10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

Real-Time Financial Fraud Intelligence Using Graph Learning, Behavioral Analytics, and Privacy-Aware AI

Chaitanya Nagabhyru

12-1006, Rahmath Nagar 2nd Line, Chilakaluripet, Guntur
Pin: 522616, Andhra Pradesh, India
chaitanyanagabhyru@gmail.com

Abstract: Payment and banking fraud is increasingly organised. Rings of mule accounts, synthetic identities and compromised devices act together, and their individual transactions often look normal. Models that score transactions one at a time miss this coordination, they decay as fraudsters change tactics, and they are often trained on centralised copies of sensitive customer data. This paper presents a design for real-time fraud intelligence that combines four elements: graph anomaly detection over the network of accounts, devices and merchants; behavioural analytics that profile each customer against their own history; concept-drift detection and adaptation; and explainable risk scoring that produces reason codes for analysts and customers. Privacy is built into the design through data minimisation, keyed pseudonymisation of identifiers, differentially private training, federated learning across institutions and privacy-preserving evaluation. The system uses a two-tier architecture so that the online decision stays within a latency budget while the graph is refreshed in near real time. The paper specifies an evaluation protocol on appropriately licensed or public datasets, with time-ordered splits, and it measures fraud recall at a fixed false-decline rate, false declines, processing latency and financial loss avoided net of the cost of false alarms. **This manuscript presents the design, the datasets, the cost model and the pre-specified protocol. It does not report experimental results: the result tables in Section VI are templates to be completed from actual runs, and the numerical examples in Section IV are labelled illustrations of the cost model and not findings.** The paper argues that fraud models should be judged by net value after the cost of false alarms and by robustness over time, and not by accuracy on a random split.

Keywords: *Financial Fraud Detection, Graph Neural Networks, Anomaly Detection, Concept Drift, Explainable AI, Differential Privacy, Federated Learning*

I. INTRODUCTION

Cards, instant payments, digital wallets and open-banking interfaces have made it cheap and fast to move money, and they have done the same for criminals. Consumers lose large sums each year to unauthorised card use, account takeover, authorised-push-payment scams and mule networks that move stolen funds out of the system. Fraud also imposes costs that do not appear in loss figures: declined legitimate purchases, frozen accounts, investigation workloads and loss of customer trust. Regulators and consumer groups

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

increasingly treat fraud prevention and fair treatment of customers as joint obligations of financial institutions [1].

Detection systems have evolved from fixed rules, through supervised classifiers on transaction features, to deep models. Gradient-boosted trees on engineered features are still a strong baseline for card fraud and are widely deployed [2]. Their weakness is structural. A transaction is scored mainly on its own attributes and on summaries of the account's recent behaviour, so a coordinated scheme in which many accounts each make a small, plausible transfer to a common set of recipients may not trigger any single alert. The coordination is visible only in the relations between accounts: shared devices, shared addresses, fan-in and fan-out patterns, rapid circular flows. Graph learning is designed to use exactly that information [3], [4].

Three further problems limit the use of research models in production. The first is non-stationarity. Fraud is adversarial: when a pattern is blocked, it changes. A model trained on last year's data is tested by this year's tactics, and performance can fall within weeks. The second is the cost of false alarms. Declining a legitimate payment annoys the customer, may lead to abandoned purchases and calls to the contact centre, and a high false-decline rate can cost more than the fraud that is prevented. Research that reports only the area under a curve says little about these costs. The third is privacy. Fraud models are most powerful when they link many data sources, but linking increases the exposure of personal data, the risk of misuse and the burden of compliance with data protection law, and institutions cannot easily pool data across organisations even though fraud rings operate across them [5].

This paper proposes a design that addresses these problems together, and a protocol for testing it honestly. The research questions are as follows. **RQ1** Does graph-based and behavioural information raise fraud recall at a fixed false-decline rate beyond transaction-level models, especially for coordinated schemes? **RQ2** Can drift detection and adaptation preserve performance when the fraud pattern changes? **RQ3** Can the system meet a real-time latency budget at realistic throughput? **RQ4** What is the net financial value after counting the costs of false alarms and investigation? **RQ5** What does privacy protection cost in detection performance, and can evaluation itself be done without exposing sensitive records?

