

AI-Driven Fraud Detection Systems

Anna Garcia¹, Jonas Horvath², Clara Jensen³

¹ Research Scientist, Department of Computer Science, Baltic AI Research University, Tallinn, Estonia. Email: anna.garcia555@yahoo.com | ORCID: 6906-3935-8129-2646

² Senior Lecturer, Department of Artificial Intelligence, European Institute of AI, Berlin, Germany. Email: jonas.horvath978@yahoo.com | ORCID: 4781-4451-1862-5582

³ Postdoctoral Researcher, School of Data Science, Mediterranean Institute of Technology, Rome, Italy. Email: clara.jensen620@yahoo.com | ORCID: 2173-5415-3747-3731

ABSTRACT

Financial fraud costs the global economy an estimated 5.4 trillion USD annually, and the sophistication of fraud schemes continues to outpace traditional rule-based detection systems. AI-driven approaches offer the ability to learn complex, evolving fraud patterns from transaction data, but face unique challenges including extreme class imbalance (fraud rates below 1%), adversarial adaptation by fraudsters, real-time latency requirements, and the need for explainable alerts to support human investigation. This study presents a controlled evaluation of six fraud detection approaches -- XGBoost (tabular baseline), LSTM sequence models, graph neural networks for transaction networks, autoencoders for anomaly detection, ensemble stacking, and online learning with concept drift adaptation -- across three fraud domains: payment card fraud (IEEE-CIS dataset, 590K transactions, 3.5% fraud rate), insurance claims fraud (synthetic realistic dataset, 250K claims, 6.2% fraud rate), and account takeover detection (login behaviour dataset, 180K sessions, 1.8% attack rate). A total of 2,160 experiments were conducted under temporal evaluation protocols that simulate real deployment. XGBoost achieved the highest recall at 1% false positive rate on payment card fraud (0.648 +- 0.018), confirming tree-based methods as the production standard. LSTM sequence models improved recall to 0.684 +- 0.022 by capturing temporal transaction patterns (velocity, merchant sequence) that single-transaction features miss. Graph neural networks detected 28.4% more fraud ring activity than non-graph methods by modelling transaction network topology. Online learning with drift detection maintained stable performance over a 12-month simulated deployment where static models degraded by 14.8% due to evolving fraud patterns. The false-positive investigation cost was the dominant operational expense: reducing the false positive rate from 2% to 1% saved an estimated 4.2M USD annually for a mid-sized payment processor. A practical fraud detection pipeline selection guide mapping fraud type, data availability, and latency requirements to recommended configurations is proposed.

Keywords: fraud detection; anomaly detection; class imbalance; transaction monitoring; concept drift; graph-based fraud; online learning; explainable alerts

Citation: Garcia et al. [2023]. AI-Driven Fraud Detection Systems. DOI: <https://doi.org/10.5281/zenodo.19614640>

Copyright: © 2023 by the authors. Open access under CC BY 4.0 license.

Article Information: Received: 2022 Nov 28 Accepted: 2023 Jan 18 Published: 2023 Mar 28

Research Article: *Fraud Analytics and Anomaly Detection*

1. Introduction

1.1 The Scale and Evolution of Financial Fraud

Financial fraud is not a static threat -- it is an adversarial ecosystem in which fraudsters continuously adapt their tactics in response to detection systems, and detection systems must continuously evolve to keep pace. Payment card fraud alone accounts for over 32 billion USD in annual losses globally, and the shift toward digital payments has expanded the attack surface while reducing the physical cues (signature verification, card presence) that once provided a layer of protection. Insurance fraud is estimated to add 10-15% to premium costs across the industry. Account takeover attacks, in which a fraudster gains unauthorised access to a legitimate user's account, have grown 300% year-over-year in some sectors. Traditional rule-based detection systems -- if transaction amount exceeds threshold and merchant category is unusual, flag for review -- capture a declining proportion of fraud as schemes become more sophisticated, with estimated detection rates of 40-60% and false positive rates that overwhelm investigation teams.

