**Global Multidisciplinary Journal**
eISSN: 2791-173X pISSN: 2791-2760

VOLUME04 ISSUE11
*Published 30-11-2025* *Page No. 111-119*

# Real-Time Credit Card Fraud Detection With Streaming Analytics: A Convergent Framework Using Kafka, Deep Learning, And Hybrid Provenance

**Dr. Anika Moreau**

**Department of Computer Science, University of Melbourne, Australia**

## ABSTRACT

**This article develops a comprehensive, publication-ready synthesis and original framework for near real-time credit card fraud detection grounded in streaming analytics, deep learning, and pragmatic system design. Drawing from empirical and methodological literature on real-time fraud detection, streaming platforms (Kafka, Spark, Flink), deep learning architectures, large-scale anomaly detection, and operational constraints in financial systems, the paper articulates a resilient architectural pattern that balances latency, detection accuracy, explainability, and data governance (Abakarim et al., 2018; Rajeshwari & Babu, 2016; Martín Hernández, 2015; Hebbar, 2025). The proposed Convergent Streaming Detection Framework emphasizes a tiered detection pipeline: ultrafast rule-based triage in the streaming path, lightweight explainable models for immediate scoring, and contextual deep models (including sequence and graph-based learners) operating on enriched windows for elevated scrutiny (Nicholls et al., 2021; Zhou et al., 2019). Practical considerations include feature engineering for streaming contexts, approaches to class imbalance and concept drift, strategies for low-latency model serving, and hybrid provenance and logging to preserve forensic trails without violating privacy or incurring prohibitive storage and throughput costs (Saxena & Gupta, 2017; Nguyen et al., 2020). The article also details rigorous evaluation metrics appropriate to streaming fraud contexts, an experimental design for realistic pilot deployments, adversarial threat modeling, and a multi-year research agenda emphasizing red-team testing and socio-technical evaluation. The synthesis stresses that engineering trade-offs—between latency and model complexity, explainability and predictive performance, and on-chain/off-chain evidence storage—must be made transparently and governed by regulatory and user-centric considerations (The Business Research Company, 2025; Udeh et al., 2024). The contribution is a practically actionable blueprint for researchers and practitioners seeking to deploy deep-learning-driven fraud detection in production-grade streaming environments.**

## KEYWORDS

**Real-time fraud detection; Kafka streams; streaming analytics; deep learning; credit card fraud; anomaly detection; model explainability.**

## INTRODUCTION

Digital payments and credit card transactions now form a central artery of global commerce. The ubiquity of card-not-present transactions, mobile wallets, and online marketplaces has expanded payment convenience while simultaneously inflating the opportunity surface for fraudsters to exploit vulnerabilities in identity, authentication, and transaction processing (The Business Research Company, 2025; Udeh et al., 2024). The persistent growth of transaction volume demands detection systems that operate at streaming timescales; batch analysis is insufficient to stop losses in-flight or to prevent cascading reputational and regulatory harm (Rajeshwari & Babu, 2016; Martín Hernández, 2015).

A well-established research trajectory seeks to harness machine learning and, more recently, deep learning to detect fraudulent transactions by learning patterns that deviate from legitimate behavioral norms (Abakarim et al., 2018; Nicholls et al., 2021). Simultaneously, systems engineering advancements—particularly the maturation of streaming platforms such as Apache Kafka, Apache Flink, and Apache Spark Streaming—provide the infrastructural substrate to operationalize low-latency detection pipelines (Saxena & Gupta, 2017). Together, these domains suggest the possibility of real-time, AI-driven fraud detection that can scale with transaction throughput while delivering robust detection quality.

Nevertheless, the literature highlights persistent tensions. High-capacity models, such as deep sequence learners or graph neural networks, can improve detection accuracy but often impose latency and resource costs that conflict with the need to act within seconds (Zhou et al., 2019; Abakarim et al., 2018). Conversely, lightweight models and rule engines are fast but can generate false positives that disrupt genuine customers and impose human review burdens (Rajeshwari & Babu, 2016). Moreover, streaming contexts introduce complications in feature computation, class imbalance, and concept drift—where attacker tactics evolve and degrade static models over time (Nicholls et al., 2021). Practical systems must therefore reconcile predictive sophistication with operational constraints and regulatory obligations related to privacy, explainability, and dispute resolution.