The contributions are: an integrated architecture combining graph anomaly detection, behavioural profiling, drift adaptation and explainable scoring; a privacy design with explicit mechanisms at the data, model and evaluation levels; a cost model that puts detection and false alarms on one financial scale; and a pre-specified, time-ordered evaluation protocol with success criteria. The remainder of the paper is organised as follows. Section II reviews the background. Section III covers related work. Section IV describes the methodology. Section V describes the datasets and setup. Section VI gives the result structure. Section VII discusses the implications and limitations, and Section VIII concludes.

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

II. LITERATURE REVIEW

A. Graph Neural Networks and Graph Anomaly Detection

Graph neural networks learn node representations by passing messages along edges and aggregating the messages from neighbours. Variants include sampling-based aggregation for large graphs [6], attention over neighbours, and relational versions that handle several edge types. For fraud, a known difficulty is camouflage: fraudsters connect to legitimate accounts or mimic normal features to hide, so the neighbourhood of a fraudulent node is mixed. Methods that filter or re-weight neighbours by their similarity or predicted label were proposed to address it [3]. Unsupervised and semi-supervised graph anomaly detection methods, such as autoencoders over the graph that flag nodes poorly reconstructed, address the problem that fraud labels are rare and delayed [7]. Temporal graph networks maintain a memory for each node updated by events, which suits streaming transactions [8].

B. Concept Drift

Concept drift is a change over time in the relation between inputs and the label. In fraud, it takes the form of new tactics (a change of the virtual cards used, of the merchants targeted), seasonal changes in legitimate behaviour (holidays, salary days) and changes in the customer base. Detectors monitor either the data stream or the error stream. Methods based on the error rate, such as the drift detection method and adaptive windowing, signal a change when the error departs from its historical level [9]. Because fraud labels arrive late (a chargeback can take weeks), error-based detection is delayed, and monitoring the input distribution and the score distribution is also needed. Adaptation strategies include periodic retraining on a sliding window, weighting recent data more, incremental updates and training ensembles of models of different ages. Delayed labels also bias the most recent window, since confirmed fraud is under-represented.

C. Explainability and Fair Treatment

Credit and payment decisions are subject to expectations that decisions can be explained and challenged. Local explanation methods attribute a score to features using Shapley values [10], and for graph models, subgraph explainers find the small set of edges and nodes most responsible for a prediction [11]. A good explanation for a fraud analyst is a short list of reasons, such as “device shared with 14 other accounts in 48 hours” or “amount ten times the customer’s usual,” and a picture of the relevant subgraph. Explanations for customers must avoid disclosing the detection logic to attackers, and so need a different, more limited form. Fair treatment requires that false-decline rates do not fall unevenly on groups of customers, for instance by geography, age or income, and this should be monitored.

D. Privacy-Enhancing Technologies

Several techniques limit exposure. Data minimisation keeps only the attributes needed. Keyed pseudonymisation replaces identifiers by values computed with a secret key, so that

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

records can be linked without revealing the identifier, and the key can be rotated. Differential privacy bounds how much any single person's record can influence an output, and differentially private stochastic gradient descent adds calibrated noise to clipped gradients during training [12]. Federated learning trains a shared model without moving raw data between institutions, sending only model updates, which can be protected by secure aggregation [13]. Private set intersection lets two institutions learn which identifiers they have in common without revealing the others. Each technique has costs in accuracy, complexity or latency, and none replaces governance and access control. Models can still leak information about their training data, and membership-inference attacks test whether a given record was used in training [14].