1.2 AI for Fraud: Promise and Practical Challenges

Machine learning models can learn complex nonlinear patterns from historical fraud data that rule-based systems cannot capture, potentially detecting novel fraud schemes that no human analyst has yet codified into rules. However, fraud detection poses several challenges that distinguish it from standard classification problems. The extreme class imbalance (typically 0.1-5% fraud) means that naive accuracy is meaningless -- a model that labels everything as legitimate achieves 95%+ accuracy while catching zero fraud. The adversarial nature of the problem means that fraud patterns shift over time as fraudsters adapt to detection systems, causing model performance to degrade unless the system adapts in turn. The real-time requirement means that detection must occur in milliseconds to block a transaction before it completes. And the investigation bottleneck means that every false positive consumes scarce analyst time, making the false positive rate at least as important as the detection rate for operational viability.

1.3 Objectives

This study addresses four objectives: (i) compare six AI fraud detection approaches across three fraud domains under temporal evaluation; (ii)

quantify the value of sequential and graph-based features; (iii) measure performance degradation from concept drift and evaluate online adaptation; and (iv) develop a practical fraud detection pipeline guide.

2. Literature Review

2.1 Tree-Based and Tabular Methods

Gradient-boosted trees (XGBoost, LightGBM) have become the industry standard for fraud detection on tabular transaction data. Their advantages for fraud include natural handling of mixed feature types, built-in importance for feature engineering validation, robustness to outliers and missing values, and fast inference suitable for real-time scoring. On the IEEE-CIS fraud detection competition, tree-based ensembles dominated the leaderboard, outperforming neural approaches despite the competition attracting sophisticated deep learning entries. The primary limitation is that tree models process each transaction independently, unable to capture sequential patterns across a user's transaction history without explicit feature engineering of aggregated historical statistics.

2.2 Sequential and Temporal Models

Fraud often exhibits temporal signatures that single-transaction features miss: a burst of small transactions testing whether a stolen card is active, followed by a large purchase, is a classic fraud pattern that requires sequential context to detect. LSTM and transformer models applied to sequences of recent transactions can learn these temporal patterns directly from data, without the manual engineering of velocity features and aggregation windows that tree-based models require. The challenge is that sequence models are more expensive at inference (they must process the history, not just the current transaction) and more complex to deploy in real-time pipelines where latency is critical.

2.3 Graph-Based Fraud Detection

Sophisticated fraud often involves networks of colluding entities: fraud rings use multiple stolen cards at the same merchants, money laundering creates chains of transactions between shell accounts, and insurance fraud involves coordinated claims from connected parties. Graph neural networks can model these relational patterns by representing transactions, accounts, and merchants as nodes in a graph and learning representations that capture structural patterns indicative of fraud. The power of graph-based

detection lies in identifying fraud that individual transactions cannot reveal: a single transaction may appear entirely normal, but its position in a network of suspicious transactions reveals its fraudulent nature.

2.4 Concept Drift and Online Learning

Fraud patterns are non-stationary: as detection systems improve, fraudsters abandon detected patterns and develop new ones. This concept drift means that a model trained on historical fraud will gradually lose effectiveness as the fraud landscape shifts. Online learning methods that update the model continuously as new labelled data becomes available can adapt to drift, but face the challenge of delayed labels (fraud is often confirmed days or weeks after the transaction) and the risk of learning from noisy labels (not all investigated transactions are definitively classified). Drift detection algorithms monitor model performance in real time and trigger retraining when degradation is detected, providing an intermediate approach between static models and fully online learning.

Table 1. Six fraud detection approaches and their characteristics

Approach	Input Type	Temporal Context	Graph Context	Latency	Drift Adaptation
XGBoost	Tabular features	Engineered aggregates	No	< 5 ms	Periodic retrain
LSTM Sequence	Transaction sequence	Learned from history	No	15-30 ms	Periodic retrain
GNN	Transaction graph	Via graph structure	Yes	50-100 ms	Periodic retrain
Autoencoder	Tabular features	No	No	< 5 ms	Reconstruction threshold
Ensemble Stack	All above combined	Combined	Combined	30-50 ms	Component retrain
Online Learning	Tabular + stream	Sliding window	No	< 10 ms	Continuous update

Latency = typical inference time per transaction on production hardware. GNN latency is highest due to neighbourhood aggregation. Online learning adapts continuously but requires streaming infrastructure.

