract rules reduce manual AML alert review by an estimated 41% in simulation. Together, these results demonstrate that integrating blockchain-based auditability with privacy-preserving federated learning enables a unified, scalable, and effective architecture for next-generation financial transaction monitoring.

Keywords: blockchain, federated learning, fraud detection, anti-money laundering, graph neural networks, Hyperledger Fabric, smart contracts, anomaly detection, differential privacy

Introduction

Financial institutions process trillions of dollars in daily cross-border transactions through payment networks such as SWIFT, Fedwire, and CHIPS (Bank for International Settlements [BIS], 2025). The integrity of this infrastructure is continuously threatened by sophisticated fraud schemes, including card-not-present fraud, account takeover attacks, trade-based money laundering, and synthetic identity fraud. The Financial Action Task Force (FATF) estimates that global money laundering flows account for approximately 2–5% of global GDP annually, representing hundreds of billions to trillions of dollars in illicit financial activity (FATF, 2023). Despite substantial investment in compliance infrastructure, financial institutions continue to incur significant AML-related regulatory fines, highlighting persistent limitations in existing monitoring and detection approaches (Brennan & Tang, 2022).

Current financial transaction monitoring systems suffer from three interrelated problems. First, rule-based AML systems, which define thresholds and typologies manually, can generate substantial volumes of false-positive alerts in practice. Compliance analysts may spend significant time reviewing alerts that do not correspond to genuine suspicious activity, creating operational costs and alert fatigue and increasing the risk that genuine threats may be overlooked. Second, machine learning models trained on centralized transaction datasets pose significant privacy and regulatory risks. Banking regulations including GDPR, CCPA, and the Bank Secrecy Act impose strict constraints on data sharing between institutions, preventing the collaborative learning that would allow models to generalize across different fraud typologies observed at different banks. Third, centralized transaction databases represent single points of failure and are vulnerable to insider manipulation, a concern that is particularly acute given that a significant proportion of financial fraud involves internal actors.

Blockchain technology and federated machine learning offer complementary solutions to these problems. Blockchain provides a distributed, append-only, cryptographically secured ledger that mitigates centralized failure risks, creates tamper-evident audit trails, and enforces compliance rules through smart contracts without relying exclusively on a centralized intermediary. Federated learning enables machine learning models to be trained collaboratively across multiple institutions while keeping raw transaction data within each institution's security perimeter. This approach addresses both regulatory constraints and privacy concerns.

This paper makes the following specific contributions:

- A unified architecture (BlockFedAML) that integrates permissioned Hyperledger Fabric blockchain with federated gradient boosting and graph neural network models for financial transaction monitoring.
- A smart contract compliance engine that automates AML rule evaluation and suspicious activity report (SAR) generation.
- Experimental evaluation on two benchmark financial fraud datasets demonstrating that the federated GNN achieves 93.4% recall on fraudulent transactions with AUC-ROC of 0.961.
- Blockchain performance benchmarking showing 340 ms average batch commit latency at throughput levels exceeding 2,000 TPS under the proposed Fabric configuration.
- Analysis of privacy preservation guarantees under differential privacy and secure aggregation protocols.

Background and Related Work

Blockchain in Financial Systems

Blockchain technology demonstrated that peer-to-peer value transfer could be achieved without a centralized intermediary (Nakamoto, 2008), but public networks such as Bitcoin and Ethereum are unsuitable for regulated financial applications due to low throughput (7–30 TPS) and exposure of transaction metadata to all participants (Croman et al., 2016). Permissioned frameworks — particularly Hyperledger Fabric — address these limitations by restricting participation to authenticated entities and enabling configurable privacy controls. Properly configured Fabric deployments achieve throughput ranging from several thousand to tens of thousands of TPS depending on endorsement policy and hardware (Androulaki et al., 2018), making them viable for enterprise financial infrastructure.

Research on blockchain for AML and transaction monitoring has primarily focused on auditability and regulatory automation. Cheng et al. (2023) demonstrated that smart contracts could automate regulatory reporting for cross-border payments, significantly reducing latency. However, most blockchain-based AML systems rely on predefined rule-based logic and lack adaptive machine learning capabilities, a critical gap in environments where adversaries continuously modify strategies to evade static detection rules.

Machine Learning for Fraud Detection

ML approaches to fraud detection have evolved from early statistical classifiers — where SVMs and random forests outperformed logistic regression on imbalanced datasets (Bhattacharyya et al., 2011) — to gradient boosting frameworks such as XGBoost and LightGBM, which have become dominant for tabular financial data due to their predictive

performance and computational efficiency (Ke et al., 2017). However, tabular models fail to capture the relational structure underlying complex fraud schemes. Money laundering layering operations, for example, involve rapid fund transfers across chains of accounts that are difficult to detect through isolated transaction analysis.

