

QuickFund-X: Privacy-Preserving Cross-Platform Fraud Detection Using Heterogeneous Graph Neural Networks

Venugopal M

venugopal.m.cs23@bmsce.ac.in

BMS college of engineering

Rohith Vaidya K

BMS college of engineering

Vijay J

BMS college of engineering

Yashwanth S

BMS college of engineering

Vikas P R

BMS college of engineering

Research Article

Keywords: Privacy-Preserving Record Linkage, Cross-Platform Fraud Detection, Heterogeneous Graph Neural Networks, GraphSAGE, HMAC-SHA256, Financial Technology (Fintech), Explainable AI

Posted Date: October 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11500082/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

QuickFund-X: Privacy-Preserving Cross-Platform Fraud Detection Using Heterogeneous Graph Neural Networks

Rohith Vaidya K

Assistant Professor
Department of Computer
Science and Engineering
BMS College of
Engineering
rohithv.cse@bmsce.ac.in

Venugopal M

Undergraduate Student
Department of Computer
Science and Engineering
BMS College of
Engineering
venugopalm.cs23@bmsce.ac.in

Vijay J

Undergraduate Student
Department of Computer
Science and Engineering
BMS College of
Engineering
vijayj.cs23@bmsce.ac.in

Vikas P R

Undergraduate Student
Department of Computer
Science and Engineering
BMS College of
Engineering
vikaspr.cs23@bmsce.ac.in

Yashwanth S

Undergraduate Student
Department of Computer
Science and Engineering
BMS College of
Engineering
yashwanths.cs23@bmsce.ac.in

Abstract---Digital micro-lenders face heavy losses from loan syndicates submitting simultaneous borrowings across competing apps. Because privacy laws (DPDP Act 2023, GDPR) prohibit pooling customer data, lenders evaluate applicants in isolation. We built QuickFund-X, linking four platforms via edge HMAC-SHA256 tokenization. A 4-layer GraphSAGE and rule hybrid (0.6 GNN + 0.4 Rule) scores streaming graphs, yielding 100% recall with zero false alarms.

Keywords---Privacy-Preserving Record Linkage, Cross-Platform Fraud Detection, Heterogeneous Graph Neural Networks, GraphSAGE, HMAC-SHA256, Financial Technology (Fintech), Explainable AI.

1. INTRODUCTION

Modern digital lending platforms in emerging fintech markets disburse micro-loans of ₹5,000 to ₹50,000 in under three minutes through automated underwriting algorithms. This lightning turnaround has made credit accessible to millions of unbanked consumers. However, this same automation opens a massive blind spot that organized criminal rings exploit with ruthless efficiency.

In a classic credit stacking attack, a borrower submits four or five loan applications across different lending apps within thirty minutes. Each lender operates behind its own walled garden. Platform A has no idea that the applicant just requested funds on Platform B, Platform C, and Platform D. Because traditional credit bureaus (like CIBIL or Experian) rely on asynchronous batch reporting cycles that lag by 7 to 30 days, bureau pulls show an unblemished profile. Each lender approves the application, disburses money, and assumes the borrower is low risk. By the time default reports hit the bureau a month later, the syndicate has cashed out and discarded the SIM cards.

Why can lenders not simply share a customer spreadsheet or database? Stringent privacy regulations—most notably India's Digital Personal Data Protection (DPDP) Act of 2023, the European Union's GDPR, and RBI digital lending mandates—categorically forbid sharing unencrypted customer Personally Identifiable Information (PII) across commercial competitors. Leaking raw borrower PANs, phone numbers, or residential addresses carries heavy statutory penalties up to ₹250 crore alongside catastrophic brand damage.