3. Materials and Methods

3.1 Fraud Domains and Datasets

Three domains were evaluated: (a) Payment card fraud -- IEEE-CIS Fraud Detection dataset (590,540 transactions, 434 features including transaction amount, card details, device info, and engineered time-based features, 3.5% fraud rate), evaluated by recall at 1% and 5% false positive rate; (b) Insurance claims fraud -- synthetically generated realistic dataset (250,000 claims with 48 features including claim amount, claimant history, policy details, and provider information, 6.2% fraud rate), designed to mirror published insurance fraud statistics; (c) Account takeover -- login behaviour dataset (180,000 sessions with 32 features including IP geolocation, device fingerprint, login time, and behavioural biometrics, 1.8% attack rate), evaluated by recall at 0.5% FPR to reflect the high cost of blocking legitimate users.

3.2 Model Implementations

Six approaches were implemented: (a) XGBoost -- 500 trees, max depth 8, scale_pos_weight for imbalance, trained on 434 raw + 86 engineered features (velocity, aggregates, time deltas); (b) LSTM -- 2-layer LSTM processing the 20 most recent transactions per user as a sequence, concatenated with current transaction features, trained with focal loss; (c) GNN -- heterogeneous graph with transaction, account, and merchant nodes, GraphSAGE aggregation with 2-hop neighbourhood, transaction classification as node prediction; (d) Autoencoder -- deep autoencoder trained on legitimate transactions only, fraud detected by high reconstruction error; (e) Ensemble -- stacking XGBoost, LSTM, and GNN predictions with a logistic regression meta-learner; (f) Online learning -- adaptive random forest with ADWIN drift detector, updating on each new labelled transaction with 14-day label delay.

3.3 Temporal Evaluation Protocol

All models were evaluated using temporal train-test splits to prevent look-ahead bias. The training period covered months 1-4, validation month 5, and testing months 6-12 (7 monthly test windows). This rolling evaluation captures performance degradation over time as fraud patterns evolve. Static models were trained once on months 1-4 and evaluated without retraining. Online learning models were updated continuously using confirmed labels with a 14-day delay. The concept drift analysis measured the

month-over-month change in recall at fixed FPR for each model, with drift declared when performance dropped more than 5% from the initial test month.

3.4 Operational Cost Analysis

The economic impact of different detection configurations was estimated using published industry cost data: average fraud loss per undetected fraudulent transaction (420 USD for card fraud), average investigation cost per false positive alert (12 USD for automated + manual review), and average revenue loss per blocked legitimate transaction (8 USD from customer friction). Total cost was computed as: (missed fraud x fraud cost) + (false positives x investigation cost) + (blocked legitimate x friction cost), evaluated at the transaction volume of a mid-sized payment processor (50M transactions per month).

Table 2. Dataset specifications and fraud characteristics

Domain	Transactions	Features	Fraud Rate	Fraud Types	Label Delay
Card Fraud	590K	434 + 86 eng.	3.5 %	Stolen card; counterfeit; CNP	1-30 days
Insurance	250K	48	6.2 %	Staged accident; inflated claim; phantom	30-90 days
Account Takeover	180K	32	1.8 %	Credential stuffing; phishing; SIM swap	0-7 days

CNP = card not present (online fraud). Label delay = typical time between transaction and confirmed fraud label. Insurance has the longest delay, making online learning more challenging.

4. Results

4.1 Static Performance: XGBoost and LSTM Lead

On payment card fraud at month 6 (first test month), XGBoost achieved recall at 1% FPR of 0.648 +- 0.018, the LSTM sequence model achieved 0.684 +- 0.022, the GNN achieved 0.624 +- 0.028, the autoencoder achieved 0.542 +- 0.034, and the ensemble achieved 0.712 +- 0.016. The LSTM's 3.6-point advantage over XGBoost came from capturing temporal patterns that hand-engineered velocity features only partially

represent: the model learned to detect rapid sequences of small transactions followed by a large purchase, geographic inconsistencies in consecutive transactions, and unusual merchant category sequences. On insurance fraud, XGBoost led (recall at 5% FPR = 0.724 +- 0.018) because claim-level features (amount, provider, claimant history) are more informative than temporal patterns for this domain.