This article addresses those tensions by offering a convergent framework for real-time credit card fraud detection that explicitly integrates streaming architectural choices, a tiered model pipeline balancing speed and depth, practical feature engineering for streaming data, on-line learning strategies, robust evaluation metrics, and governance constructs for forensic evidence and regulatory reporting. The framework is informed by contemporary empirical work and operational case studies (Abakarim et al., 2018; Martín Hernández, 2015; Saxena & Gupta, 2017) and is positioned to guide both experimental pilot deployments and production-grade implementations.

The remainder of the paper is structured as follows. The methodology section explains the conceptual synthesis approach and the design rationale for the system blueprint. The results section describes the proposed Convergent Streaming Detection Framework in operational detail, enumerating model pipelines, feature design, and logging strategies. The discussion interprets trade-offs, threat models, and governance requirements, while the conclusion synthesizes recommendations, implications for practice, and a targeted research agenda. Throughout, claims and design choices are grounded in the referenced literature.

## METHODOLOGY

The methodology for this research is a structured conceptual synthesis oriented toward engineering applicability. The approach integrates three complementary activities: (1) systematic literature assimilation focusing on streaming analytics and fraud detection, (2) extraction of engineering patterns and constraints from applied reports and technical literature, and (3) formulation of an operational architecture and experimental evaluation strategy that can be implemented in realistic financial environments.

Literature assimilation involved curating peer-reviewed studies and high-quality technical reports that address real-time fraud detection, streaming infrastructure, deep learning architectures in fraud contexts, and operational considerations such as latency, scalability, and governance. Key academic sources included works proposing deep-learning real-time fraud models (Abakarim et al., 2018), practical descriptions of streaming analytics for credit-card fraud (Rajeshwari & Babu, 2016; Martín Hernández, 2015), scalable big-data analytics for e-commerce fraud (Zhou et al., 2019), and surveys of deep learning approaches to financial cybercrime (Nicholls et al., 2021). Complementary technical sources and industry analyses (Saxena & Gupta, 2017; Hivemind Technologies, 2024; The Business Research Company, 2025) informed infrastructure-level choices and business-context constraints.

From these sources, engineering patterns were extracted: stream partitioning and windowing strategies, acceptable latency budgets for transaction authorization, typical feature sets used in fraud scoring (transaction amount, merchant, time-of-day, device fingerprint, geolocation), approaches to handling severe class imbalance (oversampling, synthetic minority generation, cost-sensitive learning), and the role of human adjudication in reducing false positive impacts (Rajeshwari & Babu, 2016; Abakarim et al., 2018). These patterns were synthesized with platform capabilities—Kafka's partitioning semantics, exactly-once processing guarantees when configured correctly, and its suitability for decoupled producer-consumer designs—to derive practical implementation advice (Saxena & Gupta, 2017; Hivemind Technologies, 2024).

Finally, an architecture and experimental plan were constructed. The architecture specifies a tiered detection pipeline composed of (A) streaming rule-based triage for immediate action; (B) lightweight explainable models for primary scores in the authorization path; and (C) computationally heavier deep models operating on aggregated windows, graphs, or sequences for deferred elevated analysis and retraining (Abakarim et al., 2018; Zhou et al., 2019). The experimental plan outlines data requirements, evaluation metrics tailored for streaming contexts (time-aware precision/recall, detection latency distributions, cost-weighted loss utility), and adversarial test scenarios (poisoning, low-and-slow fraud, and mimicry) to validate robustness (Nicholls et al., 2021).

Throughout the methodology, explicit attention was paid to operational constraints and ethical-legal considerations: privacy-preserving logging, governance of model decisions, and transparent dispute-resolution workflows consistent with banking regulation and user expectations (Udeh et al., 2024; The Business Research Company, 2025).

## RESULTS

The principal output is the Convergent Streaming Detection Framework (CSDF), a concrete, implementable blueprint combining streaming infrastructure, a tiered modeling pipeline, feature engineering strategies, logging and forensic commitments, and an evaluation regimen. Below, the framework is described in operational detail and linked to supporting literature.

Architectural Overview and Dataflows

At its core CSDF organizes processing into three operational planes: the fast-path (authorization-time operations), the enrichment & contextualization plane, and the forensic & learning plane.