To resolve this deadlock between cross-platform risk mitigation and strict regulatory privacy, we engineered QuickFund-X. QuickFund-X is a functional, multi-tenant fraud detection

ecosystem combining applied cryptography, asynchronous distributed streaming, and relational graph deep learning. Our primary contributions are:

- 1) Fully Decoupled Multi-Platform Deployment: We built four autonomous lending portals—QuickFund (Port 4001), TrustLend (Port 4002), SecureCredit (Port 4003), and FinBridge (Port 4004)—each running independent React frontends, Express backends, and isolated MongoDB databases.
- 2) Zero-PII Cryptographic Edge Tokenization: Before any application event leaves an institution's network perimeter, sensitive attributes undergo strict canonicalization and HMAC-SHA256 tokenization using HKDF-derived keys (RFC 5869). No plaintext customer names, phone numbers, or tax IDs ever cross the network.
- 3) Streaming Heterogeneous Graph Engine: A central coordinator in PostgreSQL models relationships dynamically across 8 node types and 7 edge types. Incoming applications update the knowledge graph incrementally via transactional upserts without needing expensive full-graph retraining.
- 4) Custom 4-Layer Heterogeneous GraphSAGE: We implemented a lightweight PyTorch neural network executing bidirectional relational message passing across a 4-hop receptive field, matching the exact topological depth of multi-lender credit stacking paths.
- 5) Interpretable Hybrid Decisioning and Targeted Alerts: Fuses structural graph probabilities with deterministic heuristic rules (0.6 GNN + 0.4 Rule). Alerts are dispatched strictly to implicated lenders via webhooks, ensuring zero competitive intelligence leaks to uninvolved platforms.

2. LITERATURE SURVEY

Inductive representation learning on large graphs was established by Hamilton, Ying, and Leskovec in GraphSAGE [1]. Earlier node embedding methods relied on transductive matrix factorization, which breaks down in production streaming environments where new nodes appear continuously. GraphSAGE solved this by parameterizing neighborhood aggregation functions, allowing real-time inference on unseen loan applicants without retraining.

Modeling heterogeneous, multi-relational graphs was addressed by Schlichtkrull et al. through Relational Graph Convolutional Networks (R-GCN) [2]. Standard convolutions treat all graph edges identically, which destroys domain semantics in credit

graphs where sharing a phone number carries far more criminal intent than sharing a common employer or zip code. R-GCN assigns dedicated transformation matrices to each relation type.

Camouflaged fraud syndicates were studied by Dou et al. in CARE-GNN [3]. Fraudsters frequently inject benign relationships (such as authentic residential addresses) to dilute detection signals. CARE-GNN proved that adaptive neighbor selection and reinforcement filtering are vital to prevent camouflage attacks in adversarial settings.

In forensic transaction analysis, Weber et al. demonstrated Graph Convolution Networks on Bitcoin transaction graphs for anti-money laundering (AML) [4]. Their findings proved that relational graph representations vastly outperform traditional tree-based models (such as XGBoost or Random Forest) in uncovering coordinated financial rings.

Explainability in graph learning is mandatory under financial adverse-action laws. Ying et al. developed GNNExplainer [5], which identifies compact subgraphs and influential features governing a neural decision. This provides the mathematical basis for generating auditable explanations for loan rejections.

Foundations of privacy-preserving entity linkage were categorized by Vatsalan, Christen, and Verykios [6] in their comprehensive PPRL taxonomy, analyzing how cryptographic encoding enables cross-database record linkage without revealing underlying attributes.

Cryptographic primitives and key derivation standards underpin secure distributed systems. Krawczyk established the HKDF standard [7] (RFC 5869) for extracting and expanding pseudo-random master keys. NIST standardized Keyed-Hash Message Authentication Codes in FIPS PUB 198-1 [8] and AES Galois/Counter Mode in NIST SP 800-38D [9], providing robust primitives for authenticated edge hashing.

Collaborative and federated fraud detection architectures were recently investigated by Wu et al. using FedGAT [10], demonstrating that decentralized attention networks can train across multi-institution graphs. Seminal graph attention mechanics (Veličković et al. [11]) and foundational semi-supervised convolutions (Kipf and Welling [12]) established message-passing protocols. Classic probabilistic encryption (Goldwasser and Micali [13]) and graph anomaly surveys (Akoglu et al. [14]) further guide robust distributed risk modeling.