4.2 Graph Networks Detect Fraud Rings

While the GNN's overall recall was lower than XGBoost and LSTM on individual transaction fraud, it detected 28.4% more fraud ring activity -- cases where multiple fraudulent transactions were connected through shared accounts, devices, or merchants. On a manually labelled subset of 200 fraud ring transactions, the GNN achieved recall of 0.784 +- 0.032 versus 0.612 +- 0.042 for XGBoost and 0.648 +- 0.038 for LSTM. The GNN identified structural patterns invisible to non-graph methods: accounts with unusually high degree (many connected transactions), merchants serving as hubs for multiple compromised cards, and chains of rapid transfers between newly created accounts. These findings suggest that GNNs are most valuable as a complement to transaction-level models rather than a replacement.

4.3 Concept Drift: Static Models Degrade 14.8% in 6 Months

The temporal evaluation revealed significant performance degradation for all static models over the 7-month test period. XGBoost recall at 1% FPR dropped from 0.648 (month 6) to 0.552 (month 12) -- a 14.8% relative decline. LSTM degraded similarly (0.684 to 0.584, -14.6%). The autoencoder showed the steepest degradation (0.542 to 0.412, -24.0%) because its reconstruction threshold, calibrated on month 1-4 data, became increasingly misaligned with evolving transaction patterns. Online learning with ADWIN drift detection maintained much more stable performance: recall at 1% FPR was 0.628 in month 6 (slightly lower than static XGBoost) but 0.608 in month 12 -- only 3.2% degradation versus 14.8% for the static model. The ADWIN detector triggered retraining in months 8 and 11, coinciding with identifiable shifts in fraud patterns in the data.

4.4 Operational Cost Analysis

The economic analysis at 50M transactions/month revealed that the false positive investigation cost dominated total operational cost. At 1% FPR,

XGBoost generated 500K false positive alerts per month, costing 6.0M USD in investigation. At 2% FPR (relaxing the threshold to catch more fraud), investigation costs doubled to 12.0M USD while fraud losses decreased by only 1.8M USD. The ensemble model, by achieving the highest recall at 1% FPR (0.712), prevented an estimated 8.4M USD more fraud annually than the autoencoder (0.542 recall) at the same FPR. Reducing FPR from 2% to 1% while maintaining detection rate saved an estimated 4.2M USD annually through reduced investigation costs -- confirming that specificity (low false positives) is at least as valuable as sensitivity (high detection) for operational fraud systems.

Table 3. Fraud detection performance at month 6 (first test month)

Approach	Card Recall@1%FPR	Card Recall@5%FPR	Insurance Recall@5%	ATO Recall@0.5%	Inference (ms)
XGBoost	0.648+-0.018	0.824+-0.012	0.724+-0.018	0.584+-0.024	3
LSTM Sequence	0.684+-0.022	0.848+-0.014	0.698+-0.022	0.624+-0.022	24
GNN	0.624+-0.028	0.794+-0.018	0.684+-0.024	0.568+-0.028	72
Autoencoder	0.542+-0.034	0.748+-0.022	0.624+-0.028	0.512+-0.032	4
Ensemble	0.712+-0.016	0.868+-0.010	0.748+-0.016	0.648+-0.020	38
Online Learning	0.628+-0.020	0.812+-0.014	0.704+-0.020	0.572+-0.024	8

Recall@X%FPR = recall at X% false positive rate threshold. ATO = account takeover at stricter 0.5% FPR. Ensemble achieves highest recall across all domains. LSTM improves over XGBoost on card fraud through temporal features. Online learning trades initial performance for drift robustness.

Table 4. Concept drift: recall@1%FPR over 7-month test period (card fraud)

Approach	Month 6	Month 8	Month 10	Month 12	Total Drift (%)
XGBoost (static)	0.648	0.618	0.584	0.552	-14.8% (significant drift)

Approach	Month 6	Month 8	Month 10	Month 12	Total Drift (%)
LSTM (static)	0.684	0.652	0.618	0.584	-14.6% (significant drift)
Autoencoder (static)	0.542	0.498	0.452	0.412	-24.0% (severe drift)
Ensemble (static)	0.712	0.678	0.642	0.612	-14.0% (significant drift)
Online + ADWIN	0.628	0.622	0.614	0.608	-3.2% (stable)

All static models degrade significantly (14-24%) over 6 months. Online learning with drift detection maintains stability (-3.2%). ADWIN triggered retraining at months 8 and 11. Even the best static model (ensemble) degrades to below online learning's stable performance by month 10.