III. RELATED WORK

Graph methods have been applied to several settings. Weber et al. [4] released a graph of Bitcoin transactions in which a subset of nodes is labelled illicit or licit, and showed that graph convolutional networks and random forests with local features both gave useful but limited results, with a model on raw features a strong baseline. Dou et al. [3] proposed a graph neural network that filters neighbours to resist camouflage in review and transaction graphs and reported gains over unfiltered aggregation. Liu et al. [15] developed a graph neural network for fraud detection at a large online platform, with attention to heterogeneous relations and to scale. Altman et al. [16] created a synthetic anti-money-laundering dataset with explicit laundering patterns such as fan-out, fan-in and cycles, to enable the development and comparison of graph methods where real data cannot be shared.

The gaps are as follows. Graph, drift, explainability and privacy are mostly studied separately. Evaluations use random splits and report area-under-curve measures without costs. Latency is seldom measured even though the decision usually has to be made in a fraction of a second. Privacy mechanisms are tested for their effect on accuracy but rarely for what they protect against, and the evaluation of models itself is assumed to be done on raw data. The present work is designed around these gaps.

IV. METHODOLOGY

A. System Overview and Two-Tier Architecture

The system has two tiers. The **online tier** receives a transaction or event, retrieves the precomputed features and embeddings of the entities involved, computes a risk score and decides within a latency budget, for example under 100 milliseconds at the 99th percentile, which is typical for card authorisation. The **near-real-time tier** maintains the graph: it ingests events in micro-batches every few seconds, updates the structural features and node embeddings, runs the detection of suspicious communities, and writes the results to a low-latency feature store that the online tier reads. A third, **offline tier** retrains models, runs heavy community analysis and recomputes baselines nightly or when a drift alarm is raised. Separating the tiers keeps expensive graph computation out of the decision path,

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

and the cost is that the information in the graph is a few seconds old when the decision is made, which the evaluation measures.

B. Data Model and Privacy-by-Design Ingestion

Identifiers such as account number, card number, device identifier, address and phone number are replaced at ingestion by keyed pseudonyms, produced with a secret key held in a hardware security module, and a separate key for each type of identifier so that pseudonyms of different types cannot be joined by accident. Names and free-text fields are not ingested. Amounts, timestamps, merchant category and coarse geography are retained, and precise location is generalised to the region. Each field has a documented purpose, and a field with no use in the models is not collected. Access to the mapping between pseudonym and identity is restricted to the investigation team and is logged, so analysts work on pseudonyms and request re-identification only when a case requires it.

C. Behavioural Analytics

Each customer has a behavioural profile built from their own history: typical amounts, merchants and categories, times of day, devices, locations, and the speed of transactions. The profile is kept as running summaries (means, spreads, counts, sets of recent devices) and not as the raw history, which saves memory and limits exposure. At scoring time the transaction is compared with the profile, and the deviations are features: the amount relative to the customer's usual range, a new device or merchant, a change in location faster than travel allows, and an unusual burst of transactions. Peer-group profiles, for customers with similar patterns, give a reference for new customers who have little history. Behavioural signals on the session, such as typing rhythm or navigation pattern in online banking, can be used where customers have given permission, but they raise privacy concerns and are treated as optional.

D. Graph Construction and Features

A heterogeneous graph is built whose nodes are accounts, devices, merchants and other pseudonymised entities, and whose edges are transactions (directed, timestamped, with amount) and shared-attribute links (an account used on a device, two accounts with the same address pseudonym). To limit size and noise, edges older than a configured window are aged out, and very high-degree nodes, such as large merchants and public Wi-Fi addresses, are treated as hubs whose edges are down-weighted. Structural features are computed for each node over several windows: in-degree and out-degree, the number of distinct counterparties, the ratio of money in to money out, the speed with which funds leave after arriving, the number of accounts that share its devices, and counts of small patterns such as fan-in, fan-out, and short cycles. Community-detection results give each node a group identifier and group-level statistics such as size, age of accounts and the share of funds that circulate inside the group.