Fast-Path (Authorization-Time Operations): Transactions flow from point-of-sale or online checkout systems into a Kafka topic partitioned by card or account identifier to maintain event locality and ordering (Saxena & Gupta, 2017). The fast-path performs ultralow-latency computations: rule-based heuristics (velocity limits, merchant blacklists), device fingerprint mismatches, and a lightweight scoring model (e.g., logistic regression or small decision-tree ensemble) trained to be explainable and computationally trivial (Rajeshwari & Babu, 2016; Abakarim et al., 2018). Actions in this path include immediate declines, step-up authentication prompts, soft-blocks, or allow-with-monitoring. The design principle is that any decision here must meet strict latency constraints—typically sub-200 millisecond added latency to the authorization path, consistent with industry tolerances for customer experience (Hivemind Technologies, 2024).

Enrichment & Contextualization Plane: The fast-path emits event pointers to Kafka topics consumed by enrichment services. These services augment transactions with richer context: historical aggregates (e.g., rolling spend over last 24 hours), behavioral embeddings from user history, device reputational scores, and merchant risk indicators (Zhou et al., 2019). Enriched events feed a mid-path scoring layer composed of moderately complex models—gradient boosted trees or compact neural nets—that can run under slightly relaxed latency budgets (e.g., 500ms–2s) and that yield more refined risk probabilities. This plane also supports link-analysis features where transaction graphs reveal suspicious connectivity (e.g., multiple cards transacting with a novel merchant and shared device fingerprint), which frequently signal coordinated fraud rings (Zhou et al., 2019).

Forensic & Learning Plane: Heavy models (sequence models, graph neural networks, variational models for anomaly scoring) consume streams of enriched events aggregated into windows (e.g., last 30 minutes to 24 hours) and run asynchronously. These models prioritize accuracy and contextual reasoning over sub-second latency and provide labels that drive human adjudication workflows and model retraining pipelines (Abakarim et al., 2018; Nicholls et al., 2021). This plane also implements hybrid logging: comprehensive raw logs reside in secure off-line storage while cryptographic commitments (e.g., digest hashes) or summarized evidence are persisted to an immutable audit trail to preserve forensic integrity without exposing sensitive data or imposing on-chain scalability costs (Nguyen et al., 2020).

Tiered Modeling Strategy and Rationale

The framework’s tiered modeling strategy directly addresses the latency-accuracy trade-off. The first tier sacrifices model complexity for interpretability and speed, reducing immediate customer friction while catching obvious fraud cases. The second tier refines scoring with additional context and is the principal decision input for soft enforcement actions. The third tier focuses on complex patterns and cross-session behavior that only become apparent over time or across multiple entities—for example, linkages between merchant accounts, devices, and card clusters that indicate coordinated fraud networks (Zhou et al., 2019; Abakarim et al., 2018).

Feature Engineering for Streaming Contexts

The CSDF emphasizes feature families that are feasible to compute in a streaming environment:

Temporal Velocity Features: Counts and sums over sliding windows (per minute, per hour) capturing sudden spikes in transaction rates or amounts (Rajeshwari & Babu, 2016). Efficiently maintaining these requires windowed aggregations implemented in streaming processors.

Behavioral Embeddings: Compact, incremental user embeddings derived from recent transaction sequences. These embeddings can be updated online with lightweight recurrent updates or reservoir sampling techniques to capture short-term behavioral drift without reprocessing entire histories (Nicholls et al., 2021).

Device and Channel Signatures: Deterministic device fingerprints, IP geolocation signals, and channel identifiers (card-present vs. card-not-present) that correlate strongly with fraud risk (Martín Hernández, 2015).

Graph-Based Connectivity Metrics: Real-time approximations of graph metrics (e.g., shared devices across cards) computed using streaming graph sketches or micro-batching to detect collaborative fraud behavior (Zhou et al., 2019).

Merchant Risk Signals: Aggregated merchant-level statistics—chargeback rates, sudden spike in high-value transactions—that serve as contextual priors for transaction-level models.

Handling Class Imbalance and Concept Drift

Class imbalance—fraud being a tiny fraction of transactions—is a central modeling challenge. CSDF recommends a combination of strategies:

Windowed Resampling and Synthetic Data: Over-sampling rare events within sliding windows and using synthetic minority generation methods for training to maintain model sensitivity to recent fraud patterns (Abakarim et al., 2018).