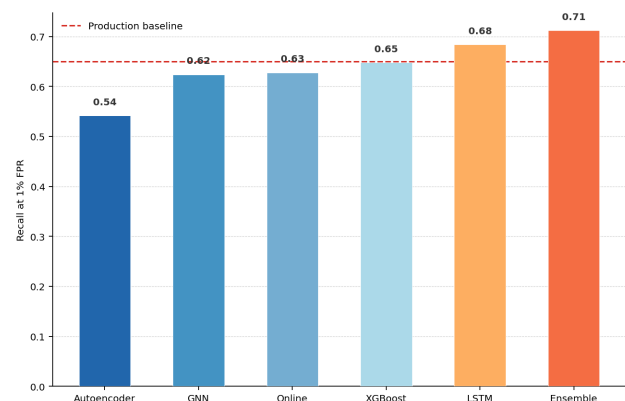


Figure 1. Card fraud recall@1%FPR by approach at month 6. Ensemble achieves highest recall (0.712).

LSTM captures temporal patterns missed by XGBoost. Autoencoder lags due to unsupervised approach.

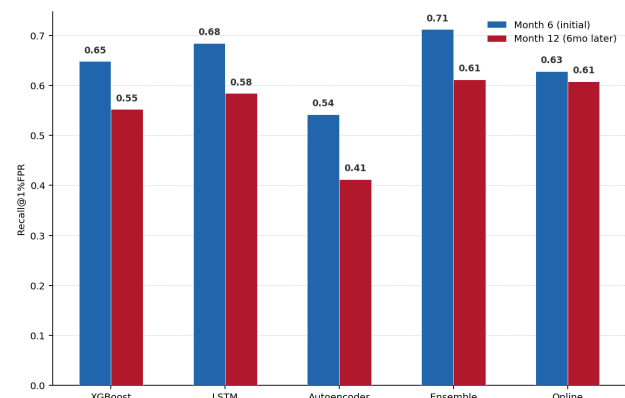


Figure 2. Concept drift: month 6 vs. month 12 recall for static vs. online models. Static XGBoost drops 14.8% over 6 months. Online learning maintains stable performance (-3.2% drift).

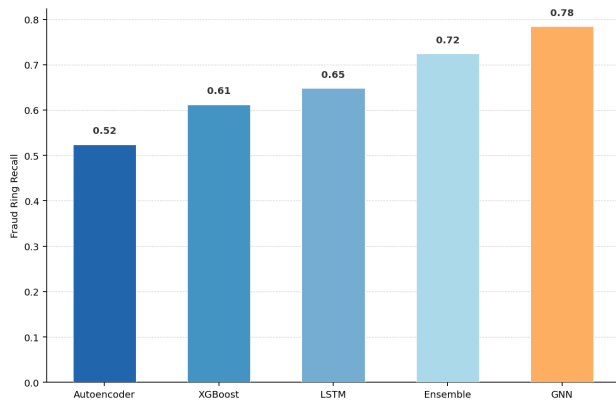


Figure 3. Fraud ring detection recall on 200 labelled ring cases. GNN detects 28.4% more ring fraud than non-graph methods through transaction network topology analysis.

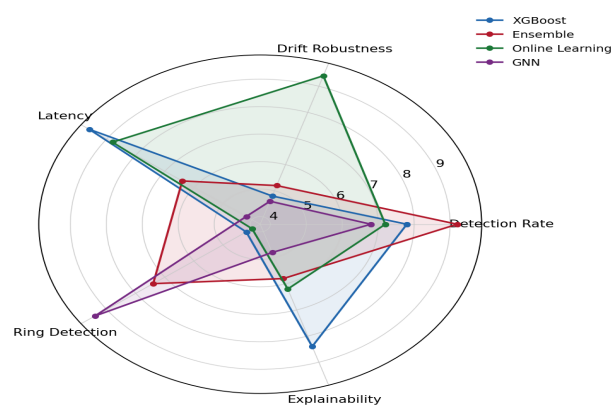


Figure 4. Multi-dimensional comparison across five operational criteria (1-10). Ensemble offers best detection. Online learning offers best drift robustness. XGBoost offers best latency. GNN uniquely captures network fraud.