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

E. Graph Anomaly Detection Models

Two complementary model families are used. The **supervised model** is a relational graph neural network that aggregates the features of neighbours of each type, uses sampling to bound the work per node, and filters neighbours by similarity so that links to legitimate accounts do not dilute the signal of a fraudulent node. It is trained on labelled fraud with a loss that weights the rare class heavily. The **unsupervised model** is a graph autoencoder that learns to reconstruct the structure and the features of normal neighbourhoods, and its reconstruction error is the anomaly score; it can flag new patterns for which no labels exist. For streaming operation, a node memory is updated by each event, so that the embedding of an account reflects what happened up to a few seconds ago. The scores of the supervised model, the unsupervised model, the behavioural deviation features and the transaction features are combined by a final gradient-boosted ranker, which is a robust and fast tabular model, and which produces the final risk score. This design keeps the strength of tree models on tabular data while adding the graph information as features.

F. Concept-Drift Detection and Adaptation

Drift is monitored on three streams. The input stream is monitored by comparing the distribution of each important feature in a recent window with a reference window, with a summary statistic of the shift. The score stream is monitored by the distribution of the final scores, since a shift in the share of high scores indicates either an attack or a model fault. The error stream is monitored through confirmed outcomes when they arrive, using an adaptive-window detector. When a monitor crosses its warning level, the system increases the frequency of retraining and puts more weight on recent data. When an alarm level is crossed, a candidate model trained on recent data is trained, compared against the current model on a recent validation window, and promoted only if it is better by a margin that is fixed in advance, with the old model kept as a fallback. Because labels are delayed, the recent window is corrected for label delay: transactions too recent to have mature labels are used for unsupervised monitoring and for semi-supervised updates, and not as confirmed negatives. A rolling ensemble of models trained on windows of different lengths provides stability, since a short-window model reacts quickly and a long-window model protects against forgetting.

G. Explainable Risk Scoring

The final score is a probability-like value that is calibrated on recent data, so that a score of 0.9 means that about nine in ten such cases are confirmed as fraud, which makes thresholds and costs interpretable. Every alert comes with reason codes, produced in two ways. For the tabular ranker, Shapley-value attributions give the main contributing features, which are mapped to a controlled vocabulary of reasons that analysts and quality assurance staff can read. For the graph component, a subgraph explainer extracts the small neighbourhood that most influenced the score, and the analyst interface shows it as a picture with a short caption, for instance “eleven accounts opened within one week, all sending funds to the same two recipients.” The reason codes are logged with the decision

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

for audit. Customer-facing messages use a limited set of generic reasons that do not reveal detection logic to attackers, and every adverse decision has a path to human review.

H. Privacy-Preserving Training and Evaluation

Privacy is applied at three levels. At the **data level**, minimisation and keyed pseudonymisation as described in Section IV-B. At the **model level**, two options are tested. The first is differentially private training of the neural components, with gradient clipping and noise, at several privacy budgets, to quantify the loss in detection performance as the budget tightens. The second is federated training across institutions, in which each institution trains on its own data and shares only protected model updates, with the graph linked across institutions through a private set intersection on pseudonymised identifiers so that shared devices and counterparties can be found without revealing the other identifiers. At the **evaluation level**, models are evaluated inside the institution's secure environment, and what leaves it is aggregated: confusion counts, curves, and cost totals with noise added to the small cells so that no individual case can be inferred. A membership-inference test is run against each trained model, and the attack success rate over chance is reported as a measure of leakage. This level of privacy-preserving evaluation is what allows results to be shared between institutions and with regulators without sharing records.

I. Cost Model for Detection and False Alarms

The cost model puts fraud prevented and customer harm on one scale. The financial value of the system over a period equals the fraud loss avoided, less the costs created by the system. Fraud loss avoided is the sum of the amounts of fraudulent transactions that were blocked or reversed in time, and in the protocol it is measured against the loss that the existing system would have incurred on the same transactions. The costs created are of four kinds. The first is the cost of a false decline: the margin lost on the abandoned sale, the cost of contact-centre handling, and an estimate of customer attrition caused by the decline, which is usually the largest part. The second is the cost of a review: the analyst time for each alert sent to manual investigation. The third is the cost of a missed fraud, which is the amount lost together with the chargeback and handling fees, and which is counted already in the loss not avoided. The fourth is the operating cost of the system, with infrastructure and maintenance. The net value is then the loss avoided minus the false-decline cost, the review cost and the operating cost.

