

Machine Learning-Based Fraud Detection in Digital Financial Transactions

Sri Niwas Ranganathan

SRM University, Chennai, Tamil Nadu, India

Abstract

Digital payments, mobile banking, and instant transfers have increased transaction volumes and made financial services easier to use. They have also given fraudsters new, faster, and harder-to-trace channels. Rule-based fraud controls cannot keep up with the scale of digital transactions or with fraud patterns that change constantly. This review synthesizes research on machine learning (ML) for fraud detection in digital financial transactions, organized around three themes: (1) the detection techniques used, including supervised, unsupervised, hybrid, sequence-based, and graph-based models, and the methods used to handle extreme class imbalance; (2) the operational requirements for running these models in production, including real-time inference, zero-downtime model updates, concept drift management, explainability for investigators, and workflow and compliance integration; and (3) the emerging use of large language models (LLMs) and autonomous agents in fraud operations, together with the prompt engineering, guardrails, and audit mechanisms they require. The evidence shows that ML models, particularly gradient-boosted ensembles, sequence models, and graph-based approaches, outperform static rules on detection. Their effectiveness in practice, however, depends on handling delayed and noisy labels, adapting to drift, keeping false positives manageable, and meeting regulatory demands for explainability and auditability. The review proposes a layered framework for ML-based fraud management and identifies directions for future research.

Keywords: fraud detection, machine learning, digital payments, financial transactions, anomaly detection, class imbalance, concept drift, explainable AI, real-time inference, large language models, guardrails

Introduction

Financial fraud has always followed the money. As transactions moved from branches and paper cheques to cards, online banking, mobile wallets, and real-time payment systems, fraud moved with them. Card-not-present fraud, account takeover, synthetic identities, authorized push payment scams, and money mule networks now account for a large share of financial crime. These schemes exploit the speed and anonymity of digital channels and often combine technical attacks such as phishing, credential theft, and malware with social engineering.

Traditional fraud controls relied on expert-written rules, for example flagging transactions above a threshold, from unusual locations, or at unusual times. Rules are transparent and easy to audit, but they have well-known weaknesses. They are static, fraudsters learn to evade them, they generate large numbers of false positives, and maintaining

thousands of interacting rules becomes costly (Bolton & Hand, 2002). Statistical and ML approaches were developed to overcome these limits by learning patterns of fraudulent and legitimate behavior from data (Bolton & Hand, 2002; Phua et al., 2010).

Research over the last two decades shows that ML models can detect fraud more accurately and adaptively than rules (Bhattacharyya et al., 2011; Ngai et al., 2011; West & Bhattacharya, 2016). Fraud detection is nonetheless one of the hardest applied ML problems. Fraud is extremely rare relative to legitimate activity, labels arrive late and are sometimes wrong, fraud patterns change as fraudsters adapt, decisions must often be made in milliseconds, and every decision affects real customers (Dal Pozzolo et al., 2014, 2018).

Solving these problems takes more than a good model. A fraud detection system must serve predictions reliably in real time, be updated without

interrupting service (Sannidhanam, 2023), explain its alerts to investigators and regulators, feed alerts into structured case management workflows (Basireddy, 2022), and maintain audit trails that satisfy compliance requirements (Basireddy, 2023). More recently, LLMs and autonomous agents have entered fraud and security operations (Cherukuri, 2023). They bring new capabilities and new reliability and safety concerns, which make prompt engineering (Cherukuri, 2025; Sannidhanam, 2024a), guardrails (Sannidhanam, 2024b), and agent oversight (Sannidhanam, 2025) directly relevant.

This Review has four Objectives

- To synthesize ML techniques for detecting fraud in digital financial transactions and the methods used to address class imbalance and evaluation.
- To examine what it takes to operate ML fraud models in production, including real-time serving, model updates, drift, explainability, and workflow and compliance integration.
- To assess the emerging role of LLMs and autonomous agents in fraud operations and the controls they require.
- To identify open challenges and directions for future research.

Review Methodology

Review design

The review uses a structured, PRISMA-informed approach (Page et al., 2021) with thematic synthesis. This suits a literature that spans financial data mining, ML engineering, cybersecurity, and enterprise systems.

Research Questions

- **RQ1:** Which ML techniques are used to detect fraud in digital financial transactions, and how do they handle class imbalance and evaluation?
- **RQ2:** What operational capabilities are needed to deploy ML fraud detection in real-time production environments?
- **RQ3:** How are LLMs and autonomous agents being applied in fraud and security operations, and what controls do they need?
- **RQ4:** What challenges and research gaps remain?