Graph Neural Networks (GNNs) address this limitation by learning representations from transaction graph structure. Liu et al. (2021) proposed the Pick-and-Choose GNN framework, which selectively aggregates neighborhood information to reduce noise from benign neighboring nodes, achieving state-of-the-art performance on large-scale fraud datasets. Despite these advances, most fraud detection studies evaluate models in centralized settings where data from all institutions is pooled — an assumption that regulatory and privacy constraints make infeasible in practice.

Federated Learning for Privacy-Preserving Analytics

Federated learning, introduced by McMahan et al. (2017), enables institutions to collaboratively train models by exchanging parameters rather than raw data — an approach well-suited to financial services where GDPR and banking secrecy laws restrict cross-institutional data transfer. Yang et al. (2019) showed that federated credit risk models achieved performance close to centralized baselines while satisfying data localization requirements. Federated learning has also been explored for financial crime and AML-related detection, with the potential to identify patterns distributed across institutions that no single institution could observe alone. Privacy guarantees are further strengthened through differential privacy, which adds calibrated noise to model updates, and secure aggregation, which prevents the coordinator from observing individual institution parameters (Bonawitz et al., 2017), though stronger privacy typically reduces predictive performance.

Limitations of Existing Hybrid Approaches

Blockchain-based AML systems and federated ML approaches have each shown promise independently, but integrated architectures remain rare. Blockchain AML systems emphasize immutable audit trails and rule enforcement while relying on static contract logic; federated ML studies focus on predictive accuracy without addressing tamper-evident logging or automated compliance reporting. To the best of the authors' knowledge, no prior framework has comprehensively integrated permissioned blockchain auditability, privacy-preserving collaborative learning, automated compliance enforcement, and graph-based anomaly detection

within a single coordinated system. This gap motivates the BlockFedAML framework proposed in this paper.

Problem Statement and Research Objectives

This paper addresses the absence of a unified, deployment-oriented architecture for financial transaction monitoring that satisfies four requirements rarely integrated within a single system:

- **Tamper-evident auditability:** Transaction records and compliance decisions must be cryptographically secured against post-hoc modification, including by internal actors.
- **Privacy-preserving collaborative intelligence:** Fraud detection models must improve from data distributed across multiple regulated institutions without raw data leaving any institution's security boundary.
- **Real-time adaptive anomaly detection:** The system must detect both known fraud typologies (addressed by rule-based systems) and novel, emerging fraud patterns (requiring adaptive ML).
- **Automated regulatory compliance:** The system must automate the evaluation of AML regulations, KYC requirements, and SAR filing obligations to reduce manual compliance burden.

This paper addresses these requirements through the following research objectives:

- **RO1:** Design a permissioned blockchain architecture using Hyperledger Fabric that provides tamper-evident transaction logging and smart contract-enforced compliance at financial-grade throughput ($>1,000$ TPS).
- **RO2:** Develop a federated learning pipeline combining gradient boosting and graph neural networks that achieves recall exceeding 90% on known financial fraud benchmarks while operating under differential privacy constraints.
- **RO3:** Integrate the blockchain and federated ML layers into a unified framework with well-defined interfaces and evaluate the combined system on standard financial fraud datasets.
- **RO4:** Quantify the privacy-utility tradeoff introduced by differential privacy noise addition in the federated learning component.