An **illustration** shows how the model works and why the false-decline rate matters. These parameters are hypothetical and are not findings of this study. Suppose a portfolio processes ten million transactions a month, of which 0.1% are fraud, so 10,000 are fraudulent, with an average loss of 120 currency units each. A system that blocks 70% of the fraud avoids 840,000 units of loss. If the same system also declines 0.2% of the legitimate transactions, that is about 20,000 false declines, and if each costs 25 units in lost margin, handling and attrition, the cost is 500,000 units, so that the net value is 340,000 before review and operating costs. If a better model blocks the same 70% of fraud but cuts false declines to 0.1%, the false-decline cost halves to 250,000 and the net value rises to

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

590,000. The example shows that a change of a tenth of a per cent in the false-decline rate can be worth as much as a large change in recall, and that a model should be compared with others at equal false-decline rates or by net value, not by recall alone.

J. Evaluation Protocol

All splits are by time: the model is trained on an earlier period, validated on the next, and tested on the following one, and the test period is never used in tuning. For streaming evaluation, the models are run in a replay of the test period in time order, with labels released only after the delay that holds for the data, so that drift adaptation is tested under realistic label delays. Because fraud is rare, the test sets are not subsampled, and evaluation uses the whole stream. Measures follow the brief. **Fraud recall** is reported at fixed false-decline rates (0.05%, 0.1% and 0.5% of legitimate transactions) and also as the share of fraud value caught. **False declines** are reported as a rate on legitimate transactions and as a count per million. **Processing latency** is reported as the median, the 95th and the 99th percentile for the online decision, together with the throughput in transactions per second and the staleness of the graph features. **Financial loss avoided** is reported with the net value after the costs of Section IV-I, with the cost parameters varied over stated ranges. Additional measures are the area under the precision–recall curve, the recovery time after a drift event (the time for recall to return to within a stated margin of its prior level), performance on coordinated schemes as a separate stratum, and differences in the false-decline rate between customer segments. Differences between systems are assessed with confidence intervals obtained by bootstrap resampling in blocks of days, to respect the dependence in time.

V. DATASETS AND EXPERIMENTAL SETUP

A. Candidate Datasets

Table 1 lists the candidate datasets. The statistics are those given by the dataset publishers and must be verified at the time of use, together with the licence terms.

TABLE 1. Candidate Datasets, Their Nature and Limitations

Dataset	Type	Approximate size and fraud share (as published)	Graph content	Main limitation
European card transactions (ULB, public)	Real, anonymised	About 285,000 transactions over two days; 492 frauds (0.17%)	None	Features are anonymised components; two days only, so drift cannot be studied
IEEE-CIS fraud detection (competition	Real, anonymised	About 590,000 labelled transactions;	Entity linkage through card, device and	Competition licence; features masked; labels

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

data)		about 3.5% fraud	address fields	derived from chargebacks
Elliptic Bitcoin transaction graph	Real graph, labelled subset	About 204,000 nodes and 234,000 edges; about 4,500 illicit and 42,000 licit labelled	Directed transaction graph over 49 time steps	Cryptocurrency, not banking; most nodes unlabelled; features anonymised
PaySim (synthetic mobile-money simulator)	Synthetic	About 6.4 million transactions; about 8,200 frauds	Account-to-account transfers	Synthetic; fraud patterns simple
Synthetic anti-money-laundering transactions (IBM, public)	Synthetic with embedded laundering patterns	Several sizes, with labelled laundering transactions	Account-to-account flows with fan-out, fan-in and cycle patterns	Synthetic; patterns known by construction
Review and transaction fraud graphs (Yelp and Amazon sets from the graph-fraud literature)	Real graph, labelled	Tens of thousands of nodes	Multi-relation graphs	Domain is reviews and not payments