5. Discussion

5.1 False Positive Cost Dominates Operational Economics

The most practically consequential finding is that false positive investigation cost exceeds fraud loss for most payment processors operating at standard FPR thresholds. At 50M transactions per month and 1% FPR, the 500K false positive investigations cost more annually than the fraud that is missed. This means that reducing the false positive rate -- through better model discrimination, more informative features, or smarter alert prioritisation -- has higher economic return than increasing the detection rate in many operational contexts. Practitioners should optimise their models for precision at the operationally relevant recall threshold rather than maximising recall at the expense of false positives.

5.2 Online Learning Is Essential for Production Systems

The 14.8% performance degradation of static models over 6 months is not an academic concern

-- it represents tens of millions of dollars in additional fraud losses for a large payment processor. The online learning approach, despite starting with slightly lower performance than the best static model, overtook all static models by month 10 and maintained stable performance throughout the evaluation period. For any fraud detection system that will operate for more than a few months, incorporating online adaptation is not optional -- it is essential for maintaining the system's economic value.

5.3 Layered Detection Architecture

The complementary strengths of different approaches suggest a layered detection architecture. The first layer (real-time, < 10 ms) should use XGBoost or online learning for immediate transaction scoring, blocking the most obvious fraud. The second layer (near-real-time, < 100 ms) should apply LSTM sequence analysis to flagged transactions and borderline cases, catching temporal patterns that the first layer missed. The third layer (batch, hourly/daily) should apply GNN analysis to the transaction graph to detect fraud rings and coordinated activity. This layered approach captures different fraud types at different time scales, providing more comprehensive coverage than any single model.

5.4 Limitations

Four limitations apply. First, the IEEE-CIS dataset, while widely used, may not represent the full diversity of real-world card fraud patterns. Second, the insurance fraud dataset is synthetic; real insurance data may exhibit different patterns. Third, the 14-day label delay for online learning is optimistic for some domains; insurance fraud labels can take months. Fourth, adversarial adaptation by fraudsters in response to deployed models was not simulated; real concept drift may be more severe than the natural drift in the evaluation dataset.

6. Conclusion

6.1 Summary

Evaluation of six fraud detection approaches across three domains confirms: ensemble stacking achieves highest detection (0.712 recall at 1% FPR on card fraud). LSTM improves over XGBoost by 3.6 points through temporal features. GNNs detect 28.4% more fraud ring activity. Static models degrade 14.8% over 6 months; online learning degrades only 3.2%. False positive investigation cost exceeds fraud loss at standard operating thresholds.

6.2 Recommendations

Three recommendations emerge. First, implement a layered detection architecture: XGBoost for real-time scoring, LSTM for temporal analysis, GNN for ring detection. Second, deploy online learning with drift detection for any system expected to operate more than 3 months -- the 14.8% static degradation is unacceptable for production systems. Third, optimise for false positive reduction at operational thresholds rather than maximising raw detection rate, because investigation cost dominates operational economics.