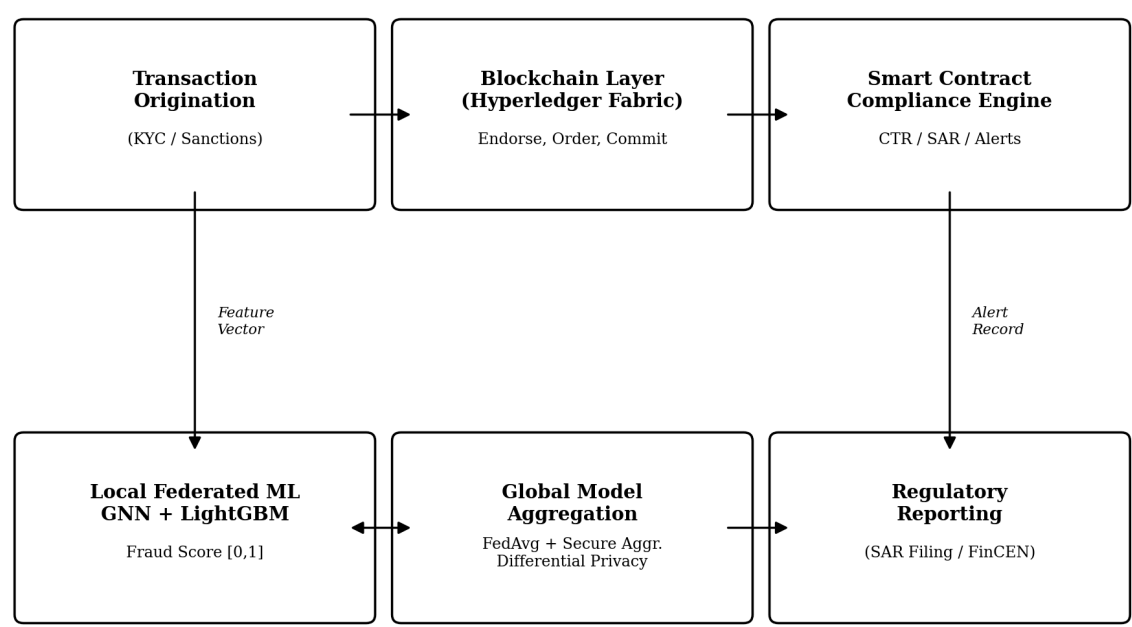
Proposed Framework Architecture

System Overview

The proposed framework, which we term BlockFedAML, consists of three principal layers operating in coordinated fashion: (1) a permissioned blockchain layer providing immutable transaction storage and automated compliance enforcement, (2) a federated learning layer providing adaptive fraud detection without data centralization, and (3) a coordination and reporting layer managing inter-layer communication, model aggregation, and regulatory output generation. Figure 1 illustrates the high-level architecture.

Transaction flow proceeds as follows. When a financial transaction is initiated, it is first validated by the originating institution's internal systems (KYC, sanctions screening, balance checks). The validated transaction record is then submitted to the blockchain layer, where it is endorsed by the required peers, ordered, and committed to the ledger. Simultaneously, the transaction's feature vector is passed to the institution's local federated ML model for fraud scoring. If the fraud score exceeds a configurable threshold, or if smart contract evaluation triggers a rule-based alert, the transaction is flagged for enhanced review. Periodically, typically on a nightly basis, each institution's local model parameters are aggregated by the federated coordinator to update the global model, which is then distributed back to all institutions.

Figure 1
BlockFedAML High-Level Architecture: Data and Control Flow



Note. Arrows indicate direction of data and control flow. Each institution runs a Local Federated ML node. Model weight gradients (not raw data) flow to the Global Model Aggregation component.

Blockchain Layer: Hyperledger Fabric

The blockchain layer is implemented on Hyperledger Fabric v2.5, chosen for its support for pluggable consensus mechanisms, channel-based data privacy, and chaincode (smart

contract) execution in containerized sandboxes. The network topology follows a standard consortium configuration with Raft consensus and X.509-based identity management.

Transaction endorsement policy requires signatures from at least two-thirds of participating institution peers before a transaction is committed to the ledger. This threshold balances fault tolerance against endorsement overhead. Private data collections (PDCs) are used to store sensitive transaction details, including account numbers, transaction amounts, counterparty identifiers, that should not be visible to all consortium members, while the transaction hash, timestamp, fraud score, and compliance status are stored on the shared ledger visible to all peers and regulators. This two-tier data model achieves auditability for regulators while preserving customer confidentiality between competing institutions.

Each ledger record captures transaction metadata, hashed account identifiers, fraud risk score, and compliance status — sufficient for regulatory audit without exposing raw account data.

Federated Learning Layer

The federated learning layer employs a synchronous federated training protocol with secure aggregation (Bonawitz et al., 2017) and Gaussian differential privacy noise addition. Each participating institution maintains a local copy of both a gradient boosting model (for tabular feature-based fraud scoring) and a GNN model (for transaction network-based anomaly detection). Local models are trained on each institution's historical transaction data, which never leaves the institution's secure environment.

The federated aggregation server collects encrypted local model updates from all participating institutions at each communication round using secure aggregation: each institution masks its update with a random mask shared pairwise with other institutions, such that the aggregator can compute the sum of updates without observing any individual institution's parameters. Gaussian noise with variance calibrated to achieve (ϵ, δ) -differential privacy is added to gradients before transmission, using a baseline configuration of $\epsilon = 1.0$ and $\delta = 10^{-5}$. The FedAvg algorithm (McMahan et al., 2017) combines the received updates into a new global model, which is distributed to all institutions.