B. Role of Each Dataset

The datasets cover different questions. The European card data and the IEEE-CIS data serve to measure transaction-level and behavioural performance and the cost trade-offs on real card fraud, with the caveats that the ULB data span two days and cannot test drift, and that the IEEE-CIS data contain entity fields that allow a limited graph. The Elliptic graph tests the graph models on real labelled illicit flows with a time structure, and it has a well-known time-based split, with a change in the illicit pattern late in the series that makes it a useful natural test of drift. The synthetic anti-money-laundering data and PaySim permit the study of coordinated schemes, latency at scale and federated settings, where the data can be partitioned among simulated institutions. The review-graph sets are used only to compare graph architectures against published results. Where an institution provides data, it is used as a separate evaluation under its own governance. No result on a synthetic dataset should be presented as evidence on real banking fraud.

VI. RESULTS: REPORTING STRUCTURE AND SUCCESS CRITERIA

Important note. No experiments have been run for this version of the manuscript, and **no measured result is reported here.** The tables below define how results will be reported

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

once the experiments are completed. All numerical cells are intentionally empty. They must be filled only with values from the pre-specified protocol, and the abstract, discussion and conclusion must be revised to match them. A result that does not meet the criteria below should be reported as it is.

A. Pre-Specified Success Criteria

The criteria are fixed in advance. For RQ1, the proposed system should have higher fraud recall than the best tree-based baseline at each fixed false-decline rate on the real-data benchmarks, with the confidence interval of the difference excluding zero, and the gain should be reported separately for the coordinated-scheme stratum. For RQ2, after a drift event in the replay, recall should return to within a stated margin of its pre-drift level faster than the static model, and the adaptive model should never fall below the static model by more than the margin. For RQ3, the 99th-percentile online latency should be within the stated budget at the target throughput, and the staleness of graph features should be reported. For RQ4, net value should be positive and higher than the baseline across the stated range of cost parameters, or the break-even cost should be stated clearly. For RQ5, the loss in recall at the chosen privacy budget should be reported, and the membership-inference attack should not exceed chance by more than a stated margin. If a criterion is not met, the corresponding claim is withdrawn.

B. Fraud Recall and False Declines

Table 2 is the template for the main detection results on each real-data benchmark, at fixed false-decline rates.

TABLE 2. Fraud Recall at Fixed False-Decline Rates (to be completed; one block per dataset)

System	Recall at 0.05% false declines	Recall at 0.1%	Recall at 0.5%	PR-AUC	Recall on coordinated-scheme stratum
Rule set	[ ]	[ ]	[ ]	[ ]	[ ]
Gradient-boosted trees (transaction and behavioural features)	[ ]	[ ]	[ ]	[ ]	[ ]
Supervised graph network alone	[ ]	[ ]	[ ]	[ ]	[ ]
Graph autoencoder alone	[ ]	[ ]	[ ]	[ ]	[ ]
Proposed system without drift adaptation	[ ]	[ ]	[ ]	[ ]	[ ]
Proposed system (full)	[ ]	[ ]	[ ]	[ ]	[ ]

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

C. Concept Drift

Table 3 is the template for the drift replay. Each row is a drift event, either natural (such as the late change in the illicit pattern in the Elliptic series) or injected in a simulated dataset by changing the fraud tactic at a known time.

TABLE 3. Drift Replay Results (to be completed)

Drift event	System	Recall before	Minimum recall after	Time to recover (days)	Retrainings triggered
[event 1]	Static model	[ ]	[ ]	[ ]	0
[event 1]	Periodic retraining	[ ]	[ ]	[ ]	[ ]
[event 1]	Proposed adaptive	[ ]	[ ]	[ ]	[ ]
[event 2]	Static model	[ ]	[ ]	[ ]	0
[event 2]	Proposed adaptive	[ ]	[ ]	[ ]	[ ]

D. Latency and Throughput

Table 4 is the template for latency, measured end to end under load.