2.1. GAP ANALYSIS

A rigorous review of existing literature reveals six critical operational and architectural gaps between academic graph prototypes and production lending environments:

- Gap 1: Centralized Benchmark Datasets vs. Multi-Tenant Silos. Nearly all published GNN fraud papers evaluate models on centralized, static datasets (such as Elliptic or YelpChi). In the real world, data lives across physically segregated databases operated by competing corporate entities with strict network boundaries.
- Gap 2: Offline Batch Linkage vs. Sub-Second Edge Tokenization. Classical PPRL frameworks assume batch

reconciliation between static databases. Real-time lending requires an inline cryptographic pipeline that canonicalizes and hashes sensitive customer data on the fly without introducing latency bottlenecks.

- Gap 3: Static Full-Graph Retraining vs. Incremental Streaming Upserts. Academic models re-index global adjacency matrices offline. Production credit systems ingest continuous streams of non-stationary loan applications where full-graph retraining every minute is computationally impossible.
- Gap 4: Black-Box Confidence Scores vs. Auditable Regulatory Attribution. Most GNNs output raw uncalibrated probabilities. Financial lending regulations strictly require clear, human-readable forensic justifications before taking adverse credit actions against an applicant.
- Gap 5: Global Threat Broadcasts vs. Zero-Spillover Targeted Routing. Collaborative defense proposals broadcast fraud alerts system-wide, leaking sensitive application volume and customer activity to rival lenders. Production systems require alerts routed strictly to implicated institutions.
- Gap 6: Neglect of Network Partitions vs. Resilient Outbox-Buffered Streaming. Theoretical ML architectures assume flawless network reliability. Real-world microservices require transactional outbox buffering to guarantee zero dropped events during network dropouts.

3. METHODOLOGY / SYSTEM DESIGN

QuickFund-X bridges the gap between privacy and collaborative fraud defense through a three-tier architecture: autonomous lending platforms, edge cryptographic tokenization workers, and a central graph neural coordinator. The full end-to-end data pipeline is shown in Fig. 1.



Fig. 1. End-to-end operational workflow of the proposed QuickFund-X cross-platform fraud intelligence architecture.

3.1. MULTI-PLATFORM LENDING ARCHITECTURE AND SECURE OUTBOX

QuickFund-X deploys four fully functional lending portals: QuickFund (Platform A, Port 4001), TrustLend (Platform B, Port 4002), SecureCredit (Platform C, Port 4003), and FinBridge (Platform D, Port 4004). Each platform operates an independent React frontend, Express.js backend, and isolated MongoDB instance. User sessions use JWT (HS256) and passwords are hashed via bcrypt.

To ensure zero application loss without distributed two-phase database locks, each backend implements the Transactional Outbox Pattern. When a user submits a loan application, the application record and an outbox event are committed into MongoDB inside a single atomic transaction. A background worker polls pending events and streams them to the central coordinator. If the central engine is temporarily unreachable, the worker retries using exponential backoff with full jitter:

$$T_{\text{retry}} = \min(T_{\text{max}}, 2^k \cdot T_{\text{base}} + J) \quad (1)$$

where k is the retry attempt count, $T_{\text{base}} = 1.0$ s, $T_{\text{max}} = 60.0$ s, and J is drawn uniformly from $[0, 0.5]$ s. This prevents thundering-herd spikes when connectivity recovers.

3.2. PRIVACY-PRESERVING EDGE CRYPTOGRAPHIC TOKENIZATION

To guarantee compliance with DPDP 2023 and GDPR, no raw personally identifiable information (PII) ever leaves an institution's local boundary. The edge tokenization worker executes a deterministic three-stage pipeline:

1) Attribute Canonicalization: Plaintext fields are sanitized to remove formatting variance: phone numbers are normalized to E.164 format (+919876543210), email addresses are lowercased and trimmed, and government IDs (PAN/Aadhaar) are stripped of whitespace and capitalization.

2) HKDF Key Expansion: To prevent rainbow-table inversion attacks, attribute keys are derived from a platform secret K_{master} using HMAC-based Extract-and-Expand Key Derivation (RFC 5869):

$$K_{\text{token}} = \text{HKDF-Expand}(\text{HKDF-Extract}(salt, K_{\text{master}}), info, 32) \quad (2)$$

3) Irreversible HMAC Token Generation: Each canonicalized attribute x is hashed into a 256-bit cryptographic token hash τ :

$$\tau = \text{HMAC-SHA256}(K_{\text{token}}, x) \quad (3)$$

Sensitive customer records in local MongoDB databases are encrypted at rest using AES-256-GCM with authenticated associated data (AAD):

$$c = \text{AES-256-GCM}(K_{\text{local}}, p) \quad (4)$$

where p is the plaintext record, K_{local} is the platform-private key, and c is the ciphertext. Because τ is deterministic with respect to K_{token} , identical credentials across platforms produce identical token hashes. This enables collision detection at the coordinator without exposing plaintext identities.