Cost-Sensitive Learning: Loss functions weighted by the business cost of false negatives versus false positives, making models directly optimize for economic impact rather than raw accuracy (Nicholls et al., 2021).

Online and Incremental Learners: Deploy models that support incremental updates—either via online gradient updates or periodic mini-batch retraining using recent labeled examples—to respond to evolving attack patterns (Nicholls et al., 2021).

Drift Detection and Adaptive Thresholding: Monitor distributions of features and model score outputs; trigger retraining or threshold recalibration when statistical divergence exceeds tolerance bands. Such monitoring reduces silent degradation in detection performance (Rajeshwari & Babu, 2016).

Model Explainability and Human-in-the-Loop Workflows

Because false positives directly impact customer experience, CSDF embeds explainability and adjudication mechanisms. Lightweight models provide feature-attribution outputs (e.g., SHAP-like approximations) for human analysts to understand why a transaction was flagged (Abakarim et al., 2018). Human adjudicators resolve ambiguous cases and their decisions feed supervised signals back into retraining pipelines. This loop is essential both for operational quality and for compliance obligations that require explainable decision rationale in dispute proceedings (Udeh et al., 2024).

Deployment and Latency Engineering on Kafka Streams

Kafka's partitioning and consumer-group semantics facilitate ordering guarantees by account id, enabling deterministic feature windows and stateful stream processing (Saxena & Gupta, 2017). Exactly-once semantics, while operationally complex, prevent double-counting and ensure consistent state in the face of consumer restarts. CSDF recommends stateless fast-path microservices tightly integrated with Kafka Streams for ultrafast rules, while stateful processors and model servers back the mid- and deep-path layers with controlled concurrency to meet latency budgets.

Forensic Logging and Governance

To reconcile forensic needs with privacy and storage constraints, CSDF employs hybrid logging: verbose logs retained in secure, access-controlled object stores for a bounded retention window (e.g., 2–7 years depending on regulation) and cryptographic digests or compact evidence summaries persisted to an immutable, append-only ledger or a tamper-resistant audit service (Nguyen et al., 2020). The digest approach preserves the ability to prove log integrity while avoiding exposure of personal data in immutable stores. Governance policies specify access controls, audit trails, and dispute-resolution timelines consistent with banking regulations and consumer protection requirements (The Business Research Company, 2025).

Evaluation Metrics and Experimental Design

Standard batch metrics (precision, recall, ROC AUC) are necessary but insufficient for streaming fraud contexts. CSDF proposes streaming-sensitive metrics:

Time-to-Detection Distribution: Histogram of detection latency from transaction time to first high-confidence flag, which reflects operational responsiveness.

Cost-Weighted Utility: Economic loss prevented minus remediation costs and customer support overhead, evaluated per time window to capture operational trade-offs.

False Positive Impact Score: A composite measure that accounts for customer complaint rates, manual review time, and conversion loss.

Model Calibration Drift Index: Statistical measure of change in score distributions relative to baseline, used to proactively trigger retraining.

An experimental pilot design samples anonymized production transaction streams or synthetic workloads mirroring real-world distributions, with staged injections of fraud patterns to evaluate detection under varying signal-to-noise ratios and adversarial tactics (Nicholls et al., 2021; Zhou et al., 2019).

## DISCUSSION

The CSDF operationalizes a suite of practical and theoretical insights from the referenced literature. This discussion examines the framework's implications, trade-offs, and limitations, and articulates an agenda for empirical validation and incremental adoption.

Balancing Latency and Model Complexity

A core engineering tension is the trade-off between fast decisions and sophisticated inference. CSDF resolves this through tiering: keep the fast-path simple but capable, and push complexity to asynchronous layers where computationally expensive models can deliver improved precision and forensic evidence without disrupting the customer experience. This architectural pattern is consistent with empirical findings that ensemble and deep sequence models improve detection but are often unsuitable for millisecond-level decisions (Abakarim et al., 2018; Zhou et al., 2019).

Explainability, Accountability, and Regulatory Compliance

Financial institutions operate under strict regulatory scrutiny. Black-box models without explainability can undermine dispute resolution and invite regulatory sanctions. CSDF's emphasis on explainable first- and second-tier models and human adjudication is a direct response to literature stressing the need for interpretable decisions in high-impact domains (Nicholls et al., 2021; The Business Research Company, 2025). Moreover, transparent governance for logging and audit ensures evidentiary integrity for regulatory examinations.