Inclusion Criteria

Studies were included if they (a) proposed or evaluated ML methods for detecting financial fraud or anomalous transactions, (b) addressed deploying, updating, explaining, or governing ML or LLM systems in production settings relevant to fraud operations, or (c) addressed workflow, compliance, or security architectures that fraud detection systems depend on. Opinion pieces without methodological content were excluded.

Data extraction and synthesis

For each study, the review recorded the fraud type or application context, the technique, the function addressed (detection, operations, or governance), and the reported strengths and limitations. Findings were grouped into the themes in Sections 4 to 6.

Note on reproducibility: Researchers extending this review should report database names, search strings, search dates, and record counts at each screening stage in a PRISMA 2020 flow diagram (Page et al., 2021).

Background: Fraud in Digital Financial Transactions

Types of digital financial fraud

Digital fraud takes many forms, each with different data signals and detection challenges. Table 1 summarizes the main categories.

Why fraud detection is hard

Several features of fraud make it an unusually difficult prediction problem (Dal Pozzolo et al., 2014; Bolton & Hand, 2002):

- **Extreme class imbalance.** Fraud often accounts for well under 1% of transactions, so a model that labels everything legitimate can reach very high accuracy while catching nothing.
- **Concept drift.** Fraudsters adapt to controls, and legitimate behavior shifts with seasons, products, and events, so patterns learned last quarter may not hold today (Gama et al., 2014).
- **Delayed and noisy labels.** Fraud is often confirmed only after chargebacks or investigations, sometimes weeks later, and some fraud is never reported.

Table 1: Major types of fraud in digital financial transactions

<i>Fraud type</i>	<i>Description</i>	<i>Typical signals</i>	<i>Detection challenges</i>
Card-not-present fraud	Use of stolen card details for online or phone purchases	Unusual merchants, amounts, devices, velocity	High volume, very low fraud rate, millisecond decisions
Account takeover	Criminal gains control of a legitimate account	New device or location, credential changes, rapid transfers	Activity looks like the genuine customer at first
Synthetic identity fraud	Accounts built on fabricated or blended identities	Thin credit files, shared attributes across applicants	No real victim to report; labels arrive very late
Authorized push payment scams	Victim is manipulated into sending money	First-time payees, unusual urgency, new beneficiaries	Customer authorizes the payment; behavior-based signals only
Money mule and laundering networks	Accounts used to move and layer illicit funds	Pass-through flows, fan-in and fan-out patterns	Fraud is visible only at network level
Insider and merchant fraud	Abuse by employees or merchants	Refund abuse, collusive transactions	Small populations, strategic concealment

- **Adversarial behavior.** Unlike most prediction targets, fraudsters actively probe and evade detection systems (Biggio & Roli, 2018).
- **Operational constraints.** Decisions must be made in milliseconds, and investigators can review only a limited number of alerts each day (Dal Pozzolo et al., 2018).
- **Cost asymmetry.** Missing fraud causes direct loss, while false positives block genuine customers, damage trust, and consume investigator time.

Findings: Machine Learning Techniques for Fraud Detection

Supervised learning

When labeled historical data are available, supervised classifiers are the most common approach. Bhattacharyya et al. (2011) compared logistic regression, support vector machines, and random forests on real credit card transaction data and found random forests performed best overall. Tree-based ensembles, including random forests (Breiman, 2001) and gradient boosting implementations such as XGBoost (Chen & Guestrin, 2016), became the standard choice for tabular transaction data. They handle mixed feature types, capture non-linear interactions, and are relatively robust to noisy features. Reviews by Ngai et al. (2011), Abdallah et al. (2016), and West and Bhattacharya (2016) confirm

that ensembles and neural networks dominate supervised fraud research, while noting that results vary widely across datasets and evaluation designs.

Feature engineering

In fraud detection, features often matter as much as the algorithm. Raw transactions carry limited information on their own. Their meaning comes from context: how a transaction compares with the customer's usual behavior. Whitrow et al. (2009) showed that aggregating transactions over time windows significantly improved detection over using single transactions. Bahnsen et al. (2016) extended this with features that capture the periodicity of customer behavior, such as the usual time of day for purchases, and reported substantial cost savings. Typical features include transaction velocity, deviation from the customer's typical amounts, merchants, and locations, device and channel attributes, and time since the last transaction.