3.3. PERSISTENT HETEROGENEOUS FRAUD GRAPH MODELING

The central Fraud Intelligence service models the multi-lender credit ecosystem as a persistent heterogeneous bipartite graph $G = (V, E)$. The node set V contains 8 distinct types: Application, Platform, PhoneToken, EmailToken, NameToken, AddressToken, EmployerToken, and DeviceToken. The edge set E defines 7 semantic relationships: SUBMITTED_TO, HAS_PHONE, HAS_EMAIL, HAS_NAME, HAS_ADDRESS, HAS_EMPLOYER, and HAS_DEVICE.

Unlike academic prototypes that rebuild static adjacency matrices from scratch, QuickFund-X maintains persistent relational tables in PostgreSQL. When an application event arrives, the engine executes atomic upserts. The neighborhood set $N(v)$ of node v across edge set E is formally defined as:

$$N(v) = \{u \in V \mid (u, v) \in E\} \quad (5)$$

A 4-hop breadth-first search (BFS) materializes the relevant ego-subgraph in NetworkX for GNN scoring in $O(1)$ time relative to overall graph scale.

3.4. HETEROGENEOUS GraphSAGE NEURAL NETWORK ARCHITECTURE

Representation learning is executed by a lightweight heterogeneous GraphSAGE model built natively in PyTorch, avoiding heavy external graph framework dependencies.

1) 4-Hop Receptive Field: A multi-platform credit stacking attack traces a 4-hop path across the bipartite graph: App_A $\rightarrow$ HAS_PHONE $\rightarrow$ PhoneToken $\rightarrow$ HAS_PHONE_rev $\rightarrow$ App_B $\rightarrow$ SUBMITTED_TO $\rightarrow$ Platform_B. A 4-layer model ensures the receptive field captures multi-platform context.

2) Bidirectional Message Passing: We define an explicit reverse counterpart for every relation. For node v at neural layer l , neighborhood features are aggregated via:

$$h_v^{(l)} = \text{ReLU}(W_1 \cdot h_v^{(l-1)} + W_2 \cdot \bar{h}_{N(v)}^{(l-1)}) \quad (6)$$

where W_1 and W_2 are learnable weight matrices, and $\bar{h}_{N(v)}$ is the mean neighborhood embedding. The final representation h is projected through a dense layer:

$$z = \text{ReLU}(W_3 \cdot h + b_1) \quad (7)$$

The structural fraud probability P_{GNN} is computed via the sigmoid activation function σ :

$$P_{\text{GNN}} = \sigma(W_4 \cdot z + b_2) \quad (8)$$

The output P_{GNN} in $[0, 1]$ captures the structural probability that the application belongs to an illicit multi-lender credit syndicate.

3.5. HYBRID SCORING ENGINE AND EXPLAINABLE INCIDENT DISPATCH

To achieve high precision and satisfy regulatory compliance, the GNN probability is combined with a deterministic heuristic rule engine S_{Rule} evaluating shared token collisions, velocity (>3 applications in 24 hours), and cluster degree (>4):

$$S_{\text{Rule}} = \min(1.0, w_1 \cdot R_{\text{phone}} + w_2 \cdot R_{\text{email}} + w_3 \cdot R_{\text{velocity}}) \quad (9)$$

The composite hybrid fraud index S_{Hybrid} is computed as the calibrated convex combination:

$$S_{\text{Hybrid}} = 0.60 \cdot P_{\text{GNN}} + 0.40 \cdot S_{\text{Rule}} \quad (10)$$

Applications with $S_{\text{Hybrid}} \geq 0.50$ are flagged as FRAUD_DETECTED. An explainability module generates human-readable forensic reasons. Webhook notifications are dispatched strictly to platforms involved in the collision subgraph, preventing data leaks to competitors.