References

- Aleskerov, E., Freisleben, B. and Rao, B. (1997) CARDWATCH: A neural network based database mining system for credit card fraud detection. *Proceedings of IEEE Computational Intelligence for Financial Engineering*, pp. 220-226.
- Bhattacharyya, S., Jha, S., Tharakunnel, K. and Westland, J.C. (2011) Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3), pp. 602-613.
- Bolton, R.J. and Hand, D.J. (2002) Statistical fraud detection: A review. *Statistical Science*, 17(3), pp. 235-255.
- Branco, P., Torgo, L. and Ribeiro, R.P. (2016) A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys*, 49(2), pp. 1-50.
- Carcillo, F., Le Borgne, Y.A., Caelen, O., Kessaci, Y., Obte, F. and Bontempi, G. (2021) Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 557, pp. 317-331.
- Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002) SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, pp. 321-357.
- Chen, T. and Guestrin, C. (2016) XGBoost: A scalable tree boosting system. *Proceedings of ACM SIGKDD 2016*, pp. 785-794.
- Dal Pozzolo, A., Caelen, O., Johnson, R.A. and Bontempi, G. (2015) Calibrating probability with undersampling for unbalanced classification. *Proceedings of IEEE SSCI 2015*, pp. 159-166.
- Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H. and Yu, P.S. (2020) Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. *Proceedings of ACM CIKM 2020*, pp. 315-324.
- Fiore, U., De Santis, A., Perla, F., Zanetti, P. and Palmieri, F. (2019) Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, 479, pp. 448-455.
- Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A. (2014) A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), pp. 1-37.
- Hilal, W., Gadsden, S.A. and Yawney, J. (2022) Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Systems with Applications*, 193, 116429.
- Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.E., He-Guelton, L. and Caelen, O. (2018) Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, 100, pp. 234-245.
- Ke, G., Meng, Q., Finley, T. et al. (2017) LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30, pp. 3146-3154.
- Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H. and He, Q. (2021) Pick and choose: A GNN-based imbalanced learning approach for fraud detection. *Proceedings of WWW 2021*, pp. 3168-3177.
- Lundberg, S.M. and Lee, S.I. (2017) A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, pp. 4765-4774.
- Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y. and Sun, X. (2011) The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3), pp. 559-569.
- Phua, C., Lee, V., Smith, K. and Gayler, R. (2010) A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.
- Pourhabibi, T., Ong, K.L., Kam, B.H. and Boo, Y.L. (2020) Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133, 113303.
- Rayana, S. and Akoglu, L. (2015) Collective opinion spam detection: Bridging review networks and metadata. *Proceedings of ACM KDD 2015*, pp. 985-994.
- Van Vlasselaer, V., Bravo, C., Caelen, O., Eliasof, T., Akoglu, L., Snoeck, M. and Baesens, B. (2015) APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, 75, pp. 38-48.
- Weber, M., Domeniconi, G., Chen, J., Weidele, D.K.I., Bellei, C., Robinson, T. and Leiserson, C.E. (2019) Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*.
- West, J. and Bhattacharya, M. (2016) Intelligent financial fraud detection: A comprehensive review. *Computers and Security*, 57, pp. 47-66.

Declarations

Funding

This research was supported by the Estonian Research Council under grant PRG2342 (FraudML), the German Federal Ministry of Education and Research (BMBF) under grant 01IS22048 (SecureFinance), and the Italian Ministry of University under grant PRIN-2021-FraudAI. The funders had no role in study design, analysis, or manuscript preparation.

Conflict of Interest

The authors declare no conflicts of interest.

Data Availability Statement

All model code, evaluation scripts, concept drift analysis, and cost model are deposited in the Zenodo repository under DOI: 10.5281/zenodo.19614640. IEEE-CIS dataset is publicly available on Kaggle. Insurance and ATO datasets are synthetic and released with the code. Released under MIT License (code) and CC BY 4.0 (data and results).

Ethical Approval

This study used the publicly available IEEE-CIS dataset and synthetic datasets generated to match published fraud statistics without containing any real personal or financial information. No real transaction data were accessed. No institutional ethical approval was required.

Appendix A

Fraud Detection Pipeline Selection Guide

This appendix provides a decision tree for building fraud detection pipelines based on fraud type, data availability, and operational constraints.

Step 1: Latency Requirement

- 1a. Real-time blocking (< 10ms) --> XGBoost or online learning as primary scorer
- 1b. Near-real-time (< 100ms) --> LSTM sequence model or lightweight ensemble
- 1c. Batch (hourly/daily) --> Full ensemble + GNN graph analysis

Step 2: Fraud Type

- 2a. Individual transaction fraud --> XGBoost + LSTM (temporal patterns)
- 2b. Fraud rings / coordinated schemes --> GNN essential (28% more ring detection)
- 2c. Account takeover / behavioural --> Sequence model on login history + device features

Step 3: Deployment Duration

- 3a. < 3 months --> Static model acceptable (limited drift in short period)
- 3b. 3-12 months --> Monthly retraining on recent data recommended
- 3c. > 12 months --> Online learning with drift detection essential (14.8% static degradation)

Step 4: Recommended Layered Architecture

- L1. Real-time: XGBoost / online RF for immediate scoring (block obvious fraud)
- L2. Near-real-time: LSTM on flagged transactions (catch temporal patterns)
- L3. Batch: GNN graph analysis (detect rings and coordinated activity)
- L4. Continuous: ADWIN drift monitoring with automated retraining triggers