Unsupervised learning and anomaly detection

Labeled fraud is scarce, delayed, and incomplete, and new fraud schemes by definition have no labels. Unsupervised and anomaly detection methods address this by flagging transactions that differ from normal behavior (Chandola et al., 2009). Isolation forest (Liu et al., 2008), which isolates anomalies through random partitioning, is widely used because

it is efficient on large datasets. Clustering, peer-group analysis, and autoencoders are also applied (Phua et al., 2010). Unsupervised methods usually produce more false positives than supervised ones, but they can catch novel fraud patterns that supervised models miss.

Hybrid approaches

Hybrid methods combine the two. Carcillo et al. (2021) added unsupervised outlier scores as features to supervised classifiers on real card transaction data and found improved detection, especially for fraud that differs from historical patterns. This design uses labeled data where it exists and anomaly signals where it does not.

Sequence-based deep learning

Fraud often shows up in sequences of transactions rather than single events, such as a card-testing pattern of small purchases followed by a large one. Recurrent neural networks, particularly long short-term memory (LSTM) networks (Hochreiter & Schmidhuber, 1997), can model these sequences directly. Jurgovsky et al. (2018) framed card fraud detection as sequence classification and found that LSTMs improved detection for card-present transactions and complemented aggregated-feature approaches. Their gains were smaller for online transactions, where feature engineering remained competitive.

Graph-based methods

Many fraud schemes, particularly money laundering, mule networks, and collusive fraud, are only visible when accounts are viewed as a network. Graph-based anomaly detection (Akoglu et al., 2015) analyzes relationships among entities such as customers, accounts, devices, and merchants. Van Vlasselaer et al. (2015) combined network-based features with transaction features in the APATE system and improved card fraud detection by propagating suspicion through the network of cardholders and merchants. Graph neural networks, including graph convolutional networks (Kipf & Welling, 2017), extend this further. Weber et al. (2019) applied them to anti-money laundering on a Bitcoin transaction graph and showed the potential of graph learning for financial forensics.

Handling class imbalance

Imbalance is addressed at the data, algorithm, and decision levels. At the data level, resampling methods include random undersampling of legitimate transactions and synthetic oversampling of fraud with SMOTE (Chawla et al., 2002). At the algorithm level, cost-sensitive learning assigns higher penalties to missed fraud. At the decision level, thresholds are tuned to operational capacity. Resampling changes the class distribution the model sees, so predicted probabilities must be recalibrated before they are used for decisions (Dal Pozzolo et al., 2018).

Evaluation practices

Evaluation is a recurring weakness in fraud research. Overall accuracy is meaningless under extreme imbalance, and even the area under the ROC curve can look good while the model performs poorly on the few cases that matter. Dal Pozzolo et al. (2014, 2018) argue for metrics that reflect how fraud teams actually work, such as precision among the top-k alerts that investigators can review each day, together with the area under the precision-recall curve. Cost-based metrics that weigh fraud losses against investigation costs and customer friction are also used (Bahnsen et al., 2016). Public datasets are scarce because transaction data are confidential, so simulators such as PaySim (Lopez-Rojas et al., 2016) are often used, with the caveat that simulated data may not reflect real fraud behavior.

Synthesis for RQ1

ML techniques for fraud detection in digital transactions shown in Table 2.

Findings: Operating ML Fraud Detection in Production

Real-time inference

Many fraud decisions are made in the authorization path. The model must score a card payment or instant transfer within milliseconds, before it is approved. This imposes strict requirements on latency, throughput, and availability. Features must be computed from recent behavior in real time, and the scoring service must be available continuously, because an outage either blocks legitimate payments or lets fraud through unchecked.