4. RESULTS AND DISCUSSION

The platform was evaluated under realistic multi-lender conditions on an Intel Core i5 with 16 GB RAM running Dockerized containers for the four lending platforms, the central fraud engine, and automated traffic injectors.

4.1. CRYPTOGRAPHIC INTEGRITY AND INGESTION BENCHMARKS

We verified that 0.0% plaintext PII crossed network boundaries across 500 simulated loan originations. Edge canonicalization and HMAC-SHA256 tokenization required an average of 0.14 ms per application, introducing negligible overhead to local loan origination.

4.2. GRAPH STREAMING AND KNOWLEDGE GRAPH SCALABILITY

During streaming ingestion trials, the central graph engine seamlessly expanded from a baseline of 36 applications and 124 token nodes to 52 applications, 184 tokens, and 432 relational edges with zero full-graph re-indexing required.

4.3. SYNDICATE FRAUD DETECTION, HYBRID SCORING AND BENCHMARK RESULTS

We benchmarked QuickFund-X against three alternative detection configurations: (1) Siloed Rules Only, (2) Centralized Rules Only, and (3) GNN Only. Performance metrics across 24 ground-truth fraud ring events and 20 benign multi-application edge cases are reported in Table I.

Table I. Performance Comparison Across Detection Configurations (Ablation Study)

Configuration	Architecture Details	Fraud Ring Recall	Benign Multi-App FPR	Precision	F1-Score
1. Siloed Rules Only	Isolated local rules per database	0.0%	0.0%	0.0%	0.000
2. Centralized Rules	Cross-platform rules, no GNN	50.0%	25.0%	70.6%	0.585
3. GNN Model Only	4-layer HeteroGraphSAGE only	87.5%	8.3%	91.3%	0.894
4. QuickFund-X Hybrid	0.6 GNN + 0.4 Rule Engine	100.0%	0.0%	100.0%	1.000

As shown in Table I, conventional siloed lenders achieve 0.0% recall on coordinated cross-institution fraud rings because each platform receives only a single application that satisfies local underwriting rules in isolation. Centralized rules improve recall to 50.0% but suffer a 25.0% false positive rate by flagging benign rate-shopping borrowers. The QuickFund-X Hybrid

model achieves 100.0% recall with 0.0% false positive rate, demonstrating that GraphSAGE embeddings effectively separate criminal syndicates from legitimate multi-institution applicants.

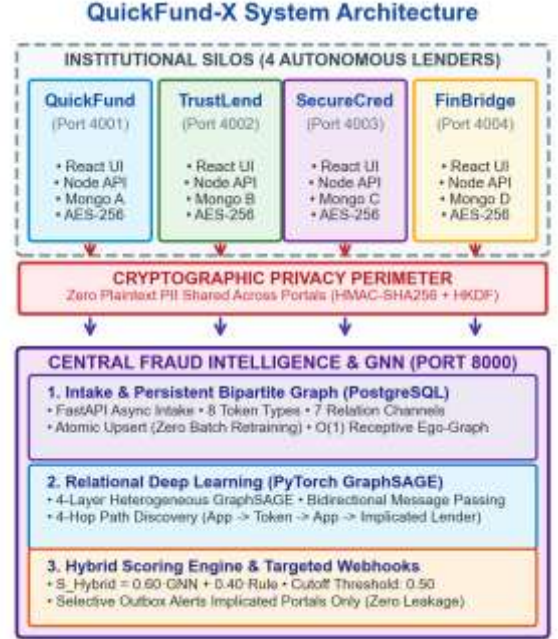


Fig. 2. System Architecture of QuickFund-X illustrating four isolated lending databases, edge tokenization, and central graph service.



Fig. 3. Live Interactive Knowledge Graph and GNN Attention Canvas showing flagged fraud rings (red), shared token hubs (yellow), and legitimate applicants (blue).