Adversarial Threats and Resilience

Attackers continually adapt. The literature documents varied adversarial techniques—including mimicry attacks, poisoning, and low-and-slow behaviors—that reduce detection efficacy (Nicholls et al., 2021). CSDF recommends integrating adversarial testing into the development lifecycle, including red-team simulations that probe both model vulnerabilities and operational workflows. Continuous monitoring, ensemble diversity, and adversarially robust training techniques (where feasible) are part of the resilience toolkit.

Data Privacy and Forensics: A Tightly Coupled Trade-off

Comprehensive logs are valuable for forensics but pose privacy risks if not properly managed. Hybrid logging that stores detailed logs off-chain in controlled repositories while preserving integrity via cryptographic commitments aligns with best practices in balancing auditability with privacy (Nguyen et al., 2020). Legal teams and compliance officers must set retention policies and access governance calibrated to jurisdictional mandates and consumer rights.

Operational Costs and Organizational Readiness

Implementing CSDF requires organizational capabilities—data engineering for streaming pipelines, MLops for model serving and retraining, security teams for log governance, and compliance for legal oversight. Smaller institutions may find the requisite investments challenging; consortium-based or managed-service approaches can disperse costs but raise governance and vendor-risk questions. The literature underscores that successful deployments typically require cross-functional teams and incremental rollouts with measured KPIs (Saxena & Gupta, 2017; The Business Research Company, 2025).

Limitations of the Proposed Framework

While CSDF synthesizes best practices, several limitations must be acknowledged. First, robust empirical validation in production-scale environments is required; many published studies are lab-scale or rely on historical datasets that do not capture real-time complexities or adversarial adaptation (Abakarim et al., 2018; Nicholls et al., 2021). Second, the framework depends on accurate and timely enrichment signals: where upstream telemetry (device fingerprints, merchant metadata) is incomplete or noisy, detection performance will degrade. Third, legal constraints around data sharing and cross-border data flows may restrict the ability to use consortium-led forensic ledgers or to aggregate cross-institutional graphs that reveal fraud networks (The Business Research Company, 2025; Udeh et al., 2024).

Research and Implementation Roadmap

To operationalize and validate CSDF, the following practical roadmap is recommended:

Sandbox Pilots: Deploy a staged pilot in a sandbox environment with real-time feeds and realistic user behavior. Measure time-to-detection, cost-weighted utility, and false positive impacts.

Adversarial Stress Testing: Conduct red-team exercises simulating poisoning, mimicry, and low-and-slow attacks to evaluate resilience.

Interoperability Experiments: Explore consortium-level graph analytics for cross-institution fraud detection while trialing privacy-preserving protocols (e.g., secure multiparty computation for feature aggregation) to assess feasibility.

User-Centered Evaluations: Study customer perceptions of step-up authentication, dispute resolution flows, and communication strategies to minimize churn and dissatisfaction following false positives.

Standards and Compliance Frameworks: Work with regulators and industry bodies to define standards for logging, retention, explainability thresholds, and acceptable latency-cost trade-offs (The Business Research Company, 2025).

## CONCLUSION

Real-time credit card fraud detection is both a pressing operational challenge and a rich research frontier. The Convergent Streaming Detection Framework presented in this article synthesizes streaming infrastructure, tiered detection pipelines, practical feature engineering, governance-conscious logging, and evaluation metrics into a coherent blueprint for practitioners and researchers. Key engineering principles include: maintain a minimal-latency fast-path for immediate triage; enrich and refine scoring in mid-path systems with moderate latency tolerance; reserve computationally intensive deep models for asynchronous contextual analysis; and employ hybrid logging to reconcile forensic needs with privacy and cost constraints (Abakarim et al., 2018; Rajeshwari & Babu, 2016; Saxena & Gupta, 2017; Nguyen et al., 2020).

The path from conceptual design to robust production systems requires disciplined empirical evaluation—sandbox pilots, adversarial testing, careful monitoring for drift, and user-centered assessments of remediation workflows. Future research should prioritize large-scale field validations, cross-institutional experiments for network-level detection, and legal-technical studies of privacy-preserving cross-border cooperation. With deliberate architectural choices and governance frameworks, streaming analytics and deep learning can substantially reduce fraud losses while preserving customer trust and regulatory compliance.