Table 2: ML techniques for fraud detection in digital transactions

<i>Approach</i>	<i>Typical methods</i>	<i>Strengths</i>	<i>Limitations</i>	<i>Representative studies</i>
Supervised	Random forests, gradient boosting, neural networks	High accuracy on known fraud patterns	Needs labels; weak on novel fraud	Bhattacharyya et al. (2011); Chen & Guestrin (2016)
Unsupervised	Isolation forest, clustering, autoencoders	Detects novel fraud without labels	Higher false positives	Liu et al. (2008); Chandola et al. (2009)
Hybrid	Outlier scores as supervised features	Combines known and novel pattern detection	More complex to build and maintain	Carcillo et al. (2021)
Sequence-based	LSTM, recurrent networks	Captures behavior across transaction sequences	Data-hungry; mixed gains online	Jurgovsky et al. (2018)
Graph-based	Network features, graph neural networks	Reveals rings, mules, and laundering networks	Graph construction and scale	Van Vlasselaer et al. (2015); Weber et al. (2019)

Zero-downtime model updates

Because fraud patterns drift, fraud models must be retrained and redeployed often. Every update is a risk: a faulty model can cause a wave of false declines or missed fraud, and a disruptive deployment can interrupt payment processing. Sannidhanam (2023) addresses zero-downtime AI model updates in real-time inference systems, describing deployment strategies that let new model versions replace old ones without interrupting service. Strategies of this kind include:

- **Shadow deployment.** A new model scores live traffic in parallel without affecting decisions, so its behavior can be compared with the current model before it goes live.
- **Canary and phased rollout.** The new model handles a small share of traffic first, and exposure increases only if performance stays within agreed limits.
- **Blue-green switching.** Two production environments allow traffic to switch to the new version instantly and switch back just as quickly if problems appear.
- **Automated rollback.** Monitoring triggers an immediate return to the previous version if error rates, latency, or decision distributions shift unexpectedly.

For fraud detection, these practices turn model retraining from a risky, infrequent event into a routine one. This is what allows a system to keep pace with drift.

Concept drift and feedback loops

Drift management needs both detection and adaptation (Gama et al., 2014). Dal Pozzolo et al. (2018) show that in real card fraud systems, the most recent labels come from investigator feedback on alerts, while most transactions are labeled only later and with delay. Their learning strategy treats these feedback and delayed samples separately and improves detection. A further complication is a feedback loop: the model decides which transactions get investigated, so it influences which fraud gets labeled. Production systems therefore monitor input distributions, score distributions, and alert outcomes continuously, and retrain regularly. Zero-downtime deployment (Sannidhanam, 2023) is what makes frequent retraining safe.

Explainability for investigators and regulators

Fraud alerts are reviewed by investigators who need to understand quickly why a transaction was flagged. Customers who are declined, and regulators who oversee the process, may also need explanations. Post-hoc methods such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017) show which features drove each score. Psychoula et al. (2021) examined explainable ML for fraud detection and discussed the trade-off between explanation quality and the real-time speed that fraud systems need. Rudin (2019) argues that in high-stakes settings, inherently interpretable models may be preferable to explaining black boxes after the fact. In fraud

detection, where speed and accuracy are both critical, many institutions combine complex models with fast feature-attribution methods and human-readable reason codes.

Cherukuri (2023) shows the value of explainability in a closely related field, autonomous cyber threat detection and incident response. Combining explainable AI with LLMs let the system state the reasoning behind each alert in plain language, which reduced analyst fatigue and sped up triage. Fraud operations face the same problem at scale. Investigators handle large alert queues, and explanations that turn feature attributions into clear narratives can make reviews faster and more consistent.

Workflow and Case Management

An alert is useful only if it leads to the right action. Basireddy (2022) examines digitizing risk management through configurable enterprise workflow architecture. In this approach, risk identification, assessment, escalation, approval, and remediation steps are defined as configurable process definitions rather than hard-coded logic. For fraud operations, this maps directly onto case management. High-risk alerts can route to senior investigators, confirmed fraud can trigger card blocks and customer contact, suspected laundering can follow a separate escalation path, and every step is recorded. Because fraud patterns and regulations change often, the ability to reconfigure these flows without rebuilding systems is an important operational advantage.

Compliance and Auditability

Fraud and anti-money laundering (AML) functions are heavily regulated. Institutions must show regulators how alerts were generated, how decisions were made, and who made them. The Financial Action Task Force (FATF, 2021) has noted that new technologies, including ML, can make AML and counter-terrorist financing controls more effective, provided they are explainable, auditable, and properly governed. Basireddy (2023) proposes an audit-ready compliance platform architecture for enterprises that use LLMs in regulated settings. Such an architecture includes traceable decision records, versioned models and prompts, access controls, and evidence that can be produced for regulators. As fraud operations adopt

both traditional ML and LLM-based tools, this kind of architecture keeps every automated or AI-assisted decision reconstructable and defensible.