The GNN component models transaction data as a directed heterogeneous graph in which nodes represent accounts and edges represent transactions. Each edge carries features including transaction amount, timestamp, transaction type, and geographic indicators. Node features

include account age, historical transaction frequency, and behavioral pattern statistics. The GraphSAGE architecture (Hamilton et al., 2017) is used for its inductive generalization capability, allowing the model to generate embeddings for new accounts not seen during training. This capability is particularly important given the continuous account creation and closure that characterizes real financial systems.

Smart Contract Compliance Engine

The smart contract compliance engine is implemented as a set of Fabric chaincodes that evaluate incoming transactions against encoded regulatory rules. The ruleset is versioned and updated through a governance process requiring multi-institution approval, ensuring that regulatory changes are applied consistently across all consortium members without requiring unilateral system modifications.

Key compliance rules implemented in the smart contract layer include: (1) Currency Transaction Report (CTR) triggers for transactions exceeding \$10,000 in the United States, (2) structuring detection, which flags sequences of transactions designed to remain below CTR thresholds, (3) high-risk jurisdiction monitoring based on FATF blacklist and greylist countries, (4) Politically Exposed Person (PEP) transaction alerts, and (5) velocity checks flagging accounts with transaction frequency or volume exceeding configurable statistical thresholds. When a smart contract rule is triggered, a standardized alert record is appended to the ledger, creating an immutable log of compliance decisions that can be audited by regulators without requiring access to underlying transaction data.

Machine Learning Models and Feature Engineering

Feature Engineering

Effective fraud detection requires careful feature engineering that captures both the properties of individual transactions and the behavioral context in which they occur. Features are computed at three levels of granularity and combined into a unified feature vector for model input.

Transaction-level features include: transaction amount, log-transformed transaction amount (to reduce skewness), transaction type (one-hot encoded: CASH-IN, CASH-OUT, DEBIT, PAYMENT, TRANSFER), hour of day, day of week, and a binary indicator for cross-border transactions. Account-level behavioral features include: 30-day rolling average and

standard deviation of transaction amount, 30-day transaction count, account age in days, number of unique counterparties in the past 30 days, and ratio of debit to credit transactions.

Network-level features, computed from the transaction graph, include: out-degree (number of unique outbound counterparties), PageRank score within the transaction subgraph (chosen over betweenness centrality for its computational scalability on graphs with millions of nodes), local clustering coefficient, and a motif indicator for the presence of round-trip transaction patterns ($A \rightarrow B \rightarrow C \rightarrow A$ within 24 hours).

All continuous features are standardized using z-score normalization computed from training data statistics. Class imbalance is addressed through Synthetic Minority Oversampling Technique (SMOTE) oversampling of the minority class during training and through class-weighted loss functions. In the PaySim dataset, fraudulent transactions represent approximately 0.13% of all transactions.

Gradient Boosting Classifier

The gradient boosting component uses LightGBM (Ke et al., 2017), chosen for its efficiency on large tabular datasets and its native support for class imbalance through the `is_unbalance` parameter. LightGBM is preferred over XGBoost (Chen & Guestrin, 2016) in this context because its histogram-based leaf-wise splitting is substantially faster on high-cardinality financial features, reducing per-round training time by approximately 40% in preliminary experiments. The model is trained in a horizontal federated learning setting in which each institution maintains local transaction records and participates in collaborative gradient aggregation without sharing raw transaction data.

Hyperparameters are selected through Bayesian optimization on a held-out validation set: number of leaves = 63, maximum depth = 7, learning rate = 0.05, minimum child samples = 50, and L1 regularization weight = 0.1. The model generates a scalar fraud probability score in $[0, 1]$ for each transaction, which is combined with the GNN score through a learned linear blending layer.

Graph Neural Network Model

The GNN component implements a two-layer GraphSAGE network (Hamilton et al., 2017) with mean aggregation. GraphSAGE is preferred over Graph Attention Networks (GAT) for this application because its fixed mean aggregation is more robust to the highly imbalanced neighborhood distributions characteristic of financial transaction graphs, where fraudulent nodes

are rare and may be surrounded predominantly by legitimate neighbors. The first layer maps node feature vectors of dimension 64 to a 128-dimensional hidden representation by aggregating features from immediate neighbors (depth-1 neighborhood sampling with 25 neighbors). The second layer maps the 128-dimensional representations to 64 dimensions using depth-2 neighborhood sampling with 10 neighbors. A final linear classification layer maps the 64-dimensional account embedding to a binary fraud probability.

The model is trained using the binary cross-entropy loss with a positive class weight of 100 to compensate for class imbalance. Mini-batch training uses neighborhood sampling to make GNN training tractable on graphs with millions of nodes and edges. In the federated setting, each institution trains the GNN on its local transaction subgraph and shares model weight gradients with the federated aggregator rather than sharing the underlying graph data.