TABLE 4. Online Decision Latency and Throughput (to be completed; state hardware)

Load (× expected peak)	Throughput (transactions/s)	Median latency (ms)	p95 (ms)	p99 (ms)	Graph feature staleness (s)
1×	[ ]	[ ]	[ ]	[ ]	[ ]
2×	[ ]	[ ]	[ ]	[ ]	[ ]
5×	[ ]	[ ]	[ ]	[ ]	[ ]

E. Financial Value Net of False-Alarm Cost

Table 5 is the template for the cost results. Because the cost parameters are assumptions, each system is evaluated at low, central and high values, and the break-even false-decline cost is reported.

TABLE 5. Financial Loss Avoided and Net Value (to be completed; cost parameters stated)

System	Loss avoided	False-decline cost (low / central /	Review cost	Operating cost	Net value (low / central /	Break-even false-decline cost

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

		high)			high)	
Rule set	[ ]	[ ]	[ ]	[ ]	[ ]	—
Gradient-boosted trees	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Proposed system (full)	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

F. Privacy, Fairness and Explanation Quality

Table 6 is the template for the privacy and fairness results. The privacy cost is measured as the change in recall relative to the non-private system, and leakage as the success of a membership-inference attack above chance.

TABLE 6. Privacy Cost, Leakage and Fairness (to be completed)

Configuration	Privacy budget	Recall at 0.1% false declines	Membership-inference attack advantage	Largest gap in false-decline rate between segments
Non-private, centralised	none	[ ]	[ ]	[ ]
Differentially private training, loose budget	[ ]	[ ]	[ ]	[ ]
Differentially private training, tight budget	[ ]	[ ]	[ ]	[ ]
Federated, two institutions	[ ]	[ ]	[ ]	[ ]

VII. DISCUSSION

The design in this paper rests on a view of fraud detection as a system problem and not as a classification problem. Four ideas are central.

First, the unit of detection should sometimes be the group and not the transaction. Coordinated schemes are made of individually unremarkable events, and the information that distinguishes them lies in relations. The graph components are therefore expected to add value mainly for such schemes, and the protocol evaluates them separately, because a modest overall gain could hide a large gain in the schemes that cause the largest losses, or the reverse: a graph model could add cost and latency while contributing little on a portfolio where most fraud is individual. The result should say which.

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

Second, cost is the right common scale. A recall figure without its false-decline rate is not interpretable, and the example in Section IV-I shows that small differences in the false-decline rate can matter as much as large differences in recall. Reporting net value over a range of cost assumptions, with a break-even value, avoids a false precision about costs that are in reality uncertain, and tells an institution whether the conclusion would survive its own numbers.

Third, a model is a process and not a file. Because fraudsters adapt, the lifetime of a model is limited, and its value depends on how fast it can be monitored, retrained and replaced safely. The delay in labels is the main practical obstacle, because the signal that would trigger retraining arrives weeks after the behaviour it concerns. This is why the design monitors inputs and scores as well as errors, and uses unsupervised detection to cover the gap.

Fourth, privacy protection is a design choice with a price, and the price should be measured. Pseudonymisation and minimisation cost little and should be standard. Differentially private training and federated learning cost some accuracy and complexity, and the protocol quantifies how much, so that an institution can choose a budget with its eyes open. Evaluation also needs to be private, which is often forgotten: sharing results across institutions or with regulators should not require sharing records.

The limitations are clear. Public datasets differ from the data of a real institution: they are anonymised, partial, sometimes synthetic, and mostly lack information on the cost of declines and on the conditions of the real authorisation flow. Results on synthetic data demonstrate what a method can find when patterns exist by construction. The cost parameters are assumptions. Latency depends on the deployment, and a result on one platform is not a guarantee for another. Adversaries respond to deployed models, and a replay on historical data cannot capture how criminals would adapt to this particular system, so that adversarial testing and continuing monitoring after deployment are needed. Finally, the claim of consumer protection and resilience at a national level would require evidence from real deployments across institutions, not from benchmark datasets.