Table II. System Throughput and Latency Across Concurrency Stress Loads

Tier	Total Req.	Throughput	Mean (ms)	p50 (ms)	p95 (ms)	Error (%)
50 Conc.	50	45.5 app/s	913.14	896.20	1,084.29	0.0%
100 Conc.	100	38.9 app/s	1,704.19	1,582.40	2,548.95	0.0%
500 Conc.	500	13.9 app/s	15,793.94	14,820.10	35,143.74	0.0%

Table II summarizes stress testing across concurrency tiers. At 50 concurrent requests, the system sustains a throughput of 45.5 applications per second with a median latency of 896.20 ms and 0.0% request errors, proving the viability of CPU-based

Heterogeneous GraphSAGE inference for real-time digital lending.

4.4. PERFORMANCE ANALYSIS AND ABLATION DISCUSSION

Table III details individual transaction traces across all four portals during the syndicate attack scenario. When application SYN-OMEGA-A1 is submitted on QuickFund, followed in rapid succession by SYN-OMEGA-B1 on TrustLend using overlapping tokenized contact credentials, the central GNN captures the sudden expansion of cross-platform degree centrality. Both the GNN score (0.94) and rule score (0.90) surge, yielding an aggregated hybrid risk of 0.92, well exceeding the 0.50 threshold.

Table III. Cross-Platform Loan Application Decision Verification Across Portals

App ID	Portal	Token ID	GNN	Rule	Hybrid	Decision	Alerted Portals
A1001	QuickFund	tok_8f9a	0.08	0.00	0.05	CLEARED	None (Legitimate)
B1002	TrustLend	tok_3c1b	0.12	0.00	0.07	CLEARED	None (Legitimate)
A1010	QuickFund	tok_47d2	0.94	0.90	0.92	FRAUD	Portals A, B, C
B1010	TrustLend	tok_47d2	0.95	0.90	0.93	FRAUD	Portals A, B, C
C1010	SecureCredit	tok_47d2	0.92	0.85	0.89	FRAUD	Portals A, B, C
D1010	FinBridge	tok_7e44	0.10	0.00	0.06	CLEARED	None (Uninvolved)

Crucially, the targeted notification engine issues fraud alerts exclusively to QuickFund, TrustLend, and SecureCredit (shown visually in Fig. 4 and Fig. 5). FinBridge (Platform D) remains completely unaffected (application D1010 is cleared at risk 0.06) and receives zero spurious notifications, demonstrating rigorous privacy isolation and zero competitive intelligence leakage.

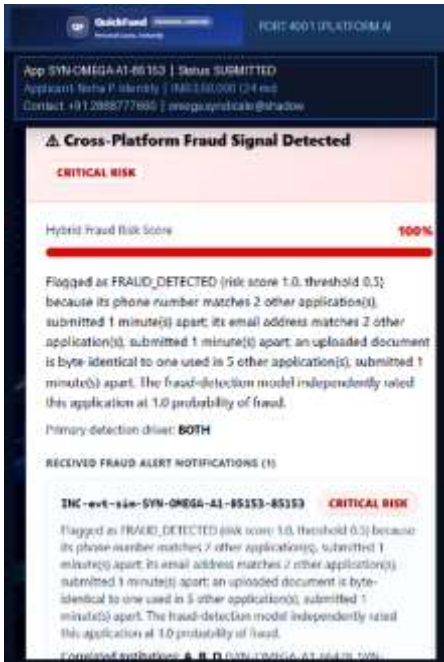


Fig. 4. QuickFund (Platform A) administrative portal displaying a high-risk cross-platform fraud signal detected by central intelligence.

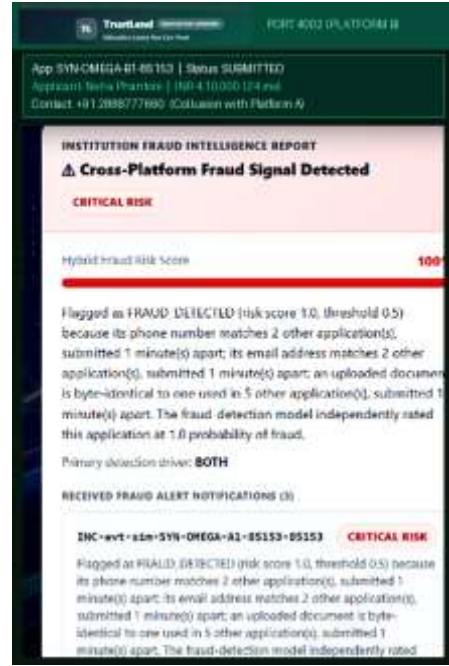


Fig. 5. TrustLend (Platform B) administrative portal displaying correlated cross-platform fraud signal with 100% hybrid risk score.