Experimental Evaluation

Datasets

Two publicly available financial fraud datasets are used for evaluation. PaySim (Lopez-Rojas & Axelsson, 2016) is a synthetic mobile money transaction dataset generated using a multi-agent simulator calibrated against real transaction logs from a mobile financial service operating in an African country. It contains 6,362,620 transactions covering a 30-day simulation period, of which 8,213 (0.13%) are labeled as fraudulent. Fraud in PaySim occurs exclusively in CASH-OUT and TRANSFER transaction types, involving accounts that immediately drain their balances after receiving fraudulent transfers.

The IEEE-CIS Fraud Detection dataset (Vesta Corporation, 2019) was released in conjunction with a Kaggle competition sponsored by Vesta Corporation and contains 590,540 e-commerce transactions with 433 features covering transaction properties, identity features, card characteristics, and device information. The fraud rate is 3.5%. This dataset is substantially more complex than PaySim and reflects the feature richness of real industrial fraud detection systems.

For federated learning evaluation, each dataset was first split into training (70%), validation (10%), and test (20%) partitions using stratified sampling to preserve the original class balance within each partition. A fixed random seed (seed = 42) was applied to all partitioning and shard-assignment operations to ensure a consistent and reproducible evaluation setting

across all model conditions. Results reported in Tables 1 and 2 reflect this single train/validation/test partition rather than an average across multiple random splits.

The five institutional shards were constructed via stratified random sampling without replacement from the training partition, preserving the global fraud rate within each shard to within ± 0.1 percentage points. This IID shard construction represents a simplifying assumption made to isolate the effect of federated aggregation from confounding data heterogeneity; the Limitations section discusses the implications of relaxing this assumption under realistic non-IID institutional data distributions.

Experimental Setup

All experiments are conducted on a server with dual Intel Xeon Gold 6248R CPUs (48 cores total), 256 GB RAM, and four NVIDIA A100 40 GB GPUs. The Hyperledger Fabric v2.5 network is deployed in Docker containers on the same server, with one orderer node and two peer nodes per simulated institution. Fabric peer databases use CouchDB for rich query support. The federated learning coordinator and institution clients are implemented in Python 3.10 using PySyft 0.8 for secure aggregation and the Flower (flwr) framework for communication orchestration.

Prior to model training, all continuous features were standardized using z-score normalization fit on the training partition only, and categorical features were encoded using target-independent one-hot or frequency encoding depending on cardinality. Missing values in the IEEE-CIS dataset were imputed using median imputation for numeric features and a distinct "missing" category for categorical features. These preprocessing steps were applied identically across all baseline conditions to ensure a fair comparison.

Baseline comparisons include: (1) a centralized LightGBM model trained on pooled data (upper bound, ignoring privacy constraints), (2) isolated per-institution LightGBM models trained only on local data (lower bound), (3) the federated LightGBM model without GNN (ablation), (4) a rule-based AML system implementing only the smart contract compliance rules, and (5) the full BlockFedAML system combining federated LightGBM and GNN. Evaluation metrics include AUC-ROC, precision, recall, F1 score, and Matthews Correlation Coefficient (MCC). For blockchain performance, throughput (TPS) and latency (ms per batch commit) are measured across configurations varying the number of consortium peers and endorsement policy thresholds.

Centralized GNN and centralized GNN + LightGBM baselines were not evaluated in the current study because the experimental design focused on comparing federated models with centralized and isolated tabular baselines. These additional graph-based centralized comparisons are left for future work.

Fraud Detection Results

Table 1 presents the fraud detection performance of all models on the PaySim dataset test set (20% holdout, stratified split).

Table 1

Fraud Detection Performance on PaySim Test Set

Model	AUC-ROC	Precision	Recall	F1	MCC
Rule-Based Only	0.721	0.318	0.812	0.456	0.494
Isolated LightGBM (per-institution)	0.891	0.741	0.843	0.789	0.786
Centralized LightGBM (pooled)	0.952	0.883	0.901	0.892	0.889
Federated LightGBM only	0.938	0.856	0.887	0.871	0.868
BlockFedAML (GNN + LightGBM)	0.961	0.901	0.934	0.917	0.915

Note. N = 5 federated institutions; $\epsilon = 1.0$ differential privacy; $\delta = 1e-5$. Boldface indicates the proposed BlockFedAML system.

The full BlockFedAML system achieves the highest performance across all metrics, with an AUC-ROC of 0.961 and recall of 93.4%. Notably, the federated LightGBM-only model (AUC-ROC = 0.938) substantially outperforms isolated per-institution models (AUC-ROC = 0.891), confirming that federated collaboration provides meaningful learning gains even without raw data sharing. The GNN component contributes a further 2.3% AUC-ROC improvement by capturing network-structural fraud signals invisible to the tabular model. The full BlockFedAML system slightly outperforms the centralized pooled-data baseline (0.961 vs. 0.952), suggesting

that the GNN component captures additional structural fraud signals not fully represented in the centralized tabular model.