VIII. CONCLUSION

This paper proposed a design for real-time financial fraud intelligence that combines graph anomaly detection, behavioural analytics, concept-drift adaptation, explainable risk scoring and privacy protection at the data, model and evaluation levels, in a two-tier architecture that separates the fast decision from the slower graph computation. It defined a cost model that puts fraud loss avoided and the costs of false declines and reviews on one financial scale, and a pre-specified, time-ordered evaluation protocol on public or licensed datasets with stated success criteria.

The manuscript does not yet contain experimental results. The next steps are to obtain the datasets under their licences, to run the experiments under the stated time splits and cost assumptions, to fill the tables of Section VI with measured values, to revise the abstract, discussion and conclusion to match them, and to seek an evaluation with real institutional

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

data under a data-use agreement. If the results meet the criteria, the contribution will be evidence that graph and behavioural information raise detection at an equal false-decline rate while meeting latency and privacy requirements. If they do not, the error analysis will show where the approach fails, which is useful to institutions that are considering similar systems.

REFERENCES

[1] Financial Action Task Force, *Opportunities and Challenges of New Technologies for AML/CFT*. Paris: FATF, 2021.

[2] A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *Proc. IEEE Symposium Series on Computational Intelligence*, 2015, pp. 159–166.

[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. ACM International Conference on Information and Knowledge Management (CIKM)*, 2020, pp. 315–324.

[4] M. Weber et al., "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," in *KDD Workshop on Anomaly Detection in Finance*, 2019.

[5] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[6] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017, pp. 1024–1034.

[7] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proc. SIAM International Conference on Data Mining*, 2019, pp. 594–602.

[8] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," in *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

[9] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.

[10] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017, pp. 4765–4774.

[11] R. 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.

10.48047/jocaaa.2024.32.01.108
ISSN:1521-1398 ,1572-9206

[12] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 308–318.

[13] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175–1191.

[14] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symposium on Security and Privacy*, 2017, pp. 3–18.

[15] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *Proc. ACM International Conference on Information and Knowledge Management (CIKM)*, 2018, pp. 2077–2085.

[16] E. Altman, J. Blanuša, L. von Niederhäusern, B. Egressy, A. Anghel, and K. Atasu, "Realistic synthetic financial transactions for anti-money laundering models," in *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

[17] Manikantha Varaprasad Inakollu, ERP AS A DIGITAL BACKBONE: REDEFINING ENTERPRISE SYSTEMS FOR CONTINUOUS VALUE CREATION, Vol. 51 No. 3 (2023): July-**September 2023**, , Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/285

[18] GAURAV KUMAR SINHA, Mitigating Hallucinations in Retrieval-Augmented Generation via Dynamic Confidence Thresholding, **Vol. 31 No. 4 (2023**): JOCAAA, DOI: https://doi.org/10.1109/ICAC3N56670.2022.10074458, https://eudoxuspress.com/index.php/pub/article/view/5916

[19] Mr. Lalith Chandra Bandaru, Mr. Mohammed Shakeer Bandrevu, Secure CI/CD Governance for Salesforce Platforms: Integrating DevSecOps Controls Across Every Stage of the Release Pipeline, Published In Volume 13, Issue 4, October-December 2022,**Published On: 2022-12-06, https://www.ijsat.org/research-paper.php?id=11155 , DOI https://doi.org/10.71097/IJSAT.v13.i4.11155**

[20] Lalith Chandra Bandaru, Threat Detection and Data Breach Analysis in Salesforce CRM Environments: The LTDF Machine Learning Framework, Date of Publication: 11.06.2021 ,
Country: USA,Pages: 1-12,Published In: Volume 7 Issue 3 June-2021,
DOI: https://doi.org/10.62970/IJIRCT.v7.i3.2605034, https://www.ijirct.org/viewPaper.php?paperId=2605034