## REFERENCES

1. Rajeshwari, U., and B. Sathish Babu. Real-time credit card fraud detection using streaming analytics. 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). IEEE, 2016.
2. Martín Hernández, Sergi. Near real time fraud detection with Apache Spark. 2015.
3. Jayanthi, D., G. Sumathi, and Sriperumbudur Sriperumbudur. A framework for real-time streaming analytics using a machine learning approach. Proceedings of national conference on communication and informatics. 2016.
4. Zhou, Hangjun, et al. A scalable approach for fraud detection in online e-commerce transactions with big data analytics. Computers, Materials & Continua 60.1 (2019): 179-192.
5. Saxena, Shilpi, and Saurabh Gupta. Practical real-time data processing and analytics: distributed computing and event processing using Apache Spark, Flink, Storm, and Kafka. Packt Publishing Ltd, 2017.
6. Nicholls, J., Kuppa, A., & Le-Khac, N.-A. Financial cybercrime: a comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape. IEEE Access, 2021.
7. Chen, Z., Van Khoa, L. D., Teoh, E. N., Nazir, A., Karuppiah, E. K., & Lam, K. S. Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: a review. Knowledge and Information Systems, 2018.
8. Jensen, R., & Iosifidis, A. Fighting money laundering with statistics and machine learning. IEEE Access, 2023.
9. Demetis, D. S. Fighting money laundering with technology: a case study of bank x in the UK. Decision Support Systems, 2018.
10. Chen, Z., Soliman, W. M., Nazir, A., & Shorfuzzaman, M. Variational autoencoders and Wasserstein generative adversarial networks for improving the anti-money laundering process. IEEE Access, 2021.
11. Abbassi, H., Abdellah, B., Mendili, S., & Youssef, G. End-to-end real-time architecture for fraud detection in online digital transactions. International Journal of Advanced Computer Science and Applications, 2023.
12. Alkhalili, M., Qutqut, M. H., & Almasalha, F. Investigation of applying machine learning for watch-list filtering in anti-money laundering. IEEE Access, 2021.
13. The Business Research Company. Financial Services Market Definition. The Business Research Company

Insight. Jan. 2025.

**14.** Stéphane Derosiaux. The Rise of Data Streaming and the Evolution of Data at Rest: 2018-2024. Medium. 5 Feb. 2025.

**15.** Ezekiel Onyekachukwu Udeh et al. The role of big data in detecting and preventing financial fraud in digital transactions. World Journal of Advanced Research and Reviews. 24 May 2024.

**16.** Hivemind Technologies. Apache Kafka in the Financial Sector: Real-Time Data Processing for Banking Operations. LinkedIn Pulse. 29 Oct. 2024.

**17.** Jim Marous. Improving the Customer Experience in Banking. Digital Banking Report. Feb. 2017.

**18.** Seshika Fernando. Real-Time Analytics in Banking and Finance: Use Cases. WSO2 Whitepaper. April 2017.

**19.** Hebbar, K. S. AI-DRIVEN REAL-TIME FRAUD DETECTION USING KAFKA STREAMS IN FINTECH. International Journal of Applied Mathematics, 38(6s), 770-782. 2025.

**20.** Wang, J., & Liu, Q. Advancements in AI and Machine Learning for Financial Fraud Detection. International Journal of Data Science, 2023.

**21.** Baah, S. S., Adu-Twum, H. T., Adjei, S. O., Ampadu, G., Martins, A. O., & Fonkem, B. Leveraging big data analytics to combat emerging financial fraud schemes in the USA: A literature review and practical implications. World Journal of Advanced Research and Reviews, 2024.

**22.** Uddin, N. Significance of Live Streaming in Shaping Business: A Critical Review and Analytical Study. Social Networking, 2024.

**23.** Federal Trade Commission. Consumer Sentinel Network Data Book. 2024.

**24.** The Motley Fool. Report on Identity Theft. 2024.

**25.** The Motley Fool. Credit Card Fraud Statistics. 2024.

**26.** Wang and Liu. Advancements in AI and Machine Learning for Financial Fraud Detection. 2023.

**27.** Baah et al. Leveraging big data analytics to combat emerging financial fraud schemes. 2024.

**28.** Uddin, N. Significance of Live Streaming in Shaping Business. 2024.