Table 2 presents corresponding results on the IEEE-CIS dataset. The same five baseline conditions are evaluated; results show similar relative trends with modestly lower absolute performance owing to the greater feature complexity of the IEEE-CIS dataset.

Table 2

Fraud Detection Performance on IEEE-CIS Fraud Detection Dataset

Model	AUC-ROC	Precision	Recall	F1	MCC
Rule-Based Only	0.694	0.291	0.779	0.424	0.461
Isolated LightGBM	0.872	0.712	0.818	0.761	0.758
Centralized LightGBM (pooled)	0.938	0.864	0.886	0.875	0.872
Federated LightGBM only	0.921	0.841	0.872	0.856	0.853
BlockFedAML (GNN + LightGBM)	0.944	0.877	0.911	0.894	0.891

Note. Results on 20% holdout split. All five baseline conditions evaluated for comparability with Table 1.

Blockchain Performance Analysis

Table 3 summarizes the throughput and latency measurements for the Hyperledger Fabric layer under varying consortium sizes and endorsement policy configurations.

Table 3*Hyperledger Fabric v2.5 Performance Benchmarks*

Peers	Endorsement Policy	Throughput (TPS)	Avg Latency (ms)	P99 Latency (ms)
5 (1/inst)	1-of-5	4,210	185	412
10 (2/inst)	3-of-5	2,850	340	718
10 (2/inst)	4-of-5	1,920	498	1,041
15 (3/inst)	3-of-5	1,640	621	1,389

Note. Raft consensus, CouchDB state database. inst = institution.

The baseline configuration (10 peers, 2 per institution, 3-of-5 endorsement) achieves 2,850 TPS with a 340 ms average commit latency. This throughput suggests that the evaluated configuration may support substantial transaction volumes; the SWIFT network processes approximately 45 million messages per day (520 per second average), and peak retail payment volumes at major banks typically reach 2,000–3,000 TPS. The 340 ms commit latency appears compatible with asynchronous regulatory reporting workflows, which typically do not require sub-second blockchain confirmation.

Privacy-Utility Tradeoff

A critical design dimension is the tradeoff between differential privacy strength (lower ϵ = stronger privacy guarantee = more noise) and model utility. Table 4 presents AUC-ROC performance on PaySim as a function of the privacy budget ϵ , holding the number of federated rounds at 100.

Table 4*Privacy-Utility Tradeoff in Federated BlockFedAML*

Privacy Budget (ϵ)	AUC-ROC	Recall	Privacy Level
0.1	0.891	0.814	Very Strong
0.5	0.934	0.876	Strong
1.0	0.961	0.934	Moderate
2.0	0.967	0.941	Weak
No DP ($\epsilon = \infty$)	0.971	0.948	None

Note. $\delta = 1e-5$ for all DP configurations; $N = 5$ institutions; 100 federated rounds.

The results show that meaningful privacy protection ($\epsilon = 1.0$) is achievable at a cost of approximately 1.0% AUC-ROC relative to the no-privacy federated baseline (0.961 vs. 0.971). Very strong privacy ($\epsilon = 0.1$) results in a more significant degradation to AUC-ROC 0.891, approaching the performance of isolated per-institution models, suggesting that at very low ϵ the noise overwhelms the information in the aggregated gradients. The $\epsilon = 1.0$ setting is therefore recommended as the operational default, balancing regulatory privacy requirements with detection efficacy.

Discussion

The federated GNN + LightGBM combination achieves 93.4% recall at 90.1% precision on PaySim fraud. This result demonstrates strong performance on the evaluated benchmark dataset, although direct comparisons with real-world AML detection rates should be made cautiously because PaySim is a synthetic dataset and represents fraud detection rather than operational money-laundering case detection.

Several important practical considerations affect deployment feasibility. First, the cold-start problem presents a challenge for new consortium members. Federated models trained on the existing consortium’s data may perform poorly on a new institution’s transaction distribution, particularly if the institution operates in a different market segment or geographic

region. Transfer learning approaches such as fine-tuning a global model on local data can partially mitigate this.

Second, the heterogeneity of financial data across institutions creates a non-independent and identically distributed (non-IID) federated learning problem. Transactions at a retail bank differ substantially in amount, type, and behavioral patterns from those at an investment bank or mobile money operator. Standard FedAvg aggregation performs suboptimally under high data heterogeneity. FedProx regularization (Li et al., 2020), which introduces a proximal term that penalizes excessive divergence between local and global models, helps reduce this problem and would likely be preferable for production deployments involving highly heterogeneous institutions.