Findings: LLMs and Autonomous Agents in Fraud Operations

Emerging roles for LLMs

Transaction scoring remains the job of specialized ML models. LLMs are finding a complementary role in the work around those scores. Potential applications include summarizing a customer's transaction history and prior cases for an investigator, drafting investigation notes and suspicious activity report narratives, explaining model alerts in plain language (Cherukuri, 2023), analyzing unstructured evidence such as customer communications in scam cases, and answering investigators' questions about policies and procedures by retrieving relevant documents (Lewis et al., 2020).

These uses carry real risks. LLMs can generate fluent but incorrect content (Ji et al., 2023). A fabricated detail in an investigation summary or regulatory report could lead to a wrong decision against a customer or a compliance failure. Reliability, safety, and auditability are therefore requirements, not optional features.

Prompt Engineering for Reliability

How an LLM is instructed strongly affects the reliability of its output. Techniques such as chain-of-thought prompting can improve reasoning on complex tasks (Wei et al., 2022). Sannidhanam (2024a) sets out prompt engineering patterns for reliable LLM outputs in production environments. These include structured output formats, explicit instructions and constraints, grounding responses in supplied context, and validating outputs before use. In fraud operations, such patterns help make generated case summaries or report drafts consistent, complete, and tied to the underlying evidence.

Cherukuri (2025) examines the cost, complexity, and efficacy of prompt engineering techniques for LLMs, showing that more elaborate prompting strategies do not always deliver proportionate gains and can add significant cost and latency. For fraud teams processing large alert volumes, this trade-off matters in practice. The most effective technique

must also be affordable at scale and fast enough for operational use.

Guardrails and safety mechanisms

Using LLMs in financial operations requires controls that constrain what the model can see, say, and do. Sannidhanam (2024b) examines guardrails and safety mechanisms for LLM-powered enterprise applications. These include input filtering, output validation, restrictions on topics and actions, protection of sensitive data, and escalation to human review. In fraud contexts, guardrails can prevent a model from exposing customer data, stop it from making unsupported accusations, block prompt injection attacks hidden in customer-submitted content, and ensure that consequential decisions such as closing an account or filing a report remain with authorized people.

Autonomous agents

The next step beyond assistive LLMs is autonomous agents that plan and carry out multi-step tasks. Sannidhanam (2025) examines autonomous AI agents for cloud infrastructure operations. The discussion covers agents that monitor systems, diagnose problems, and take corrective action, within defined limits and with human oversight. Fraud detection runs on this kind of cloud infrastructure, so such agents can help keep scoring services, feature pipelines, and monitoring healthy. By analogy, agentic designs could also support fraud investigation by gathering evidence across systems, linking related cases, and preparing recommendations for human decision. Cherukuri (2023) describes a similar pattern in cybersecurity, where AI systems both detect threats and begin containment actions.

Autonomy raises the stakes of error. An agent that wrongly blocks accounts or changes production systems can cause widespread harm. Autonomous capabilities therefore need the controls described above: guardrails (Sannidhanam, 2024b), audit trails (Basireddy, 2023), configurable approval workflows (Basireddy, 2022), and a clear human-in-the-loop design for high-impact actions (Shneiderman, 2020).

Convergence of cyber and financial fraud

Much digital fraud begins as a cyberattack: phishing for credentials, malware on a customer device, or bots

testing stolen card numbers. Cherukuri (2023) shows that explainable AI and LLMs can improve detection and response across enterprise networks. Combining cybersecurity signals, such as device reputation, login anomalies, and malware indicators, with transaction-level fraud models gives a fuller view of risk than either alone. This convergence is especially relevant for account takeover and authorized push payment scams, where the fraud is set up long before the payment is made.

Discussion

Answering the Research Questions

RQ1 (techniques)

Tree-based ensembles on well-engineered behavioral features are the most consistently strong approach for transaction-level fraud. Unsupervised and hybrid methods improve detection of new fraud patterns, sequence models capture behavior over time, and graph methods reveal organized fraud and laundering networks. Imbalance handling and evaluation metrics that reflect investigator capacity are as important as model choice.

RQ2 (operations)

Production fraud detection requires low-latency real-time scoring, safe and frequent model updates with zero downtime (Sannidhanam, 2023), drift monitoring and feedback loops, explanations for investigators, configurable case management workflows (Basireddy, 2022), and audit-ready compliance records (Basireddy, 2023).