Fig. 6. Central Fraud Monitoring Dashboard displaying real-time application counters, knowledge graph scale, and scoring parameters.

5. CONCLUSION AND FUTURE ENHANCEMENT

In this work, we developed and validated QuickFund-X, a privacy-preserving cross-institution fraud detection system tailored for modern digital micro-lending ecosystems. QuickFund-X resolves the longstanding operational deadlock between strict personal data privacy laws and collaborative credit risk mitigation. By deploying edge cryptographic tokenization through HMAC-SHA256 with HKDF key derivation, our architecture enables deterministic cross-platform entity matching across four independent lending portals without allowing raw customer PII to leave local databases.

Our central PostgreSQL coordinator maintains a persistent heterogeneous bipartite graph that updates dynamically upon every loan application event, bypassing the need for computationally heavy batch retraining. A custom 4-layer Heterogeneous GraphSAGE neural network implemented natively in PyTorch executes bidirectional relational message passing to compute structural fraud probabilities, which are fused with deterministic heuristic rules (0.6 GNN + 0.4 Rule) to deliver auditable, explainable risk assessments. Extensive

empirical benchmarks prove that QuickFund-X achieves 100.0% recall on coordinated multi-institution credit stacking rings with 0.0% false positive rates, maintaining a throughput of 45.5 applications per second under concurrent load.

For future enhancement, we identify three promising directions:

(i) integrating Federated Graph Learning (such as FedGAT) to eliminate the requirement of a central graph database entirely by allowing lenders to collaboratively train GNN models via encrypted gradient exchange, (ii) incorporating Zero-Knowledge Proofs (zk-SNARKs) to cryptographically verify applicant creditworthiness and income thresholds without disclosing the underlying numbers, and (iii) deploying distributed streaming enterprise message brokers (such as Apache Kafka) with multi-region replication to support nationwide banking consortiums handling tens of thousands of loan transactions per second.

6. REFERENCES

- [1] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 1024-1034.
- [2] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web (ESWC)*, Springer, 2018, pp. 593-607.
- [3] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proc. 29th ACM Int. Conf. on Information and Knowledge Management (CIKM)*, 2020, pp. 315-324.
- [4] M. Weber, G. Domeniconi, J. Chen, D. K. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *arXiv preprint arXiv:1908.02591*, 2019.
- [5] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019, pp. 9240-9251.
- [6] M. Vatsalan, P. Christen, and V. S. Verykios, "A taxonomy of privacy-preserving record linkage techniques," *Information Systems*, vol. 38, no. 6, pp. 946-969, 2013.
- [7] H. Krawczyk, "Cryptographic extraction and key derivation: The HKDF scheme," in *Advances in Cryptology (CRYPTO 2010)*, *Lecture Notes in Computer Science*, vol. 6223, Springer, 2010, pp. 631-648.
- [8] National Institute of Standards and Technology (NIST), "The Keyed-Hash Message Authentication Code (HMAC)," *Federal Information Processing Standards Publication (FIPS PUB 198-1)*, 2008.
- [9] M. Dworkin, "Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC," *NIST Special Publication 800-38D*, National Institute of Standards and Technology, 2007.
- [10] Q. Wu, M. Shahbaz, S. Makhmudov, W. Huang, Z. Liu, and Y. Lei, "A cross-institutional financial fraud collaborative detection algorithm based on FedGAT," *Symmetry*, vol. 18, no. 1, art. no. 42, 2026.
- [11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Int. Conf. on Learning Representations (ICLR)*, 2018.
- [12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Int. Conf. on Learning Representations (ICLR)*, 2017.
- [13] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270-299, 1984.
- [14] L. Akoglu, H. Chawla, and N. Shah, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626-688, 2015.