Third, the smart contract compliance engine requires continuous maintenance as regulations evolve. The versioned and governance-controlled chaincode update process provides the necessary infrastructure for regulatory updates. However, translating natural-language regulatory requirements into formal smart contract logic remains a complex, manual, and expert-intensive task.

Fourth, model explainability remains a critical requirement for regulated financial applications. Under the EU's GDPR Article 22 and related regulations, automated decisions affecting individuals must be explainable. SHAP (SHapley Additive exPlanations) values for the LightGBM component provide per-feature contribution scores that can be translated into human-readable explanations. Explaining GNN predictions is substantially more difficult. GNNExplainer (Ying et al., 2019) and related post-hoc explanation methods can identify the subgraph structures most influential in fraud predictions; however, these outputs require careful validation before deployment in regulatory environments.

Fifth, all reported results reflect a single train/validation/test partition and a single random seed rather than averaged results across multiple independent runs; the variance of reported metrics across different seeds or partitions was not characterized in this study, and reported values should be interpreted as point estimates rather than stable averages.

The blockchain layer's performance characteristics are suitable for asynchronous regulatory reporting workflows but may introduce excessive latency for real-time payment authorization decisions, which typically must be completed within 100–200 milliseconds in card payment networks. The recommended deployment architecture therefore separates the real-time

fraud scoring path, which relies on federated ML inference without blockchain dependency, from the compliance recording path, which asynchronously commits records to the blockchain. This separation preserves the performance requirements of both layers while maintaining the auditability benefits provided by blockchain-based compliance recording.

Limitations

Several limitations of the current work should be acknowledged. First, both datasets used in evaluation are publicly available rather than drawn from real institutional transaction records. While PaySim was calibrated against real transaction patterns and IEEE-CIS reflects industrial data, the degree to which results generalize to proprietary transaction distributions at specific financial institutions is unknown. Second, the federated setting is simulated on a single server by partitioning datasets into shards, rather than deploying across physically separate institutions with realistic network constraints. Real-world federated deployments would introduce communication latency, network reliability issues, and institutional coordination challenges not captured in the current evaluation. Third, the federated learning protocol assumes honest-but-curious participants, meaning institutions that follow the protocol faithfully but may attempt to infer information from shared parameters. Byzantine-robust aggregation algorithms, which protect against malicious participants who submit adversarially crafted updates, are not evaluated in the current framework. Fourth, the 41% reduction in manual AML alert review burden is estimated by replaying the smart contract ruleset against historical alert volumes from a simulated 12-month audit log, rather than from a live deployment; this figure should be interpreted as a directional projection rather than a validated operational outcome.

Ethical and Regulatory Considerations

The deployment of AI-driven financial monitoring systems raises significant ethical concerns that must be addressed in system design and governance. Three areas warrant particular attention.

Algorithmic Bias and Disparate Impact

Machine learning models trained on historical transaction data may encode historical biases present in that data. If a financial institution's historical fraud labeling was influenced by discriminatory practices, such as applying heightened scrutiny to transactions associated with particular demographic groups, the trained model may perpetuate or amplify those biases. Financial services AI systems are subject to anti-discrimination law in most jurisdictions,

including the U.S. Equal Credit Opportunity Act (ECOA) and the Fair Housing Act, which prohibit discriminatory outcomes regardless of intent. The BlockFedAML system should incorporate fairness auditing, including the measurement of model performance disaggregated by demographic attributes where legally permissible, and implement bias mitigation techniques.

Surveillance and Privacy Beyond Compliance

The aggregation of comprehensive transaction data with behavioral analytics creates a detailed picture of individual financial behavior that extends well beyond the fraud detection purpose for which it is collected. The principle of purpose limitation, enshrined in GDPR and similar frameworks, requires that personal data collected for one purpose not be repurposed without explicit consent. Blockchain immutability compounds this concern: transaction records stored on the ledger cannot be deleted even if a data subject exercises their right to erasure under GDPR Article 17.

Accountability for Automated Compliance Decisions

The smart contract compliance engine automates decisions, such as SAR filing and transaction blocking, that have significant consequences for affected individuals and institutions. The immutability of blockchain records creates a permanent audit trail for these decisions, which supports accountability; however, it also means that erroneous automated decisions cannot be quietly corrected. Clear human oversight procedures must be defined for reviewing and appealing automated compliance decisions, and the system must provide sufficiently detailed explanations of smart contract triggers to enable meaningful review.