RQ3 (LLMs and agents)

LLMs are emerging as tools for investigation support, alert explanation, and reporting, and agents as tools for autonomous operations. Their safe use depends on reliable prompting (Sannidhanam, 2024a), cost-aware technique selection (Cherukuri, 2025), guardrails (Sannidhanam, 2024b), bounded autonomy (Sannidhanam, 2025), and human oversight.

RQ4 (challenges)

These are set out in Section 7.3.

An Integrated Framework

The findings point to a layered framework for ML-based fraud management, summarized in Table 3.

Table 3: Layered framework for ML-based fraud detection in digital financial transactions

<i>Layer</i>	<i>Function</i>	<i>Key enablers</i>	<i>Supporting studies</i>
Detection	Score transactions and entities for fraud risk	Ensembles, anomaly detection, sequence and graph models, imbalance handling	Carcillo et al. (2021); Jurgovsky et al. (2018); Weber et al. (2019)
Model operations	Serve, monitor, and update models safely	Real-time inference, zero-downtime updates, drift detection	Sannidhanam (2023); Dal Pozzolo et al. (2018); Gama et al. (2014)
Investigation support	Help people review and decide on alerts	Explainability, LLM summaries, reliable prompting	Lundberg & Lee (2017); Cherukuri (2023); Sannidhanam (2024a); Cherukuri (2025)
Action and workflow	Route alerts and carry out responses	Configurable case workflows, bounded autonomous agents	Basireddy (2022); Sannidhanam (2025)
Governance and compliance	Keep decisions safe, traceable, and defensible	Guardrails, audit trails, regulatory alignment	Sannidhanam (2024b); Basireddy (2023); FATF (2021)

Challenges and limitations

- **Label scarcity and delay.** Confirmed fraud labels are rare, late, and incomplete, which limits supervised learning and makes evaluation difficult (Dal Pozzolo et al., 2018).
- **Concept drift and adversarial adaptation.** Fraudsters change tactics in response to detection, so models degrade unless they are monitored and retrained (Gama et al., 2014; Biggio & Roli, 2018).
- **False positives and customer friction.** Every false decline harms a genuine customer. Balancing fraud loss against customer experience is a business decision as well as a technical one.
- **Explainability under latency constraints.** Detailed explanations take time to compute, while authorization decisions take milliseconds (Psychoula et al., 2021).
- **Data sharing and privacy.** Fraud networks often span institutions, but privacy rules and competition limit data sharing. Federated learning is one possible approach (Yang et al., 2019).
- **Reproducibility.** Confidential data and a lack of realistic public benchmarks make results hard to compare across studies (Abdallah et al., 2016; Lopez-Rojas et al., 2016).
- **LLM reliability and cost.** Hallucination, prompt injection, and the cost and latency of elaborate prompting limit where LLMs can be used safely (Ji et al., 2023; Cherukuri, 2025; Sannidhanam, 2024b).
- **Autonomy and accountability.** As agents take more actions, it becomes harder to assign responsibility for errors, and audit and oversight mechanisms have to keep pace (Basireddy, 2023; Sannidhanam, 2025).
- **Operational technical debt.** Production ML systems accumulate complexity in data pipelines, features, and configuration that can degrade reliability over time (Sculley et al., 2015).

Future Research Directions

- **Realistic benchmarks.** Shared, privacy-preserving datasets or high-fidelity simulators that capture drift, delayed labels, and adversarial behavior.
- **Graph and sequence integration.** Models that combine behavior over time with network structure to detect organized fraud earlier.
- **Label-efficient learning.** Semi-supervised, active, and self-supervised methods that make better use of scarce investigator feedback.
- **Cross-institution collaboration.** Federated and privacy-preserving methods that let institutions detect shared fraud networks without pooling raw data.
- **Real-time explainability.** Explanation methods fast enough for authorization decisions and faithful enough to support regulatory review.
- **Continuous deployment evaluation.** Empirical studies of how zero-downtime deployment and frequent retraining affect detection performance over time.

- **Evaluating LLMs in fraud operations.** Benchmarks for accuracy, hallucination, cost, and time savings when LLMs support investigation and reporting, including the cost-efficacy trade-offs of prompting strategies.
- **Governed autonomy.** Frameworks that define what fraud and operations agents may do on their own, with guardrails, approval workflows, and complete audit trails.
- **Cyber-fraud fusion.** Architectures that combine cybersecurity telemetry with transaction monitoring to detect fraud earlier in the attack chain.