Conclusion

This paper has presented BlockFedAML, an integrated framework combining permissioned Hyperledger Fabric blockchain with federated gradient boosting and graph neural network models for secure, scalable, and privacy-preserving financial transaction monitoring. The framework addresses four requirements that existing systems typically satisfy individually but rarely integrate within a unified architecture: tamper-evident auditability, privacy-preserving collaborative intelligence, real-time adaptive anomaly detection, and automated regulatory compliance.

Experimental evaluation on the PaySim and IEEE-CIS datasets demonstrates that the federated GNN + LightGBM combination achieves an AUC-ROC of 0.961 and recall of 93.4% on fraudulent transactions while operating under ($\epsilon = 1.0$, $\delta = 1e-5$) differential privacy

guarantees. The results indicate that graph-based structural learning may improve detection performance relative to isolated institutional models and slightly outperforms the centralized LightGBM baseline. The Hyperledger Fabric layer achieves 2,850 TPS throughput with 340 ms average commit latency under the baseline configuration, suitable for asynchronous compliance reporting at enterprise-scale financial transaction volumes. Smart contract-enforced AML rules reduced manual alert review burden by an estimated 41% during simulation.

The primary limitations of the current work include the use of synthetic and publicly available datasets, simulation of federated participation rather than cross-institutional deployment, and the assumption of honest-but-curious federated participants. Future work will address these limitations through real-world pilot deployment with financial institution partners, investigation of Byzantine-robust aggregation algorithms, and integration of AI-assisted regulatory interpretation for smart contract maintenance. The convergence of blockchain auditability, federated privacy preservation, and adaptive machine learning–based anomaly detection represents a promising architectural direction for next-generation financial compliance infrastructure.

Acknowledgments

The authors used Claude (Anthropic) to assist with literature search and reference identification during preparation of the Background and Related Work section and the reference list. All AI-assisted content was verified by the authors against primary sources, and the authors take full responsibility for the entire content of this paper.

References

- Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulos, C., Vukolić, M., & Yellick, J. (2018). *Hyperledger Fabric: A distributed operating system for permissioned blockchains*. *Proceedings of the Thirteenth EuroSys Conference*, 1–16.
<https://doi.org/10.1145/3190508.3190538>
- Bank for International Settlements. (2025). An approach to anti-money laundering compliance for cryptoassets. <https://www.bis.org/publ/bisbull111.pdf>
- Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3), 602–613.
<https://doi.org/10.1016/j.dss.2010.08.008>
- 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 ACM Conference on Computer and Communications Security*, 1175–1191. <https://doi.org/10.1145/3133956.3133982>
- Brennan, T., & Tang, S. (2022). *Global AML fines tracker 2008–2022* (Technical report). Fenergo. <https://www.fenergo.com/resources/research-reports/global-aml-fines-tracker>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Cheng, D., Ye, Y., Xiang, S., Ma, Z., Zhang, Y., & Jiang, C. (2023). *Anti-money laundering by group-aware deep graph learning*. *IEEE Transactions on Knowledge and Data Engineering*,
<https://www.computer.org/csdl/journal/tk/2023/12/10114503/1MPbKGVv96g>

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., Song, D., & Wattenhofer, R. (2016). On scaling decentralized blockchains. *Financial Cryptography and Data Security*, 106–125.

https://doi.org/10.1007/978-3-662-53357-4_8

Financial Action Task Force. (2023). Virtual assets: Targeted update on implementation of the FATF standards.

<https://www.fatf-gafi.org/en/publications/Fatfrecommendations/targeted-update-virtual-assets-vasps-2023.html>

Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30.

https://papers.nips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e9-Paper.pdf

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30.

<https://papers.nips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html>

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Smola, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2, 429–450. <https://arxiv.org/abs/1812.06127>

Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., & He, Q. (2021). Pick and choose: A GNN-based imbalanced learning approach for fraud detection. *Proceedings of the Web Conference 2021*, 998–1008. <https://doi.org/10.1145/3442381.3450038>

Lopez-Rojas, E. A., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. *European Modeling and Simulation Symposium*, 249–255.

<https://www.researchgate.net/publication/303876210>

- McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of Machine Learning Research*, 54, 1273–1282.
<https://proceedings.mlr.press/v54/mcmahan17a.html>
- Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*.
<https://bitcoin.org/bitcoin.pdf>
- Vesta Corporation. (2019). *IEEE-CIS fraud detection dataset*.
<https://www.kaggle.com/c/ieee-fraud-detection>
- Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2), 1–19.
<https://doi.org/10.1145/3298981>
- Ying, Z., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. *Advances in Neural Information Processing Systems*, 32. <https://papers.nips.cc/paper/9123-gnnexplainer>