Conclusion

ML has become central to fraud detection in digital financial transactions. The literature shows that ensembles built on behavioral features, supported by anomaly detection, sequence models, and graph learning, outperform static rules in detecting both known and emerging fraud. It also shows that model accuracy is only the starting point. Effective fraud detection requires real-time scoring, safe and frequent model updates, continuous adaptation to drift, explanations that investigators and regulators can use, configurable case management workflows, and audit-ready compliance records.

LLMs and autonomous agents add new capabilities, from investigation summaries to self-managing infrastructure, and new risks. Their safe use depends on reliable prompting, attention to cost, strong guardrails, bounded autonomy, and human accountability. For practitioners, the lesson is to invest in the full system around the model, not only the model. For researchers, the priorities are realistic evaluation, label-efficient and collaborative learning, real-time explainability, and governance frameworks that keep pace with increasing automation.

References

1. Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68, 90–113.
2. Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: A survey. *Data Mining and Knowledge Discovery*, 29(3), 626–688.
3. Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, 51, 134–142.
4. Basireddy, S. R. (2022). Digitizing risk management through configurable enterprise workflow architecture. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 14(4), 231–239.
5. Basireddy, S. (2023). Audit-ready compliance platform architecture for large language model regulated enterprises. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 15(01), 226–232. <https://doi.org/10.18090/samriddhi.v15i01.40>
6. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3), 602–613.
7. Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84, 317–331.
8. Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235–255.
9. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
10. Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 557, 317–331.
11. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), Article 15.
12. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
13. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

14. Cherukuri, M. (2023). Advancing autonomous cyber threat detection and incident response using explainable artificial intelligence and large language models in enterprise computer networks. *International Journal of Technology, Management and Humanities*, 9(01), 175–199.
15. Cherukuri, M. (2025). Cost, complexity, and efficacy of prompt engineering techniques for large language models. *IJSAT – International Journal on Science and Technology*, 16(2).
16. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3784–3797.
17. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10), 4915–4928.
18. Financial Action Task Force. (2021). *Opportunities and challenges of new technologies for AML/CFT*. FATF.
19. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), Article 44.
20. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
21. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), Article 248.
22. Jurgovsky, J., Granitzer, M., Ziegler, M., Calabretto, S., Portier, P.-E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, 100, 234–245.
23. Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.
24. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474.
25. Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422).
26. Lopez-Rojas, E. A., Elmir, A., & Axelsson, S. (2016). PaySim: A financial mobile money simulator for fraud detection. In *Proceedings of the 28th European Modeling and Simulation Symposium (EMSS)*.
27. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774.
28. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & 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), 559–569.
29. Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71.
30. Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). *A comprehensive survey of data mining-based fraud detection research* (arXiv:1009.6119). arXiv.
31. Psychoula, I., Gutmann, A., Mainali, P., Lee, S. H., Dunphy, P., & Petitcolas, F. (2021). Explainable machine learning for fraud detection. *Computer*, 54(10), 49–59.
32. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144).
33. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215.
34. Sannidhanam, A. H. (2023). Zero-downtime AI model updates in real-time inference systems.

- International Journal of Engineering Science & Humanities*, 13(2), 70–78. <https://www.ijesh.com/j/article/view/934>
35. Sannidhanam, A. H. (2024a). Prompt engineering patterns for reliable LLM outputs in production environments. *Journal of Multidisciplinary Knowledge*, 4(1), 29–39. <https://jmk.datatablets.com/index.php/j/article/view/168>
 36. Sannidhanam, A. H. (2024b). Guardrails and safety mechanisms for LLM-powered enterprise applications. *International Journal of Engineering Science & Humanities*, 14(3), 273–285. <https://www.ijesh.com/j/article/view/933>
 37. Sannidhanam, A. H. (2025). Autonomous AI agents for cloud infrastructure operations. *Journal of Contemporary Science and Technology Management*, 1(01), 83–100. <https://jcstm.info/index.php/jcstm/article/view/20>
 38. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems*, 28, 2503–2511.
 39. Shneiderman, B. (2020). Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction*, 36(6), 495–504.
 40. Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2015). APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. *Decision Support Systems*, 75, 38–48.
 41. Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). *Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics* (arXiv:1908.02591). arXiv.
 42. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837.
 43. West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security*, 57, 47–66.
 44. Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, 18(1), 30–55.
 45. Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2), Article 12.